UNIVERSITÉ GRENOBLE I - JOSEPH FOURIER

U.F.R. D'INFORMATIQUE
ET DE MATHÉMATIQUES APPLIQUÉES

MODÈLES À STRUCTURE CACHÉE : INFÉRENCE, SÉLECTION DE MODÈLES ET APPLICATIONS

THÈSE

pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ JOSEPH FOURIER

Discipline : Mathématiques Appliquées

Présentée et soutenue publiquement

par

Jean-Baptiste DURAND

Le 31 janvier 2003

Directeur de thèse : Gilles CELEUX

Jury

Serge DÉGERINE	Président
Elisabeth GASSIAT	Rapporteur
Christian ROBERT	Rapporteur
Yann GUÉDON	Examinateur

Thèse réalisée au sein du projet IS2 de l'Inria Rhône-Alpes.

Où est-il ?
Où se cache-t-il ?
Que ferai-je pour le trouver ?

L'Avare, Molière (1668).
Acte IV, scène 7.

À PJH,

à ma famille,

à Marion, la figure 3.9 c).

Remerciements

Le bon déroulement de cette thèse, jusqu'à son heureux dénouement, sont en grande partie imputables à Gilles Celeux. Je le remercie donc très chaleureusement, aussi bien pour avoir dirigé mes travaux avec talent que pour m'avoir accompagné amicalement dans ce cheminement et même, à l'occasion, en dehors de mes activités professionnelles.

Ces remerciements sont à partager avec Yann Guédon, qui s'est impliqué de manière remarquable dans cette thèse et lui a donné sa dimension actuelle, grâce à ses conseils experts et ses relectures approfondies.

Elisabeth Gassiat et Christian Robert ont droit à toute ma gratitude pour leur rapport détaillé et constructif, et pour leur intérêt à l'égard de ces travaux.

Merci également à Serge Dégerine pour avoir accepté de présider le jury et pour ses annotations lors de la relecture du présent document. J'associe à ces remerciements Alain Le Breton, qui a en premier lieu accepté cette charge.

J'ai réellement apprécié d'avoir l'occasion de collaborer avec Florence Forbes, Paulo Gonçalvès, Christian Lavergne et Olivier Gaudoin. Merci à eux pour leur amitié, leur disponibilité et leur implication dans cette thèse.

Je dois beaucoup à mes anciens enseignants de statistique, qui m'ont donné mon goût pour cette discipline, en particulier A. Antoniadis et G. Grégoire. J'ai eu grand plaisir à avoir des contacts professionnels avec nombre d'entre eux, dont O. François et J. Diebolt. J'ai trouvé une aide très chaleureuse auprès de J.-L. Soler et de Maryse Béguin pour mes enseignements à l'Ensimag. Après avoir pu admirer, en tant qu'élève, leurs qualités pédagogiques, je n'ai pas été surpris de découvrir chez eux tant de qualités humaines. Merci enfin à J.-L. Roch pour son tutorat et à A. Bienvenüe, Y. Pigeonnat, J. Monnier et V. Perrier pour leur aide bienveillante à la préparation de mes enseignements.

Tous les membres du projet IS2 ont, chacun à leur manière, contribué à créer des conditions de travail (et de détente !) des plus agréables.

Merci donc à mes collègues de bureau(x) et désormais ami(e)s : Cécile Delhumeau, Nathalie Peyrard, Myriam Garrido, Jérôme Écarnot et Grégory Noulin. J'adresse toutes mes amitiés aux membres passés, actuels et peut-être futurs du projet. Ainsi, j'ai eu le plaisir de côtoyer Christophe Biernacki (à qui je dois plusieurs idées et figures de ce document), Christophe Ambroise, Claudine Robert, Anatoli Iouditski, Henri Bertholon, Philippe Garat, Stéphane Girard, Émilie Lebarbier, Edwige Allain, Gérard Boudjema, Guillaume Bouchard, Matthieu Vignes, Julien Jacques, Carine Véra, Franck Corset, Olivier Martin, Isabel Brito, Anne Guérin-Dugué, Cyril Goutte, Yann Vernaz, Catherine Trottier, Véronique Équy, Ollivier Taramasco et Mhamed Elaroui. Merci, de plus, à Matthieu Thivin et Mohamed El Khayari (qui m'ont aidé par leurs travaux de stage) ainsi qu'à Franz Chouly et Cyril Martin.

Ces remerciements s'adressent également à mes voisins de l'Inria, dont Claude Lemaréchal, Pierre-Brice Wieber, Jérôme Malick, Aris Daniilidis, Éric Rutten et Alain Girault, ainsi qu'à ceux du LMC et du TIMC, en particulier Frédérique Letué, Jeff Cœurjolly, Sophie Fontaine & Sophie Achard, Laurent Doyen, Jérémie Bigot, Anne Bilgot, Vincent Janicot, Vincent Guigues, Ouadia Radouane et François Gannaz.

Enfin, j'ai été touché par le soutien, tout au long de cette thèse, de mes amis du Sappey, de l'harmonie d'Eybens, des *Rainbow Swingers*... sans oublier les encouragements d'Assia, de Dan et de Georges. Un grand merci à ceux et celles qui ont poussé l'amitié jusqu'à venir m'encourager à l'occasion de ma soutenance.

Table des matières

Glossaire

Principales notations

$(\Omega, \mathcal{F}, P)$	:	espace probabilisé
X	:	variable aléatoire (observée ou cachée)
$\boldsymbol{X}$	:	processus aléatoire (observé ou caché)
x	:	réalisation de la variable aléatoire X
$\boldsymbol{x}$	:	réalisation du processus aléatoire $\boldsymbol{X}$
$\mathcal{X}$	:	ensemble des valeurs prises par le processus $\boldsymbol{X}$
N	:	nombre total de variables aléatoires du modèle
Y	:	variable aléatoire observée
$\boldsymbol{Y}$	:	processus aléatoire observé
y	:	réalisation de la variable aléatoire observée Y
$\boldsymbol{y}$	:	réalisation du processus aléatoire observé $\boldsymbol{Y}$
$\mathcal{Y}$	:	ensemble des valeurs prises par le processus $\boldsymbol{Y}$
n	:	nombre de variables aléatoires observées du modèle
R	:	nombre de réalisations indépendantes du processus $\boldsymbol{Y}$
S	:	variable aléatoire cachée
$\boldsymbol{S}$	:	processus aléatoire caché
s	:	réalisation de la variable aléatoire cachée S
$\boldsymbol{s}$	:	réalisation du processus aléatoire caché $\boldsymbol{S}$
$\mathcal{S}$	:	ensemble des valeurs prises par le processus $\boldsymbol{S}$
$N_{\boldsymbol{S}}$	:	nombre de variables aléatoires cachées du modèle
$(\mathcal{X}_{\boldsymbol{X}}, \mathcal{F}_{\boldsymbol{X}}, P_{\boldsymbol{X}})$	:	espace image de $(\Omega, \mathcal{F}, P)$ par $\boldsymbol{X}$
$f_{\boldsymbol{X}}$	:	densité de $P_{\boldsymbol{X}}$
$P(\boldsymbol{X} = \boldsymbol{x})$	:	notation pour $P_{\boldsymbol{X}}(\boldsymbol{x})$ ou pour $f_{\boldsymbol{X}}(\boldsymbol{x})$
A, B, C, D	:	parties de l'ensemble des sommets $\mathcal{U}$
$\boldsymbol{X}_A$	:	restriction du processus $\boldsymbol{X}$ aux sommets du sous-ensemble A de $\mathcal{U}$
$\mathcal{X}_A$	:	ensemble des valeurs prises par le processus $\boldsymbol{X}_A$
$\boldsymbol{Y}_A$	:	restriction du processus $\boldsymbol{Y}$ aux sommets observés de A, où $A \subset \mathcal{U}$
$\mathcal{Y}_A$	:	ensemble des valeurs prises par le processus $\boldsymbol{Y}_A$
$\boldsymbol{S}_A$	:	restriction du processus $\boldsymbol{S}$ aux sommets cachés de A, où $A \subset \mathcal{U}$
$\mathcal{S}_A$	:	ensemble des valeurs prises par le processus $\boldsymbol{S}_A$

$\perp\!\!\!\perp$	:	relation d'indépendance conditionnelle
$\mathcal{U}$	:	ensemble des sommets d'un graphe ou des indices d'un processus
$\mathcal{E}$	:	ensemble des arêtes d'un graphe
$\mathcal{G} = (\mathcal{U}, \mathcal{E})$	:	graphe (généralement la structure d'un modèle graphique)
t, u, v	:	sommets d'un graphe ou indices d'un processus
X_u	:	variable aléatoire (observée ou cachée) associée au sommet u
X_v^u	:	dans un processus indexé par un arbre, sommet de profondeur u et de position v
$(\mathcal{U}, \mathcal{E}^M)$	:	graphe moral associé au graphe $(\mathcal{U}, \mathcal{E})$
$\mathcal{G}(A)$	:	graphe engendré par le sous-ensemble de sommets A
$\mathrm{An}(A)$	:	plus petit ensemble ancestral contenant A
$\mathrm{pa}(u)$	:	ensemble des parents d'un sommet u de $\mathcal{G}$
$\mathcal{C}$	:	clique d'un graphe $\mathcal{G}$
$X_i^{(l)}$	:	le sommet de la clique $\mathcal{C}_i$ possédant $l-1$ parents dans $\mathcal{C}_i$
$\to$	:	relation de parenté entre sommets d'un graphe
$\leftrightarrow$	:	relation d'adjacence entre sommets d'un graphe
$\psi_{\mathcal{C}}$	:	fonction potentiel de la clique $\mathcal{C}$
$V_{\mathcal{G}}$	:	ensemble des cliques du graphe $\mathcal{G}$
$N_{\mathcal{C}}$	:	nombre de sommets de la clique $\mathcal{C}$
$\mathrm{NC}(\mathcal{G})$	:	nombre de cliques du graphe $\mathcal{G}$
j, k	:	valeurs des états cachés
K	:	nombre de valeurs possibles d'un état caché
$\boldsymbol{a}$	:	valeur des parents de X_u
$\theta_{\boldsymbol{a}}$	:	paramètre d'émission tel que $P(Y_u = y \mid \boldsymbol{X}_{\mathrm{pa}(u)} = \boldsymbol{a}) = P_{\theta_{\boldsymbol{a}}}(y)$
Θ	:	espace des paramètres d'émission
$(P_\theta)_{\theta\in\Theta}$	:	famille paramétrique de lois d'émission
$P = (p_{\boldsymbol{a},i})_{\boldsymbol{a},i}$	:	matrice de transition
$\pi = (\pi_i)_i$	:	loi d'un état initial ou loi stationnaire
λ	:	paramètre d'un modèle
Λ	:	espace des paramètres
$\mathbb{E}_\lambda$	:	espérance sous la loi P_λ
$\mathbb{E}_{\tilde{P}}$	:	espérance sous la loi $\tilde{P}$
$\mathcal{T}$	:	arbre de jonction
a	:	arête de l'arbre de jonction
$\mathcal{C}_0$	:	clique racine de l'arbre de jonction

$\mathcal{S}_a$	:	séparateur de sommets associé à l'arête a
$\mathcal{T}_a$	:	partie de l'arbre séparée par l'arête a et ne contenant pas $\mathcal{C}_0$
$\mathcal{T}_a^c$	:	partie de l'arbre séparée par l'arête a et contenant $\mathcal{C}_0$
$\bar{\mathcal{K}}_a$	:	ensemble des sommets de $\mathcal{G}$ dans $\mathcal{T}_a$
$\mathcal{K}_a$	:	ensemble des sommets de $\mathcal{G}$ dans $\mathcal{T}_a \setminus \mathcal{S}_a$
$\bar{\mathcal{K}}_a^c$	:	ensemble des sommets de $\mathcal{G}$ dans $\mathcal{T}_a^c$
$\mathcal{K}_a^c$	:	ensemble des sommets de $\mathcal{G}$ dans $\mathcal{T}_a^c \setminus \mathcal{S}_a$
$\mathcal{D}$	:	ensemble des modèles de Markov cachés orientés acycliques à structure morale
${}^t\Sigma$	:	transposée de la matrice Σ

Introduction générale

Des modèles graphiques et modèles de mélanges aux modèles de Markov cachés

L'objet de cette thèse est l'étude d'une famille de processus aléatoires, à des fins de modélisation. Plus spécifiquement, nous nous intéressons à des processus mettant en jeu un nombre fini (mais arbitrairement grand) de variables aléatoires, pouvant être indexées par les sommets d'un graphe. Nous portons une attention plus particulière aux cas où les dépendances entre ces variables aléatoires se traduisent par l'existence de zones homogènes dans le graphe, au sens où des variables adjacentes ont tendance à avoir un comportement similaire. Un cadre privilégié dans ce type de situation est celui des modèles graphiques comportant des variables cachées.

D'après Lauritzen, 1996 [75], les modèles graphiques trouvent leur origine, entre autres, dans la physique statistique et dans les idées de Gibbs, 1902 [59]. Il s'agit de modèles probabilistes dans lesquels les variables aléatoires sont représentées par les sommets d'un graphe, les relations d'indépendance conditionnelle entre ces variables étant définies par les arêtes du graphe. La génétique a également contribué au développement des modèles graphiques, avec en particulier les travaux de Wright dans les années 1920–1930 (voir [123] par exemple). L'étude des interactions dans les tableaux de contingence à trois entrées a aussi été une source de meilleure connaissance de ces modèles (voir Darroch *et al.*, 1980 [35]). Enfin, Pearl, 1982 [97] a contribué à la généralisation et à l'approche bayésienne des modèles graphiques, ce qui depuis leur a également valu le nom de *réseaux bayésiens*. L'ajout, dans les modèles graphiques, de variables aléatoires discrètes autres que celles directement observées, et appelées pour cette raison *variables cachées*, permet d'une part de représenter, à l'aide des modèles de mélange, les inhomogénéités dans le comportement des variables observées. D'autre part, la définition, à l'aide de graphes, des dépendances entre variables cachées, induit des relations de dépendance entre les variables observées, qui permettent par exemple la prise en compte des zones homogènes. Ainsi, la notion d'influence entre variables observées voisines est traduite indirectement grâce aux variables cachées, qui définissent finalement la structure de dépendance du modèle. Ces modèles graphiques à structure cachée sont appelés *modèles de Markov cachés* et peuvent être également vus comme des champs de Gibbs à données manquantes (voir Besag, 1974 [8]).

Les modèles de Markov cachés constituent ainsi une vaste classe de modèles probabilistes dont l'un des atouts est la facilité d'interprétation due aux variables cachées. En effet, une application majeure des modèles de mélange est la classification automatique, où une variable cachée est interprétée comme la classe de la variable observée correspondante. De la même manière, dans un modèle de Markov caché, les zones du graphe où les variables cachées voisines ont même valeur induisent des zones homogènes dans le processus observé. La traduction des connaissances a priori du modélisateur sur la manière dont les classes se propagent passe donc par la définition graphique de la notion de voisinage entre variables aléatoires.

Ce fort potentiel des modèles de Markov cachés explique leur utilisation répandue en analyse et segmentation d'images, à l'aide de champs de Markov cachés, avec par exemple l'article de référence de Besag, 1986 [9]. Les chaînes de Markov cachées sont un cas particulier de modèle de Markov caché - historiquement le premier modèle de cette famille introduisant des dépendances entre variables observées. Élaboré entre les années 1960 et 1970 par Baum *et al.* [6], ce modèle a connu une utilisation intensive en reconnaissance automatique de la parole à partir de 1970, par exemple avec Baker, 1975 [4] et Jelinek *et al.*, 1975 [65] (voir le tutoriel de Rabiner, 1989 [102] au sujet de l'utilisation des chaînes de Markov cachées en reconnaissance de la parole). Cette utilisation a été étendue à d'autres domaines de la reconnaissance des formes, par exemple la reconnaissance de gestes, qui a fait l'objet de la thèse de Braffort [14]. Les chaînes de Markov cachées sont également utilisées en analyse du génome (voir Churchill, 1989 [29]). Les arbres de Markov cachés, définis à l'origine par Crouse, Nowak et Baraniuk, 1998 [32], qui sont une extension simple des chaînes de Markov cachées, ont été utilisés en traitement du signal basé sur l'analyse en ondelettes, puis en catégorisation de documents par Diligenti, Frasconi et Gori, 2001 [40].

Il est important de remarquer que les variables cachées n'ont pas, a priori, d'existence physique dans le phénomène observé : elles sont avant tout utilisées pour créer des modèles flexibles. Cependant, après analyse des données en regard du modèle, elles trouvent souvent une interprétation concrète a posteriori : phonème dans la reconnaissance de parole, zone codante dans l'analyse de génome, signal ou bruit dans le débruitage par ondelettes.

L'inférence dans les modèles de Markov cachés

La flexibilité de ces modèles les rend très intéressants en pratique, mais la présence de variables cachées complique l'inférence statistique. Le calcul, pour un modèle entièrement spécifié, de la loi jointe d'un ensemble de variables observées (typiquement le calcul de la vraisemblance), nécessite des algorithmes de complexité au mieux polynomiale, au pire exponentielle en fonction du nombre de variables du modèle. L'estimation des paramètres basée sur la vraisemblance est rendue difficile par l'absence de formule explicite pour le maximum de vraisemblance, et requiert en général des algorithmes itératifs, comme l'algorithme EM. De plus, dans ce cas, chaque itération met elle-même en jeu les algorithmes de calcul de probabilités évoqués ci-dessus. Il existe des méthodes d'estimation de paramètre alternatives à la maximisation de la vraisemblance. Citons par exemple l'approche dite *Minimum Discrimination Information* ou *MDI* qui consiste à minimiser un critère d'entropie relative, parmi toutes les distributions vérifiant des contraintes concernant des moments donnés (voir Shore et Johnson, 1980 [109]). L'approche *MDI* est une généralisation de l'approche par maximum d'entropie de Cover et Thomas, 1991 [30] et peut également servir de base à des méthodes de sélection de modèles. Enfin, l'interprétation fine des modèles de Markov cachés nécessite le plus souvent la restauration des états cachés, généralement par maximisation de leur probabilité conditionnelle, sachant les données observées (principe du Maximum A Posteriori ou MAP).

Comme les auteurs l'expliquent dans Smyth, Heckerman et Jordan, 1997 [110] ou dans Lucke, 1996 [87], un algorithme générique de complexité polynomiale existe pour le calcul de probabilités dans les modèles de Markov cachés, dans le cas où le graphe d'indépendance conditionnelle (parfois appelé *structure*) est triangulé. Il s'agit de l'algorithme *d'arbre de jonction*. Dans ce cas, il existe également un algorithme EM pour l'estimation des paramètres, dont chaque itération est également de complexité polynomiale. Enfin, un algorithme du MAP de complexité polynomiale pour la restauration des états cachés existe également.

Les algorithmes génériques cités dans Smyth, Heckerman et Jordan, 1997 [110], s'ils permettent en théorie de résoudre tous les problèmes d'inférence de manière efficace, ont cependant quelques inconvénients. D'une part, ils n'effectuent pas leurs calculs à partir de probabilités (conditionnelles ou non) mais à partir de potentiels de cliques, ce qui a pour premier inconvénient de ne pas prendre en compte les paramètres naturels du modèle, qui sont les probabilités de transition. Du coup, il est difficile d'interpréter les calculs intermédiaires de l'algorithme et, dans le cas où plusieurs calculs de probabilités consécutifs sont nécessaires, de réutiliser les calculs déjà effectués. Il est également extrêmement délicat d'utiliser ces formules pour faire du calcul analytique (et non numérique) afin de déterminer, par exemple, l'espérance de variables aléatoires en fonction de paramètres. Un problème également handicapant en pratique est lié au principe de l'algorithme d'arbre de jonction, qui décompose la loi jointe des variables observées. Ceci cause des instabilités numériques dès que le nombre de variables aléatoires du modèle est modérément grand.

Notons que certains modèles de Markov cachés peuvent être assimilés à des modèles à espace d'états (voir Künsch, 2001 [74] dans le contexte des chaînes de Markov cachées). C'est le cas, en particulier, si la structure est orientée et que chaque variable aléatoire observée admet un unique prédécesseur. Pour ces modèles de Markov cachés, on bénéficie alors de la méthodologie et des algorithmes récursifs de calcul de probabilités développés de manière générale pour les modèles à espace d'états – principalement les algorithmes de filtrage et de lissage qui, par nature, ne sont pas assujettis aux limitations numériques évoquées ci-dessus.

Sélection d'un modèle de Markov caché

Les techniques d'estimation de paramètres mentionnées ci-dessus permettent de déterminer un modèle à utiliser pour l'inférence statistique dans le cas où la structure du modèle est fixée, dont, entre autres, le graphe d'indépendance conditionnelle, le nombre de valeurs possibles pour les états cachés et la famille des lois conditionnelles des variables observées sachant les états cachés (ou *lois d'émission*). Dans la plupart des situations pratiques, cette structure est a priori inconnue, même si en principe le modélisateur va envisager plusieurs structures possibles. Il est alors confronté à un problème de sélection de modèles : à partir de quelle(s) structure(s) va-t-il fonder l'inférence statistique ? Les critères et techniques de sélection dépendent a priori de l'usage qu'on veut faire du modèle mais on peut énoncer quelques principes généraux. Lorsqu'il s'agit d'estimer la loi jointe des variables aléatoires à partir de réalisations de ces variables, l'usage de critères de choix de modèles basés sur la divergence de Kullback-Leibler est justifié et conduit à des critères de vraisemblance pénalisée (comme dans Burnham et Anderson, 1998 [17]) ou de validation croisée (introduite comme outil de validation et de comparaison de modèles dans Stone, 1974 [113]). Puisque nous nous intéressons à des modèles pouvant être utilisés pour la classification, le recours à des méthodes de choix de modèles basés sur une pénalisation de la vraisemblance complétée, comme dans Biernacki, Celeux et Govaert, 2001 [11], est également justifié.

La présence de variables cachées complique le problème de sélection de modèles lorsque le nombre d'états cachés K du modèle est inconnu. En effet, la quasi-totalité des critères de choix de modèles se base sur l'hypothèse que les données disponibles sont des réalisations d'un modèle de Markov caché avec $K = K_0$. Dans le cas où des modèles avec $K_0 < K$ sont dans l'ensemble des modèles candidats, l'hypothèse de normalité asymptotique de l'estimateur de maximum de vraisemblance, justifiant le critère de Schwarz [107] ou les tests du khi deux basés sur le logarithme du rapport de vraisemblance, n'est pas vérifiée. Des détails sur les conséquences de cette difficulté sur la sélection de modèles de mélanges indépendants par des tests sont rapportés dans Aitkin et Rubin, 1985 [2]. L'article de Geiger *et al.*, 2001 [58] propose une approche basée sur la topologie pour expliquer les origines de la difficulté de la sélection de modèles de Markov cachés, principalement par le critère de Schwarz. Les auteurs définissent une notion de régularité pour des classes de modèles graphiques et constatent que la présence d'états cachés dégrade la régularité des modèles, rendant par là-même plus difficile la preuve de la consistance de ce critère. D'autre part, d'après Gassiat, 2002 [52], certains critères de vraisemblance marginale pénalisée permettent, sous des conditions assez fortes (entre autres, la stationnarité et la β-mélangeance), d'obtenir une estimation consistante du nombre d'états cachés.

Cependant, l'hypothèse que la vraie loi des variables aléatoires observées est celle d'un modèle de Markov caché reste un inconvénient dès lors qu'il s'agit de données réelles, vu le manque de réalisme de cette hypothèse. Dans de telles situations, il peut paraître plus adapté de choisir le modèle de Markov caché le plus proche de la vraie loi des variables observées, en un certain sens (habituellement celui de la divergence de Kullback-Leibler).

Objectifs de la thèse

Le premier objectif de ce travail est de définir une classe de modèles de Markov cachés permettant la modélisation de processus observés à valeurs discrètes ou continues, admettant une paramétrisation naturelle et aisément interprétable, et pour lesquels existent des algorithmes d'inférence efficaces.

Notre but est ensuite :
- de développer ces algorithmes d'inférence sous la contrainte qu'ils tiennent compte de la paramétrisation choisie, qu'ils se basent sur des probabilités conditionnelles et non sur des potentiels de cliques, qu'ils explicitent le lien fonctionnel entre les paramètres et les probabilités conditionnelles utilisées, qu'ils soient stables numériquement et qu'ils ne répètent pas de calculs inutilement ;

– d'examiner l'intérêt, d'un point de vue théorique et pratique, de techniques de sélection de modèles déjà utilisées dans le cadre des modèles de Markov cachées, voire uniquement dans d'autres contextes. En particulier, nous examinons la justification théorique de ces critères et leur domaine de validité. Entre autres aspects, nous vérifions si le fait que, parmi les modèles en compétition, figure celui ayant généré les données, fait partie des hypothèses effectuées lors de l'élaboration des critères. Nous proposons également des techniques de choix de modèles basées sur la validation croisée (voir Celeux et Durand, 2001 [22]). Il s'agit d'une méthodologie classique mais pas encore mise en œuvre dans le cas des modèles de Markov cachés, où la structure de dépendance en complique l'implémentation.

Plan de la thèse

Le chapitre 1 est dédié à la définition d'une famille de modèles de Markov cachés répondant aux exigences énoncées ci-dessus. Dans un premier temps, nous rappelons les concepts propres aux graphes et aux modèles probabilistes graphiques, avec une brève présentation de l'algorithme d'arbre de jonction et de ses variantes. Nous présentons ensuite la famille de modèles sur laquelle nous travaillons dans les chapitres suivants, en décrivant ses propriétés les plus immédiates.

Les problèmes d'inférence statistique dans ces modèles sont traités au chapitre 2. Après avoir évoqué les problèmes d'identifiabilité, nous présentons des techniques d'estimation du maximum de vraisemblance des paramètres par l'algorithme EM et ses variantes. L'étape E met en évidence la nécessité de calculer la loi jointe des états cachés de chaque clique, conditionnellement aux données observées. L'une des contributions de cette thèse est l'élaboration d'algorithmes efficaces et interprétables, permettant l'implémentation de l'étape E, et plus généralement le calcul de probabilités, dans ces modèles de Markov cachés. Nous présentons une première procédure, analogue à *l'algorithme avant-arrière* développé pour les chaînes de Markov cachées. Elle s'appuie fortement sur l'étude des propriétés de la famille de modèles considérée, en particulier des caractéristiques de leur graphe d'indépendance conditionnelle. Ces propriétés sont établies au préalable. Nos algorithmes se basent sur la décomposition de probabilités jointes, ce qui est cause de limitations numériques, quand le nombre de données observées augmente. Nous montrons en quoi l'usage, dans les algorithmes avant-arrière, de probabilités analogues à celles de lissage dans les modèles à espace d'états, résout ces problèmes d'instabilité numérique tout en conservant la complexité polynomiale des algorithmes. Enfin, nous rappelons l'intérêt d'un algorithme de restauration globale des états cachés, puis en proposons un de complexité égale à celui de l'algorithme arrière (ou avant-arrière). Nous en donnons également une interprétation probabiliste. Ce chapitre est largement illustré par l'application de ces algorithmes au cas des chaînes et des arbres de Markov cachés. En particulier, nous présentons la méthode de Durand, Gonçalvès et Guédon, 2002 [46] pour la détection de changements de régularité locale de processus. Cette méthode est basée sur la modélisation par arbres de Markov cachés de la loi des coefficients de la transformée en ondelettes. Nous présentons, dans cette application, les étapes de modélisation, d'estimation des paramètres, de restauration des états cachés et d'interprétation du modèle.

Dans le chapitre 3, nous abordons le problème de la sélection de modèles en rappelant des techniques usuelles, leur fondement et leur validité. Ainsi, nous étudions le choix de modèles basé sur les tests, le critère d'information bayésienne BIC, le critère d'Akaike AIC (tous deux de type vraisemblance pénalisée), les critères de vraisemblance marginale pénalisée de Gassiat, 2002 [52] et la validation croisée multiple (en particulier le demi-échantillonnage). Nous nous plaçons essentiellement dans le cadre du choix du nombre d'états cachés. La mise en œuvre de la validation croisée dans un cadre où les variables aléatoires sont dépendantes les unes des autres est délicate, tout particulièrement quand elles sont à valeurs continues. C'est pourquoi nous présentons des algorithmes de calcul de probabilités et d'estimation des paramètres, dans le cas d'observations supprimées de manière déterministe ou totalement aléatoire. Nous étudions également l'adaptation au cas de mélanges non indépendants du critère ICL, pénalisant la vraisemblance complétée, adapté à la recherche du nombre de classes à des fins de classification automatique. Puis nous comparons le comportement de ces critères sur des données simulées et réelles. Ces dernières sont les durées inter-défaillances de logiciels, le processus des défaillances et des corrections étant modélisé par une chaîne de Markov cachée (voir Durand et Gaudoin, 2002 [45]).

Dans le chapitre 4, nous présentons une application complète des chaînes de Markov cachées pour l'étude de courbes de consommation électrique fournies par EDF. Nous présentons les différentes étapes de cette étude : de l'élaboration du modèle à sa validation, en passant par la sélection. L'un des buts de l'étude est, pour EDF, d'arriver à estimer la part de la consommation globale due aux différents appareils ménagers. Ainsi, l'intérêt de cette application est la prise en compte de cet objectif lors de la sélection de modèle.

Chapitre 1

Modèles de Markov cachés

1.1 Introduction

Dans ce chapitre, nous rappelons la définition des modèles probabilistes graphiques orientés et non orientés ainsi que la sémantique de ces graphes. Ce formalisme nous permet ensuite de définir les modèles de Markov cachés comme des modèles graphiques comportant des variables aléatoires discrètes non observées. Comme illustration, nous rappelons la définition des modèles de Markov cachés les plus célèbres, en nous appuyant sur leur graphe d'indépendance conditionnelle et leurs paramètres. Nous présentons les algorithmes classiques de calcul numérique de probabilités dans ces modèles, tout en insistant sur leurs limites en termes d'interprétation, de stabilité numérique et celles concernant les possibilités de calcul analytique de probabilités, limites qui rendent plus difficile l'inférence dans ces modèles. Enfin, nous caractérisons une classe de modèles de Markov cachés qui satisfait systématiquement à nos objectifs, à savoir

- telle que la donnée de la structure d'indépendance conditionnelle suffise à obtenir, sans aucun calcul théorique, une paramétrisation *naturelle* et *interprétable* (en particulier d'un point de vue probabiliste) ;
- l'existence d'algorithmes d'inférence tenant compte de cette paramétrisation, se basant sur des probabilités conditionnelles et non sur des potentiels de clique (pour des raisons d'interprétation et pour permettre le calcul analytique) ;
- l'explicitation par ces algorithmes du lien fonctionnel entre les paramètres et les probabilités conditionnelles utilisées ;
- la stabilité de ces algorithmes ainsi que la détection et la suppression des calculs répétés inutilement.

Les algorithmes répondant à ces objectifs sont présentés au chapitre 2.

1.2 Modèles graphiques

Nous rappelons dans cette section comment les graphes peuvent constituer un outil pour définir des modèles probabilistes. De tels modèles, nommés *modèles graphiques*, offrent des avantages dans le domaine :

- de la description du modèle. Les modèles graphiques constituent un moyen naturel

et intuitif pour définir ou représenter les relations d'indépendance conditionnelle entre variables aléatoires, à travers leur *structure*,
- des algorithmes de calcul des probabilités. Étant donné un modèle graphique entièrement spécifié, il est possible de développer des algorithmes efficaces pour calculer des quantités d'intérêt, telles que la loi jointe d'un sous-ensemble de variables aléatoires du modèle. Cependant, les algorithmes courants ont des inconvénients que nous détaillerons par la suite (en particulier leur instabilité numérique et leur interprétation probabiliste ardue) et qui limitent leur utilisation.

Définitions et notations

Nous supposons dans tout ce qui suit que nous voulons définir un modèle pour des variables aléatoires définies sur un espace probabilisé commun $(\Omega, \mathcal{F}, P)$. Nous utilisons des lettres majuscules pour désigner des variables aléatoires et des lettres minuscules pour désigner des réalisations de ces variables aléatoires. Une variable aléatoire X définie sur Ω et à valeurs dans $\mathcal{X}$ induit un espace probabilisé $(\mathcal{X}, \mathcal{F}_{\mathcal{X}}, P_X)$ où $\mathcal{F}_{\mathcal{X}}$ désigne la tribu engendrée par les ouverts de $\mathcal{X}$ pour une certaine topologie et P_X désigne la loi de X. Nous utiliserons parfois la notation $\mathcal{X}_X$ pour désigner l'ensemble des valeurs de X. Nous supposons que toutes les lois admettent une densité p_X par rapport à une mesure σ-finie μ. Lorsque la mesure μ est la mesure de Lebesgue, nous notons f_X la densité de P_X. Ce cas, ainsi que celui où μ est la mesure de comptage, constituent deux applications d'un intérêt particulier pour les modèles graphiques. En général, lorsque le contexte ne permet pas d'ambiguïté, nous utiliserons la notation $P(X = x)$ pour désigner indifféremment $P_X(x)$ et $f_X(x)$, afin de simplifier les notations dans les modèles mettant en jeu à la fois des variables aléatoires à valeurs discrètes et continues. Un processus aléatoire à temps discret est désigné par $\boldsymbol{X} = (X_u)_{u \in \mathcal{U}}$ où $\mathcal{U}$ est une partie de $\mathbb{N}$. Si V est une partie de $\mathcal{U}$, on note $\boldsymbol{X}_V = (X_u)_{u \in V}$ et $\mathcal{X}_V$ l'ensemble des valeurs de $\boldsymbol{X}_V$.

Graphes d'indépendance conditionnelle non orientés

D'après la terminologie de Smyth, Heckerman et Jordan, 1997 [110], la *structure* d'un modèle graphique $P_{\boldsymbol{X}}$ est un graphe représentant les relations d'indépendance conditionnelle entre les variables de $\boldsymbol{X}$. Étant donné trois sous-ensembles deux à deux disjoints B, C et D de $\mathcal{U}$, $\boldsymbol{X}_D$ et $\boldsymbol{X}_B$ sont dits conditionnellement indépendants sachant $\boldsymbol{X}_C$ si et seulement si

$$\begin{aligned} &\forall(\boldsymbol{x}_D, \boldsymbol{x}_B, \boldsymbol{x}_C),\ P(\boldsymbol{X}_D = \boldsymbol{x}_D \cap \boldsymbol{X}_B = \boldsymbol{x}_B | \boldsymbol{X}_C = \boldsymbol{x}_C) \\ &\quad = P(\boldsymbol{X}_D = \boldsymbol{x}_D | \boldsymbol{X}_C = \boldsymbol{x}_C) P(\boldsymbol{X}_B = \boldsymbol{x}_B | \boldsymbol{X}_C = \boldsymbol{x}_C). \end{aligned}$$

On note $\boldsymbol{X}_D \perp\!\!\!\perp \boldsymbol{X}_B | \boldsymbol{X}_C \, [P_{\boldsymbol{X}}]$ l'indépendance conditionnelle, au sens de la loi $P_{\boldsymbol{X}}$, de $\boldsymbol{X}_D$ et $\boldsymbol{X}_B$ sachant $\boldsymbol{X}_C$. Quand la loi de référence est évidente, nous notons simplement $\boldsymbol{X}_D \perp\!\!\!\perp \boldsymbol{X}_B | \boldsymbol{X}_C$. Soit $\mathcal{G} = (\mathcal{U}, \mathcal{E})$ un graphe non orienté dont les sommets sont les indices du processus $\boldsymbol{X}$ et l'ensemble des arêtes $\mathcal{E}$ une partie de $\mathcal{P}_2(\mathcal{U})$, l'ensemble des parties de cardinal 2 de $\mathcal{U}$. Nous identifierons parfois les sommets du graphe et les variables aléatoires associées, autrement dit nous considérerons parfois le graphe $(\boldsymbol{X}, \mathcal{E})$ à la place de $(\mathcal{U}, \mathcal{E})$.

Soient B, C et D trois parties deux à deux disjointes de $\mathcal{U}$. On dit que C sépare B de D dans $\mathcal{G}$ si et seulement si pour tout sommet dans D, tout sommet dans B et toute chaîne entre ces deux sommets, cette chaîne passe par un sommet de C. Ainsi, dans la figure 1.1, $\{2, 3\}$ sépare $\{1\}$ de $\{4, 5\}$. On dit que $\mathcal{G}$ est un graphe d'indépendance conditionnelle (parfait) pour $P_{\boldsymbol{X}}$ si et seulement si pour tous les sous-ensembles disjoints B, C et D de $\mathcal{U}$, $\boldsymbol{X}_D \perp\!\!\!\perp \boldsymbol{X}_B | \boldsymbol{X}_C \Leftrightarrow C$ sépare B de D dans $\mathcal{G}$. Dans ce cas, nous dirons que $P_{\boldsymbol{X}}$ est un modèle graphique. Ainsi, dans la figure 1.1, si $\mathcal{G}$ est un graphe d'indépendance conditionnelle parfait pour $P_{\boldsymbol{X}}$, alors $\{X_1\}$ est indépendant de $\{X_4, X_5\}$ sachant $\{X_2, X_3\}$.

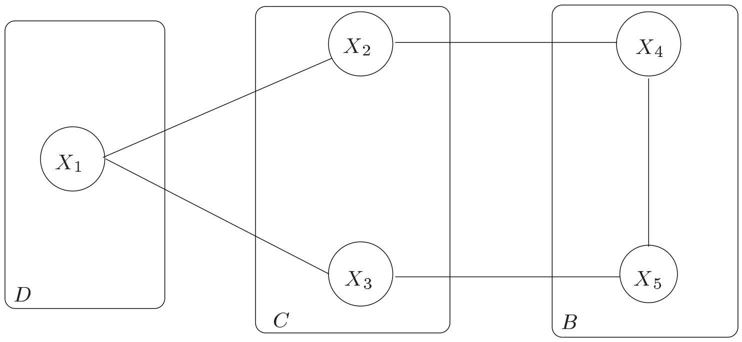


FIG. 1.1 – *Un graphe d'indépendance conditionnelle non orienté.* $\{X_1\}$ *est indépendant de* $\{X_4, X_5\}$ *sachant* $\{X_2, X_3\}$.

Les graphes d'indépendance conditionnelle non orientés sont plutôt utilisés pour modéliser des relations symétriques d'interaction entre variables aléatoires, comme des corrélations. Par exemple, nous verrons ci-dessous que de tels graphes sont adaptés à la modélisation des dépendances entre les pixels d'une image.

Graphes d'indépendance conditionnelle orientés acycliques

D'une manière similaire, les graphes orientés acycliques peuvent être utilisés pour définir des modèles graphiques. L'ensemble $\mathcal{E}$ des arcs est alors une partie de $\mathcal{U} \times \mathcal{U}$. Notons que les arcs en boucle du type (u, u) où $u \in \mathcal{U}$ n'ont pas de signification dans les graphes d'indépendance conditionnelle non orientés - ceci reviendrait à vouloir modéliser la dépendance entre X_u et X_u. Nous supposerons donc que $\mathcal{E}$ ne contient pas de telles boucles. L'appellation de *graphe orienté acyclique* est un peu inadaptée au sens où ces graphes sont caractérisés par l'absence de *circuit* (au sens des graphes orientés). En revanche, les graphes orientés acycliques peuvent admettre des cycles, cette notion étant propre aux graphes non orientés. La notion de séparation a une définition différente dans les graphes orientés et les graphes non orientés. Il est commode d'associer à $\mathcal{G} = (\mathcal{U}, \mathcal{E})$ son *graphe moral* $(\mathcal{U}, \mathcal{E}^M)$ où $\mathcal{E}^M$ est obtenu à partir de $\mathcal{E}$ en plaçant, à chaque nœud, des arêtes (non orientées) entre ses parents non adjacents, puis en supprimant l'orientation de tous les autres arcs de $\mathcal{E}$. Le terme de *graphe moral* est une allusion au *mariage* des parents ayant un enfant commun. Soit A une partie de $\mathcal{U}$. Si pour tout u dans A, A contient les parents de u, alors l'ensemble A est dit *ancestral*. La propriété de séparation dans un graphe orienté acyclique s'énonce ainsi : C sépare B de D dans $\mathcal{G}$ si et seulement si C sépare B de D dans le graphe moral engendré par le plus petit ensemble ancestral contenant B, C et D (on notera $\mathrm{An}(A)$le plus petit ensemble ancestral contenant A). On se ramène ainsi à la définition de la séparation dans un graphe non orienté. De même que précédemment, $\mathcal{G}$ est un graphe d'indépendance conditionnelle (parfait) pour $P_{\boldsymbol{X}}$ si et seulement si pour tous les sous-ensembles disjoints B, C et D de $\mathcal{U}$, $\boldsymbol{X}_D \perp\!\!\!\perp \boldsymbol{X}_B | \boldsymbol{X}_C \Leftrightarrow C$ sépare B de D dans $\mathcal{G}$ (au sens des graphes orientés). Dans ce cas également, nous dirons que $P_{\boldsymbol{X}}$ est un modèle graphique. Ainsi, dans la figure 1.2, si $\mathcal{G}$ est un graphe d'indépendance conditionnelle parfait pour $P_{\boldsymbol{X}}$, alors $\{X_5\}$ est indépendant de $\{X_4, X_1\}$ sachant $\{X_2, X_3\}$. Il a été montré (Lauritzen *et al.*, 1990 [77]) que cette définition d'un graphe

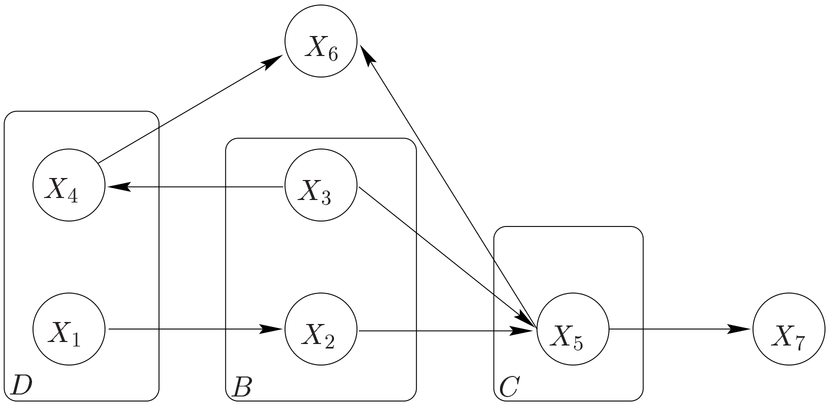


FIG. 1.2 – *Un graphe d'indépendance conditionnelle orienté acyclique.* $\{X_5\}$ *est indépendant de* $\{X_4, X_1\}$ *sachant* $\{X_2, X_3\}$.

d'indépendance conditionnelle est équivalente à la définition qui suit, plus intuitive quant à la propriété de séparation : conditionnellement à ses parents dans le graphe $\mathcal{G}$, une variable aléatoire est indépendante de toutes les variables autres que ses descendantes.

Les graphes d'indépendance conditionnelle orientés acycliques sont utilisés de manière privilégiée pour modéliser des relations asymétriques d'interaction ou d'influence entre variables aléatoires, ce que Lauritzen, 1999 [76] appelle la *causalité*. On peut représenter ainsi des hiérarchies entre variables aléatoires, comme la notion de succession, utile pour les séries chronologiques par exemple. L'ensemble des modèles graphiques non orientés et l'ensemble des modèles graphiques orientés acycliques ne sont pas égaux : Smyth, Heckerman et Jordan, 1997 [110] donnent un exemple de modèle graphique non orienté qui n'admet pas de modèle graphique orienté acyclique équivalent (en termes de relations d'indépendance conditionnelle) et un exemple de modèle graphique orienté acyclique qui n'admet pas de modèle graphique non orienté équivalent.

Définir une loi de probabilité à partir d'un graphe

Il n'est pas toujours possible d'associer un graphe d'indépendance conditionnelle parfait à une loi $P_{\boldsymbol{X}}$ donnée. Smyth, Heckerman et Jordan donnent des exemples de lois de probabilité qui ne sont pas des modèles graphiques, dans le cas de graphes non orientés aussi bien que dans celui de graphes orientés acycliques. Réciproquement, il est possible d'associer une et en général plusieurs lois de probabilité à un graphe d'indépendance conditionnelle. Un modèle graphique est défini par sa *structure* et par des *paramètres*, qui sont des spécifications numériques telles que $P_{\boldsymbol{X}}(F)$ soit définie de manière unique pour tout F dans $\mathcal{F}_{\mathcal{X}_{\mathcal{U}}}$. La loi $P_{\boldsymbol{X}}$ appartient alors une famille paramétrique donnée. Nous désignerons par Λ l'ensemble des paramètres. Quand il y a lieu d'expliciter la dépendance de $P_{\boldsymbol{X}}$ vis-à-vis des paramètres λ, nous noterons $P_\lambda(\boldsymbol{X} \in F)$ pour tout F dans $\mathcal{F}_{\mathcal{X}_{\mathcal{U}}}$. En général, $\Lambda \subset I\!\!R^d$.

1.3 Modèles de Markov cachés

Après avoir défini les modèles de Markov cachés, nous montrons l'appartenance à cette famille de plusieurs modèles classiques à structure cachée : modèles de mélanges indépendants puis chaînes, arbres et champs de Markov cachés.

Définition

Nous définissons les modèles de Markov cachés comme des modèles graphiques comportant des variables observées, constituant un processus noté $\boldsymbol{Y}$, et des variables cachées (processus $\boldsymbol{S}$). Il ne s'agit pas d'un processus ayant une existence réelle mais dont l'observation serait rendue impossible, pour des raisons physiques par exemple. Ces variables cachées peuvent plutôt être vues comme des variables indicatrices d'un type de régime pour les variables observées. Pour des raisons d'interprétation et d'identifiabilité des modèles, ces variables cachées sont ici supposées discrètes. Dans le graphe d'indépendance conditionnelle, elles contribuent à définir les dépendances entre variables observées, permettant ainsi d'obtenir des modèles plus flexibles. En principe, les variables cachées sont à valeurs dans un ensemble fini. Dans de nombreux cas, les dépendances entre variables observées sont entièrement définies par les dépendances entre variables cachées : les variables composant le processus $\boldsymbol{Y}$ sont indépendantes conditionnellement à $\boldsymbol{S} = \boldsymbol{s}$. Autrement dit, par définition du graphe d'indépendance conditionnelle, les variables $\boldsymbol{Y}$ sont déconnectées dans le graphe quand on supprime les variables $\boldsymbol{S}$. Nous montrerons que, dans ce cas et sous certaines hypothèses, la loi marginale des variables observées est une loi de mélange fini. Or les modèles de mélanges finis indépendants ont une grande importance en classification automatique (Wolfe, 1970 [121]). En effet, dans ce cas, une variable cachée est associée à chaque variable observée et permet de représenter sa classe. La classe k est alors définie par l'ensemble des variables observées dont la modalité de la variable cachée est égale à k. Ces variables observées ont en fait même loi conditionnelle.

De même, dans le cas général, les variables cachées peuvent encore être interprétées comme des indicatrices de la classe des variables observées, où la notion de classe est ici dynamique, au sens où elle désigne un type de régime possible du processus observé. Rappelons que le processus observé est en général indexé par les nœuds d'un graphe (nous ne parlons pas encore du graphe d'indépendance conditionnelle mais simplement des indices du processus observé) qui traduit l'organisation spatiale où temporelle des données : graphe linéaire pour un processus temporel, grille pour un champ aléatoire, etc. Les modèles de Markov cachés peuvent alors être utilisés pour modéliser l'existence de zones homogènes dans ce graphe. Le graphe d'indépendance conditionnelle est obtenu en ajoutant des variables aléatoires cachées (également appelées *états cachés*) à l'ensemble des variables observées. Les zones homogènes sont alors formées par des variables observées d'indices contigus et de même classe, c'est-à-dire avec une variable cachée de même modalité. La façon dont les classes se propagent entre variables voisines au fil de l'indexation est déterminée par le graphe d'indépendance conditionnelle et la valeur des paramètres, comme nous le verrons par la suite. Le modélisateur peut ainsi, en définissant la structure d'indépendance conditionnelle du modèle de Markov caché, traduire graphiquement ses connaissances a priori sur le comportement des zones homogènes. Enfin, même si les états cachés ont une signification inconnue a priori, l'analyse des variables observées à l'aide du modèle, en particulier par la restauration des états cachés, peut parfois fournir une interprétation concrète du processus caché (voir par exemple la section 3.8.2 et le chapitre 4).

Nous présentons ci-dessous des exemples classiques de modèles de Markov cachés en les définissant par leur graphe d'indépendance conditionnelle et leurs paramètres. Dans chaque cas, il s'agit de modèles de Markov cachés obtenus en calquant la structure de dépendance du processus caché sur le graphe des indices (respectivement un graphe déconnecté, linéaire, arborescent et une grille). Les variables observées sont conditionnellement indépendantes sachant le processus caché.

Modèle de mélange indépendant

Le modèle de mélange indépendant est défini par le graphe d'indépendance conditionnelle de la figure 1.3. Nous verrons ultérieurement que le modèle obtenu en supprimant l'orientation des arcs dans ce graphe est identique. Dans les modèles de Markov cachés, nous adoptons la convention graphique qui suit : les variables aléatoires à valeurs discrètes sont représentées par des carrés et les variables aléatoires à valeurs continues par des cercles. Notons que les variables aléatoires cachées sont toujours discrètes dans notre contexte, alors que les variables observées sont indifféremment discrètes ou continues. Dans le modèle de la figure 1.3, les variables observées sont mutuellement indépendantes ; les variables cachées également (le graphe comporte autant de composantes connexes que de variables observées). Si K est le nombre d'états cachés, le modèle est défini par les paramètres :
- $(\pi_k)_{1\leq k\leq K} = (P(S_u = k))_{1\leq k\leq K}$;
- $(\theta_1, \ldots, \theta_K)$, tels que $P(Y_u = y|S_u = k) = P_{\theta_k}(y)$, où $\theta \in \Theta$ est l'index d'une famille paramétrique de lois de probabilités. Nous nommerons ces paramètres les *paramètres d'émission*, car les lois conditionnelles des variables observées sachant les états cachés sont en général appelées les *lois d'émission* ou *d'observation*.

Nous proposons sur la figure 1.4 des réalisations de variables suivant un tel modèle, pour la famille des lois gaussiennes dans $\mathbb{R}^2$, et les paramètres suivants :

$$\begin{bmatrix} \pi_1 & \pi_2 \end{bmatrix} = \begin{bmatrix} 0,7 & 0,3 \end{bmatrix} \quad \mu_1 = \begin{bmatrix} -3 \\ 5 \end{bmatrix} \Sigma_1^2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \mu_2 = \begin{bmatrix} 3 \\ 5 \end{bmatrix} \Sigma_2^2 = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}.$$

Cette figure fait apparaître les deux classes associées aux deux composantes.

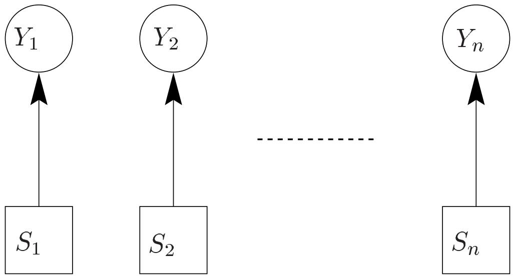


FIG. 1.3 – *La structure des modèles de mélange indépendants.*

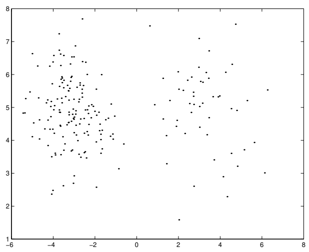


FIG. 1.4 – *Réalisation d'un modèle de mélange indépendant.*

Chaînes de Markov cachées

Les modèles de chaîne de Markov cachée sont définis par le graphe d'indépendance conditionnelle de la figure 1.5. Nous verrons ultérieurement que le modèle obtenu en supprimant l'orientation des arcs dans ce graphe est identique. Le processus caché est une chaîne de Markov supposée *homogène* ; autrement dit, nous faisons l'hypothèse que $P(S_{t+1} = j|S_t = i)$ ne dépend pas de t. Si K est le nombre d'états cachés, le modèle est défini par les paramètres :

- $(\pi_k)_{1\leq k\leq K} = (P(S_1 = k))_{1\leq k\leq K}$;
- $p_{ij} = P(S_{t+1} = j|S_t = i)$. La matrice des $(p_{ij})_{(i,j)\in\{1,\ldots,K\}^2}$ est désignée par P ;
- $(\theta_1, \ldots, \theta_K)$ tels que $P(Y_t = y|S_t = k) = P_{\theta_k}(y)$ (paramètres d'émission).

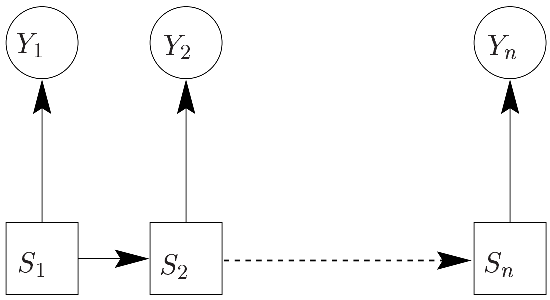


FIG. 1.5 – *La structure des modèles de chaîne de Markov cachée.*

Nous proposons en figure 1.6 une réalisation de chaîne de Markov cachée pour la famille des lois gaussiennes et les paramètres suivants :

$$\pi_1 = 1 \quad \pi_2 = 0 \quad \mu_1 = -3 \quad \Sigma_1^2 = 1 \quad \mu_2 = 3 \quad \Sigma_2^2 = 2 \quad P = \begin{bmatrix} 0,98 & 0,02 \\ 0 & 1 \end{bmatrix}.$$

Cette figure fait apparaître deux périodes homogènes, correspondant aux deux états cachés.

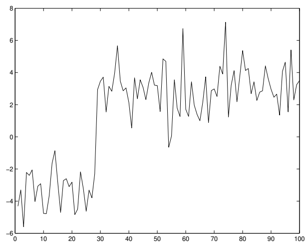


FIG. 1.6 – *Réalisation d'un modèle de chaîne de Markov cachée.*

Arbres de Markov cachés

Les modèles d'arbres de Markov cachés sont adaptés aux processus indexés de manière arborescente. On suppose qu'il existe une racine Y_1 et que toutes les autres variables aléatoires observées Y_u admettent dans l'arbre un parent désigné par $Y_{\rho(u)}$. Les arbres de Markov cachés ont pour graphe d'indépendance conditionnelle celui représenté figure 1.7. Nous verrons ultérieurement que le modèle obtenu en supprimant l'orientation des arcs dans ce graphe est identique. Les notations utilisées pour indexer le processus observé s'étendent au processus caché. Ce dernier est un arbre de Markov supposé homogène, autrement dit, $P(S_u = j | S_{\rho(u)} = i)$ ne dépend pas de u.

Si K est le nombre d'états cachés, le modèle est défini par les paramètres :
- $(\pi_k)_{1 \leq k \leq K} = (P(S_1 = k))_{1 \leq k \leq K}$;
- $p_{ij} = P(S_u = j | S_{\rho(u)} = i)$ pour $1 \leq i \leq K$ et $1 \leq j \leq K$;
- $(\theta_1, \ldots, \theta_K)$ tels que $P(Y_u = y | S_u = k) = P_{\theta_k}(y)$ (paramètres d'émission).

Nous proposons en figure 1.8 une réalisation d'un arbre de Markov caché, pour la famille des lois gaussiennes et les paramètres suivants :

$$\pi_1 = 1 \quad \pi_2 = 0 \quad \mu_1 = -3 \quad \Sigma_1^2 = 1 \quad \mu_2 = 3 \quad \Sigma_2^2 = 2 \quad P = \begin{bmatrix} 0,9 & 0,1 \\ 0 & 1 \end{bmatrix}.$$

Cette figure a un nombre de lignes égal à la profondeur de l'arbre. Puisqu'il s'agit d'un arbre binaire, chaque valeur de l'arbre est représentée sur un rectangle dont la largeur est divisée par deux chaque fois que la profondeur augmente d'une unité. La valeur de la racine est représenté par le rectangle au sommet du graphe. Le niveau de gris représente la valeur de Y_u : du gris foncé (état caché 1) au gris clair (état caché 2) suivant que sa valeur est très négative à très positive. La figure traduit le fait que la valeur d'une variable cachée parente a tendance à être héritée par ses descendants.

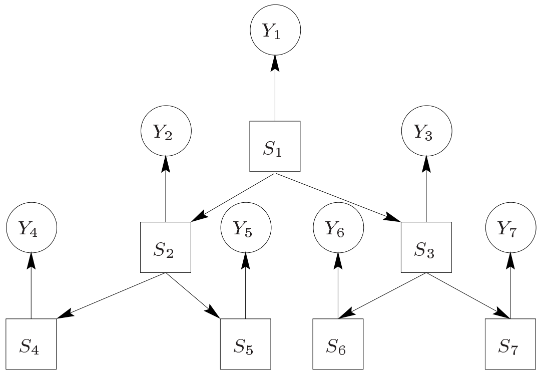


FIG. 1.7 – *La structure des modèles d'arbres de Markov cachés.*

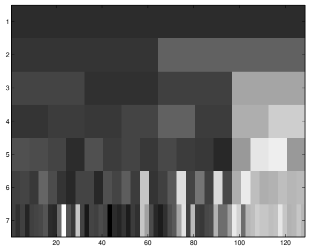


FIG. 1.8 – *Réalisation d'un modèle d'arbre de Markov caché. La valeur de chaque sommet de l'arbre est représentée sur un rectangle dont la largeur est divisée par deux chaque fois que la profondeur augmente d'une unité. La valeur de la racine est représenté par le rectangle au sommet du graphe. Le niveau de gris représente la valeur de Y_u : du gris foncé (état caché 1) au gris clair (état caché 2) suivant que sa valeur est très négative à très positive.*

Champs de Markov cachés

Les modèles de champs de Markov cachés sont définis par le graphe d'indépendance conditionnelle de la figure 1.9. Le processus caché est un champ de Markov. Par exemple, $\boldsymbol{S}$ peut suivre le modèle de Potts (dont certaines propriétés sont étudiées dans Chandler, 1987 [26]) :

$$P(\boldsymbol{S} = \boldsymbol{s}) = W_\beta^{-1} \exp(-\beta \sum_{u \sim v} s_u {}^t s_v)$$

où $s_u \in \{(0\ 1), (1\ 0)\}$, le vecteur ${}^t s_u$ désignant le vecteur transposé de s_u, la relation $\sim$ désignant la relation de voisinage et W_β étant une constante de normalisation. Notons que les modèles de chaîne et d'arbre de Markov cachés vus ci-dessus peuvent à la fois être considérés comme des modèles graphiques orientés ou non orientés. Cela n'est pas le cas des champs de Markov cachés qui ne peuvent être vus comme des modèles orientés.

Dans le cas plus général d'un nombre d'états cachés $K \geq 2$, le modèle s'étend en prenant pour ensemble des valeurs de S_v la base canonique $(e_1, \ldots, e_K)$ de $I\!R^K$. Le modèle est alors défini par les paramètres :

- β, paramètre du modèle de Potts ;
- les paramètres d'émission $(\theta_1, \ldots, \theta_K)$ tels que $P(Y_u = y | S_u = e_k) = P_{\theta_k}(y)$.

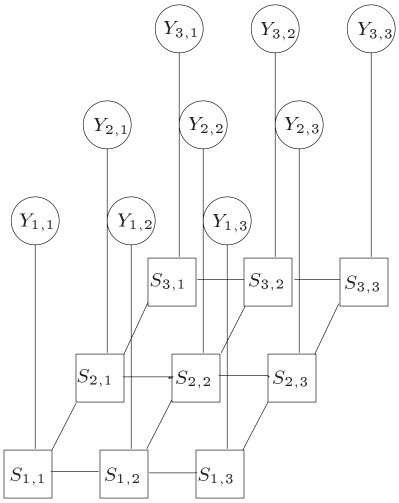


FIG. 1.9 – *La structure des modèles de champ de Markov cachés.*

Nous proposons en figure 1.10 une réalisation de champ de Markov caché, pour la famille des lois gaussiennes et les paramètres suivants :

$$\beta = 0,7 \quad \mu_1 = -3 \quad \Sigma_1^2 = 1 \quad \mu_2 = 3 \quad \Sigma_2^2 = 2.$$

Les valeurs obtenues pour le champ ont ensuite été réduites à 256 niveaux représentés sur

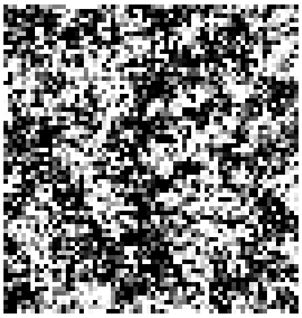


FIG. 1.10 – *Réalisation d'un modèle de champ de Markov caché. La valeur des observations est représentée par l'intensité du niveau de gris : gris foncé pour des valeurs très négatives (état caché e_1) et gris clair pour des valeurs très positives (état caché e_2).*

la figure par l'intensité d'un niveau de gris : gris foncé pour des valeurs très négatives (état caché e_1) et gris clair pour des valeurs très positives (état caché e_2). On remarque que sur la figure, les pixels sombres ont tendance à se regrouper, de même que les pixels clairs. Ceci traduit le fait que des variables cachées voisines ont tendance à prendre des valeurs égales, pour β strictement positif. Si β est strictement négatif, des variables cachées voisines vont avoir tendance à prendre des valeurs opposées. Le cas $\beta = 0$ correspond à l'indépendance des variables cachées (et l'équiprobabilité des états cachés).

1.4 Calcul des probabilités dans les modèles de Markov cachés

La section précédente a mis en évidence un certain nombre de points communs entre les modèles de mélanges finis indépendants et les chaînes, arbres et champs de Markov cachés. Il s'agit de modèles où les dépendances existant dans le processus observé $\boldsymbol{Y}$ sont entièrement définies par celles du processus caché $\boldsymbol{S}$, les variables observées étant conditionnellement indépendantes sachant les variables cachées. Par conséquent, les paramètres du modèles sont rattachés soit au processus caché, soit au lien entre le processus caché et le processus observé (paramètres d'émission). La définition du graphe d'indépendance conditionnelle est liée, pour les chaînes, arbres et champs de Markov cachés, à la manière de représenter ou d'indexer le processus observé (respectivement par une partie de $\mathbb{N}$, un arbre, une partie de $\mathbb{N}^2$).

La problématique du calcul des probabilités

Cependant, du point de vue du calcul des probabilités, ces quatre modèles ont des propriétés sensiblement différentes. Intéressons-nous, pour chacun des modèles, au problème du calcul de la vraisemblance d'un paramètre λ, définie pour les modèles à variables cachées par $P_\lambda(\boldsymbol{Y} = \boldsymbol{y})$. Tout d'abord, la plupart des modèles graphiques comportent un nombre de variables aléatoires N non fixé, qui peut croître arbitrairement, comme c'est le cas pour les modèles présentés ci-dessus. Les modèles d'indépendance sont particuliers du point de vue du calcul de la vraisemblance, puisque dans ce cas, il en existe une expression explicite,

$$P_\lambda(\boldsymbol{Y} = \boldsymbol{y}) = \prod_{u=1}^{n} \sum_{i=1}^{K} \pi_i P_{\theta_i}(y_u),$$

qui permet de calculer $P_\lambda(\boldsymbol{Y} = \boldsymbol{y})$ en environ Kn opérations si n représente le nombre de variables observées et K le nombre de valeurs possibles pour les états cachés. Pour les modèles avec des dépendances entre variables cachées, il est en théorie possible de calculer la vraisemblance grâce à la formule des probabilités totales, en conditionnant par $\boldsymbol{S} = \boldsymbol{s}$:

$$P_\lambda(\boldsymbol{Y} = \boldsymbol{y}) = \sum_{\boldsymbol{s} \in \{1,\ldots,K\}^n} P_\lambda(\boldsymbol{Y} = \boldsymbol{y} | \boldsymbol{S} = \boldsymbol{s}) P_\lambda(\boldsymbol{S} = \boldsymbol{s}) \tag{1.1}$$

où $P_\lambda(\boldsymbol{Y} = \boldsymbol{y} | \boldsymbol{S} = \boldsymbol{s})$ et $P_\lambda(\boldsymbol{S} = \boldsymbol{s})$ ont des expressions explicites. L'inconvénient de la formule (1.1) est qu'elle requiert en général au moins K^n opérations, ce qui correspond au nombre de termes de la somme. Une telle complexité de calcul empêche son utilisation. Dans ce qui suit, nous allons étudier l'existence d'algorithmes de complexité polynomiale en n. Dans le cas de modèles graphiques à structure non connexe, la vraisemblance s'écrit comme le produit de probabilités des composantes connexes. Nous pouvons donc nous restreindre à l'étude de graphes connexes, ce que nous ferons désormais.

Graphes triangulés, cliques et potentiels

Un algorithme de calcul des probabilités, nommé algorithme d'*arbre de jonction*, a été développé par Jensen, Lauritzen et Olesen vers 1990 [67]. Cet algorithme est utilisable dans les modèles graphiques non orientés à structure triangulée comportant uniquement des variables aléatoires à valeurs finies. Un graphe non orienté est dit *triangulé* (ou *cordal*) si et seulement si tout cycle de longueur strictement supérieure à trois admet une corde. Une corde, dans un cycle, est une arête du graphe joignant deux sommets non consécutifs dans le cycle. La figure 1.11 représente un graphe avec un cycle admettant une corde : le cycle $[1, 2, 3, 4, 5, 1]$ admet la corde $\{1, 3\}$ (et également la corde $\{1, 4\}$). La figure 1.12 représente un graphe non triangulé, car le cycle $[1, 2, 3, 4, 5, 6, 1]$, de longueur 6, n'admet pas de corde. Remarquons que la paire $\{3, 6\}$, par exemple, n'est pas une corde car ce n'est pas une arête du graphe. Un graphe complet est un graphe $(\mathcal{U}, \mathcal{E})$

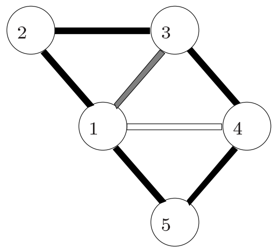


FIG. 1.11 – *Exemple de corde dans un cycle. L'arête* $\{1, 3\}$ *est une corde du cycle* $[1, 2, 3, 4, 5, 1]$ *dans le graphe ci-dessus.*

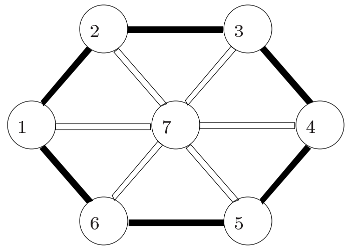


FIG. 1.12 – *Un exemple de graphe non triangulé. Le graphe ci-dessus n'est pas triangulé car le cycle* $[1, 2, 3, 4, 5, 6, 1]$*, de longueur strictement supérieure à trois, n'admet pas de corde.*

tel que $\forall (u, v) \in \mathcal{U} \times \mathcal{U}$, $u \neq v \Rightarrow \{u, v\} \in \mathcal{E}$. Autrement dit, tous les sommets sont deux à deux reliés. Une clique d'un graphe $\mathcal{G}$ non orienté est un sous-graphe complet de $\mathcal{G}$, maximal (au sens où il n'est pas lui-même sous-graphe d'un sous-graphe complet de $\mathcal{G}$). Ainsi, dans la figure 1.11, le graphe engendré par l'ensemble de sommets $\{1, 2, 3\}$ est une clique. Ce n'est pas le cas du graphe engendré par $\{1, 2\}$, qui est complet mais pas maximal. Nous notons $V_{\mathcal{G}}$ l'ensemble des cliques de $\mathcal{G}$.

L'algorithme de l'arbre de jonction est basé en partie sur la factorisation de la loi jointe d'un modèle graphique $P_{\boldsymbol{X}}$ sous la forme

$$P(\boldsymbol{X} = \boldsymbol{x}) = \prod_{\mathcal{C} \in V_{\mathcal{G}}} \psi_{\mathcal{C}}(\boldsymbol{x}_{\mathcal{C}}) \tag{1.2}$$

où les fonctions $\psi_{\mathcal{C}}$ sont des fonctions positives appelées *potentiels de clique*, définies à une constante multiplicative près. L'algorithme est également basé sur l'existence d'un *arbre de jonction* ou *arbre de cliques* pour $\mathcal{G}$. Soit $\mathcal{A}_{\mathcal{G}} = (V_{\mathcal{G}}, \mathcal{E}_{\mathcal{G}})$ un arbre (graphe connexe sans cycle) non orienté dont les sommets sont les cliques de $\mathcal{G}$. Pour toute paire $(\mathcal{C}, \mathcal{C}')$ de sommets de cet arbre, il existe un unique chemin dans $\mathcal{A}_{\mathcal{G}}$ entre $\mathcal{C}$ et $\mathcal{C}'$. L'arbre $\mathcal{A}_{\mathcal{G}}$ est par définition un arbre de clique si et seulement si pour toute paire $\{\mathcal{C}, \mathcal{C}'\}$, l'intersection des cliques $\mathcal{C}$ et $\mathcal{C}'$ est contenue dans toute clique du chemin entre $\mathcal{C}$ et $\mathcal{C}'$. Le graphe central de la figure 1.13 représente un arbre de jonction associé au graphe de gauche. Par contre, le graphe de droite n'est pas un arbre de jonction. En effet, $\mathcal{C}_2 \cap \mathcal{C}_3 = \{2,3,4\} \cap \{3,4,6\} = \{3,4\}$ doit être contenu dans chacune des cliques du chemin entre $\mathcal{C}_2$ et $\mathcal{C}_3$, c'est-à-dire $[\mathcal{C}_2, \mathcal{C}_1, \mathcal{C}_4, \mathcal{C}_3]$. Or $\mathcal{C}_4 = \{3,5\}$ ne contient pas $\mathcal{C}_2 \cap \mathcal{C}_3$. Notons que dans cet exemple, le graphe admet plusieurs arbres de jonction : l'arbre obtenu en remplaçant, dans le graphe central, l'arête $\{\mathcal{C}_2, \mathcal{C}_4\}$ par l'arête $\{\mathcal{C}_1, \mathcal{C}_4\}$, est également un arbre de jonction. L'intersection de deux cliques $\mathcal{C}$ et $\mathcal{C}'$ formant les extrémités d'une arête $a = \{\mathcal{C}, \mathcal{C}'\}$ d'un arbre de jonction est appelée un *séparateur de cliques* $\mathcal{S}$. Ainsi, nous noterons $\mathcal{S}_a = \mathcal{C} \cap \mathcal{C}'$. Nous noterons $\mathcal{S}_{\mathcal{G}}$ l'ensemble des séparateurs de cliques de $\mathcal{G}$.

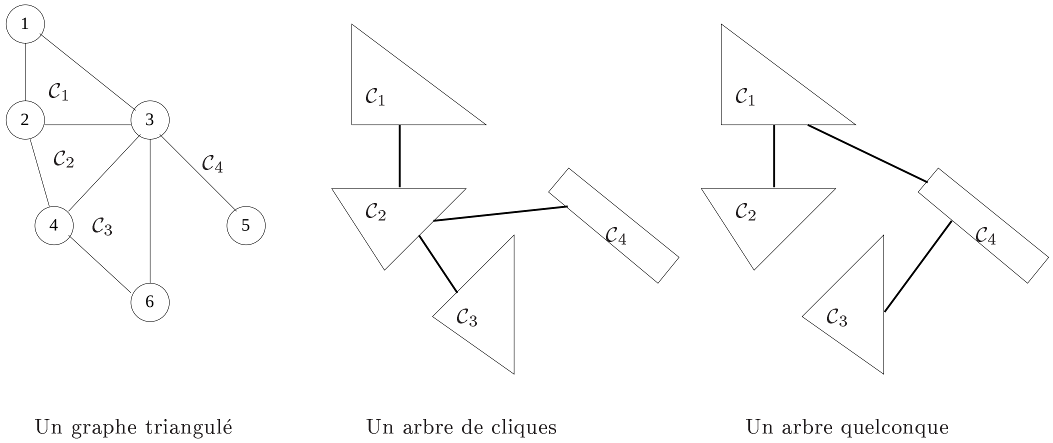


FIG. 1.13 – *La notion d'arbre de jonction*

Tout graphe triangulé admet un arbre de jonction, et réciproquement, comme cela a été montré en parallèle par Buneman, 1974 [15], par Gavril, 1974 [57] et par Walter, 1972 [118]. Le problème de sa construction est abordé dans Smyth, Heckerman et Jordan, 1997 [110] : on définit le poids d'une arête entre deux cliques comme le nombre de variables dans leur intersection. Un arbre défini sur $V_{\mathcal{G}}$ est un arbre de jonction si et seulement si c'est un arbre engendrant $\mathcal{G}$ de poids maximal. L'arbre de jonction peut alors être construit en ajoutant successivement des arêtes de poids maximal, sans prendre celles qui créeraient des cycles. L'algorithme est d'une complexité de $\mathcal{O}(N^2 \log(N))$ où N désigne le nombre de sommets du graphe (donc de variables aléatoires dans le modèle).

L'algorithme d'arbre de jonction

L'algorithme d'arbre de jonction utilise les relations d'indépendance conditionnelle entre les variables aléatoires $(X_u)_{u\in\mathcal{U}}$ d'un modèle graphique pour réduire la complexité du calcul des probabilités dans les modèles graphiques non orientés à structure triangulée comportant uniquement des variables aléatoires à valeurs finies. Il est basé sur la factorisation de la loi jointe sous la forme de potentiels de clique (équation (1.2)), sur une autre factorisation de la loi jointe faisant intervenir la loi des séparateurs de cliques

$$P(\boldsymbol{X} = \boldsymbol{x}) = \frac{\prod\limits_{\mathcal{C}\in V_{\mathcal{G}}} P(\boldsymbol{X}_{\mathcal{C}} = \boldsymbol{x}_{\mathcal{C}})}{\prod\limits_{\mathcal{S}\in \mathcal{S}_{\mathcal{G}}} P(\boldsymbol{X}_{\mathcal{S}} = \boldsymbol{x}_{\mathcal{S}})} \tag{1.3}$$

et sur l'arbre de jonction. D'après Lucke, 1996 [87], l'ensemble des séparateurs de cliques est égal à l'ensemble des séparateurs de sommets minimaux de $\mathcal{G}$, où un *séparateur de sommets* est défini comme un sous ensemble $\mathcal{V}$ de $\mathcal{U}$ tel que la suppression, dans $\mathcal{G}$, des sommets de $\mathcal{V}$ et des arêtes incidentes à ces sommets, rend $\mathcal{G}$ non connexe. Nous verrons au chapitre 2 en section 2.4.2 (remarque 2.8) que la complexité des calculs de probabilité est réduite par l'existence de séparateurs de sommets, mais est une fonction croissante de la taille de ces séparateurs de sommets. L'arbre de cliques permet donc d'utiliser des séparateurs de sommets de taille minimale : les séparateurs de cliques.

L'algorithme d'arbre de jonction s'appuie sur la notion de *flot d'information* entre des cliques $\mathcal{C}$ et $\mathcal{C}'$ adjacentes dans l'arbre de jonction, définie comme suit : soit $\mathcal{S} = \mathcal{C} \cap \mathcal{C}'$. On dispose d'une valeur courante des potentiels de clique $(\psi_{\mathcal{C}})_{\mathcal{C}\in V_{\mathcal{G}}}$ et des potentiels de séparateur de cliques $(\psi_{\mathcal{S}})_{\mathcal{S}\in\mathcal{S}_{\mathcal{G}}}$. Le potentiel ψ_S est mis à jour par le potentiel

$$\psi^*_{\mathcal{S}}(\boldsymbol{x}_{\mathcal{S}}) = \sum_{\boldsymbol{x}_{\mathcal{C}\setminus\mathcal{S}}\in\mathcal{X}_{\mathcal{C}\setminus\mathcal{S}}} \psi_{\mathcal{C}}(\boldsymbol{x}_{\mathcal{C}}) \tag{1.4}$$

puis le facteur de mise à jour est défini par :

$$\zeta_{\mathcal{S}}(\boldsymbol{x}_{\mathcal{S}}) = \frac{\psi^*_{\mathcal{S}}(\boldsymbol{x}_{\mathcal{S}})}{\psi_{\mathcal{S}}(\boldsymbol{x}_{\mathcal{S}})}.$$

Enfin, le potentiel $\psi_{\mathcal{C}'}$ est mis à jour par le potentiel

$$\psi^*_{\mathcal{C}'}(\boldsymbol{x}_{\mathcal{C}'}) = \psi_{\mathcal{C}'}(\boldsymbol{x}_{\mathcal{C}'})\zeta_{\mathcal{S}}(\boldsymbol{x}_{\mathcal{S}}).$$

Si les potentiels $(\psi_{\mathcal{C}})_{\mathcal{C}\in V_{\mathcal{G}}}$ et $(\psi_{\mathcal{S}})_{\mathcal{S}\in\mathcal{S}_{\mathcal{G}}}$ sont correctement initialisés, on peut appliquer successivement l'algorithme ci-dessus en effectuant un parcours de l'arbre de jonction, de manière à obtenir $P(\boldsymbol{X} = \boldsymbol{x})$. Dans le cas de modèles de Markov cachés, un premier parcours de l'arbre de jonction permet de calculer la vraisemblance $P_\lambda(\boldsymbol{Y} = \boldsymbol{y})$. En parcourant ensuite l'arbre de jonction dans le sens contraire, on peut également calculer les probabilités $P(\boldsymbol{S}_{\mathcal{C}} = \boldsymbol{s}_{\mathcal{C}}, \boldsymbol{Y} = \boldsymbol{y})$, utiles pour l'estimation des paramètres comme nous le verrons dans le chapitre 2 (remarque consécutive à la propriété 1). La complexité de l'algorithme est en $\mathcal{O}(\sum\limits_{\mathcal{C}\in\mathcal{V}_{\mathcal{G}}} \text{taille}(\mathcal{C}))$ où la taille d'un ensemble de variables aléatoires $\boldsymbol{X}_{\mathcal{V}}$

est définie comme le nombre de valeurs possibles de $\boldsymbol{X}_{\mathcal{V}}$. Typiquement, chaque variable aléatoire du modèle peut prendre au plus K valeurs, chaque clique $\mathcal{C}$ a un nombre de variables aléatoires $N_{\mathcal{C}}$ borné par une constante L (indépendante du nombre total N de variables aléatoires du modèle) et le nombre de cliques est une fonction polynomiale de N, donc appartient à un certain $\mathcal{O}(N^m)$. Dans ce cas, la complexité de calcul de l'algorithme d'arbre de jonction est $\mathcal{O}(K^L N^m)$. Pour le calcul de la vraisemblance dans les modèles de Markov cachés, la sommation dans l'équation (1.4) de l'algorithme se restreint aux variables cachées du modèle. Par conséquent, il n'est pas nécessaire que toutes les variables aléatoires du modèle soient à valeurs finies : il suffit en fait que toutes les variables cachées le soient.

Si l'on revient à présent sur le problème du calcul de la vraisemblance dans les modèles de Markov cachés classiques présentés dans la section 1.3, on a donc trois catégories de modèles :

1. le modèle d'indépendance pour lequel le calcul est direct, d'une complexité en $\mathcal{O}(nK)$;
2. les modèles de chaîne et d'arbre de Markov cachés, qui ont une structure arborescente (voir figures 1.5 et 1.7) donc triangulée (il n'existe pas de cycle dans une telle structure). On peut donc appliquer l'algorithme d'arbre de jonction à ces modèles. Comme le nombre de cliques est, dans chacun des cas, équivalent à $2n$ et que $N_{\mathcal{C}}$ est égal à deux pour chaque clique $\mathcal{C}$, la complexité du calcul de la vraisemblance est en $\mathcal{O}(nK^2)$ pour les deux modèles ;
3. les champs de Markov cachés. Il s'agit d'un modèle à structure non triangulée pour lequel il n'y a pas d'algorithme connu de calcul de la vraisemblance avec une complexité polynomiale par rapport à n. En pratique, on calcule la vraisemblance de manière approchée en approximant la vraie loi jointe originale par une loi approximante sous forme factorisée appartenant à une certaine famille paramétrique. On détermine alors la loi factorisée de cette famille la plus proche de la loi originale en un certain sens, par exemple au sens de la divergence de Kullback-Leibler. C'est la loi factorisée qui est utilisée pour approcher la vraisemblance.

Une variante de l'algorithme d'arbre de jonction pour les modèles de Markov cachés, de même complexité, a été proposée dans Lucke, 1996 [87]. Cette variante a l'avantage de remplacer l'usage des potentiels de clique par des probabilités jointes et généralise l'algorithme avant-arrière utilisé pour les chaînes de Markov cachées. Elle s'appuie également sur un parcours de l'arbre de jonction, ainsi que sur des quantités α qui associent, à chaque clique, la loi jointe des variables aléatoires observées déjà rencontrées dans le parcours et des variables aléatoires de cette clique. Une formule de récurrence permet de passer de la quantité α associée à une clique, à celle de son successeur dans l'arbre de jonction.

Paramétrisation dans les modèles graphiques orientés acycliques

Rappelons que nous avons défini les paramètres d'un modèle graphique comme des spécifications numériques associées au graphe d'indépendance conditionnelle telles que la loi jointe $P_{\boldsymbol{X}}$ soit définie de manière unique. Nous avons vu ci-dessus que l'algorithme d'arbre de jonction s'appuie sur les potentiels de clique, c'est-à-dire sur la factorisation (1.2) de la loi jointe. Il suffit alors de disposer d'une spécification numérique, par exemple paramétrique, des potentiels de clique $\psi_{\mathcal{C}}$, pour obtenir une définition unique de $P_{\boldsymbol{X}}$. C'est ce type de paramétrisation qui est utilisée dans le modèle de champ de Markov caché de Potts, où pour les cliques $\mathcal{C} = \{S_u, S_v\}$ avec $u \sim v$, on a $\psi_{\mathcal{C}}(s_u, s_v) = \exp(-\beta {s_u}^t s_v)$. Lucke (1996), dans [87], propose une paramétrisation basée sur la loi jointe des $\boldsymbol{X}_{\mathcal{C}}$ pour chaque clique $\mathcal{C}$.

Cependant, dans les modèles graphiques orientés acycliques, il existe pour les variables aléatoires discrètes une notion naturelle de transition de parents à enfant. Ainsi, les probabilités de transition p_{ij} entre états cachés dans les chaînes de Markov cachées correspondent à l'existence d'un arc de S_t à S_{t+1}. De même, dans les arbres de Markov cachés, les probabilités de transition p_{ij} entre états cachés correspondent à l'existence d'un arc de $S_{\rho(u)}$ à S_u. Cette paramétrisation est due à la factorisation de la loi jointe des modèles graphiques sous la forme

$$P(\boldsymbol{X} = \boldsymbol{x}) = \prod_{u \in \mathcal{U}} P(X_u = x_u | \boldsymbol{X}_{\mathrm{pa}(u)} = \boldsymbol{x}_{\mathrm{pa}(u)}) \tag{1.5}$$

où pour un sommet u de $\mathcal{G} = (\mathcal{U}, \mathcal{E})$, $\mathrm{pa}(u)$ désigne l'ensemble des parents de u, soit l'ensemble $\{v \in \mathcal{U} | (v, u) \in \mathcal{E}\}$. Par convention, si $\mathrm{pa}(u) = \emptyset$, on pose $P(X_u = x_u | \boldsymbol{X}_{\mathrm{pa}(u)} = \boldsymbol{x}_{\mathrm{pa}(u)}) = P(X_u = x_u)$. L'équation (1.5) montre que $P_{\boldsymbol{X}}$ est entièrement spécifiée par la donnée des lois conditionnelles $P(X_u = x_u | \boldsymbol{X}_{\mathrm{pa}(u)} = \boldsymbol{x}_{\mathrm{pa}(u)})$. Dans le cas de variables aléatoires à valeurs finies, $P(X_u = x_u | \boldsymbol{X}_{\mathrm{pa}(u)} = \boldsymbol{x}_{\mathrm{pa}(u)})$ peut être représentée par un tenseur, de dimension égale à la taille de $\boldsymbol{X}_{\{u\} \cup \mathrm{pa}(u)}$, correspondant à des *probabilités de transition*. Si $\mathrm{np}(u)$ désigne le cardinal de $\mathrm{pa}(u)$ et que $\mathrm{pa}(u) = \{X_{i_1}, \ldots, X_{i_{\mathrm{np}(u)}}\}$ alors on notera

$$P(X_u = k | X_{i_1} = j_1, \ldots, X_{i_{\mathrm{np}(u)}} = j_{\mathrm{np}(u)}) = p^{(u)}_{j_1, \ldots, j_{\mathrm{np}(u)}, k} = p^{(u)}_{\boldsymbol{x}_{\mathrm{pa}(u)}, x_u}.$$

Dans le cas où le processus $\boldsymbol{X}_{\mathrm{pa}(u)}$ est à valeurs finies et où X_u est à valeurs continues, on peut choisir $P(X_u = x_u | \boldsymbol{X}_{\mathrm{pa}(u)} = \boldsymbol{x}_{\mathrm{pa}(u)})$ dans une famille paramétrique de lois de probabilités $\{P_\theta | \theta \in \Theta\}$. Si $\mathrm{pa}(u) = \{X_{i_1}, \ldots, X_{i_{\mathrm{np}(u)}}\}$ alors

$$P(X_u = x_u | X_{i_1} = j_1, \ldots, X_{i_{\mathrm{np}(u)}} = j_{\mathrm{np}(u)}) = P_{\theta^{(u)}_{j_1, \ldots, j_{\mathrm{np}(u)}}}(x_u) = P_{\theta^{(u)}_{\boldsymbol{x}_{\mathrm{pa}(u)}}}(x_u).$$

Remarque 1.1 *Lien entre les modèles de Markov cachés et les mélanges de lois. Par la formule des probabilités totales, on peut écrire la loi de Y_u comme*

$$P(Y_u = y) = \sum_{\boldsymbol{x}_{pa(u)} \in \mathcal{X}_{pa(u)}} P(\boldsymbol{X}_{pa(u)} = \boldsymbol{x}_{pa(u)}) P(Y_u = y | \boldsymbol{X}_{pa(u)} = \boldsymbol{x}_{pa(u)}),$$

c'est-à-dire

$$P(Y_u = y) = \sum_{\boldsymbol{x}_{pa(u)} \in \mathcal{X}_{pa(u)}} P(\boldsymbol{X}_{pa(u)} = \boldsymbol{x}_{pa(u)}) p^{(u)}_{\boldsymbol{x}_{pa(u)}, y}$$

si Y_u est à valeurs discrètes et

$$P(Y_u = y) = \sum_{\boldsymbol{x}_{pa(u)} \in \mathcal{X}_{pa(u)}} P(\boldsymbol{X}_{pa(u)} = \boldsymbol{x}_{pa(u)}) P_{\theta^{(u)}_{\boldsymbol{x}_{pa(u)}}}(y)$$

si Y_u est à valeurs continues. Si les variables aléatoires observées sont indépendantes conditionnellement aux processus caché et si la loi de $\boldsymbol{X}_{pa(u)}$ est indépendante de u (hypothèse de stationnarité), alors $\boldsymbol{X}_{pa(u)}$ est le processus caché $\boldsymbol{S}_{pa(u)}$; nous notons π sa distribution. Alors la loi marginale de Y_u est une loi de mélange, de proportions π et de famille de lois conditionnelles $\{P_{\theta^{(u)}}|\theta^{(u)} \in \Theta^{(u)}\}$ ou l'ensemble des matrices stochastiques $\{p^{(u)}\}$.

L'article de Smyth, Heckerman et Jordan, 1997 [110] aborde la question du passage de la paramétrisation par les probabilités de transition à la paramétrisation basée sur les potentiels de clique. En effet, ce problème intervient dans l'initialisation de l'algorithme d'arbre de jonction. Une paramétrisation basée sur les lois conditionnelles $P(X_u = x_u|\boldsymbol{X}_{\mathrm{pa}(u)} = \boldsymbol{x}_{\mathrm{pa}(u)})$ a l'avantage, par rapport aux deux autres propositions, d'avoir une interprétation probabiliste et de tenir compte pleinement de la structure du modèle, en particulier de l'orientation des arcs. Elle permet de mieux appréhender la dynamique du modèle, c'est-à-dire la manière dont les états se propagent d'un sommet à ses voisins. Enfin, si – comme c'est souvent le cas pour des graphes d'indépendance conditionnelle orientés acycliques – le but est d'étudier, en particulier, l'influence (asymétrique) d'un ensemble de variables aléatoires sur une autre, une telle paramétrisation est essentielle pour l'interprétation du modèle.

Limites de l'algorithme d'arbre de jonction

L'algorithme d'arbre de jonction est un algorithme générique pour les modèles graphiques, permettant en théorie de résoudre de manière efficace tous les problèmes de calcul de probabilités dans les modèles de Markov cachés à structure non orientée triangulée. En fait, il permet aussi de traiter des structures orientées acycliques, par moralisation puis triangulation du graphe d'origine, d'après Smyth, Heckerman et Jordan, 1997 [110]. Il possède cependant quelques inconvénients. D'une part, les calculs ne sont pas effectués à partir de probabilités (conditionnelles ou non) mais à partir de potentiels de clique, ce qui a déjà l'inconvénient de ne pas prendre en compte les paramètres naturels du modèle (à savoir les probabilités de transition dans le cas d'une structure orientée acyclique). Du coup il est difficile d'interpréter les calculs et les quantités intermédiaires de l'algorithme de manière probabiliste (voir équation (1.4) et les deux suivantes). Notons que l'algorithme de Lucke résout ce dernier problème.

Dans certaines situations, plusieurs calculs de probabilités consécutifs sont nécessaires. Par exemple, nous verrons dans le chapitre 2, section 2.3.5, qu'il est souhaitable de savoir simuler le processus caché $\boldsymbol{S}$ suivant sa loi conditionnelle $P(\boldsymbol{S} = \boldsymbol{s}|\boldsymbol{Y} = \boldsymbol{y})$, pour des algorithmes stochastiques d'estimation des paramètres. Ceci est fait en simulant S_1 suivant $P(S_1|\boldsymbol{Y} = \boldsymbol{y})$, puis S_2 suivant $P(S_2|\boldsymbol{Y} = \boldsymbol{y}, S_1 = s_1)$, etc. Si l'on utilise l'algorithme d'arbre de jonction, il est difficile de réutiliser les calculs déjà effectués à cause de son manque d'interprétation. En général, utiliser une passe de l'algorithme d'arbre de jonction pour simuler chacun des S_u conduit à répéter les mêmes calculs de très nombreuses fois. En fait, une seule passe d'algorithme de type avant-arrière suffit souvent.

Il est également extrêmement délicat d'utiliser les formules de ces algorithmes génériques, tout particulièrement celui d'arbre de jonction, pour faire des calculs analytiques.

Par exemple, il est fréquent d'avoir à calculer l'espérance de variables aléatoires du modèle en fonction des paramètres, ou de désirer connaître la fonction de répartition d'une variable aléatoire conditionnellement à d'autres variables.

Enfin, un problème également handicapant et commun à tous ces algorithmes génériques est dû au principe même de factorisation de la loi jointe. L'algorithme de Lucke met en évidence le fait que les quantités calculées sont les probabilités jointes d'un nombre de variables qui croît quand on parcourt l'arbre de jonction, et qui s'expriment comme des produits de probabilités jointes. Ces quantités tendent vers 0 quand n tend vers $+\infty$, ce qui cause des instabilités numériques, même quand le nombre de variables aléatoires du modèle reste raisonnable. Ce problème est traité, pour les chaînes de Markov cachées, dans Devijver, 1985 [37], en passant aux probabilités des états cachés sachant les t premières observations (*probabilités de filtrage*) et à la factorisation de la loi des états cachés sachant la totalité des observations (*probabilités de lissage*).

1.5 Une famille "opérationnelle" de modèles de Markov cachés

Quelle est la généricité des méthodes pour les modèles de Markov cachés ?

La conclusion de la section précédente est que de façon standard, on se concentre sur des modèles de Markov cachés génériques, c'est-à-dire pour lesquels il n'y ait qu'à définir la structure d'indépendance conditionnelle pour en déduire, sans refaire aucun calcul théorique, une paramétrisation du modèle, des algorithmes efficaces et numériquement stables de calcul des probabilités, d'estimation de paramètres, de restauration des états cachés et des méthodes de sélection de modèles. Il nous paraît souhaitable d'ajouter à ces exigences celle de facilité d'interprétation de la paramétrisation et de la transparence des algorithmes, pour permettre le calcul d'espérances ou de fonctions de répartitions conditionnelles et pour supprimer des calculs redondants.

Comme nous l'avons vu au cours de ce chapitre, ces objectifs ne sont pas toujours atteints par les modèles et les méthodes actuels. Ceci s'explique à la fois par la très grande diversité des modèles de Markov cachés, qui rend difficile le traitement de tous les cas par une même méthode, et peut-être par une certaine inadéquation des méthodes elles-mêmes. C'est pourquoi nous nous proposons comme objectif de définir une famille de modèles de Markov cachés, que nous noterons $\mathcal{D}$, pour laquelle on puisse assurer l'existence de méthodes qui répondent aux exigences énoncées ci-dessus.

Proposition d'une famille de modèles de Markov cachés

La famille $\mathcal{D}$ est constituée des modèles de Markov cachés vérifiant les hypothèses suivantes :

– la structure est orientée et acyclique. Nous avons vu que l'intérêt des modèles graphiques orientés acycliques, par rapport aux modèles graphiques non orientés, est d'offrir une plus grande facilité d'interprétation et de permettre une paramétrisation naturelle basée sur des probabilités de transition. De plus, le calcul de probabilités (par exemple de la vraisemblance) dans ces graphes ne requiert pas d'approximation ;

– la structure est morale. Un graphe (orienté ou non orienté) est dit moral si et seulement si pour chaque sommet admettant au moins deux parents, tous les parents sont deux à deux adjacents. Notons que Lauritzen, 1996 [75] utilise le terme de *graphe parfait* pour de tels graphes, mais cette terminologie entre en conflit avec celle de Smyth *et al.*, 1997 [110] pour qui les *graphes parfaits* sont des graphes tels qu'il y a équivalence entre les propriétés de séparation et les propriétés d'indépendance conditionnelle. L'hypothèse d'une structure morale provient du constat que les calculs directs dans les modèles graphiques orientés acycliques sont difficiles et complexes en nombre d'opérations élémentaires si le graphe n'est pas moral, comme le notent Smyth, Heckerman et Jordan, 1997 [110]. Ceci se comprend, entre autres, par la définition même des relations d'indépendance conditionnelle dans les graphes orientés acycliques, qui fait appel au graphe moral (non orienté) du graphe original. C'est la raison pour laquelle la méthode préconisée en général, par exemple dans Smyth *et al.*, consiste à moraliser les graphes orientés acycliques, à en supprimer l'orientation puis à les trianguler (ce qui est la cause profonde de l'usage de potentiels de clique à la place de probabilités). C'est pourquoi nous nous restreignons directement à des graphes moraux. D'après Whittaker, 1990 [120] et Pearl *et al.*, 1990 [98], les propriétés d'indépendance conditionnelle de ces modèles graphiques restent identiques quand on supprime l'orientation de leurs arcs. C'est la raison pour laquelle l'orientation des arcs pour les modèles des figures 1.5 et 1.7 peut être supprimée. De plus, nous avons vu que pour des modèles graphiques non orientés, il suffit que la structure du graphe soit triangulée pour qu'il existe des algorithmes de calcul de probabilités ayant une complexité polynomiale en fonction du nombre de variables aléatoires observées du modèle, sous certaines conditions quant à la taille des cliques. A priori, ces algorithmes ne s'appliquent pas directement aux modèles graphiques orientés acycliques mais l'hypothèse de moralité du graphe d'indépendance conditionnelle nous permettra d'appliquer ces résultats aux modèles de la famille $\mathcal{D}$. En effet, un graphe orienté moral et sans circuit (*i.e.* soi-disant *acyclique*) est triangulé car pour tout cycle $[X_1, \ldots, X_n, X_1]$ de longueur n strictement supérieure à trois, le graphe engendré par $\{X_1, \ldots, X_n\}$ est également orienté et sans circuit, donc par un résultat classique, contient un puits X_t (sommet sans arc sortant). Les sommets X_{t-1} et X_{t+1} sont donc des parents de X_t et sont adjacents, par moralité du graphe. Par suite, $\{X_{t-1}, X_{t+1}\}$ est une corde du cycle : le graphe d'origine est triangulé et admet par conséquent un arbre de jonction ;

– la structure est connexe. Si la structure du graphe n'est pas connexe, on peut travailler successivement sur chacune des composantes connexes du graphe (cas de plusieurs processus mutuellement indépendants). C'est pourquoi il est toujours possible de se ramener à des modèles graphiques à structure connexe ;

– il n'existe pas d'arc entre variables aléatoires à valeurs continues. À part dans le cas de lois conditionnelles de la famille exponentielle, les modèles graphiques comportant des arcs entre variables aléatoires continues sont délicats à paramétrer. D'autre part, nous serons conduits à considérer des situations où certaines des données $\boldsymbol{Y}$, observées en principe, se trouvent être en réalité manquantes (en particulier en sélection de modèles par validation croisée). Ceci revient à considérer des modèles avec des données cachées à valeurs continues. Nous verrons alors que s'il n'existe aucun arc entre deux variables à valeurs continues, l'estimation des paramètres et le calcul de probabilités demeurent possibles, même quand la valeur de ces variables est inconnue.

Il peut être entièrement justifié d'avoir recours à des modèles graphiques non orientés pour modéliser des processus à interactions symétriques entre variables aléatoires. D'après Smyth, Heckerman et Jordan, 1997 [110], on peut montrer que pour des modèles graphiques *non orientés et triangulés*, il existe un graphe d'indépendance conditionnelle orienté acyclique traduisant les mêmes relations d'indépendance conditionnelle que le graphe non orienté. On pourra alors appliquer les méthodes et les résultats suivants aux modèles graphiques non orientés et triangulés, dès lors qu'il existe un graphe d'indépendance conditionnelle orienté acyclique équivalent et vérifiant les hypothèses ci-dessus. Notons qu'à partir de ce graphe orienté acyclique, il est possible de paramétrer le modèle par les probabilités conditionnelles $P(X_u = x_u | \boldsymbol{X}_{\mathrm{pa}(u)} = \boldsymbol{x}_{\mathrm{pa}(u)})$, mais ces paramètres n'ont a priori pas grand sens par rapport au phénomène modélisé, c'est pourquoi nous avons préféré faire l'hypothèse que le graphe est orienté et acyclique. Il est également possible de paramétrer les modèles graphiques non orientés et triangulés par la valeur des probabilités jointes de chaque clique, comme dans Lucke, 1996 [87]. L'adaptation de nos algorithmes est alors évidente, voir chapitre 2, section 2.4.7

L'hypothèse de moralité est sensiblement plus restrictive que les autres hypothèses mais reste satisfaite par de nombreux modèles, dont les chaînes et les arbres de Markov cachés. Nous verrons un modèle d'intérêt où cette hypothèse n'est pas vérifiée et où nos exigences en termes d'algorithmes d'inférence et d'interprétation sont néanmoins satisfaites (voir chapitre 5.

Notons que la famille ainsi définie autorise l'existence d'arcs de variables aléatoires observées vers des variables aléatoires cachées, ce qui est inhabituel dans les modèles de Markov cachés. Cependant, les modèles que nous considérons permettent de traiter des modèles graphiques dont le but est d'implémenter des systèmes experts, comme l'exemple ASIA de Lauritzen et Spiegelhalter, 1988 [78]. Cet exemple est destiné à illustrer l'intérêt des modèles graphiques dans l'aide automatisée au diagnostic médical. Nous en reproduisons ici une version modifiée afin d'en diminuer le nombre de variables aléatoires et d'en faire correspondre le graphe d'indépendance conditionnelle à nos hypothèses. Les quatre variables aléatoires de ce modèle sont

- F qui représente le fait de fumer ;
- B qui représente le fait d'avoir une bronchite ;
- P qui représente le fait d'avoir un cancer du poumon ;
- D qui représente le fait d'avoir une dyspnée.

Toutes ces variables sont binaires. Le graphe d'indépendance conditionnelle est représenté figure 1.14. La variable F est supposée toujours connue, les trois autres pouvant être connues ou inconnues. Par exemple, le fait que le patient souffre d'une bronchite peut être connu ou non, selon les cas et l'utilisation du modèle. Par exemple, on pourrait justement désirer connaître la probabilité que le patient souffre de bronchite sachant qu'il fume et n'a pas de dyspnée, en absence de tout renseignement sur l'existence d'un cancer du poumon. En autorisant des arcs de variables cachées vers des variables observées, on

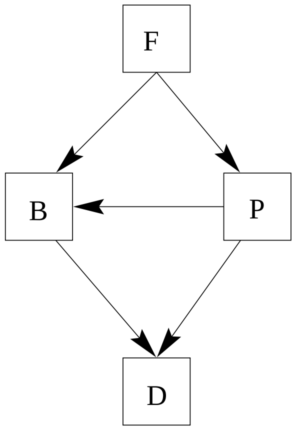


FIG. 1.14 – *Un exemple factice et simpliste de système expert pour l'aide au diagnostic médical. Les variables aléatoires binaires F, B, P et D représentent respectivement le fait de fumer, d'avoir une bronchite, un cancer du poumon et une dyspnée.*

permet l'application des méthodes présentées par la suite à de telles situations, où le fait qu'une variable aléatoire soit cachée ou observée dépend du contexte dans lequel le modèle est utilisé.

Paramétrisation

Comme nous l'avons vu dans la section 1.4, la loi jointe des modèles de Markov cachés de la famille $\mathcal{D}$, du fait de leur structure orientée acyclique, s'écrit sous forme d'un produit de probabilités conditionnelles des sommets sachant leurs parents (équation (1.5)). Le modèle est alors entièrement spécifié par la donnée :

- des lois initiales des variables aléatoires sources. Les sources d'un graphe sont les sommets n'ayant aucun arc entrant. Par connexité de la structure du modèle, elles ont au moins un arc sortant. Comme les variables aléatoires à valeurs continues n'ont aucun arc sortant par hypothèse, les sources sont à valeurs finies et on peut représenter leur loi par des vecteurs

$$(P(X_u = k))_{k \in \mathcal{X}_u} = (\pi_k^{(u)})_{k \in \mathcal{X}_u} \ ;$$

- des probabilités de transition entre parents (cachés ou observés) et état caché

$$P(S_u = k | \boldsymbol{S}_{\mathrm{pa}(u)} = \boldsymbol{s}_{\mathrm{pa}(u)}, \boldsymbol{Y}_{\mathrm{pa}(u)} = \boldsymbol{y}_{\mathrm{pa}(u)}) = P(S_u = k | \boldsymbol{X}_{\mathrm{pa}(u)} = \boldsymbol{x}_{\mathrm{pa}(u)}) = p^{(u)}_{\boldsymbol{x}_{\mathrm{pa}(u)}, k} \ ;$$

- des paramètres d'émission $\theta^{(u)}_{\boldsymbol{x}_{\mathrm{pa}(u)}}$ tels que

$$P(Y_u = y | \boldsymbol{S}_{\mathrm{pa}(u)} = \boldsymbol{s}_{\mathrm{pa}(u)}, \boldsymbol{Y}_{\mathrm{pa}(u)} = \boldsymbol{y}_{\mathrm{pa}(u)}) = P(Y_u = y | \boldsymbol{X}_{\mathrm{pa}(u)} = \boldsymbol{x}_{\mathrm{pa}(u)}) = P_{\boldsymbol{\theta}^{(u)}_{\boldsymbol{x}_{\mathrm{pa}(u)}}}(y).$$

Dans le cas particulier où Y_u est à valeurs finies, des lois d'émission multinomiales sont fréquemment utilisées et la loi

$$P(Y_u = k | \boldsymbol{S}_{\mathrm{pa}(u)} = \boldsymbol{s}_{\mathrm{pa}(u)}, \boldsymbol{Y}_{\mathrm{pa}(u)} = \boldsymbol{y}_{\mathrm{pa}(u)}) = q^{(u)}_{\boldsymbol{x}_{\mathrm{pa}(u)}, k}$$

peut être représentée par un tenseur $q^{(u)}$ de dimension taille($\{Y_u\} \cup \boldsymbol{S}_{\mathrm{pa}(u)} \cup \boldsymbol{Y}_{\mathrm{pa}(u)}$). Il y a donc autant de paramètres vectoriels que de variables aléatoires dans le modèle, ce qui rend celui-ci pratiquement impossible à estimer à partir d'une seule réalisation du processus observé $\boldsymbol{Y}$, alors que le problème de l'estimation à partir d'une seule réalisation se pose souvent. En réalité, les modèles de Markov cachés sont fréquemment construits à partir de la répétition périodique, un nombre de fois arbitrairement grand, d'un même motif graphique. De tels modèles sont parfois appelés modèles de Markov cachés *dynamiques* (voir Cowell *et al.*, 1999 [31]). C'est d'ailleurs cette répétition qui permet de s'intéresser aux propriétés asymptotiques du modèle, puisqu'on peut alors construire des modèles graphiques avec un nombre de variables observées n arbitrairement grand. Dans ce cas, nous ferons l'hypothèse que les paramètres ne dépendent que du motif et non de la position du motif dans le graphe (autrement dit de l'indice de la réplication). C'est ce qu'on appelle l'hypothèse d'*homogénéité*. On notera alors $\pi_k^{(u)} = \pi_k$, $p^{(u)}_{\boldsymbol{x}_{\mathrm{pa}(u)},k} = p_{\boldsymbol{x}_{\mathrm{pa}(u)},k}$ et $\theta^{(u)}_{\boldsymbol{x}_{\mathrm{pa}(u)}} = \theta_{\boldsymbol{x}_{\mathrm{pa}(u)}}$. À titre d'exemple, pour les chaînes de Markov cachées (figure 1.5), les probabilités de transition $p_{ij} = P(S_{t+1} = j | S_t = i)$ ne dépendent pas de t. On dit que la chaîne de Markov $(S_t)_{t \in I\!N}$ est homogène. La figure 1.15 représente un exemple de modèle de Markov caché vérifiant les conditions ci-dessus. Ses paramètres sont :

- $(\pi_i)_i$ tel que $P(S_{1,1} = i) = \pi_i$;
- $\theta_1, \ldots, \theta_K$ tels que $P(Y_{u,1} = y | S_{u,1} = i) = P_{\theta_i}(y)$;
- $(q_{i,j})_{i,j}$ tel que $P(Y_{u,2} = j | S_{u,1} = i) = q_{i,j}$;
- $(p_{i,j,k})_{i,j,k}$ tel que $P(S_{u,2} = k | Y_{u,2} = j, S_{u,1} = i) = p_{i,j,k}$;
- $(r_{i,j})_{i,j}$ tel que $P(S_{u+1,1} = j | S_{u,2} = i) = r_{i,j}$.

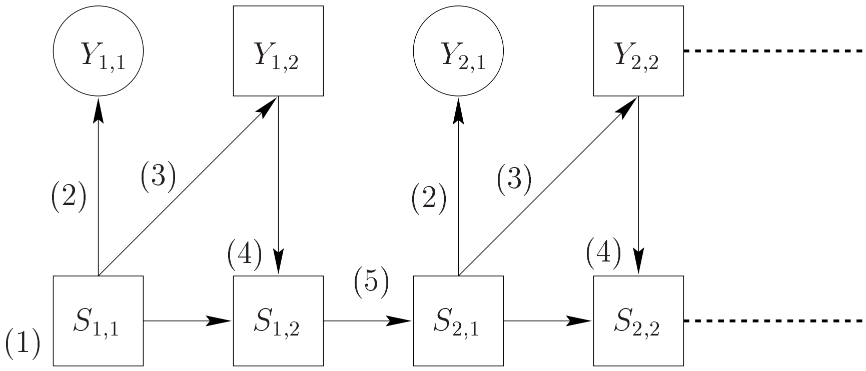


FIG. 1.15 – *Un exemple de modèle de Markov caché vérifiant les hypothèses de la section 1.5. (1) Le paramètre π_i est associé à la source $S_{1,1}$ du graphe. (2) Le paramètre θ_i est associé à la loi de $Y_{u,1}$ sachant $S_{u,1} = i$. (3) Le paramètre $q_{i,j}$ est associé à la loi de $Y_{u,2}$ sachant $S_{u,1} = i$. (4) Le paramètre $p_{i,j,k}$ est associé à la loi de $S_{u,2}$ sachant $S_{u,1} = i$ et $Y_{u,2} = j$. (5) Le paramètre $r_{i,j}$ est associé à la loi de $S_{u+1,1}$ sachant $S_{u,2} = i$.*

Il s'agit d'un modèle de Markov caché dynamique, homogène : les probabilités de transition $P(S_{u,2} = k | S_{u,1} = i, Y_{u,2} = j) = p_{i,j,k}$ ne dépendent pas de u. De même pour les probabilités de transition $r_{i,j}$ et les paramètres d'émission θ_i et $q_{i,j}$. L'unique source $S_{1,1}$ engendre un unique paramètre π vectoriel.

1.6 Conclusion

Nous avons donc défini une famille de modèles de Markov cachés $\mathcal{D}$ caractérisée par les propriétés suivantes :

- possibilité de modéliser des processus aléatoires à valeurs continues sans hypothèse sur les lois d'émission ;
- existence d'une paramétrisation interprétable (probabilités de transition et d'émission) ;
- existence d'un algorithme efficace de calcul des probabilités (arbre de jonction).

Dans le chapitre suivant, nous proposons des algorithmes d'inférence pour ces modèles qui sont efficaces, numériquement stables et avec une interprétation probabiliste. Au chapitre 3 sont exposées des méthodes de sélection de modèles.

Chapitre 2

Estimation des paramètres

2.1 Introduction

Dans le chapitre précédent, nous avons défini une famille de modèles de Markov cachés $\mathcal{D}$ pour lesquels la structure d'indépendance conditionnelle permet, comme nous l'avons montré, de définir une paramétrisation naturelle et interprétable, ceci sans avoir à refaire aucun calcul théorique. Dans la perspective d'obtenir une famille de modèles opérationnels, nous montrons dans le présent chapitre que les modèles de la famille $\mathcal{D}$ admettent des algorithmes basés sur le principe de l'arbre de jonction (voir section 1.4) mais qui ne connaissent pas ses limitations dues à l'absence de prise en compte des paramètres du modèle et à une instabilité numérique. Nous introduisons ces algorithmes dans le cadre de l'estimation du maximum de vraisemblance.

Nous donnons tout d'abord des conditions nécessaires et des conditions suffisantes pour l'identifiabilité de ces modèles. Puis nous proposons une introduction à l'algorithme EM (*Expectation-Maximization*) de maximisation de l'espérance conditionnelle de la vraisemblance évaluée sur les données complètes, sachant les données observées. Cette introduction met en évidence le principe intuitif de l'algorithme et son interprétation en terme d'algorithme de restauration-maximisation. Nous présentons d'autres stratégies de restauration des données manquantes que celle de l'algorithme EM. Cette section met en évidence l'utilité de calculer la loi conditionnelle des variables cachées des cliques du modèle, sachant les données observées. La section suivante présente des algorithmes dynamiques de type arrière-avant pour les calculs de probabilités dans les modèles de la famille $\mathcal{D}$, qui évitent les écueils énoncés ci-dessus ; nous proposons ensuite un algorithme (dit du Maximum A Posteriori, ou MAP) pour la restauration des états cachés. Nous revenons alors sur l'algorithme EM et ses propriétés, ainsi que celles de ses variantes. Enfin, nous présentons une application des arbres de Markov cachés à la détection de changements de régularité locale d'un processus. Cette application montre l'intérêt des arbres de Markov cachés pour la modélisation de la loi d'une transformée en ondelette ; elle met également en œuvre l'ensemble des algorithmes présentés dans ce chapitre, ce qui permet de voir comment nos algorithmes génériques sont instanciés dans le cas particulier des arbres de Markov cachés.

2.2 Identifiabilité

Nous considérons un modèle de Markov caché de structure $\mathcal{G}$ fixée. Nous supposons également fixés l'ensemble $\mathcal{S}$ des valeurs prises par le processus caché et la famille $(P_\theta)_{\theta\in\Theta}$ des lois d'émission. Le modèle P_λ est donc caractérisé par le paramètre $\lambda \in \Lambda$.

Définition de l'identifiabilité

En général, un *modèle est dit identifiable* si et seulement si $\lambda \longrightarrow P_\lambda$ est une application injective définie sur Λ. Dans le cas de modèles de Markov cachés, pour toute valeur $\boldsymbol{y}$ dans $\boldsymbol{Y}$, la probabilité $P_\lambda(\boldsymbol{Y} = \boldsymbol{y})$ est invariante par permutation des états cachés, ce qui fait qu'au sens de la définition ci-dessus, un modèle n'est jamais identifiable, sauf cas particulier. C'est pourquoi on contourne ce problème en considérant la relation d'équivalence d'égalité des paramètres à une permutation près des états cachés, puis en se plaçant dans l'espace quotient des paramètres. La définition devient la suivante : un *modèle de Markov caché est dit identifiable* si et seulement si pour tout $(\lambda, \lambda') \in \Lambda^2$, $[P_\lambda = P_{\lambda'}] \Rightarrow \lambda = \lambda'$ à une permutation près des états cachés.

Pour éviter de distinguer les cas où les Y_u sont à valeurs discrètes ou continues et les cas où il existe ou non des variables aléatoires observées parmi les parents de certains sommets, nous considérons le cas où les Y_u sont conditionnellement indépendants sachant $\boldsymbol{S} = \boldsymbol{s}$, la loi conditionnelle de $\boldsymbol{Y}$ étant donnée par

$$P_\lambda(\boldsymbol{Y} = \boldsymbol{y}|\boldsymbol{S} = \boldsymbol{s}) = \prod_u P_{\theta^{(u)}_{\boldsymbol{s}_{\mathrm{pa}(u)}}}(y_u). \tag{2.1}$$

Causes de non identifiabilité

L'identifiabilité des modèles de Markov cachés est liée à l'identifiabilité des modèles de mélange. La *famille des mélanges finis de* $(P_\theta)_{\theta\in\Theta}$ *est dite identifiable* si et seulement si pour tout K, pour tout K', pour tout $\pi = (\pi_k)_{1\leq k\leq K} \in]0;1[$ tel que $\sum \pi_k = 1$, pour tout $\rho = (\rho_{k'})_{1\leq k'\leq K'}$ vérifiant la même condition, pour tous paramètres $\theta = (\theta_k)_{1\leq k\leq K}$ et $\varphi = (\varphi_{k'})_{1\leq k'\leq K'}$ (où les θ_k sont deux à deux distincts, de même pour les $\varphi_{k'}$),

$$\left[\sum_{k=1}^{K} \pi_k P_{\theta_k}(y) = \sum_{k'=1}^{K'} \rho_{k'} P_{\varphi_{k'}}(y)\ \mu - \text{p.-p.}\right] \Rightarrow \left[K = K' \text{ et } \sum_{k=1}^{K} \pi_k \delta_{\theta_k} = \sum_{k'=1}^{K'} \rho_{k'} \delta_{\varphi_{k'}}\right], \tag{2.2}$$

c'est-à-dire que les paramètres π et ρ d'une part, θ et φ d'autre part, sont égaux à une permutation commune des états cachés près.

La question de l'identifiabilité des modèles de Markov cachés peut être traitée en se basant sur celle des mélanges, d'après Ephraim et Merhav, 2002 [47]. En effet, la loi d'un modèle de Markov caché s'écrit

$$\begin{aligned} P_\lambda(\boldsymbol{Y} = \boldsymbol{y}) &= \sum_{\boldsymbol{s}} P_\lambda(\boldsymbol{Y} = \boldsymbol{y}|\boldsymbol{S} = \boldsymbol{s}) P_\lambda(\boldsymbol{S} = \boldsymbol{s}) \\ &= \sum_{\boldsymbol{s}} P_\lambda(\boldsymbol{S} = \boldsymbol{s}) \prod_u P_{\theta^{(u)}_{\boldsymbol{s}_{\mathrm{pa}(u)}}}(y_u) \end{aligned} \tag{2.3}$$

d'après l'équation (2.1). De plus, $P_\lambda(\boldsymbol{S} = \boldsymbol{s})$ s'exprime comme un produit de paramètres du modèle, d'après l'équation (1.5).

Les causes suivantes sont sources de non identifiabilité des modèles de Markov cachés :

– non identifiabilité de la famille des mélanges finis de $(P_\theta)_{\theta\in\Theta}$. Par exemple, si la famille des lois d'émission est une famille de lois uniformes, le modèle de Markov caché n'est en général pas identifiable ;
– réductibilité du processus caché ou présence d'états absorbants. Si $P_\lambda(\boldsymbol{S} = \boldsymbol{s})$ est telle que pour certains sommets u, $P_\lambda(\boldsymbol{S}_{\mathrm{pa}(u)} = \boldsymbol{s}_{\mathrm{pa}(u)}) = 0$, alors la vraisemblance est indépendante de $\theta^{(u)}_{\boldsymbol{s}_{\mathrm{pa}(u)}}$.

Condition suffisante d'identifiabilité

L'identifiabilité des chaînes de Markov cachées est traitée par exemple dans Leroux, 1992 [81]. Elle résulte essentiellement d'un théorème de Teicher, 1967 [116] selon lequel si la famille des mélanges finis de $(P_\theta)_{\theta\in\Theta}$ est identifiable, alors la famille des mélanges finis de densités produits de la forme (2.3) est identifiable. Ce résultat intervient de la même manière pour prouver l'identifiabilité des modèles de Markov cachés plus généraux.

En définitive, si le processus caché est irréductible (par exemple si $\forall \boldsymbol{s},\ P_\lambda(\boldsymbol{S} = \boldsymbol{s}) > 0$) et si la famille des mélanges finis de $(P_\theta)_{\theta\in\Theta}$ est identifiable, alors le modèle de Markov caché est identifiable.

Remarque 2.1 *Dans la définition (2.2), on suppose que les paramètres d'émission sont deux à deux distincts. Si le processus caché est irréductible, l'absence de cette condition entraîne en général la non identifiabilité. Dans le cas d'un processus caché réductible, l'égalité de paramètres d'émission peut ne pas causer la non identifiabilité. Par exemple, le modèle de chaîne de Markov caché de matrice de transition*

$$P = \begin{bmatrix} a & 1-a & 0 \\ 0 & a' & 1-a' \\ 0 & 0 & 1 \end{bmatrix}$$

avec $a \in]0;1[$ et $a' \in]0;1[$, l'état initial ayant pour loi $\pi = [1 \quad 0 \quad 0]$ et avec les paramètres d'émission $(\theta, \theta', \theta)$ d'une famille identifiable, avec $\theta \neq \theta'$, est identifiable.

Le problème de l'identifiabilité des chaînes de Markov cachées où la chaîne cachée est réductible ou non, est traité dans Ito et al., 1992 [62].

2.3 Introduction à l'algorithme EM

Nous nous plaçons dans le cadre d'un modèle de paramètre λ inconnu. Nous supposons dans ce chapitre que la structure $\mathcal{G}$ du modèle est connue, ainsi que l'ensemble $\mathcal{S}$ des valeurs prises par le processus caché et la famille $(P_\theta)_{\theta\in\Theta}$ des lois d'émission. Nous nous intéressons, dans cette section, à l'estimation de λ par maximum de vraisemblance à partir d'une réalisation $\boldsymbol{y}$ du processus observé. Nous envisagerons également le cas où R réalisations indépendantes $(\boldsymbol{y}^{(1)}, \ldots, \boldsymbol{y}^{(R)})$ sont disponibles pour estimer les paramètres. Dans un premier temps, nous rappelons les propriétés de l'estimateur de maximum de vraisemblance dans le cas des chaînes de Markov cachées. En supposant que ces propriétés sont, dans une certaine mesure, généralisables à des modèles de Markov cachés autres que les chaînes, nous présentons le principe de la maximisation de la vraisemblance par l'algorithme EM. Nous donnons les étapes de cet algorithme pour la famille $\mathcal{D}$ dans le cas de modèles non homogènes, puis dans le cas de modèles homogènes. Nous remarquons ensuite que dans certaines situations, l'algorithme EM consiste à remplacer par leur espérance conditionnelle les quantités rendues indisponibles par la présence des états cachés. Ceci permet de considérer EM comme un algorithme de restauration-maximisation, dans ce cas particulier. Nous envisageons ensuite d'autres stratégies de restauration. Enfin, nous détaillons l'application de notre algorithme aux arbres de Markov cachés de Crouse, Nowak et Baraniuk, 1998 [32], en montrant que nos formules génériques pour l'étape M coïncident, lorsqu'elles sont instanciées pour ce modèle particulier, avec celles des auteurs.

La vraisemblance est définie, dans le cas de modèles à données cachées $\boldsymbol{S}$, par

$$\mathcal{L}_{\boldsymbol{y}}(\lambda) = P_\lambda(\boldsymbol{Y} = \boldsymbol{y}).$$

Cette expression est rendue explicite par la formule des probabilités totales, en faisant intervenir les valeurs cachées

$$\mathcal{L}_{\boldsymbol{y}}(\lambda) = \sum_{\boldsymbol{s} \in \mathcal{S}} P_\lambda(\boldsymbol{Y} = \boldsymbol{y}, \boldsymbol{S} = \boldsymbol{s}), \tag{2.4}$$

où $P_\lambda(\boldsymbol{Y} = \boldsymbol{y}, \boldsymbol{S} = \boldsymbol{s})$ s'exprime comme un produit de probabilités faisant intervenir les paramètres du modèle, d'après l'équation (1.5) de la section 1.4. Quand la dépendance vis-à-vis de $\boldsymbol{y}$ n'a pas besoin d'être explicitée, nous noterons la vraisemblance $\mathcal{L}(\lambda)$. Pour une valeur $\boldsymbol{s}$ du processus caché, nous appellerons la quantité $\mathcal{L}_{\boldsymbol{y},\boldsymbol{s}}(\lambda) = P_\lambda(\boldsymbol{Y} = \boldsymbol{y}, \boldsymbol{S} = \boldsymbol{s})$ la *vraisemblance de λ évaluée sur les données complètes* $(\boldsymbol{y}, \boldsymbol{s})$, ou plus simplement la *vraisemblance complétée*.

2.3.1 Propriétés de l'estimateur de maximum de vraisemblance

Supposons que la vraisemblance soit deux fois dérivable par rapport à λ. Pour trouver une valeur de λ maximisant la vraisemblance, on peut en théorie chercher une valeur annulant sa dérivée puis s'assurer qu'il s'agit d'un maximum en étudiant sa dérivée seconde en la valeur trouvée.

L'existence d'une solution $\hat{\lambda}$ à ces équations (dites *équations de la vraisemblance*) et les propriétés de convergence en probabilité vers la “vraie” valeur λ_0 du paramètre (sous l'hypothèse que $\boldsymbol{Y}$ suit effectivement la loi P_{λ_0}) ou de normalité asymptotique, lorsque le nombre de variables observées tend vers l'infini (pour autant que cela ait un sens, voir section 1.5), sont des questions difficiles dépassant le cadre de cette thèse. On peut également envisager d'étudier $\hat{\lambda}$ quand R vers l'infini, ce qui paraît a priori un cas plus simple à traiter (réalisations indépendantes d'un processus).

Cependant, les travaux de Leroux, 1992[81], Bickel, Ritov et Rydén, 1998 [10], Le Gland et Mevel, 2000 [79], Jensen et Petersen, 1999 [68] et Douc et Matias, 2001 [41] donnent des conditions sous lesquelles l'existence d'une solution consistante et asymptotiquement gaussienne des équations de la vraisemblance est assurée dans le cas de chaînes de Markov cachées. En particulier, l'hypothèse d'ergodicité est nécessaire à la consistance de l'estimateur. L'idée est que si toutes les transitions entre états cachés ne sont pas effectuées par le processus une infinité de fois quand le nombre de données observées tend vers l'infini, avec une probabilité égale à un (processus caché non irréductible), des probabilités de transition ne vont pas pouvoir être estimées de manière consistante. Il semble que la preuve de la consistance de $\hat{\lambda}$ dans des modèles de Markov cachés dynamiques repose sur l'extension à ces modèles de la notion d'ergodicité.

Enfin, la vraisemblance n'est pas toujours bornée (voir en particulier Robert, 1992 [105]). Par exemple, c'est le cas pour un modèle de mélange indépendant gaussien ($\{P_\theta | \theta \in \Theta\}$ est la famille des lois gaussiennes). En choisissant autant de valeurs possibles pour les états cachés que de données observées et en prenant pour espérance des lois gaussiennes les données elles-mêmes, on peut faire croître la vraisemblance arbitrairement en prenant des variances tendant vers 0.

2.3.2 Principe de l'algorithme EM

Il s'agit donc de maximiser en λ l'expression (2.4), qui est une somme de produits comportant taille($\mathcal{X}_{\boldsymbol{S}}$) termes, typiquement $K^{N_{\boldsymbol{S}}}$ où $N_{\boldsymbol{S}}$ désigne le nombre de variables cachées du modèle. Chaque paramètre intervient potentiellement dans plusieurs termes de cette somme et plusieurs fois dans chaque terme. C'est pourquoi la résolution des équations de la vraisemblance ne peut que rarement se faire de manière analytique. On a alors recours à des méthodes itératives d'optimisation, dont l'algorithme EM est un exemple classique.

L'algorithme EM a été introduit par Baum *et al.*, 1970 [7] dans le contexte des chaînes de Markov cachées. Puis il a été étendu à des modèles plus généraux par Dempster, Laird et Rubin, 1977 [36]. Il s'agit, dans le cadre d'un modèle à données incomplètes, de créer une suite $(\hat{\lambda}^{(\eta)})_{\eta \in I\!N}$ telle que $(\ln(\mathcal{L}(\hat{\lambda}^{(\eta)})))_{\eta \in I\!N}$ soit croissante. Idéalement, on souhaite que $(\hat{\lambda}^{(\eta)})_{\eta \in I\!N}$ converge vers une solution des équations de vraisemblance. Pour construire cette suite, on part d'une valeur initiale $\lambda^{(0)}$, supposée connue, du paramètre. À l'itération η de l'algorithme, nous disposons d'une valeur courante $\lambda^{(\eta-1)}$ du paramètre. La valeur suivante est obtenue en maximisant, en la variable λ, l'espérance conditionnelle de la log-vraisemblance complétée du paramètre λ, sachant les variables observées, calculée par rapport à la loi $P_{\lambda^{(\eta-1)}}$. On définit donc la fonction Q par

$$Q(\lambda, \lambda^{(\eta-1)}) = I\!E_{\lambda^{(\eta-1)}}[\ln(P_\lambda(\boldsymbol{S}, \boldsymbol{Y})) | \boldsymbol{Y} = \boldsymbol{y}] \tag{2.5}$$

qu'on peut voir comme une notation pour

$$Q(\lambda, \lambda^{(\eta-1)}) = \sum_{\boldsymbol{s} \in \mathcal{S}} \ln(P_\lambda(\boldsymbol{S} = \boldsymbol{s}, \boldsymbol{Y} = \boldsymbol{y})) P_{\lambda^{(\eta-1)}}(\boldsymbol{S} = \boldsymbol{s} | \boldsymbol{Y} = \boldsymbol{y}) \tag{2.6}$$

L'idée générale est que la log-vraisemblance ne peut être maximisée directement, alors que la log-vraisemblance complétée est facile à calculer et à maximiser. La présence de données cachées suggère l'utilisation de l'espérance conditionnelle de la log-vraisemblance complétée, considérée en tant que fonction des variables aléatoires cachées. Cette espérance est conditionnelle aux données observées et évaluée sous la loi $P_{\lambda^{(\eta-1)}}$ où $\lambda^{(\eta-1)}$ est la valeur courante du paramètre. La valeur suivante du paramètre est alors définie par

$$\hat{\lambda}^{(\eta)} = \arg\max_{\lambda\in\Lambda} Q(\lambda, \lambda^{(\eta-1)})$$

Soit $\mathcal{G} = (\mathcal{U}, \mathcal{E})$ la structure du modèle graphique, supposée connue. Pour $u \in \mathcal{U}$, on note X_u la variable aléatoire associée au sommet u du graphe, quand il n'est pas nécessaire de distinguer entre les variables cachées et observées. Nous désignons par $\boldsymbol{X}$ l'ensemble des variables aléatoires, cachées ou observées, du modèle. L'ensemble des indices des variables aléatoires cachées (respectivement, des variables aléatoires observées) est noté $\mathcal{U}_{\boldsymbol{S}}$ (respectivement $\mathcal{U}_{\boldsymbol{Y}}$). Nous notons $\mathcal{S}$ l'ensemble des valeurs possibles pour le processus caché $\boldsymbol{S}$ et $\mathcal{Y}$ l'ensemble des valeurs possibles pour le processus observé $\boldsymbol{Y}$. Soit $A \subset U$: nous notons $\boldsymbol{S}_A$ l'ensemble des variables aléatoires cachées dont les indices sont dans A, et $\mathcal{S}_A$ l'ensemble des valeurs prises par le processus $\boldsymbol{S}_A$. Ces notations s'étendent aux processus $\boldsymbol{Y}$ et $\boldsymbol{X}$ ainsi qu'aux éléments de $\mathcal{X}$, $\mathcal{S}$ et $\mathcal{Y}$. Par exemple si $\boldsymbol{a} \in \mathcal{X}$, la projection orthogonale de $\boldsymbol{a}$ sur $\mathcal{X}_A$ est désignée par $\boldsymbol{a}_A$.

Dans un premier temps, nous faisons l'hypothèse que $\ln(P_\lambda(\boldsymbol{S} = \boldsymbol{s}, \boldsymbol{Y} = \boldsymbol{y}))$ est défini pour toute valeur $\boldsymbol{s} \in \mathcal{S}$ et pour toute valeur $\lambda \in \Lambda$. Cela revient à supposer, comme le montre l'équation (1.5), que toutes les probabilités $(P(X_u = x_u | \boldsymbol{X}_{\mathrm{pa}(u)} = \boldsymbol{x}_{\mathrm{pa}(u)}))_{u\in\mathcal{U}}$ sont strictement positives. Nous rappelons que ces probabilités sont associées aux paramètres du modèle, ce qui rend facile la vérification de cette hypothèse de positivité. Nous verrons dans la section 2.4.8 que lorsque certaines de ces probabilités sont nulles, l'estimation reste possible.

2.3.3 Mise en œuvre dans le cas de modèles de Markov cachés

En réalité, l'expression (2.6) n'est pas d'une grande utilité tant qu'elle reste sous cette forme de somme comportant taille($\mathcal{X}_{\boldsymbol{S}}$) termes. D'autre part, plus que la fonction Q, c'est la valeur suivante du paramètre $\hat{\lambda}^{(\eta)}$, fonction de $\lambda^{(\eta-1)}$, qui nous est utile. Il est connu que dans les modèles de Markov cachés à structure non orientée, définis par leur structure et par les potentiels des cliques, ces potentiels peuvent être estimés grâce à l'algorithme EM (voir Smyth, Heckerman et Jordan, 1997 [110]). Ici, nous montrons que dans les modèles de Markov cachés à structure orientée acyclique, définis par leur structure et par des probabilités de transition, les paramètres peuvent être estimés grâce à l'algorithme EM. Cependant, les formules de réestimation de l'algorithme EM donnant $\hat{\lambda}^{(\eta)}$ en fonction de $\lambda^{(\eta-1)}$ font en général intervenir des probabilités d'états cachés conditionnellement aux données observées et la manière dont peuvent être calculées ces probabilités n'est pas toujours explicite, ce qui rend l'algorithme EM applicable en théorie seulement. D'autre part, les paramètres d'un modèle de Markov caché général, comme ceux considérés dans Smyth, Heckerman et Jordan (1997), ne sont pas toujours définis de manière précise. Cette imprécision se répercute dans l'algorithme EM. Dans le travail qui suit, cette ambiguïté est levée par le recours à une paramétrisation explicite des modèles de la famille $\mathcal{D}$ (voir section 1.5). Enfin, ce n'est que dans la section 2.4 que nous présenterons un algorithme qui permet effectivement de calculer toutes les quantités intervenant dans l'algorithme EM.

Simplification de l'étape E

En considérant l'équation (1.5), on constate que pour tout u dans $\mathcal{U}$, $P(X_u = x_u|\boldsymbol{X}_{\mathrm{pa}(u)} = \boldsymbol{x}_{\mathrm{pa}(u)})$ est associé à l'un des paramètres du modèle :

- soit $P(X_u = x_u) = \pi^{(u)}_{x_u}$ lorsque u appartient à l'ensemble $\mathcal{U}_\pi$ des sommets sans parents,
- soit $P(X_u = x_u|\boldsymbol{X}_{\mathrm{pa}(u)} = \boldsymbol{x}_{\mathrm{pa}(u)}) = p^{(u)}_{\boldsymbol{x}_{\mathrm{pa}(u)},x_u}$ lorsque u appartient à l'ensemble $\mathcal{U}_{\boldsymbol{S}}\backslash\mathcal{U}_\pi$, noté $\mathcal{U}_p$, des sommets correspondants aux variables aléatoires discrètes admettant au moins un parent,
- soit $P(X_u = x_u|\boldsymbol{X}_{\mathrm{pa}(u)} = \boldsymbol{x}_{\mathrm{pa}(u)}) = P_{\theta^{(u)}_{\boldsymbol{x}_{\mathrm{pa}(u)}}}(x_u)$, associé au paramètre $\theta^{(u)}_{\boldsymbol{x}_{\mathrm{pa}(u)}}$, lorsque X_u est à valeurs continues, ce qui est possible uniquement si $X_u = Y_u$ est l'une des variables aléatoires observées, d'après nos hypothèses. On notera $\mathcal{U}_\theta$ l'ensemble des sommets correspondants.

Ainsi, à partir de l'équation (1.5) on obtient l'expression suivante de la log-vraisemblance complétée

$$\ln(\mathcal{L}_{\boldsymbol{x}}(\lambda)) = \sum_{u\in\mathcal{U}_\theta}\sum_{\boldsymbol{a}\in\mathcal{X}_{\mathrm{pa}(u)}} \ln(P_{\theta^{(u)}_{\boldsymbol{a}}}(y_u))\mathbb{I}_{\{\boldsymbol{x}_{\mathrm{pa}(u)}=\boldsymbol{a}\}} \tag{2.7}$$
$$+\sum_{u\in\mathcal{U}_p}\sum_{\boldsymbol{a}\in\mathcal{X}_{\mathrm{pa}(u)}}\sum_{i\in\mathcal{X}_u}\ln(p^{(\bar{u})}_{\boldsymbol{a},i})\mathbb{I}_{\{x_u=i,\boldsymbol{x}_{\mathrm{pa}(u)}=\boldsymbol{a}\}} + \sum_{u\in\mathcal{U}_\pi}\sum_{i\in\mathcal{X}_u}\ln(\pi^{(u)}_i)\mathbb{I}_{\{x_u=i\}}$$

De cette expression et de la définition (2.5), en remarquant que $\{\mathcal{U}_\pi, \mathcal{U}_p, \mathcal{U}_\theta\}$ constitue une partition de $\mathcal{U}$, on déduit l'expression suivante de $Q(\lambda, \lambda^{(\eta-1)})$

$$Q(\lambda, \lambda^{(\eta-1)}) = \tag{2.8}$$
$$\sum_{u\in\mathcal{U}_\theta}\sum_{\boldsymbol{a}\in\mathcal{X}_{\mathrm{pa}(u)}} \ln(P_{\theta^{(u)}_{\boldsymbol{a}}}(y_u))P_{\lambda^{(\eta-1)}}(\boldsymbol{S}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}}, \boldsymbol{Y}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}}|\boldsymbol{Y} = \boldsymbol{y})$$
$$+\sum_{u\in\mathcal{U}_p}\sum_{\boldsymbol{a}\in\mathcal{X}_{\mathrm{pa}(u)}}\sum_{i\in\mathcal{X}_u}\ln(p^{(u)}_{\boldsymbol{a},i})P_{\lambda^{(\eta-1)}}(X_u = i, \boldsymbol{S}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}}, \boldsymbol{Y}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}}|\boldsymbol{Y} = \boldsymbol{y})$$
$$+\sum_{u\in\mathcal{U}_\pi}\sum_{i\in\mathcal{X}_u}\ln(\pi^{(u)}_i)P_{\lambda^{(\eta-1)}}(X_u = i|\boldsymbol{Y} = \boldsymbol{y}),$$

étant entendu que pour deux ensembles $A \subset \mathcal{U}$ et $B \subset \mathcal{U}$, et pour une valeur $a \in \mathcal{X}$ du processus complet $\boldsymbol{X}$,

$$P_{\lambda^{(\eta-1)}}(\boldsymbol{S}_A = \boldsymbol{a}_A, \boldsymbol{Y}_B = \boldsymbol{a}_B|\boldsymbol{Y} = \boldsymbol{y}) = P_{\lambda^{(\eta-1)}}(\boldsymbol{S}_A = \boldsymbol{a}_A|\boldsymbol{Y} = \boldsymbol{y})\mathbb{I}_{\{\boldsymbol{y}_B=\boldsymbol{a}_B\}}.$$

Par exemple, si X_u est la variable cachée S_u, alors la quantité $P_{\lambda^{(\eta-1)}}(S_u = i|\boldsymbol{Y} = \boldsymbol{y})$ intervient dans le dernier terme de l'expression (2.8) mais si X_u est la variable observée Y_u, alors c'est la quantité $\mathbb{I}_{\{y_u=i\}}$ qui intervient en (2.8).

Remarquons que les expressions (2.7) et (2.8) se décomposent en des termes dépendant à chaque fois d'un seul paramètre. Cette propriété de séparabilité, commune à tous les modèles de Markov cachés à structure orientée acyclique, explique en partie la faisabilité de l'algorithme EM.

En définitive, les probabilités apparaissant dans l'expression (2.8) se résument aux quantités

$$P_{\lambda^{(\eta-1)}}(\boldsymbol{S}_{\mathrm{pa}(u)} = \boldsymbol{s}_{\mathrm{pa}(u)}|\boldsymbol{Y} = \boldsymbol{y}),\ P_{\lambda^{(\eta-1)}}(S_u = i, \boldsymbol{S}_{\mathrm{pa}(u)} = \boldsymbol{s}_{\mathrm{pa}(u)}|\boldsymbol{Y} = \boldsymbol{y})$$
$$\text{et } P_{\lambda^{(\eta-1)}}(S_u = i|\boldsymbol{Y} = \boldsymbol{y}).$$

En principe, l'algorithme d'arbre de jonction permet de les calculer pour tous les sommets u du graphe et toutes les valeurs possibles $\boldsymbol{s}_{\mathrm{pa}(u)}$ et i des états cachés. Cette étape est appelée l'*étape E* de l'algorithme EM. Remarquons que la valeur de $Q(\hat{\lambda}^{(\eta)}, \lambda^{(\eta-1)})$ n'a pas besoin d'être calculée numériquement : seule la valeur de $\hat{\lambda}^{(\eta)}$, qui se déduit des probabilités conditionnelles des états cachés ci-dessus, doit être calculée. Nous verrons que dans le cas de modèles de la classe $\mathcal{D}$, pour tout sommet $u \in \mathcal{U}$ du graphe, $\{u\} \cup \mathrm{pa}(u)$ est inclus dans l'une des cliques de l'arbre de jonction (voir la propriété 1 et la remarque qui suit sa démonstration). Il suffira donc de savoir calculer $P(\boldsymbol{S}_{\mathcal{C}} = \boldsymbol{s}_{\mathcal{C}}|\boldsymbol{Y} = \boldsymbol{y})$ pour toutes les cliques $\mathcal{C}$ du graphe $\mathcal{G}$. Le calcul se fait, pour les modèles admettant un arbre de jonction, au moyen d'un algorithme de type arrière-avant. Celui-ci est décrit en section 2.4

Étape M :
lois des sources dans le cas non homogène

Le calcul du paramètre $\hat{\lambda}^{(\eta)}$ à l'itération suivante η de l'algorithme EM passe par la maximisation de cette fonction $Q(\lambda, \lambda^{(\eta-1)})$ par rapport aux quantités $\theta_{\boldsymbol{a}}^{(u)}$, $p_{\boldsymbol{a},i}^{(u)}$ et $\pi_i^{(u)}$. Nous nous plaçons, pour commencer, dans le cadre dit non homogène où les paramètres dépendent de u. Nous établissons les formules de réestimation dans ce cadre non homogène pour mettre en évidence l'impossibilité ou la trivialité de l'estimation par maximum de vraisemblance pour ces modèles à partir d'une seule réalisation du processus observé (voir remarque 2.2). L'origine et les conséquences de cette difficulté ne sont pas forcément immédiates à appréhender ; de plus les calculs ci-dessus utilisent des principes qui seront repris dans le cas homogène, c'est pourquoi nous les développons malgré leur manque pratique d'utilité.

Pour un sommet $u \in \mathcal{U}$ donné dans le graphe, et pour $i \in \mathcal{X}_u$, le paramètre $\pi_i^{(u)}$ intervient exactement dans un seul terme de la somme (2.8). Comme X_u est à valeurs finies, on peut par exemple supposer, sans perte de généralité, que $\mathcal{X}_u = \{1, \ldots, K\}$. Nous sommes alors amenés à maximiser

$$\sum_{i=1}^{K} \ln(\pi_i^{(u)}) P_{\lambda^{(\eta-1)}}(X_u = i|\boldsymbol{Y} = \boldsymbol{y}) \tag{2.9}$$

sous la contrainte $\sum_{j=1}^{K} \pi_j^{(u)} = 1$, car les $(\pi_j^{(u)})_{1 \leq j \leq K}$ définissent une loi de probabilité. Soit ξ le multiplicateur de Lagrange associé à cette contrainte. Les valeurs maximisant (2.9) sont données par la résolution des équations

$$\forall i \in \{1, \ldots, K\},\ \nabla_{\pi_i^{(u)}} \left[\sum_{j=1}^{K} \ln(\pi_j^{(u)}) P_{\lambda^{(\eta-1)}}(X_u = j|\boldsymbol{Y} = \boldsymbol{y}) + \xi \left(\sum_{j=1}^{K} \pi_j^{(u)} - 1 \right) \right] = 0$$

ce qui équivaut à

$$\forall i \in \{1,\ldots,K\}, \quad \frac{P_{\lambda^{(\eta-1)}}(X_u = i|\boldsymbol{Y} = \boldsymbol{y})}{\pi_i^{(u)}} = -\xi$$

On déduit de la contrainte $\sum_{j=1}^{K} \pi_j^{(u)} = 1$ les équations

$$\begin{aligned} \xi &= -\sum_{i=1}^{K} P_{\lambda^{(\eta-1)}}(X_u = i|\boldsymbol{Y} = \boldsymbol{y}) \\ \text{et } \forall i \in \{1,\ldots,K\}, \quad \hat{\pi}_i^{(u)} &= \frac{P_{\lambda^{(\eta-1)}}(X_u = i|\boldsymbol{Y} = \boldsymbol{y})}{\sum_j P_{\lambda^{(\eta-1)}}(X_u = j|\boldsymbol{Y} = \boldsymbol{y})}, \end{aligned}$$

ce qui constitue la valeur du paramètre $\pi_i^{(u)}$ à l'itération η de l'algorithme EM. Comme

$$\sum_j P_{\lambda^{(\eta-1)}}(X_u = j|\boldsymbol{Y} = \boldsymbol{y}) = 1,$$

on obtient $\hat{\pi}_i^{(u)} = P_{\lambda^{(\eta-1)}}(X_u = i|\boldsymbol{Y} = \boldsymbol{y})$.

Notons que dans le cas où $X_u = Y_u$ est l'une des variables aléatoires observées, la formule de réestimation de $\hat{\pi}^{(u)}$ est donnée par $\hat{\pi}_i^{(u)} = P_{\lambda^{(\eta-1)}}(Y_u = i|\boldsymbol{Y} = \boldsymbol{y}) = \delta_{i,y_u}$.

Estimation des probabilités de transition dans le cas non homogène

On procède de la même manière pour estimer $p_{\boldsymbol{a},i}^{(u)}$ où $i \in \mathcal{X}_u$ et $\boldsymbol{a} \in \mathcal{X}_{\mathrm{pa}(u)}$ sont fixés. Nous notons $\{1,\ldots,K\}$ l'ensemble des valeurs possibles de X_u. Rappelons que $\boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}}$ désigne la projection orthogonale de $\boldsymbol{a}$ sur $\mathcal{Y}$. Dans le cas où $\mathrm{pa}(u) \cap \mathcal{U}_{\boldsymbol{Y}} \neq \emptyset$, il est évident que $p_{\boldsymbol{a},i}^{(u)}$ ne peut être estimé par maximum de vraisemblance que si $\boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}} = \boldsymbol{y}_{\mathrm{pa}(u)}$ (dans le cas contraire, la vraisemblance est indépendante de ce paramètre), ce qui n'est vérifié que pour une seule valeur de $\boldsymbol{a}$ parmi toutes celles possibles quand $\boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}}$ décrit $\mathcal{Y}_{\mathrm{pa}(u)}$. Si cette condition est vérifiée, alors $\hat{p}_{\boldsymbol{a},i}^{(u)}$ est déterminé par la maximisation de la quantité suivante

$$\sum_{j=1}^{K} \ln(p_{\boldsymbol{a},j}^{(u)}) P_{\lambda^{(\eta-1)}}(X_u = j, \boldsymbol{S}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}}, \boldsymbol{Y}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}}|\boldsymbol{Y} = \boldsymbol{y})$$

sous la contrainte $\sum_j p_{\boldsymbol{a},j}^{(u)} = 1$.

En s'inspirant du cas de la loi des sources traité précédemment, on obtient finalement, pour $i \in \{1,\ldots,K\}$,

$$\hat{p}_{\boldsymbol{a},i}^{(u)} = \frac{P_{\lambda^{(\eta-1)}}(X_u = i, \boldsymbol{S}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}}, \boldsymbol{Y}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}}|\boldsymbol{Y} = \boldsymbol{y})}{\sum_j P_{\lambda^{(\eta-1)}}(X_u = j, \boldsymbol{S}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}}, \boldsymbol{Y}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}}|\boldsymbol{Y} = \boldsymbol{y})}$$

ce qui donne en fait

$$\hat{p}_{\boldsymbol{a},i}^{(u)} = P_{\lambda^{(\eta-1)}}(X_u = i|\boldsymbol{S}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}}, \boldsymbol{Y} = \boldsymbol{y}).$$

Notons que dans le cas où $X_u = Y_u$ est l'une des variables aléatoires observées, la formule de réestimation de $\hat{p}^{(u)}_{\boldsymbol{a},i}$ est donnée par $\hat{p}^{(u)}_{\boldsymbol{a},i} = \delta_{i,y_u}$.

Estimation des paramètres d'émission dans le cas non homogène

Soit $\boldsymbol{a} \in \mathcal{X}_{\mathrm{pa}(u)}$, fixé. Comme pour les probabilités de transition, dans le cas où $\mathrm{pa}(u) \cap \mathcal{U}_{\boldsymbol{Y}} \neq \emptyset$, le paramètre $\theta^{(u)}_{\boldsymbol{a}}$ ne peut être estimé par maximum de vraisemblance que si $\boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}} = \boldsymbol{y}_{\mathrm{pa}(u)}$ (dans le cas contraire, la vraisemblance est indépendante de ce paramètre). Dans ce cas, l'estimateur $\hat{\theta}^{(u)}_{\boldsymbol{a}}$ à l'itération η de l'algorithme EM est donné par la résolution des équations :

$$\nabla_{\theta^{(u)}_{\boldsymbol{a}}} \ln(P_{\theta^{(u)}_{\boldsymbol{a}}}(y_u)) P_{\lambda^{(\eta-1)}}(\boldsymbol{S}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}}, \boldsymbol{Y}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}} | \boldsymbol{Y} = \boldsymbol{y}) = 0$$

soit encore, puisque l'on a supposé $P_{\lambda^{(\eta-1)}}(\boldsymbol{S}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}}, \boldsymbol{Y}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}} | \boldsymbol{Y} = \boldsymbol{y}) > 0$ (la quantité $Q(\lambda, \lambda^{(\eta-1)})$ étant indépendante de $\theta^{(u)}_{\boldsymbol{a}}$ dans le cas contraire),

$$\frac{\nabla_{\theta^{(u)}_{\boldsymbol{a}}} P_{\theta^{(u)}_{\boldsymbol{a}}}(y_u)}{P_{\theta^{(u)}_{\boldsymbol{a}}}(y_u)} = 0$$

d'où, dans le cas non homogène, $\hat{\theta}^{(u)}_{\boldsymbol{a}} = \arg\max_{\theta^{(u)}_{\boldsymbol{a}}} P_{\theta^{(u)}_{\boldsymbol{a}}}(y_u)$.

La réestimation des paramètres à l'itération η à partir des paramètres obtenus à l'itération $\eta - 1$ par maximisation de la fonction $Q(., \lambda^{(\eta-1)})$ est appelée *étape M* de l'algorithme EM.

Remarque 2.2 *Les formules de réestimation mettent en évidence le peu d'intérêt d'un modèle non homogène, au moins dans le cas où une seule réalisation du processus observé est utilisée pour estimer les paramètres. En effet, la plupart des probabilités de transition associées à un sommet ne peuvent être estimées dès lors que ce sommet admet des variables aléatoires observées parmi ses parents. Les lois des variables aléatoires observées discrètes sont estimées par des lois de Dirac. Les lois des sources cachées S_u (variables aléatoires cachées dépourvues de parents, appelées également états initiaux) sont estimées par les probabilités $P_{\lambda^{(\eta-1)}}(S_u = i|\boldsymbol{Y} = \boldsymbol{y})$ mais on peut montrer aisément que l'estimateur de maximum de vraisemblance induit une loi de Dirac pour la loi de S_u : en effet, d'après les équations (2.4) et (1.5), et si l'on note $\mathcal{S}_u = \{1, \ldots, K\}$, la vraisemblance est linéaire en chacune des composantes $\pi^{(u)}_i$ donc admet son maximum en l'un des sommets du simplexe $\{(\pi_1, \ldots, \pi_K) \in [0;1]^K \mid \sum_j \pi_j = 1\}$. C'est également le cas des probabilités de transition puisque chaque paramètre intervient au plus une fois dans la vraisemblance, qui est une somme de produits de paramètres. Enfin, les paramètres d'émission sont estimés à l'aide d'une seule donnée observée.*

Estimation dans les modèles homogènes

En réalité, ces estimateurs triviaux s'expliquent par le fait que le nombre de paramètres à estimer est plus important que le nombre d'observations : autant de paramètres vectoriels que de sommets dans le graphe ! Si l'on s'intéresse à des modèles de Markov cachés dynamiques, constitués par des successions de motifs identiques, il est possible de faire l'hypothèse que les probabilités de transition et d'émission ne dépendent pas de la position du motif dans le graphe. Ainsi, la façon dont se propagent les états cachés, des parents d'un sommet au sommet descendant, est la même pour tous les motifs ; de même pour la façon dont sont émises les observations. Ceci se traduit par l'égalité des paramètres associés à ces transitions et ces émissions. Dès lors, le paramètre $p^{(u)}_{\boldsymbol{x}_{\mathrm{pa}(u)},x_u}$ ne dépend plus de u mais seulement de la classe ${}^p\bar{u}$ de u pour la relation d'équivalence $\doteq_p$ définie sur $\mathcal{U}_p$.

$$u \doteq_p v \Leftrightarrow [\, \mathcal{X}_u = \mathcal{X}_v,\, \mathcal{X}_{\mathrm{pa}(u)} = \mathcal{X}_{\mathrm{pa}(v)} \text{ et } \forall \boldsymbol{a} \in \mathcal{X}_{\mathrm{pa}(u)},\, \forall i \in \mathcal{X}_u,\, p^{(u)}_{\boldsymbol{a},i} = p^{(v)}_{\boldsymbol{a},i} \,]$$

Nous notons alors $p^{(\bar{u})}_{\boldsymbol{a},i}$ le paramètre correspondant pour $\boldsymbol{a} \in \mathcal{X}_{\mathrm{pa}(u)}$ et $i \in \mathcal{X}_u$. Ces espaces étant par définition dépendants de la classe d'équivalence de u seulement, nous les notons respectivement $\mathcal{X}_{\mathrm{pa}(\bar{u})}$ et $\mathcal{X}_{\bar{u}}$. Le paramètre $p^{(\bar{u})}_{\boldsymbol{a},i}$ est commun à tous les sommets de $\mathcal{U}_{\bar{u}}$. Nous notons $\bar{\mathcal{U}}_p$ l'ensemble quotient pour la relation $\doteq_p$. Nous définissons de la même manière la relation d'équivalence $\doteq_\pi$ sur l'ensemble $\mathcal{U}_\pi$ des sources de $\mathcal{U}$, $\pi^{(\bar{u})}_i$ les paramètres associés aux sommets de $\mathcal{U}_\pi$ et $\bar{\mathcal{U}}_\pi$ l'ensemble quotient, puis la relation d'équivalence $\doteq_\theta$ sur l'ensemble $\mathcal{U}_\theta$ des variables aléatoires à valeurs continues de $\mathcal{U}$, $\theta^{(\bar{u})}_p$ les paramètres associés aux sommets de $\mathcal{U}_\theta$ et $\bar{\mathcal{U}}_\theta$ l'ensemble quotient. Lorsque la relation d'équivalence utilisée pour un sommet $u \in \mathcal{U}$ est claire d'après le contexte, nous noterons $\bar{u}$ au lieu de ${}^p\bar{u}$, ${}^\pi\bar{u}$ ou ${}^\theta\bar{u}$.

La log-vraisemblance complétée s'écrit alors

$$\begin{aligned}\ln(\mathcal{L}_{\boldsymbol{x}}(\lambda)) &= \sum_{\bar{u}\in\bar{\mathcal{U}}_\theta}\sum_{u\in\bar{u}}\sum_{\boldsymbol{a}\in\mathcal{X}_{\mathrm{pa}(u)}} \ln(P_{\theta^{(\bar{u})}_{\boldsymbol{a}}}(y_u))\mathbb{I}_{\{\boldsymbol{x}_{\mathrm{pa}(u)}=\boldsymbol{a}\}} \\ &+ \sum_{\bar{u}\in\bar{\mathcal{U}}_p}\sum_{u\in\bar{u}}\sum_{\boldsymbol{a}\in\mathcal{X}_{\mathrm{pa}(u)}}\sum_{i\in\mathcal{X}_u} \ln(p^{(\bar{u})}_{\boldsymbol{a},i})\mathbb{I}_{\{x_u=i,\boldsymbol{x}_{\mathrm{pa}(u)}=\boldsymbol{a}\}} \\ &+ \sum_{\bar{u}\in\bar{\mathcal{U}}_\pi}\sum_{u\in\bar{u}}\sum_{i\in\mathcal{X}_u} \ln(\pi^{(\bar{u})}_i)\mathbb{I}_{\{x_u=i\}}\end{aligned} \tag{2.10}$$

On en déduit l'expression suivante de $Q(\lambda, \lambda^{(\eta-1)})$

$$\begin{aligned}&Q(\lambda, \lambda^{(\eta-1)}) = \\ &\quad \sum_{\bar{u}\in\bar{\mathcal{U}}_\theta}\sum_{u\in\bar{u}}\sum_{\boldsymbol{a}\in\mathcal{X}_{\mathrm{pa}(u)}} \ln(P_{\theta^{(\bar{u})}_{\boldsymbol{a}}}(y_u))P_{\lambda^{(\eta-1)}}(\boldsymbol{S}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}}, \boldsymbol{Y}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}}|\boldsymbol{Y} = \boldsymbol{y}) \\ &\quad + \sum_{\bar{u}\in\bar{\mathcal{U}}_p}\sum_{u\in\bar{u}}\sum_{\boldsymbol{a}\in\mathcal{X}_{\mathrm{pa}(u)}}\sum_{i\in\mathcal{X}_u} \ln(p^{(\bar{u})}_{\boldsymbol{a},i})P_{\lambda^{(\eta-1)}}(X_u = i, \boldsymbol{S}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}}, \boldsymbol{Y}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}}|\boldsymbol{Y} = \boldsymbol{y}) \\ &\quad + \sum_{\bar{u}\in\bar{\mathcal{U}}_\pi}\sum_{u\in\bar{u}}\sum_{i\in\mathcal{X}_u} \ln(\pi^{(\bar{u})}_i)P_{\lambda^{(\eta-1)}}(X_u = i|\boldsymbol{Y} = \boldsymbol{y}).\end{aligned}$$

Comme dans le cas non homogène, l'expression ci-dessus se décompose en des termes dépendant chacun d'un seul paramètre. Ainsi, elle peut être maximisée séparément par rapport à chacun des paramètres.

Étape M dans le cas homogène :
loi des sources

Soit $\bar{u} \in \bar{\mathcal{U}}_\pi$. La réestimation, à l'itération η de l'algorithme EM, de $\pi_i^{(\bar{u})}$ pour $i \in \mathcal{X}_{\bar{u}}$, se fait par maximisation de la quantité

$$\sum_{u \in \bar{u}} \sum_i \ln(\pi_i^{(\bar{u})}) P_{\lambda^{(\eta-1)}}(X_u = i | \boldsymbol{Y} = \boldsymbol{y})$$

sous la contrainte $\sum_j \pi_j^{(\bar{u})} = 1$. La solution $\hat{\pi}^{(\bar{u})}$ est donnée par la résolution des équations

$$\nabla_{\pi_i^{(\bar{u})}} \left[\sum_{u \in \bar{u}} \sum_j \ln(\pi_j^{(\bar{u})}) P_{\lambda^{(\eta-1)}}(X_u = j | \boldsymbol{Y} = \boldsymbol{y}) + \xi \left(\sum_j \pi_j^{(\bar{u})} - 1 \right) \right] = 0_{\mathbb{R}^K}$$

où ξ désigne le multiplicateur de Lagrange associé à la contrainte. Ceci conduit aux formules

$$\begin{aligned} \xi = \sum_j \sum_{u \in \bar{u}} P_{\lambda^{(\eta-1)}}(X_u = j | \boldsymbol{Y} = \boldsymbol{y}) &= \sum_{u \in \bar{u}} 1 = \mathrm{card}(\bar{u}) \\ \text{et } \hat{\pi}_i^{(\bar{u})} &= \frac{\sum_{u \in \bar{u}} P_{\lambda^{(\eta-1)}}(X_u = i | \boldsymbol{Y} = \boldsymbol{y})}{\mathrm{card}(\bar{u})}. \end{aligned} \tag{2.11}$$

Notons que dans le cas où $X_u = Y_u$ est l'une des variables aléatoires observées, la formule de réestimation de $\hat{\pi}^{(\bar{u})}$ est donnée par

$$\hat{\pi}_i^{(\bar{u})} = \frac{\sum_{u \in \bar{u}} \delta_{i, y_u}}{\mathrm{card}(\bar{u})} = \frac{\mathrm{card}(\{u \in \bar{u} \,|\, y_u = i\})}{\mathrm{card}(\bar{u})} \tag{2.12}$$

Probabilités de transition

Soit $\bar{u} \in \bar{\mathcal{U}}_p$. Il s'agit de réestimer $p_{\boldsymbol{a},i}^{(\bar{u})}$ pour $i \in \mathcal{X}_{\bar{u}}$ et $\boldsymbol{a} \in \mathcal{X}_{\mathrm{pa}(\bar{u})}$ fixés. Comme pour les modèles non homogènes, si $\mathrm{pa}(u) \cap \mathcal{U}_{\boldsymbol{Y}} \neq \emptyset$, le paramètre $p_{\boldsymbol{a},i}^{(u)}$ ne peut être estimé par maximum de vraisemblance que si $\boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}} = \boldsymbol{y}_{\mathrm{pa}(u)}$ pour l'un au moins des éléments u de $\bar{u}$. La différence avec les modèles non homogènes est que le fait que $p_{\boldsymbol{a},i}^{(\bar{u})}$ puisse être estimée pour toutes les valeurs de $\boldsymbol{a}$ n'est pas exclu d'emblée. D'autre part, la probabilité que $p_{\boldsymbol{a},i}^{(\bar{u})}$ puisse être estimée croît avec le cardinal de $\bar{u}$, donc avec le nombre de données observées n dans la plupart des cas d'intérêt.

Si $p_{\boldsymbol{a},i}^{(\bar{u})}$ peut effectivement être estimé, alors $\hat{p}_{\boldsymbol{a},i}^{(\bar{u})}$ est donné par la maximisation de la quantité

$$\sum_{u \in \bar{u}} \sum_{i \in \mathcal{X}_u} \ln(p_{\boldsymbol{a},i}^{(\bar{u})}) P_{\lambda^{(\eta-1)}}(X_u = i, \boldsymbol{S}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}}, \boldsymbol{y}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}} | \boldsymbol{Y} = \boldsymbol{y})$$

Comme précédemment, on obtient la formule de réestimation suivante

$$\hat{p}_{\boldsymbol{a},i}^{(\bar{u})} = \frac{\sum_{u\in\bar{u}} P_{\lambda^{(\eta-1)}}(X_u = i, \boldsymbol{S}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}}, \boldsymbol{Y}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}} | \boldsymbol{Y} = \boldsymbol{y})}{\sum_{u\in\bar{u}} P_{\lambda^{(\eta-1)}}(\boldsymbol{S}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}}, \boldsymbol{Y}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}} | \boldsymbol{Y} = \boldsymbol{y})} \tag{2.13}$$

où les probabilités $P_{\lambda^{(\eta-1)}}(\boldsymbol{S}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}}, \boldsymbol{Y}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}} | \boldsymbol{Y} = \boldsymbol{y})$ se calculent à partir des $P_{\lambda^{(\eta-1)}}(X_u = i, \boldsymbol{S}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}}, \boldsymbol{Y}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}} | \boldsymbol{Y} = \boldsymbol{y})$ par

$$\begin{aligned} & P_{\lambda^{(\eta-1)}}(\boldsymbol{S}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}}, \boldsymbol{Y}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}} | \boldsymbol{Y} = \boldsymbol{y}) \\ & \quad = \sum_j P_{\lambda^{(\eta-1)}}(X_u = j, \boldsymbol{S}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}}, \boldsymbol{Y}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}} | \boldsymbol{Y} = \boldsymbol{y}). \end{aligned}$$

Remarquons que

$$\begin{aligned} & P_{\lambda^{(\eta-1)}}(X_u = j, \boldsymbol{S}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}}, \boldsymbol{Y}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}} | \boldsymbol{Y} = \boldsymbol{y}) \\ & = \begin{cases} P_{\lambda^{(\eta-1)}}(S_u = j, \boldsymbol{S}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}} | \boldsymbol{Y} = \boldsymbol{y}) \mathbb{1}_{\{\boldsymbol{y}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}}\}} & \text{si } X_u = S_u \text{ est cachée,} \\ P_{\lambda^{(\eta-1)}}(\boldsymbol{S}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}} | \boldsymbol{Y} = \boldsymbol{y}) \mathbb{1}_{\{\boldsymbol{y}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}}, y_u = i\}} & \text{si } X_u = Y_u \text{ est observée.} \end{cases} \end{aligned}$$

Ainsi, comme annoncé ci-dessus, ces quantités peuvent être calculées si et seulement si $\{u \in \bar{u} \,|\, \boldsymbol{y}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}}\} \neq \emptyset$, faute de quoi les dénominateurs sont nuls.

Paramètres d'émission

Soit $\bar{u} \in \bar{\mathcal{U}}_\theta$. La réestimation, à l'itération η de l'algorithme EM, de $\theta_{\boldsymbol{a}}^{(\bar{u})}$ pour $\boldsymbol{a} \in \mathcal{X}_{\mathrm{pa}(\bar{u})}$, n'est possible, dans le cas où $\mathrm{pa}(u) \cap \mathcal{U}_{\boldsymbol{Y}} \neq \emptyset$, que si $\boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}} = \boldsymbol{y}_{\mathrm{pa}(u)}$ pour l'un au moins des éléments u de $\bar{u}$. Dans ce cas, $\hat{\theta}_{\boldsymbol{a}}^{(\bar{u})}$ est obtenu par maximisation de la quantité

$$\sum_{u\in\bar{u}} \ln(P_{\theta_{\boldsymbol{a}}^{(\bar{u})}}(y_u)) P_{\lambda^{(\eta-1)}}(\boldsymbol{S}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}}, \boldsymbol{Y}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}} | \boldsymbol{Y} = \boldsymbol{y})$$

ce qui revient à résoudre

$$\sum_{u\in\bar{u}} P_{\lambda^{(\eta-1)}}(\boldsymbol{S}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}}, \boldsymbol{Y}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}} | \boldsymbol{Y} = \boldsymbol{y}) \frac{\nabla_{\theta_{\boldsymbol{a}}^{(\bar{u})}}[P_{\theta_{\boldsymbol{a}}^{(\bar{u})}}(y_u)]}{P_{\theta_{\boldsymbol{a}}^{(\bar{u})}}(y_u)} = 0 \tag{2.14}$$

La solution dépend de la famille $\{P_\theta | \theta \in \Theta\}$ considérée. A titre d'exemple, nous donnons les formules de réestimation quand la famille considérée est gaussienne multivariée. Autrement dit, la loi conditionnelle de Y_u sachant $\boldsymbol{X}_{\mathrm{pa}(u)} = \boldsymbol{a}$ est la loi gaussienne de paramètres $\theta_{\boldsymbol{a}} = (\mu_{\boldsymbol{a}}, \Sigma_{\boldsymbol{a}})$, de densité

$$f_{\theta_{\boldsymbol{a}}}(y) = \frac{1}{(2\pi)^{\frac{p}{2}}\sqrt{|\Sigma_{\boldsymbol{a}}|}} \exp(-\frac{1}{2}{}^t(y - \mu_{\boldsymbol{a}})\Sigma_{\boldsymbol{a}}^{-1}(y - \mu_{\boldsymbol{a}})),$$

de gradients

$$\nabla_{\mu_{\boldsymbol{a}}}[f_{\theta_{\boldsymbol{a}}}(y)] = f_{\theta_{\boldsymbol{a}}}(y)\Sigma_{\boldsymbol{a}}^{-1}(y - \mu_{\boldsymbol{a}})$$

et

$$\nabla_{\Sigma_{\boldsymbol{a}}}[f_{\theta_{\boldsymbol{a}}}(y)] = -\frac{1}{2} f_{\theta_{\boldsymbol{a}}}(y) \Sigma_{\boldsymbol{a}}^{-2} \left(\Sigma_{\boldsymbol{a}} - {}^t(y - \mu_{\boldsymbol{a}})(y - \mu_{\boldsymbol{a}}) \right),$$

où la transposée d'une matrice A est notée ${}^tA$. Les équations (2.14) deviennent alors

$$\begin{cases} \sum\limits_{u \in \bar{u}} P_{\lambda^{(\eta-1)}}(\boldsymbol{S}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}}, \boldsymbol{Y}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}} | \boldsymbol{Y} = \boldsymbol{y}) \Sigma_{\boldsymbol{a}}^{-1}(y_u - \mu_{\boldsymbol{a}}) = 0 \\ -\frac{1}{2} \sum\limits_{u \in \bar{u}} P_{\lambda^{(\eta-1)}}(\boldsymbol{S}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}}, \boldsymbol{Y}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}} | \boldsymbol{Y} = \boldsymbol{y}) \Sigma_{\boldsymbol{a}}^{-2} \left[\Sigma_{\boldsymbol{a}} - {}^t(y_u - \mu_{\boldsymbol{a}})(y_u - \mu_{\boldsymbol{a}}) \right] = 0 \end{cases}$$

ce qui conduit aux formules

$$\begin{cases} \hat{\mu}_{\boldsymbol{a}} = \dfrac{\sum\limits_{u \in \bar{u}} P_{\lambda^{(\eta-1)}}(\boldsymbol{S}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}}, \boldsymbol{Y}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}} | \boldsymbol{Y} = \boldsymbol{y}) y_u}{\sum\limits_{u \in \bar{u}} P_{\lambda^{(\eta-1)}}(\boldsymbol{S}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}}, \boldsymbol{Y}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}} | \boldsymbol{Y} = \boldsymbol{y})} \\ \hat{\Sigma}_{\boldsymbol{a}} = \dfrac{\sum\limits_{u \in \bar{u}} P_{\lambda^{(\eta-1)}}(\boldsymbol{S}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}}, \boldsymbol{Y}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}} | \boldsymbol{Y} = \boldsymbol{y}) {}^t(y_u - \hat{\mu}_{\boldsymbol{a}})(y_u - \hat{\mu}_{\boldsymbol{a}})}{\sum\limits_{u \in \bar{u}} P_{\lambda^{(\eta-1)}}(\boldsymbol{S}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}}, \boldsymbol{Y}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}} | \boldsymbol{Y} = \boldsymbol{y})}. \end{cases} \tag{2.15}$$

Remarquons que, de même que pour la réestimation des probabilités de transition,

$$P_{\lambda^{(\eta-1)}}(\boldsymbol{S}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}}, \boldsymbol{Y}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}} | \boldsymbol{Y} = \boldsymbol{y}) = P_{\lambda^{(\eta-1)}}(\boldsymbol{S}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}} | \boldsymbol{Y} = \boldsymbol{y}) \mathbb{I}_{\{\boldsymbol{y}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}}\}}.$$

Ainsi, comme annoncé ci-dessus, ces quantités peuvent être calculées si et seulement si $\{u \in \bar{u} \,|\, \boldsymbol{y}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}}\} \neq \emptyset$, faute de quoi les dénominateurs sont nuls. En outre, l'équation (2.15) correspond aux formules usuelles de l'estimateur de maximum de vraisemblance dans les modèles gaussiens mais où les observations sont pondérées par les probabilités conditionnelles des états cachés. Ces formules sont en fait une conséquence du résultat de Redner et Walker, 1984 [103]. Celui-ci a été montré pour les mélanges indépendants dans le cas de lois d'émission de la famille exponentielle mais il se généralise sans difficulté aux modèles de Markov cachés étudiés ici.

Une famille $(P_\theta)_{\theta \in \Theta}$ est dite *famille exponentielle* si et seulement si ses éléments sont de la forme

$$\begin{array}{rccc} P_\theta : & \mathbb{R}^p & \longrightarrow & \mathbb{R}^+ \\ & y & \longrightarrow & \frac{1}{\alpha(\theta)} b(y) e^{{}^t\theta T(y)} \end{array}$$

où $b : \mathbb{R}^p \longrightarrow \mathbb{R}^+$ et $T : \mathbb{R}^p \longrightarrow \mathbb{R}^{p'}$, $\alpha(\theta)$ étant donné par

$$\alpha(\theta) = \int_{\mathbb{R}^p} b(y) e^{{}^t\theta T(y)} \, dy,$$

en supposant que $\alpha(\theta)$ est fini. Le paramètre θ est appelé le *paramètre naturel* du modèle. Une famille exponentielle peut être reparamétrée à partir de l'espérance

$$\Phi = \mathbb{E}_\theta[T(y)] = \int_{\mathbb{R}^p} T(y) P_\theta(y) \, dy,$$

de sorte que

$$P_\Phi(y) = \frac{1}{\alpha(\Phi)} b(y) e^{{}^t\theta(\Phi) T(y)}.$$

Le résultat de Redner et Walker ne s'appuie pas sur l'hypothèse d'indépendance des $(S_u)_{u\in\mathcal{U}_{\boldsymbol{S}}}$ mais seulement sur les propriétés suivantes :

$$\begin{aligned}
&\ln(P_\Phi(y)) = -\ln(\alpha(\Phi)) + \ln(b(y)) + {}^t\theta(\Phi)T(y),\\
&\nabla_\theta[\alpha(\theta)] = \alpha(\theta)\Phi,\\
\text{et}\qquad &\nabla_\Phi[\ln(P_\Phi(y))] = -\tfrac{\nabla_\Phi[\alpha(\Phi)]}{\alpha(\Phi)} + \nabla_\Phi[\theta(\Phi)]T(y)
\end{aligned}$$

d'où découlent les équations

$$\begin{aligned}
&\nabla_\Phi[\alpha(\theta(\Phi))] = \alpha(\theta(\Phi))\nabla_\Phi[\theta(\Phi)]\Phi\\
\text{et}\quad &\nabla_\Phi[\ln(P_\Phi(y))] = \nabla_\Phi[\theta(\Phi)](T(y) - \Phi).
\end{aligned}$$

On déduit donc de l'équation (2.14) que $\theta_{\boldsymbol{a}}$ est solution de

$$\sum_{u\in\bar{u}} P_{\lambda^{(\eta-1)}}(\boldsymbol{S}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}}, \boldsymbol{Y}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}} | \boldsymbol{Y} = \boldsymbol{y})\nabla_{\Phi_{\boldsymbol{a}}}[\theta_{\boldsymbol{a}}(\Phi_{\boldsymbol{a}})](T(y_u) - \Phi_{\boldsymbol{a}}) = 0$$

ce qui équivaut, sous réserve que $\nabla_{\Phi_{\boldsymbol{a}}}[\theta_{\boldsymbol{a}}(\Phi_{\boldsymbol{a}})]$ soit inversible, à

$$\begin{aligned}
\Phi_{\boldsymbol{a}} &\sum_{u\in\bar{u}} P_{\lambda^{(\eta-1)}}(\boldsymbol{S}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}}, \boldsymbol{Y}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}} | \boldsymbol{Y} = \boldsymbol{y})\\
&= \sum_{u\in\bar{u}} P_{\lambda^{(\eta-1)}}(\boldsymbol{S}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}}, \boldsymbol{Y}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}} | \boldsymbol{Y} = \boldsymbol{y})T(y_u),
\end{aligned}$$

d'où la formule de réestimation pour $\Phi_{\boldsymbol{a}}$

$$\hat{\Phi}_{\boldsymbol{a}} = \frac{\sum\limits_{u\in\bar{u}} P_{\lambda^{(\eta-1)}}(\boldsymbol{S}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}}, \boldsymbol{Y}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}} | \boldsymbol{Y} = \boldsymbol{y})T(y_u)}{\sum\limits_{u\in\bar{u}} P_{\lambda^{(\eta-1)}}(\boldsymbol{S}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}}, \boldsymbol{Y}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}} | \boldsymbol{Y} = \boldsymbol{y})}$$

qui généralise aux modèles de Markov cachés le résultat de Redner et Walker.

Enfin, pour conclure cette partie sur l'étape M dans le cas homogène, on peut remarquer qu'en remplaçant $\bar{u}$ par $\{u\}$ dans les formules (2.11) à (2.14) on retombe sur les formules de réestimation du cas non homogène.

Cas de plusieurs réalisations du processus caché $\boldsymbol{Y}$

Il arrive dans certaines situations de disposer, pour estimer le paramètre λ, de R réalisations indépendantes $(\boldsymbol{y}^{(1)}, \ldots, \boldsymbol{y}^{(R)})$ du processus caché $\boldsymbol{Y}$. S'il s'agit d'un modèle dynamique, il n'est pas nécessaire que toutes les réalisations comportent le même nombre de données. La vraisemblance du paramètre est alors

$$\mathcal{L}_{\boldsymbol{y}^{(1)},\ldots,\boldsymbol{y}^{(R)}}(\lambda) = P_\lambda(\boldsymbol{Y}^{(1)} = \boldsymbol{y}(1), \ldots, \boldsymbol{Y}^{(R)} = \boldsymbol{y}(r)) = \prod_{r=1}^{R} P_\lambda(\boldsymbol{Y}^{(r)} = \boldsymbol{y}(r)) = \prod_{r=1}^{R} \mathcal{L}_{\boldsymbol{y}^{(r)}}(\lambda)$$

par indépendance des différentes réalisations. Si $(\boldsymbol{s}^{(1)}, \ldots, \boldsymbol{s}^{(R)})$ sont les valeurs des processus cachés associés respectivement à chacune des réalisations, et que $\boldsymbol{X}^{(r)}$ désigne l'ensemble des variables aléatoires observées et cachées constituant le $r^{\text{ème}}$processus, alors les $(\boldsymbol{X}^{(r)})_{r\in\{1,\ldots,R\}}$ sont également indépendants mutuellement. Il s'ensuit que la log-vraisemblance complétée s'écrit

$$\begin{aligned}
&\ln(\mathcal{L}_{\boldsymbol{y}^{(1)},\boldsymbol{s}^{(1)},\ldots,\boldsymbol{y}^{(R)},\boldsymbol{s}^{(R)}}(\lambda)) \\
&\quad = \ln(P_\lambda(\boldsymbol{Y}^{(1)} = \boldsymbol{y}^{(1)}, \boldsymbol{S}^{(1)} = \boldsymbol{s}^{(1)}, \ldots, \boldsymbol{Y}^{(R)} = \boldsymbol{y}^{(R)}, \boldsymbol{S}^{(R)} = \boldsymbol{s}^{(R)})) \\
&\quad = \sum_{r=1}^{R} \ln(P_\lambda(\boldsymbol{Y}^{(r)} = \boldsymbol{y}^{(r)}, \boldsymbol{S}^{(r)} = \boldsymbol{s}^{(r)})) = \sum_{r=1}^{R} \ln(\mathcal{L}_{\boldsymbol{y}^{(r)},\boldsymbol{s}^{(r)}}(\lambda))
\end{aligned}$$

Dans le cas où l'algorithme EM est utilisé pour estimer les paramètres, la fonction Q est donnée (pour un modèle homogène) par la même expression que dans le cas où une seule réalisation du processus observé est utilisée, à ceci près que les sommations s'effectuent sur toutes les valeurs de r (c'est-à-dire pour toutes les réalisations indépendantes du processus observé). Ce résultat s'obtient en remarquant que les R processus cachés sont conditionnellement indépendants sachant les processus observés et donc

$$P_{\lambda^{(\eta-1)}}\left(\bigcap_r \{\boldsymbol{S}^{(r)} = \boldsymbol{s}^{(r)}\} | \bigcap_r \{\boldsymbol{Y}^{(r)} = \boldsymbol{y}^{(r)}\}\right) = \prod_r P_{\lambda^{(\eta-1)}}(\boldsymbol{S}^{(r)} = \boldsymbol{s}^{(r)} | \boldsymbol{Y}^{(r)} = \boldsymbol{y}^{(r)})$$

On obtient alors les formules de réestimation suivantes, pour les lois des sources,

$$\hat{\pi}_i^{(\bar{u})} = \frac{\sum_r \sum_{u\in\bar{u}^{(r)}} P_{\lambda^{(\eta-1)}}(X_u = i | \boldsymbol{Y}^{(r)} = \boldsymbol{y}^{(r)})}{\operatorname{card}(\bigcup_r \bar{u}^{(r)})},$$

ce qui se réécrit, si X_u est la variable aléatoire observée Y_u,

$$\hat{\pi}_i^{(\bar{u})} = \frac{\operatorname{card}(\{u \in \bigcup_r \bar{u}^{(r)} \mid y_u^{(r)} = i\})}{\operatorname{card}(\bigcup_r \bar{u}^{(r)})}.$$

Pour les probabilités de transition, les formules de réestimation sont

$$\hat{p}_{\boldsymbol{a},i}^{(\bar{u})} = \frac{\sum_r \sum_{u\in\bar{u}^{(r)}} P_{\lambda^{(\eta-1)}}(X_u = i, \boldsymbol{S}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}}, \boldsymbol{Y}^{(r)}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}} | \boldsymbol{Y}^{(r)} = \boldsymbol{y}^{(r)})}{\sum_r \sum_{u\in\bar{u}^{(r)}} P_{\lambda^{(\eta-1)}}(\boldsymbol{S}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}}, \boldsymbol{Y}^{(r)}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}} | \boldsymbol{Y}^{(r)} = \boldsymbol{y}^{(r)})}$$

Les formules de réestimation des paramètres d'émission sont données par la résolution de

$$\sum_r \sum_{u\in\bar{u}^{(r)}} \frac{\nabla_{\theta_{\boldsymbol{a}}^{(\bar{u})}}[P_{\theta_{\boldsymbol{a}}^{(\bar{u})}}(y_u^{(r)})]}{P_{\theta_{\boldsymbol{a}}^{(\bar{u})}}(y_u^{(r)})} P_{\lambda^{(\eta-1)}}(\boldsymbol{S}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}}, \boldsymbol{Y}^{(r)}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}} | \boldsymbol{Y}^{(r)} = \boldsymbol{y}^{(r)}) = 0$$

Nous donnons, à titre d'exemple, les formules de réestimation des paramètres dans le cas gaussien

$$\left\{\begin{array}{l} \hat{\mu}_{\boldsymbol{a}} = \dfrac{\sum\limits_r \sum\limits_{u \in \bar{u}^{(r)}} P_{\lambda^{(\eta-1)}}(\boldsymbol{S}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}}, \boldsymbol{Y}^{(r)}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}} | \boldsymbol{Y}^{(r)} = \boldsymbol{y}^{(r)}) y_u^{(r)}}{\sum\limits_r \sum\limits_{u \in \bar{u}^{(r)}} P_{\lambda^{(\eta-1)}}(\boldsymbol{S}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}}, \boldsymbol{Y}^{(r)}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}} | \boldsymbol{Y}^{(r)} = \boldsymbol{y}^{(r)})} \\ \hat{\Sigma}_{\boldsymbol{a}} = \dfrac{\sum\limits_r \sum\limits_{u \in \bar{u}^{(r)}} P_{\lambda^{(\eta-1)}}(\boldsymbol{S}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}}, \boldsymbol{Y}^{(r)}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}} | \boldsymbol{Y}^{(r)} = \boldsymbol{y}^{(r)})^t (y_u^{(r)} - \hat{\mu}_{\boldsymbol{a}})(y_u^{(r)} - \hat{\mu}_{\boldsymbol{a}})}{\sum\limits_r \sum\limits_{u \in \bar{u}^{(r)}} P_{\lambda^{(\eta-1)}}(\boldsymbol{S}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}}, \boldsymbol{Y}^{(r)}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}} | \boldsymbol{Y}^{(r)} = \boldsymbol{y}^{(r)})} \end{array}\right.$$

De la même manière que précédemment, notre généralisation aux modèles de Markov cachés du résultat de Redner et Walker, 1984 [103] pour la réestimation des paramètres d'émission dans le cas de familles exponentielles, s'applique pour donner :

$$\hat{\Phi}_{\boldsymbol{a}} = \frac{\sum\limits_r \sum\limits_{u \in \bar{u}^{(r)}} P_{\lambda^{(\eta-1)}}(\boldsymbol{S}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}}, \boldsymbol{Y}^{(r)}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}} | \boldsymbol{Y}^{(r)} = \boldsymbol{y}^{(r)}) T(y_u^{(r)})}{\sum\limits_r \sum\limits_{u \in \bar{u}^{(r)}} P_{\lambda^{(\eta-1)}}(\boldsymbol{S}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}}, \boldsymbol{Y}^{(r)}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}} | \boldsymbol{Y}^{(r)} = \boldsymbol{y}^{(r)})}$$

Ces formules permettent également, en considérant le cas $\bar{u} = \{u\}$, la réestimation dans des modèles non homogènes quand plusieurs réalisations indépendantes du processus observé sont disponibles.

Spécificités de l'estimation pour les modèles de la classe $\mathcal{D}$

Nous avons vu que l'une des caractéristiques de l'étape E, dans les modèles de Markov cachés à structure orientée acyclique, est la décomposition de l'expression $Q(\lambda, \lambda^{(\eta-1)})$ en termes dépendants chacun d'un seul paramètre. Ainsi, on peut opérer la maximisation séparée de cette fonction $Q(., \lambda^{(\eta-1)})$ par rapport à chacun des paramètres. En outre, les paramètres sont associés à des probabilités de transition de parents à enfant pour certains sommets de $\mathcal{G}$, et les quantités associées à ces paramètres dans $Q(\lambda, \lambda^{(\eta-1)})$ sont pondérées par la probabilité conditionnelle de ces sommets sachant les variables observées. Par ailleurs, la principale difficulté de l'algorithme EM est le calcul de ces probabilités conditionnelles.

Les résultats précédents ne reposent en réalité que sur l'hypothèse d'un modèle à structure orientée acyclique et sur la paramétrisation. Leur nouveauté n'est pas évidente : elle tient avant tout à notre façon de définir les modèles de Markov cachés et à la paramétrisation choisie. Nombreuses sont en effet les publications où la question de la paramétrisation est traitée de manière vague, ce qui pose évidemment des problèmes pour décrire l'estimation des paramètres. Notre apport dans ce domaine est avant tout une description rigoureuse du modèle et de sa paramétrisation, opérant une distinction entre modèles homogènes et non homogènes. Il consiste aussi en une présentation et une justification claire de l'étape M de l'algorithme EM pour l'estimation des paramètres dans ces modèles, y compris dans le contexte où l'on ignore *a priori* si certaines variables aléatoires Y_u à valeurs discrètes sont observées ou non, et dans le cas également où des variables aléatoires observées sont à valeurs continues. Dans ce dernier cas, nous avons donné des formules explicites pour la réestimation des paramètres d'émission dans le cas de familles exponentielles, en utilisant et en généralisant au cas non indépendant le résultat de Redner et Walker, 1984.

La spécificité de l'estimation dans les modèles de la classe $\mathcal{D}$ tient donc avant tout à la manière d'implémenter l'étape E. Ceci repose sur plusieurs propositions, dont la suivante.

Proposition 1 *Soit $u \in \mathcal{U}$ un sommet du graphe $\mathcal{G}$ d'indépendance conditionnelle. Alors il existe une clique $\mathcal{C}$ de l'arbre de jonction contenant $\{u\}\cup$ pa(u).*

Démonstration

Par définition, $\forall v \in \mathrm{pa}(u)$, (v, u) est un arc de $\mathcal{G}$. De plus, par moralité de $\mathcal{G}$, $\forall (v, v') \in \mathrm{pa}(u)^2$, $v \neq v' \Rightarrow$ $[(v, v')$ est un arc de $\mathcal{G}$ ou (v', v) est un arc de $\mathcal{G}]$. Par conséquent le graphe $\{u\}\cup$ pa(u) est complet. Il peut être plongé par ajout de sommets dans un graphe complet maximal $\mathcal{C}$ qui est donc une clique de l'arbre de jonction.

Par conséquent, pour pouvoir calculer les probabilités $P_{\lambda^{(\eta-1)}}(S_u = i, \boldsymbol{S}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}} | \boldsymbol{Y} = \boldsymbol{y})$ nécessitées par l'étape E de l'algorithme EM, il suffit de savoir calculer $P(\boldsymbol{S}_{\mathcal{C}} = \boldsymbol{s}_{\mathcal{C}} | \boldsymbol{Y} = \boldsymbol{y})$ pour toutes les cliques $\mathcal{C}$ de l'arbre de jonction et pour toutes les valeurs possibles $\boldsymbol{s}_{\mathcal{C}} \in \mathcal{S}_{\mathcal{C}}$.

De plus, nous montrerons dans la section 2.4.1 que la structure des modèles de la classe $\mathcal{D}$ admet une unique source (proposition 4). Il y aura donc un seul paramètre de type $(\pi_i^u)_{i\in\{1,\ldots,K\}}$ à estimer (noté π). Dans le cas où une seule réalisation du processus observé est utilisée pour estimer les paramètres, nous savons par la remarque 2.2 que l'estimateur de maximum de vraisemblance de π correspond à une loi de Dirac.

2.3.4 Interprétation a posteriori

Au vu des formules de réestimation, nous pouvons interpréter l'algorithme EM comme suit : plaçons-nous dans le contexte de la maximisation de la log-vraisemblance complétée $\ln(\mathcal{L}_{\boldsymbol{y},\boldsymbol{s}}(\lambda)) = \ln(P_\lambda(\boldsymbol{Y} = \boldsymbol{y}, \boldsymbol{S} = \boldsymbol{s}))$. Si la valeur de $\boldsymbol{S}$ était connue, la maximisation de $\ln(\mathcal{L}_{\boldsymbol{y},\boldsymbol{s}}(\lambda))$ reviendrait en substance à effectuer des comptages, puisqu'il s'agirait de maximiser l'expression (2.10). Le cadre est de nouveau celui de modèles de Markov cachés à structure orientée, paramétrés de la façon décrite précédemment. Les autres hypothèses n'interviennent pas à ce point de l'exposé. Nous traitons le cas de modèles homogènes où une seule réalisation du processus observé est utilisée pour estimer les paramètres, mais les autres cas se traitent de la même manière. La maximisation de cette quantité conduit aux *estimateurs de maximum de vraisemblance complétée*, pour les lois des sources,

$$\hat{\pi}_i^{(\bar{u})} = \frac{\sum_{u\in\bar{u}} \mathbb{I}_{\{x_u = i\}}}{\mathrm{card}(\bar{u})}$$

Pour les probabilités de transition, les formules de réestimation sont

$$\hat{p}_{\boldsymbol{a},i}^{(\bar{u})} = \frac{\sum_{u\in\bar{u}} \mathbb{I}_{\{x_u = i, \boldsymbol{x}_{\mathrm{pa}(u)} = \boldsymbol{a}\}}}{\mathrm{card}(\{u \in \bar{u} \,|\, \boldsymbol{x}_{\mathrm{pa}(u)} = \boldsymbol{a}\})}$$

Les formules de réestimation des paramètres d'émission sont données par la résolution de

$$\sum_{u \in \bar{u}} \frac{\nabla_{\theta_{\boldsymbol{a}}^{(\bar{u})}} P_{\theta_{\boldsymbol{a}}^{(\bar{u})}}(y_u)}{P_{\theta_{\boldsymbol{a}}^{(\bar{u})}}(y_u)} \mathbb{1}_{\{\boldsymbol{x}_{\mathrm{pa}(u)}=\boldsymbol{a}\}} = 0$$

Dans le cas de lois gaussiennes, on obtient les estimateurs

$$\begin{cases} \hat{\mu}_{\boldsymbol{a}} = \dfrac{\sum_{u \in \bar{u}} \mathbb{1}_{\{\boldsymbol{x}_{\mathrm{pa}(u)}=\boldsymbol{a}\}} y_u}{\mathrm{card}(\{u \in \bar{u} \mid \boldsymbol{x}_{\mathrm{pa}(u)}=\boldsymbol{a}\})} \\ \hat{\Sigma}_{\boldsymbol{a}} = \dfrac{\sum_{u \in \bar{u}} \mathbb{1}_{\{\boldsymbol{x}_{\mathrm{pa}(u)}=\boldsymbol{a}\}} {}^t(y_u - \hat{\mu}_{\boldsymbol{a}})(y_u - \hat{\mu}_{\boldsymbol{a}})}{\mathrm{card}(\{u \in \bar{u} \mid \boldsymbol{x}_{\mathrm{pa}(u)}=\boldsymbol{a}\})} \end{cases}$$

Les cardinaux intervenant aux dénominateurs peuvent être vus comme des sommes d'indicatrices également, par exemple $\mathrm{card}(\{u \in \bar{u} \mid \boldsymbol{x}_{\mathrm{pa}(u)} = \boldsymbol{a}\}) = \sum_{u \in \bar{u}} \mathbb{1}_{\{\boldsymbol{x}_{\mathrm{pa}(u)}=\boldsymbol{a}\}}$.

Comme le processus $\boldsymbol{s}$ est en réalité inobservé, on remplace les indicatrices faisant intervenir des variables cachées par leur espérance conditionnelle sachant les données observées. S'agissant d'espérances conditionnelles de variables indicatrices, on est en fait amené à calculer des probabilités conditionnelles. Ainsi

1. $\mathbb{1}_{\{x_u=i\}}$ est remplacé par $P(S_u = i | \boldsymbol{Y} = \boldsymbol{y})$ si X_u est la variable cachée S_u,
2. $\mathbb{1}_{\{\boldsymbol{x}_{\mathrm{pa}(u)}=\boldsymbol{a}\}} = \mathbb{1}_{\{\boldsymbol{s}_{\mathrm{pa}(u)}=\boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}}\}} \mathbb{1}_{\{\boldsymbol{y}_{\mathrm{pa}(u)}=\boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}}\}}$ est remplacé par
   $P(\boldsymbol{S}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}} | \boldsymbol{Y} = \boldsymbol{y}) \mathbb{1}_{\{\boldsymbol{y}_{\mathrm{pa}(u)}=\boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}}\}}$
3. $\mathbb{1}_{\{x_u=i, \boldsymbol{x}_{\mathrm{pa}(u)}=\boldsymbol{a}\}} = \mathbb{1}_{\{s_u=i, \boldsymbol{s}_{\mathrm{pa}(u)}=\boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}}\}} \mathbb{1}_{\{\boldsymbol{y}_{\mathrm{pa}(u)}=\boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}}\}} \mathbb{1}_{\mathcal{U}_{\boldsymbol{S}}}(u)$ est remplacé par
   $P(S_u = i, \boldsymbol{S}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}}\} | \boldsymbol{Y} = \boldsymbol{y}) \mathbb{1}_{\{\boldsymbol{y}_{\mathrm{pa}(u)}=\boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}}\}}$ si X_u est la variable cachée S_u et
   $\mathbb{1}_{\{\boldsymbol{s}_{\mathrm{pa}(u)}=\boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}}\}} \mathbb{1}_{\{\boldsymbol{y}_{\mathrm{pa}(u)}=\boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}}\}} \mathbb{1}_{\mathcal{U}_{\boldsymbol{Y}}}(u)$ est remplacé par
   $P(\boldsymbol{S}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}}\} | \boldsymbol{Y} = \boldsymbol{y}) \mathbb{1}_{\{\boldsymbol{y}_{\mathrm{pa}(u)}=\boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}}\}} \delta_{y_u,i}$ si X_u est la variable observée Y_u.

Or la loi $P = P_{\lambda_0}$ par rapport à laquelle on désire calculer ces probabilités est inconnue puisque le but est justement d'estimer λ_0. C'est pourquoi on a recours à un algorithme itératif qui suppose connue une valeur courante $\lambda^{(\eta-1)}$ du paramètre et calcule la valeur suivante grâce aux formules de réestimation (2.11), (2.12), (2.13) et (2.14) où les probabilités sont calculées sous la loi $P_{\lambda^{(\eta-1)}}$.

Le principe consistant à

1. calculer la loi $\tilde{P}^{(\eta-1)}(\boldsymbol{s}) = P_{\lambda^{(\eta-1)}}(\boldsymbol{S} = \boldsymbol{s} | \boldsymbol{Y} = \boldsymbol{y})$ (étape E) ;
2. maximiser $\mathbb{E}_{\tilde{P}^{(\eta-1)}}[\ln(P_\lambda(\boldsymbol{S} = \boldsymbol{s}, \boldsymbol{Y} = \boldsymbol{y}))]$ (étape M),

est étudié dans Neal et Hinton, 1988 [96], où il est présenté comme une méthode de maximisation alternée. Les auteurs montrent qu'une itération de EM est équivalente à la maximisation de la fonction F définie par

$$F(\tilde{P}, \lambda) = \mathbb{E}_{\tilde{P}}[\ln(P_\lambda(\boldsymbol{Y} = \boldsymbol{y}, \boldsymbol{S}))] - \mathbb{E}_{\tilde{P}}[\ln(\tilde{P}(\boldsymbol{S}))],$$

conjointement par rapport aux variables $\tilde{P}$ et λ. La fonction F peut également être vue comme

$$F(\tilde{P}, \lambda) = -\mathrm{KL}(P_{\lambda,\boldsymbol{y}} || \tilde{P}) + \mathcal{L}_{\boldsymbol{y}}(\lambda)$$

où

$$\mathrm{KL}(P_{\lambda,\boldsymbol{y}} || \tilde{P}) = \int \ln(\frac{\tilde{P}(\boldsymbol{s})}{P_{\lambda,\boldsymbol{y}}(\boldsymbol{s})}) \tilde{P}(\boldsymbol{s}) \, d\boldsymbol{s} = \mathbb{E}_{\tilde{P}} \left[\ln \left(\frac{\tilde{P}(\boldsymbol{S})}{P_{\lambda,\boldsymbol{y}}(\boldsymbol{S})} \right) \right]$$

désigne la divergence de Kullback-Leibler entre $P_{\lambda,\boldsymbol{y}}$ et $\tilde{P}$, $P_{\lambda,\boldsymbol{y}}(\boldsymbol{s})$ désignant $P_\lambda(\boldsymbol{S} = \boldsymbol{s}|\boldsymbol{Y} = \boldsymbol{y})$ et $\tilde{P}(\boldsymbol{s})$ désignant $\tilde{P}(\boldsymbol{S} = \boldsymbol{s})$. Étant donné que $\mathbb{E}_{\tilde{P}}[\ln(\tilde{P}(\boldsymbol{S}))]$ ne dépend pas de λ, une itération de EM est équivalente à l'itération ci-dessous :

1. calculer $\tilde{P}^{(\eta)} = \arg\max_{\tilde{P}} F(\tilde{P}, \lambda^{(\eta-1)})$ (étape E),
2. calculer $\hat{\lambda}^{(\eta)} = \arg\max_{\lambda} F(\tilde{P}^{(\eta)}, \lambda)$ (étape M).

La solution de l'étape E est $\tilde{P}^{(\eta)} = P_{\lambda^{(\eta-1)},\boldsymbol{y}}$. Ce résultat explique l'usage des probabilités conditionnelles des états cachés sachant les données observées, car c'est la distribution optimale au sens du critère F.

D'autre part, il est équivalent de considérer que, pour $u \in \mathcal{U}_{\boldsymbol{S}}$, S_u est une variable aléatoire cachée à valeurs dans $\mathcal{S}_u$, et que $\tilde{S}_u = (\tilde{S}_u^{(i)})_{i \in \mathcal{S}_{S_u}}$ est un vecteur aléatoire caché de type vecteur indicateur à valeurs dans l'ensemble des vecteurs de la base canonique $(e_1, \ldots, e_K)$ de $\mathbb{R}^K$ avec $K = \text{card}(\mathcal{S}_u)$. La $j^{\text{ème}}$composante du vecteur $\tilde{S}_u$ vaut 1 si et seulement si la valeur de l'état caché est j, et vaut 0 sinon, soit $\tilde{S}_u^{(i)} = \delta_{S_u,i}$. On a alors $P(S_u = i|\boldsymbol{Y} = \boldsymbol{y}) = \mathbb{E}[\tilde{S}_u^{(i)}|\boldsymbol{Y} = \boldsymbol{y}]$. Ce changement de notation s'étend aux processus aléatoires à valeurs discrètes. On peut alors considérer que l'algorithme EM remplace les variables aléatoires cachées par leur espérance conditionnelle sachant les données observées.

Cette interprétation, valable dans le cas des modèles de Markov cachés, ne l'est pas toujours dans le contexte d'autres modèles à données manquantes. De plus, si elle s'applique à l'estimation des lois des sources et des probabilités de transition pour tous les modèles de Markov cachés, elle peut ne pas s'appliquer à l'estimation des paramètres d'émission pour des familles de lois d'émission autres que la famille exponentielle.

2.3.5 Algorithmes de restauration-maximisation

Dans la section 2.3.4, nous avons vu que l'algorithme EM pouvait être considéré, dans le cas des modèles de Markov cachés, comme un algorithme itératif d'estimation des paramètres revenant à remplacer, dans les estimateurs de maximum de log-vraisemblance complétée, les données manquantes par leur espérance conditionnelle sachant les données observées. Cette interprétation est valable pour l'estimation des lois des sources et des probabilités de transition. Elle reste valable pour l'estimation des paramètres d'émission dans le cas de la famille exponentielle. Dans ce qui suit, nous présentons d'autres algorithmes consistant à remplacer les données manquantes dans les estimateurs de maximum de log-vraisemblance complétée : ce sont les algorithmes de *restauration-maximisation*.

Définition et stratégies de restauration

Les algorithmes de restauration-maximisation ont pour but l'estimation des paramètres λ de modèles à données incomplètes $\boldsymbol{S}$ à partir d'une (ou plusieurs) réalisation(s) $\boldsymbol{y}$ du processus observé $\boldsymbol{Y}$. Dans de tels modèles, où l'estimation par maximum de vraisemblance est souvent impossible de manière directe (typiquement, pour les modèles de Markov cachés), ces algorithmes itératifs fonctionnent comme suit (voir Qian et Titterington, 1991 [101]). À partir d'une valeur initiale $\hat{\lambda}^{(0)}$ des paramètres, ils utilisent, à l'itération η, les deux étapes suivantes :

– *Restauration* : les données manquantes $\boldsymbol{s}$ sont remplacées par la valeur $\boldsymbol{s}^{(\eta)}$, en utilisant la valeur courante $\hat{\lambda}^{(\eta)}$ du paramètre et les données observées $\boldsymbol{y}$,
– *Maximisation* : la valeur $\hat{\lambda}^{(\eta+1)}$ est déterminée par maximisation en λ de la log-vraisemblance complétée $\ln(\mathcal{L}_{\boldsymbol{y},\boldsymbol{s}^{(\eta)}}(\lambda))$.

Les algorithmes de restauration-maximisation diffèrent essentiellement par leur stratégie de restauration. Les stratégies les plus connues sont décrites ci-dessous. Les valeurs manquantes sont remplacées par :

– leur espérance conditionnelle sachant $\boldsymbol{Y} = \boldsymbol{y}$, calculée sous la loi $P_{\hat{\lambda}^{(\eta)}}$. Comme nous l'avons vu dans la section 2.3.4, c'est la stratégie de l'algorithme EM pour les modèles de Markov cachés (sous la condition, pour les lois d'émission, d'appartenance à la famille exponentielle) ;
– leur valeur la plus probable sachant $\boldsymbol{Y} = \boldsymbol{y}$ et pour la loi $P_{\hat{\lambda}^{(\eta)}}$. La quantité $\boldsymbol{s}^{(\eta)} = \arg\max_{\boldsymbol{s}} P_{\hat{\lambda}^{(\eta)}}(\boldsymbol{S} = \boldsymbol{s}|\boldsymbol{Y} = \boldsymbol{y})$ est appelée valeur du MAP (*Maximum A Posteriori*). Souvent utilisé en reconnaissance de parole par chaînes de Markov cachées, l'algorithme correspondant est appelé *algorithme de Baum-Viterbi* par Ephraim et Merhav, 2002 [47]), puisque dans ce contexte, les formules de l'étape M sont dues à Baum *et al.*, 1970 [7] et l'algorithme du MAP est dû à Viterbi, 1967 [117]. Cet algorithme est également utilisé pour l'identification de mélanges indépendants, où il est connu sous le nom CEM, pour *Classification EM* (voir Celeux et Govaert, 1992 [23]) ;
– une ou plusieurs valeur(s) aléatoire(s) tirée(s) suivant la loi $P_{\hat{\lambda}^{(\eta)}}(\boldsymbol{S} = \boldsymbol{s}|\boldsymbol{Y} = \boldsymbol{y})$ (algorithme *Stochastic EM* ou SEM, introduit par Celeux et Diebolt, 1985 [20] dans le cadre des modèles de mélange) ou suivant la loi $P_{\hat{\lambda}^{(\eta)}}(S_u = s_u|\boldsymbol{S}_{\mathcal{U}\backslash u} = \boldsymbol{s}_{\mathcal{U}\backslash u}, \boldsymbol{Y} = \boldsymbol{y})$ (algorithme *EM à la Gibbs*, introduit par Robert *et al.*, 1993 [106] dans le cadre des chaînes de Markov cachées).

Les propriétés de ces algorithmes sont détaillées dans la section 2.6.

2.3.6 Application aux arbres de Markov cachés

Le modèle des arbres de Markov cachés a été introduit par Crouse, Nowak et Baraniuk, 1998 [32]. Les auteurs proposent une paramétrisation et un algorithme EM d'estimation des paramètres par maximum de vraisemblance. Nous rappelons la structure de ce modèle présenté dans la section 1.3, figure 1.7.

Paramétrisation

La structure des modèles d'arbres de Markov cachés possède une unique source S_1 qui est un état caché. On lui associe une loi $(\pi_i)_{i\in\mathcal{S}_1} = P(S_1 = i))_{i\in\mathcal{S}_1}$. Les arcs du graphe sont de deux types ($S_{\rho(u)}$ désigne le parent de S_u dans le graphe) :

1. les arcs $(S_{\rho(u)}, S_u)$, associés aux paramètres de transition $p^{(u)}_{s_{\rho(u)},s_u}$ car $S_{\rho(u)}$ est l'unique parent de S_u ;
2. les arcs (S_u, Y_u), associés aux paramètres d'émission $\theta^{(u)}_{s_u}$ car S_u est l'unique parent de Y_u.

Dans le cas d'un modèle homogène où tous les états prennent leur valeur dans l'ensemble $\{1,\ldots,K\}$, les paramètres se restreignent à la loi π de l'état source (ou *racine* de l'arbre, ou *état initial*) S_1, les probabilités de transition $p_{i,j} = P(S_u = j|S_{\rho(u)} = i)$ et les paramètres d'émission $(\theta_i)_{i\in\{1,\ldots,K\}}$ tels que, sachant $S_u = i$, Y_u suit la loi P_{θ_i}. Nous considérons, ci-dessous, l'exemple où la famille P_{θ_i} est la famille des lois gaussiennes, indexée par les paramètres $\theta_i = (\mu_i, \Sigma_i)$.

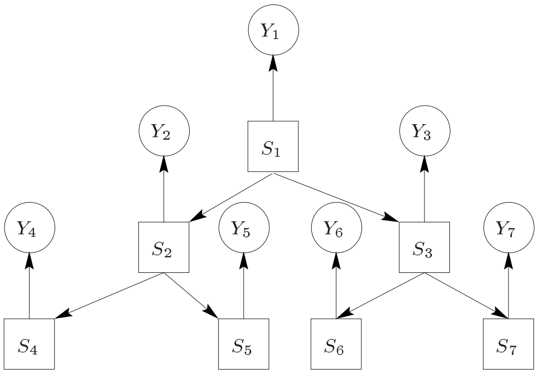


FIG. 1.7 – *La structure des modèles d'arbres de Markov cachés.*

Estimation des paramètres

L'application des résultats (2.11) et (2.13) au modèle d'arbre de Markov caché homogène défini ci-dessus conduit directement aux formules de réestimation suivantes, lorsque l'algorithme EM est utilisé pour l'estimation des paramètres :

Loi de l'état initial : $\hat{\pi}_i^{(\eta)} = P_{\lambda^{(\eta-1)}}(S_1 = i|\boldsymbol{Y} = \boldsymbol{y})$

Probabilités de transition : $$\hat{p}_{i,j}^{(\eta)} = \frac{\sum_u P_{\lambda^{(\eta-1)}}(S_u = j, S_{\rho(u)} = i|\boldsymbol{Y} = \boldsymbol{y})}{\sum_u P_{\lambda^{(\eta-1)}}(S_{\rho(u)} = i|\boldsymbol{Y} = \boldsymbol{y})}$$

Paramètres d'émission (dans le cas gaussien) :
$$\begin{cases} \hat{\mu}_i^{(\eta)} = \dfrac{\sum_u P_{\lambda^{(\eta-1)}}(S_u = i|\boldsymbol{Y} = \boldsymbol{y})y_u}{\sum_u P_{\lambda^{(\eta-1)}}(S_u = i|\boldsymbol{Y} = \boldsymbol{y})} \\ \hat{\Sigma}_i^{(\eta)} = \dfrac{\sum_u P_{\lambda^{(\eta-1)}}(S_u = i|\boldsymbol{Y} = \boldsymbol{y})^t(y_u - \hat{\mu}_i^{(\eta)})(y_u - \hat{\mu}_i^{(\eta)})}{\sum_u P_{\lambda^{(\eta-1)}}(S_u = i|\boldsymbol{Y} = \boldsymbol{y})} \end{cases}$$

Dans le cas non gaussien, les paramètres d'émission sont estimés par la résolution des équations (2.14) : $\sum_u \dfrac{\nabla_{\theta_i} P_{\theta_i}(y_u)}{P_{\theta_i}(y_u)} P_{\lambda^{(\eta-1)}}(S_u = i|\boldsymbol{Y} = \boldsymbol{y}) = 0$.

Ces formules coïncident avec celles de Crouse, Nowak et Baraniuk, 1998 [32]. Dans notre cas, il s'agit d'une simple application des résultats de la section 2.3.3. Aucun calcul théorique n'est nécessaire pour ces formules de réestimation : il suffit d'identifier, grâce au graphe d'indépendance conditionnelle, les parents de chaque sommet.

2.4 Algorithmes de type arrière-avant pour le calcul de probabilités

Dans la section 1.4, nous avons exposé la problématique du calcul de probabilités dans les modèles de Markov cachés. Il s'agit de calculer les probabilités conditionnelles $P(\boldsymbol{X}_A = \boldsymbol{x}_A | \boldsymbol{X}_B = \boldsymbol{x}_B)$ d'une sous-partie A des variables aléatoires du graphe, pour toutes les valeurs possibles $\boldsymbol{x}_A \in \mathcal{X}_A$ et $\boldsymbol{x}_B \in \mathcal{X}_B$. Dans cette section, les paramètres λ du modèle sont fixés et connus. Nous noterons alors $P = P_\lambda$. Parmi les applications de tels calculs figurent le calcul de la vraisemblance, le calcul des probabilités $P(\boldsymbol{S}_\mathcal{C} = \boldsymbol{s}_\mathcal{C} | \boldsymbol{Y} = \boldsymbol{y})$ pour toutes les cliques $\mathcal{C}$, qui sont suffisantes pour l'implémentation d'un algorithme EM (voir section 2.3, dont la proposition 1) et le calcul des lois $P_{\hat{\lambda}^{(\eta)}}(\boldsymbol{S} = \boldsymbol{s} | \boldsymbol{Y} = \boldsymbol{y})$ et $P_{\hat{\lambda}^{(\eta)}}(S_u = s_u | \boldsymbol{S}_{\mathcal{U} \backslash \{u\}} = \boldsymbol{s}_{\mathcal{U} \backslash \{u\}}, \boldsymbol{Y} = \boldsymbol{y})$ pour les variantes stochastiques de EM présentées en section 2.3.5. Nous avons vu que le calcul par sommation sur toutes les valeurs possibles des états cachés était généralement infaisable en pratique, vu le nombre d'opérations en jeu. L'algorithme d'arbre de jonction (voir Smyth, Heckerman et Jordan, 1997 [110]) permet le calcul numérique mais non analytique de ces probabilités. De plus, il n'est pas adapté à la paramétrisation naturelle des modèles graphiques probabilistes orientés acycliques et rend difficile, par le manque d'interprétation de ses calculs intermédiaires (en termes probabilistes), la détection des calculs rpts inutilement. L'algorithme avant-arrière de Lucke, 1996 [87] est plus interprétable mais il est réservé aux modèles graphiques probabilistes non orientés. En outre, lorsqu'ils sont appliqués à des modèles de Markov cachés dynamiques, les deux algorithmes sont sujets à des instabilités numériques. En effet, l'usage de probabilités jointes fait que les quantités calculées par ces algorithmes tendent vers 0 quand le nombre n de variables aléatoires observées tend vers $+\infty$.

Dans cette section, nous présentons un algorithme arrière-avant adapté aux modèles de la classe $\mathcal{D}$. Nous proposons un premier algorithme s'appuyant sur la factorisation de la loi jointe de variables observées, qui reste donc sensible au problème d'instabilité numérique. Cet algorithme rend explicite le rôle des paramètres tout en nécessitant moins de calculs que les algorithmes d'arbre de jonction et de Lucke. Il s'appuie sur l'arbre de jonction, qui a la propriété d'exhiber les séparateurs de sommets de taille minimale : les séparateurs de cliques. Nous rappelons comment cette propriété permet d'utiliser au mieux les relations d'indépendance conditionnelle, représentées par la structure du modèle, pour réduire les calculs. Il s'appuie sur plusieurs propriétés propres aux graphes de la classe $\mathcal{D}$, que nous exposons au préalable. Puis nous présentons un second algorithme de type arrière-avant répondant au problème d'instabilité numérique. Il nécessite en général une phase supplémentaire, de type avant. Nous présentons également l'application de ces algorithmes de calculs de probabilités aux arbres de Markov cachés, en montrant que notre algorithme générique basé sur la factorisation de la vraisemblance, lorsqu'il est instancié pour ce modèle, coïncide avec l'algorithme *ascendant-descendant* de Crouse, Nowak et Baraniuk, 1998 [32], spécifique à ce modèle. De même, nous montrons que notre algorithme générique basé sur la factorisation des probabilités de lissage coïncide avec l'algorithme de Durand, Gonçalvès et Guédon, 2002 [46] pour les arbres de Markov cachés. Ensuite, l'étude de nos algorithmes arrière-avant montre que leur application au calcul de probabilités (et par suite à l'estimation de paramètres par l'algorithme EM) est également possible dans le cas de modèles à structure non orientée et triangulée. D'autre part, cette étude débouche également sur une solution simple au problème de l'estimation des paramètres quand la log-vraisemblance complétée n'est pas définie pour toutes les réalisations du processus caché. Enfin, nous montrons comment nos algorithmes arrière-avant permettent l'implémentation des versions stochastiques de EM tout en évitant les calculs redondants, à savoir l'algorithme SEM, basé sur les probabilités $P_{\hat{\lambda}^{(\eta)}}(\boldsymbol{S} = \boldsymbol{s}|\boldsymbol{Y} = \boldsymbol{y})$ et l'algorithme EM à la Gibbs utilisant les probabilités $P_{\hat{\lambda}^{(\eta)}}(S_u = s_u|\boldsymbol{S}_{\mathcal{U}\setminus\{u\}} = \boldsymbol{s}_{\mathcal{U}\setminus\{u\}}, \boldsymbol{Y} = \boldsymbol{y})$. Dans le cas de l'algorithme SEM, nous proposons un algorithme de simulation du processus caché utilisant une seule exécution de l'algorithme arrière-avant (pour le calcul des probabilités $P(\boldsymbol{S}_{\mathcal{C}} = \boldsymbol{s}_{\mathcal{C}}|\boldsymbol{Y} = \boldsymbol{y})$) puis un parcours de l'arbre de jonction de type avant, pour la simulation proprement dite. Dans le cas de l'algorithme EM à la Gibbs, nous nous basons sur la solution de Spiegelhalter *et al.*, 1996 [112] qui est algorithmiquement plus simple. Nous commentons l'usage de ces deux algorithmes pour l'estimation bayésienne. En conclusion, nous réalisons une synthèse des avantages de nos algorithmes par rapport à celui de l'arbre de jonction puis nous commentons les différentes utilisations qui peuvent en être faites, avec en particulier des suggestions d'optimisation en temps de calcul pour des modèles particuliers.

2.4.1 Propriétés des graphes de la classe $\mathcal{D}$ et applications à l'algorithme arrière-avant

La terminologie d'*algorithme arrière-avant* est empruntée au vocabulaire des chaînes de Markov cachées. Dans ces modèles, les probabilités intervenant dans l'étape E de l'algorithme EM sont calculées par un parcours du processus $\boldsymbol{y} = (y_t)_{t\in\{1,\ldots,n\}}$ dans l'ordre des t croissants (phase avant) puis des t décroissants (phase arrière). Cet algorithme de calcul de probabilités est dû, à l'origine, à Chang et Hancock, 1966 [27] puis a été redécouvert par Baum *et al.*, 1970 [7]. Il se base sur une décomposition de la vraisemblance, pour le calcul de laquelle une seule des deux phases suffit. Dans cet algorithme, la phase avant et la phase arrière peuvent être exécutées dans un ordre quelconque mais par tradition, la phase avant est souvent réalisée en premier. Dans le cas des modèles de Markov cachés de la famille $\mathcal{D}$, plus généraux, c'est l'arbre des cliques qui est parcouru dans deux sens opposés. L'existence d'un arbre de cliques est assuré, dans la famille $\mathcal{D}$, par le caractère triangulé de la structure. L'équivalence entre ce caractère triangulé et l'existence d'un arbre de cliques a été démontré de manière indépendante par Buneman, 1974 [15], par Gavril, 1974 [57] et par Walter, 1972 [118]. L'ordre d'exécution des phases est imposé par le fait que la phase avant utilise les résultats de la phase arrière, qui doit donc avoir lieu en premier. Nous supposons disposer de données observées $\boldsymbol{y} \in \mathcal{Y}$ fixées. Le modèle utilisé est supposé entièrement spécifié (c'est-à-dire de structure et de paramètres connus). Le but de la phase arrière est de calculer la vraisemblance ainsi que certaines probabilités intervenant dans la phase avant (détaillées ci-dessous). Le but de la phase avant est de calculer les probabilités $P(\boldsymbol{S}_{\mathcal{C}} = \boldsymbol{s}_{\mathcal{C}}|\boldsymbol{Y} = \boldsymbol{y})$ pour toutes les cliques $\mathcal{C}$ de l'arbre de jonction et pour toutes les valeurs possibles $\boldsymbol{s}_{\mathcal{C}} \in \mathcal{S}_{\mathcal{C}}$, ce qui est suffisant pour l'implémentation d'un algorithme EM d'après la proposition 1.

Notre algorithme arrière-avant est inductif au sens où une certaine quantité est associée à chaque clique, cette quantité se déduisant de celles associées à des cliques déjà parcourues. Nous verrons ceci dans la section 2.4.2, avec en particulier la formule de propagation (2.20). L'ordre de parcours de l'arbre des cliques est déterminé par le fait que notre algorithme arrière calcule la loi d'une quantité croissante de variables aléatoires observées, conditionnellement à des variables aléatoires les séparant des cliques voisines (les *séparateurs de cliques*). Par conséquent, si la loi de ces séparateurs de cliques est connue, nous sommes en mesure de calculer la loi jointe d'ensembles de variables aléatoires observées déjà parcourues ; or la loi des séparateurs de cliques est inconnue à chaque étape du parcours de l'arbre des cliques. Cela ne pose pas de problème si le parcours de cet arbre s'achève en une clique $\mathcal{C}_0$ pour laquelle la loi de $\boldsymbol{X}_{\mathcal{C}_0}$ est connue. En effet, la loi de ses séparateurs de cliques est, dans ce cas, également connue et on en déduit la loi de toutes les variables aléatoires observées déjà parcourues, c'est-à-dire la vraisemblance. Autrement dit, la connaissance de la loi de $\boldsymbol{X}_{\mathcal{C}_0}$ permet de passer de lois conditionnelles à la loi jointe de $\boldsymbol{Y}$.

De manière plus imagée, l'algorithme arrière part du principe que pour chaque clique, on saurait calculer la loi jointe de tous les Y_u rencontrés jusqu'ici dans le parcours si la loi de ses séparateurs de cliques était connue. On peut continuer ainsi jusqu'à ce que toutes les cliques soient parcourues, mais on ne peut en déduire la vraisemblance que si à la fin, la loi des séparateurs de cliques est réellement connue. C'est le cas si la clique d'arrivée est choisie de manière judicieuse. Nous montrons dans cette section des propriétés issues des hypothèses faites sur la famille de modèles considérée. Rappelons que ces hypothèses concernent essentiellement le graphe d'indépendance conditionnelle du modèle. Ces propriétés visent à établir qu'il existe une clique $\mathcal{C}_0$ pour laquelle la loi de $\boldsymbol{X}_{\mathcal{C}_0}$ se déduit très simplement des paramètres (propriété 5). Cette propriété s'appuie en partie sur le fait que les modèles de la famille $\mathfrak{D}$ admettent un unique état initial S_0 (ou *état source*, voir propriété 4). Comme toutes les propriétés démontrées dans cette section, l'intérêt de la propriété 4 va au-delà de la démonstration de l'existence de la clique $\mathcal{C}_0$ décrite ci-dessus. Elle a des conséquences sur la compréhension des modèles de la famille $\mathfrak{D}$; en l'occurrence, l'existence et l'unicité de l'état initial entraîne une nouvelle interprétation du modèle. Elle permet de définir une notion d'origine, au sens graphique mais donc aussi au sens causal, autrement dit l'existence d'une variable aléatoire qui, au sens des graphes orientés, influence toutes les autres, directement ou non. Cette proposition implique aussi une certaine contrainte sur les paramètres (existence d'un seul paramètre de type π avec $\pi_j = P(X_{S_0} = j)$). La proposition 2 établit que les sommets d'une clique peuvent être ordonnés par leur *degré entrant* (nombre d'arcs entrants). Ceci permet, dans les propositions suivantes, de désigner les sommets du graphe de manière simple. Cette proposition sert également de base à la démonstration de la proposition 3, qui intervient dans la preuve de l'existence de la clique $\mathcal{C}_0$. La proposition 3 concerne l'étude des sommets u du graphe d'indépendance conditionnelle dont tous les parents sont dans une clique $\mathcal{C}$ contenant u. Ces sommets jouent un rôle particulier vu que les calculs de probabilités les concernant sont plus aisés, du fait qu'ils vérifient cette propriété. En effet, la recherche des parents de ces sommets est immédiate. D'autre part, la loi de $\boldsymbol{X}_{\mathcal{C}}$ fait intervenir, dans ce cas, la probabilité de transition entre les parents de u et u, qui est directement donnée par l'un des paramètres du modèle. La proposition 3 montre que pour toute clique $\mathcal{C}$, à partir d'un certain rang (au sens de l'ordre de la proposition 2), les sommets de $\mathcal{C}$ ont tous leurs parents dans $\mathcal{C}$. La clique $\mathcal{C}_0$ n'est en fait rien d'autre qu'une clique dont chacun des sommets vérifie cette propriété.

Dans tout ce qui suit, nous supposerons que le graphe $\mathcal{G} = (\mathcal{U}, \mathcal{E})$ suit les hypothèses de la classe $\mathcal{D}$. Étant donné une partie A de $\mathcal{U}$, nous noterons $\mathcal{G}(A)$ le graphe engendré par A. Nous identifierons parfois un sommet u du graphe avec la variable aléatoire X_u. Pour tout sous-graphe $\mathcal{G}'$, la notation $u \in \mathcal{G}'$ (respectivement $X_u \in \mathcal{G}'$) signifie que le sommet u (respectivement le sommet X_u) appartient à l'ensemble des sommets du graphe $\mathcal{G}'$, noté $\mathcal{U}(\mathcal{G}')$. Si $(\mathcal{G}'_i)_{i\in I}$ est une famille de sous-graphes de $\mathcal{G}$ indexée par I, nous désignerons par $\bigcup\limits_{i\in I} \mathcal{G}'_i$ le sous graphe engendré par les sommets des $\mathcal{G}'_i$. Étant donné un graphe $\mathcal{T}$ dont l'ensemble des sommets $\{\mathcal{G}'_i | i \in I\}$ est une partie de l'ensemble des sous-graphes de $\mathcal{G}$ (par exemple si $\mathcal{T}$ est un arbre de jonction), nous identifierons parfois $\mathcal{T}$ avec le graphe $\bigcup\limits_{i\in I} \mathcal{G}'_i$. Nous étendons alors la notation $\mathcal{U}(\mathcal{T})$ pour désigner $\mathcal{U}(\bigcup\limits_{i\in I} \mathcal{G}'_i)$, autrement dit l'ensemble des sommets de $\mathcal{U}$ qui apparaissent dans les sommets de $\mathcal{T}$. Nous noterons la relation de parenté par le symbole $\rightarrow$, la notation $u \rightarrow v$ (ou $X_u \rightarrow X_v$) signifiant que u est parent de v (soit : $(u, v) \in \mathcal{E}$, ce qui se traduit également par $u \in \mathrm{pa}(v)$). La notation $\leftrightarrow$ est utilisée pour désigner l'adjacence, $u \leftrightarrow v$ (ou $X_u \leftrightarrow X_v$) signifiant $[(u, v) \in \mathcal{E}$ ou $(v, u) \in \mathcal{E}]$.

Le premier résultat énonce qu'un graphe complet, orienté et acyclique admet une unique source (*i.e.* un sommet sans parent). Cette proposition servira essentiellement à numéroter les sommets d'une clique en les ordonnant par leur degré entrant (nombre d'arcs entrants d'un sommet) pour pouvoir désigner de manière compacte les relations de parenté entre sommets. Ce résultat est illustré par la figure 2.1, dans le graphe engendré par la clique $\mathcal{C}_i$ (en gras sur la figure).

Proposition 2 *Soit $\mathcal{G}' = (\mathcal{U}', \mathcal{E}')$ un graphe complet, orienté et acyclique contenant $N_{\mathcal{G}'}$ sommets. Alors $\mathcal{G}'$ admet une unique source.*

Synopsis de la preuve.
La preuve de l'existence de sources se fait par l'absurde, en montrant qu'un graphe sans source admet des chemins aussi longs qu'on veut, donc au moins un circuit. L'unicité de la source est une conséquence directe de la complétude du graphe.

Démonstration

On suppose $N_{\mathcal{G}'} \geq 2$, le résultat étant évident sinon.

Existence : si $\mathcal{G}'$ n'admet aucune source, on peut montrer par récurrence sur l que $\mathcal{G}'$ admet un chemin de longueur l. On note HR_l l'hypothèse de récurrence au rang l.

$\underline{\mathrm{HR}_l}$: il existe, dans $\mathcal{G}'$, un chemin $(X_l, \ldots, X_1)$ de longueur l.

$\underline{\mathrm{HR}_2}$: soit $X_1 \in \mathcal{U}'$. X_1 n'est pas une source de $\mathcal{G}'$ car ce graphe n'admet aucune source. Donc $\exists X_2 \in \mathcal{U}'$, $X_2 \rightarrow X_1$ et (X_2, X_1) est, dans $\mathcal{G}'$, un chemin de longueur 2.

$\underline{\mathrm{HR}_l} \Rightarrow \underline{\mathrm{HR}_{l+1}}$: par $\underline{\mathrm{HR}_l}$, il existe un chemin $(X_l, \ldots, X_1)$ de longueur l de sommets de $\mathcal{G}'$. Or X_l n'est pas une source de $\mathcal{G}'$ car ce graphe n'en admet aucune. Donc $\exists X_{l+1} \in \mathcal{U}'$, $X_{l+1} \rightarrow X_l$. La chaîne $(X_{l+1}, X_l, \ldots, X_1)$ est alors, dans $\mathcal{G}'$, un chemin de longueur $l + 1$.

Conclusion : pour tout entier $l > 1$, il existe dans $\mathcal{G}'$ un chemin $(X_l, \ldots, X_1)$ de longueur l. Or le nombre de sommets $N_{\mathcal{G}'}$ de $\mathcal{G}'$ est fini donc le chemin $(X_{N_{\mathcal{G}'}+1}, \ldots, X_1)$ passe deux fois par le même sommet. Autrement dit, ce chemin contient un circuit, ce qui est exclu car $\mathcal{G}'$ est supposé sans circuit. Donc $\mathcal{G}'$ admet une source S.
Unicité : si S et S' sont deux sources distinctes de $\mathcal{G}'$, alors comme c'est un graphe complet , il vient $S \leftrightarrow S'$. Par conséquent, soit $S \to S'$, ce qui est exclu car S' est une source dans $\mathcal{G}'$, soit $S' \to S$, ce qui est exclu car S est une source dans $\mathcal{G}'$. Par conséquent, $\mathcal{G}'$ admet une unique source.

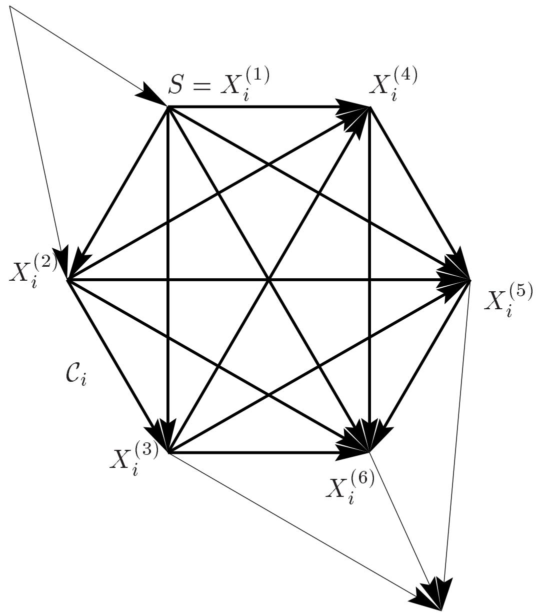


FIG. 2.1 – *La numérotation des sommets d'une clique. Les sommets de la clique $\mathcal{C}_i$ peuvent être ordonnés suivant le nombre de leurs parents dans $\mathcal{C}_i$. Ainsi, $X_i^{(1)}$ n'a aucun parent dans $\mathcal{C}_i$, $X_i^{(2)}$ a un parent dans $\mathcal{C}_i$, etc.*

Une conséquence en est qu'il existe exactement un sommet de degré entrant $l-1$ dans $\mathcal{G}'$, pour tout $l \in \{1, \ldots, N_{\mathcal{C}}\}$. En effet, $\mathcal{G}'$ admet une source $X^{(1)}$ de degré 0 d'après la proposition 2. Or $\mathcal{G}(\mathcal{U} \backslash \{X^{(1)}\})$ est un graphe complet donc la proposition 2 s'applique : ce graphe admet une source $X^{(2)}$ qui admet donc comme seul parent $X^{(1)}$ dans $\mathcal{G}'$ et est de degré 1. On crée ainsi par récurrence une suite $X^{(l)}$ de sommets de degré entrant $l-1$ dans $\mathcal{G}'$, pour $l \in \{1, \ldots, N_{\mathcal{C}}\}$ ce qui permet d'ordonner ces sommets par ordre strictement croissant de degré entrant.

Remarque 2.3 *Il est immédiat d'adapter la démonstration ci-dessus pour montrer l'existence et l'unicité d'un puits (sommet sans arc sortant) dans $\mathcal{G}'$. Ceci permet d'ordonner les sommets de $\mathcal{G}'$ en fonction de leur degré sortant.*

Soit $\mathcal{C}_i$ une clique de $\mathcal{U}$ (donc un sous-graphe complet) ; on peut ordonner les sommets de $\mathcal{G}(\mathcal{C}_i)$ de la manière décrite ci-dessus. Nous garderons la notation $\{X_i^{(1)}, \ldots, X_i^{(N_{\mathcal{C}_i})}\}$

pour les sommets de $\mathcal{C}_i$, de sorte que $X_i^{(l)}$ possède exactement $l-1$ parents dans $\mathcal{C}_i$ (voir figure 2.1). Nous avons alors, de manière immédiate, la propriété suivante :

$$\forall l \in \{2, \ldots, N_{\mathcal{C}}\}, \quad \{X_i^{(1)}, \ldots, X_i^{(l-1)}\} \subset \mathrm{pa}(X_i^{(l)}), \tag{2.16}$$

ce qui s'écrit également

$$\forall l \in \{1, \ldots, N_{\mathcal{C}}\}, \forall l' \in \{2, \ldots, N_{\mathcal{C}}\}, \quad [l < l' \Rightarrow X_i^{(l)} \to X_i^{(l')}].$$

En fait, on va montrer qu'il y a égalité dans (2.16) à partir d'un certain rang $\nu^*(\mathcal{C}_i)+1$ (au sens de l'ordre défini ci-dessus). Autrement dit, la proposition 3 ci-dessous établit qu'à partir de ce rang, les sommets de $\mathcal{C}_i$ ont tous leurs parents dans $\mathcal{C}_i$. Ceux-ci jouent un rôle particulier vu que les calculs de probabilités sont plus aisés pour les sommets vérifiant cette propriété. En effet, la recherche des parents de ce sommet est immédiate (voir remarque 2.4). D'autre part, la loi de $\boldsymbol{X}_{\mathcal{C}_i}$ fait intervenir, dans ce cas, la probabilité de transition entre les parents de u et u, qui est directement donnée par l'un des paramètres du modèle. Le sommet de $\mathcal{C}_i$ de rang $\nu^*(\mathcal{C}_i)$ joue aussi un rôle particulier dans la mesure où il admet au moins un parent dans une clique autre que $\mathcal{C}_i$. La proposition 2.2 intervient dans la preuve de l'existence d'une clique $\mathcal{C}_0$ pour laquelle la loi de $\boldsymbol{X}_{\mathcal{C}_0}$ se déduit très simplement des paramètres du modèle, ce qui permet de terminer l'algorithme arrière. Ceci peut être justifié de manière simpliste en remarquant qu'une telle clique $\mathcal{C}_0$ est une clique dont chacun des sommets a ses parents inclus dans $\mathcal{C}_0$.

Rappelons que $\mathrm{NC}(\mathcal{G})$ représente le nombre de cliques de $\mathcal{G}$ et que $V_{\mathcal{G}} = \{\mathcal{C}_0, \ldots, \mathcal{C}_{\mathrm{NC}(\mathcal{G})-1}\}$ désigne l'ensemble des cliques de $\mathcal{G}$. Par convention, nous posons $\{X_i^{(1)}, \ldots, X_i^{(0)}\} = \emptyset$. La proposition suivante est illustrée par la figure 2.2 où le sommet $X_i^{(4)}$ a un parent à l'extérieur de $\mathcal{C}_i$. À partir de $l > 4$, les sommets $X_i^{(l)}$ ont tous leurs parents dans $\mathcal{C}_i$ et $\mathrm{pa}(X_i^{(l)}) = \{X_i^{(1)}, \ldots, X_i^{(l-1)}\}$. Ainsi, $\nu^*(\mathcal{C}_i)$ vaut 4 dans cet exemple. Notons que les sommets $X_i^{(l)}$ avec $l < 4$ peuvent avoir des parents à l'extérieur de $\mathcal{C}_i$ (cas de $X_i^{(1)}$ et $X_i^{(l)}$) mais peuvent aussi avoir tous leurs parents dans $\mathcal{C}_i$ (cas de $X_i^{(3)}$).

Proposition 3 *Soit $\mathcal{C}_i \in V_{\mathcal{G}}$. Alors* $[\exists\, \nu^*(\mathcal{C}_i) \in \{1, \ldots, N_{\mathcal{C}_i}-1\}, \quad \forall l > \nu^*(\mathcal{C}_i),\ pa(X_i^{(l)}) = \{X_i^{(1)}, \ldots, X_i^{(l-1)}\}]$ *ou* $[pa(X_i^{(1)}) = \emptyset$ *et* $\forall l > 1,\ pa(X_i^{(l)}) = \{X_i^{(1)}, \ldots, X_i^{(l-1)}\}]$.

Synopsis de la preuve.
La preuve se fait en considérant l'ensemble des indices des sommets de $\mathcal{C}_i$ admettant des parents qui ne sont pas dans $\mathcal{C}_i$. Dans le cas où cet ensemble est vide, tous les sommets de $\mathcal{C}_i$ ont tous leurs parents dans $\mathcal{C}_i$. Dans le cas contraire, cet ensemble admet un plus grand élément $\nu^*(\mathcal{C}_i)$, au-delà duquel, par maximalité de $\nu^*(\mathcal{C}_i)$ et par moralité de $\mathcal{G}$, tous les sommets ont tous leurs parents dans $\mathcal{C}_i$.

Démonstration

On pose $E_i = \{l \in \{1, \ldots, N_{\mathcal{C}_i}\} \mid \mathrm{pa}(X_i^{(l)}) \subsetneq \{X_i^{(1)}, \ldots, X_i^{(l-1)}\}\}$.

<u>Cas 1</u> : $E_i = \emptyset$. Alors $\forall l \in \{1, \ldots, N_{\mathcal{C}_i}\}, \quad \mathrm{pa}(X_i^{(l)}) = \{X_i^{(1)}, \ldots, X_i^{(l-1)}\}$, et en particulier $\mathrm{pa}(X_i^{(1)}) = \emptyset$.

<u>Cas 2</u> : $E_i \neq \emptyset$. L'ensemble E_i est une partie non vide et majorée de $I\!N$ donc admet un plus grand élément $\nu^*(\mathcal{C}_i)$. $\forall l > \nu^*(\mathcal{C}_i), \quad l \notin E_i$ par maximalité de

$\nu^*(\mathcal{C}_i)$, donc $\text{pa}(X_i^{(l)}) = \{X_i^{(1)}, \ldots, X_i^{(l-1)}\}$.
Enfin, si Y est un parent de $X_i^{(N_{\mathcal{C}_i})}$ autre que l'un des $X_i^{(l-1)}$ pour $l \in \{1, \ldots, N_{\mathcal{C}_i} - 1\}$, alors par moralité de $\mathcal{G}$, $X_i^{(l)} \to X_i^{(N_{\mathcal{C}_i})}$ et $Y \to X_i^{(N_{\mathcal{C}_i})} \Rightarrow X_i^{(l)} \leftrightarrow Y$. L'ensemble de sommets $\{X_i^{(1)}, \ldots, X_i^{(N_{\mathcal{C}_i})}, Y\}$ engendre un graphe complet contenant la clique $\mathcal{C}_i$, ce qui est impossible par maximalité des cliques. D'où $N_{\mathcal{C}_i} \notin E_i$ et $\nu^*(\mathcal{C}_i) < N_{\mathcal{C}_i}$.

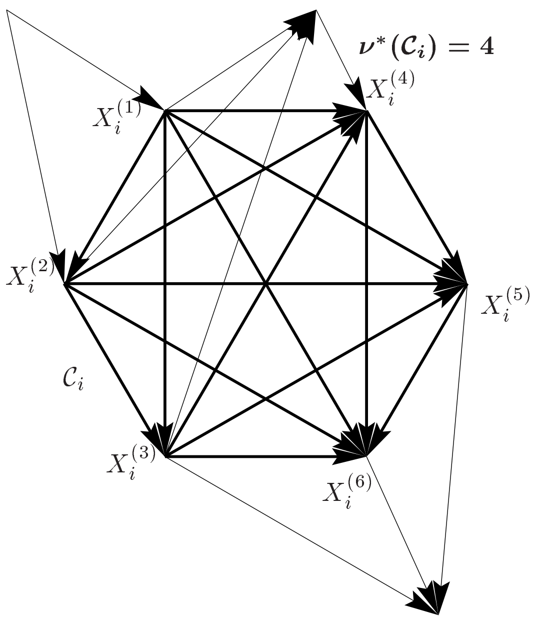


FIG. 2.2 – *Sommets d'une clique $\mathcal{C}_i$ ayant tous leurs parents dans $\mathcal{C}_i$. Le sommet $X_i^{(4)}$ a un parent à l'extérieur de $\mathcal{C}_i$. À partir de $l > 4$, les sommets $X_i^{(l)}$ ont tous leurs parents dans $\mathcal{C}_i$ et pa$(X_i^{(l)}) = \{X_i^{(1)}, \ldots, X_i^{(l-1)}\}$. Ainsi, $\nu^*(\mathcal{C}_i)$ vaut 4 sur cette figure.*

Remarque 2.4 *1. Nous noterons parfois $\nu^*(\mathcal{C}_i) = \nu_i^*$,*

2. *dans le cas où pa$(X_i^{(1)}) = \emptyset$ et $\forall l > 1$, pa$(X_i^{(l)}) = \{X_i^{(1)}, \ldots, X_i^{(l-1)}\}$, nous poserons $\nu^*(\mathcal{C}_i) = 0$,*
3. *dans tous les cas (que $\nu^*(\mathcal{C}_i)$ soit nul ou non), remarquons que $\nu^*(\mathcal{C}_i) < N_{\mathcal{C}_i}$. Or la clique $\mathcal{C}_i$ admet le puits $X_i^{(N_{\mathcal{C}_i})}$, d'après la remarque 2.3. La proposition ci-dessus prouve que $X_i^{(N_{\mathcal{C}_i})}$ a tous ses parents dans $\mathcal{C}_i$,*
4. *le calcul de probabilités est plus aisé pour les sommets $X_i^{(l)}$ dont l'ensemble des parents est inclus dans $\mathcal{C}_i$ (en particulier ceux qui vérifient $l > \nu_i^*$). En effet, l'inférence dans les modèles graphiques acycliques est basée sur la propriété qu'une variable aléatoire est indépendante de ses non descendants sachant ses parents. La détermination des parents ne requerra pas de recherche parmi toutes les cliques du graphe pour de tels $X_i^{(l)}$. En fait, plus ν_i^* sera proche de zéro, plus l'inférence sera facile pour les variables aléatoires de cette clique, ce qui explique aussi le rôle particulier des cliques vérifiant $\nu_i^* = 0$*

Nous aurons besoin, dans l'algorithme arrière de la section 2.4.2, de terminer le parcours de l'arbre de jonction par une clique stable pour la relation de parenté (autrement dit, égale à son graphe ancestral, ou encore de valeur ν^* nulle), donc contenant une source. L'intérêt d'une telle clique est rendu évident par la formule (2.25) : il peut être vu comme un algorithme récursif, et par nature, nécessite que la valeur recherchée soit connue quand l'algorithme s'arrête. Autrement dit, nous sommes dans un contexte où si nous connaissons une certaine quantité courante, alors nous savons calculer la quantité suivante. En définitive, le calcul de la quantité finale n'est possible que lorsque la quantité initiale est connue, ce qui n'est pas le cas avant la fin de l'algorithme. En l'occurrence, notre algorithme arrière calcule la loi conditionnelle des variables aléatoires observées déjà parcourues, sachant les séparateurs de cliques. Pour pouvoir calculer la loi jointe de toutes les variables aléatoires observées, il faut et il suffit de terminer par une clique $\mathcal{C}_0$ telle que la loi de $\boldsymbol{X}_{\mathcal{C}_0}$ soit connue. La loi de ses séparateurs s'obtient alors aisément, ce qui permet de passer des lois conditionnelles citées ci-dessus à la loi jointe de $\boldsymbol{Y}$.

Ainsi, nous cherchons une clique $\mathcal{C}_0$ telle que $P(\boldsymbol{X}_{\mathcal{C}_0} = \boldsymbol{x}_{\mathcal{C}_0})$ soit une fonction simple des paramètres ; c'est le cas si tous les sommets de $\mathcal{C}_0$ ont tous leurs parents dans $\mathcal{C}_0$, puisqu'alors

$$P(\boldsymbol{X}_{\mathcal{C}_0} = \boldsymbol{x}_{\mathcal{C}_0}) = P(X_0^{(1)} = x_0^{(1)}) \prod_{i>1} P(X_0^{(i)} = x_0^{(i)} | X_0^{(1)} = x_0^{(1)}, \ldots, X_0^{(i-1)} = x_0^{(i-1)}),$$

les probabilités $P(X_0^{(i)} = x_0^{(i)} | X_0^{(1)} = x_0^{(1)}, \ldots, X_0^{(i-1)} = x_0^{(i-1)})$ étant données par les paramètres du modèle, vu que $\mathrm{pa}(X_0^{(i)}) = \{X_0^{(1)}, \ldots, X_0^{(i-1)}\}$. La propriété d'existence d'une telle clique $\mathcal{C}_0$ repose sur l'existence et l'unicité d'une source dans $\mathcal{G}$. L'existence de sources dans des graphes orientés acycliques est une propriété connue. Nous pouvons en fait montrer l'unicité de la source. Cette propriété et sa démonstration sont illustrées par la figure 2.3

Proposition 4 *Le graphe orienté, acyclique, moral et (faiblement) connexe $\mathcal{G}$ admet une unique source S_0.*

Synopsis de la preuve.
La preuve se fait par l'absurde, en supposant l'existence de deux sources S_1 et S_2. On considère alors un chemin entre S_1 et S_2. On montre ensuite par récurrence, en s'appuyant sur le caractère moral de $\mathcal{G}$, qu'il existe des chemins de plus en plus courts entre S_1 et S_2. Ainsi, on arrive à montrer l'adjacence de ces deux sommets, ce qui contredit le fait que ce soient des sources.

Démonstration

Unicité : soient S_1 et S_2 deux sources distinctes de $\mathcal{G}$. Par connexité de $\mathcal{G}$, il existe une chaîne $(S_1, X_1^{(1)}, \ldots, X_N^{(1)}, S_2)$ entre S_1 et S_2, de longueur $N+2$ où $N \in \mathbb{N}^*$. On montre alors par récurrence sur k qu'il existe une chaîne de longueur $N+2-k$ entre S_1 et S_2, pour toute valeur de k.

<u>HR_k :</u> : si $1 \leq k \leq N$, il existe une chaîne $(S_1, X_1^{(k)}, \ldots, X_{N-k+1}^{(k)}, S_2)$ de longueur $N-k+3$ entre S_1 et S_2.

<u>HR_1 :</u> : traité ci-dessus.

<u>HR_k</u> $\Rightarrow$ <u>HR_{k+1}</u> : si $1 \leq k+1 \leq N$, alors par HR_k et quitte à renuméroter les sommets, il existe une chaîne $(S_1, X_1, \ldots, X_{N-k+1}, S_2)$ de longueur $N-k+3$ entre S_1 et S_2. Posons $X_0 = S_1$ et $X_{N-k+2} = S_2$. Comme S_1 et S_2 sont des sources,

$$\exists j \in \{1, \ldots, N-k+1\}, \quad [X_{j-1} \to X_j \text{ et } X_{j+1} \to X_j].$$

Par moralité de $\mathcal{G}$, on en déduit $X_{j-1} \leftrightarrow X_{j+1}$.
Donc $(S_1, X_1, \ldots, X_{j-1}, X_{j+1}, \ldots, X_{N-k+1}, S_2)$ est une chaîne de longueur $N-k+2 = N-(k+1)+3$.

<u>Conclusion</u> : pour $k = N-1$, on en déduit qu'il existe une chaîne (S_1, X, S_2) entre S_1 et S_2. Par moralité de $\mathcal{G}$, on obtient $S_1 \leftrightarrow S_2$, ce qui contredit le fait que S_1 et S_2 sont deux sources. Par conséquent, la source de $\mathcal{G}$ est unique.

Remarque 2.5 1. *Dans tout ce qui suit, la notation S_0 continuera à faire référence à l'unique source de $\mathcal{G}$.*

2. *Nous pouvons déduire directement de la démonstration le lemme suivant :*

   Lemme 1 *Soit un sommet dans un graphe orienté, acyclique, moral et (faiblement) connexe admettant deux ancêtres X et X'. Alors X et X' sont adjacents.*

   Ce lemme est également illustré par la figure 2.3.

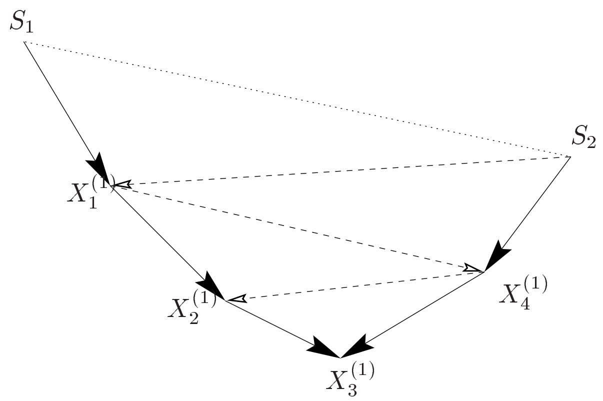


FIG. 2.3 – *L'unicité de la source dans un graphe $\mathcal{G}$ orienté, acyclique, moral et faiblement connexe. Le caractère moral de $\mathcal{G}$ fait que si le sommet $X_3^{(1)}$ admet deux ancêtres S_1 et S_2, ceux-ci doivent être adjacents. Ceci exclut le fait que S_1 et S_2 soient deux sources. On en déduit l'unicité de la source de $\mathcal{G}$.*

Nous pouvons à présent montrer qu'il existe une clique $\mathcal{C}_0$ égale à son graphe ancestral et par laquelle nous allons pouvoir terminer l'algorithme arrière. La démonstration s'appuie sur un lemme qui sera utile maintes fois par la suite. Ce lemme fait le lien entre l'arbre des cliques et les parents d'un sommet, en montrant que deux sommets adjacents appartiennent à une même clique (pas forcément unique). Une telle clique est située, dans l'arbre de jonction, entre toutes les cliques contenant X et toutes les cliques contenant X'.

Lemme 2 *Soit $\mathcal{C}$ et $\mathcal{C}'$ deux cliques de $V_{\mathcal{G}}$. Soit $X \in \mathcal{C}$ et $X' \in \mathcal{C}'$. Alors $X \leftrightarrow X' \Rightarrow$ $[X \in \mathcal{C}'$ ou $X' \in \mathcal{C}$ ou il existe $\mathcal{C}'' \in V_{\mathcal{G}}$, entre $\mathcal{C}$ et $\mathcal{C}'$ dans l'arbre des cliques et contenant X et $X']$.*

Synopsis de la preuve.
La preuve découle du fait que dans l'arbre de jonction, pour toute paire de cliques $(\mathcal{C}, \mathcal{C}')$, l'intersection des cliques $\mathcal{C}$ et $\mathcal{C}'$ est contenue dans toute clique du chemin entre $\mathcal{C}$ et $\mathcal{C}'$.
Démonstration

Le graphe $(\{X, X'\}, \{\{X, X'\}\})$ constitue un graphe complet, donc c'est un sous-graphe de l'une des cliques $\mathcal{C}''$ de $V_{\mathcal{G}}$, qui contient donc X et X'. Il existe un unique chemin dans l'arbre de jonction entre deux cliques quelconques.
<u>Cas 1</u> : le chemin entre $\mathcal{C}''$ et $\mathcal{C}$ dans l'arbre de jonction passe par $\mathcal{C}'$. Alors par définition d'un arbre de jonction, $\mathcal{C}'' \cap \mathcal{C}' \subset \mathcal{C}$. Comme $X \in \mathcal{C}$ et $X \in \mathcal{C}''$, alors $X \in \mathcal{C}'$.
<u>Cas 2</u> : le résultat s'obtient par permutation des rôles de $\mathcal{C}$ et $\mathcal{C}'$.
<u>Cas 3</u> : le chemin entre $\mathcal{C}$ et $\mathcal{C}'$ dans l'arbre de jonction passe par $\mathcal{C}''$, clique de $V_{\mathcal{G}}$ contenant X et X'.

Remarque 2.6
1. *Remarquons que dans le cas où $\mathcal{C}$ et $\mathcal{C}'$ sont voisines dans l'arbre de jonction, le dernier cas ne peut se produire et alors $X \in \mathcal{C}'$ ou $X' \in \mathcal{C}$.*
2. *Dans le cas 3, par propriété de l'arbre de jonction, toute clique entre $\mathcal{C}$ et $\mathcal{C}''$ sur l'arbre de jonction contient X et toute clique entre $\mathcal{C}'$ et $\mathcal{C}''$ contient X'.*

Ce lemme est complété par le lemme suivant, qui prouve que si X est un sommet d'une clique $\mathcal{C}$, l'ensemble des parents de X est compris dans une partie de l'arbre de jonction tronqué à partir de la clique $\mathcal{C}$. Ceci est illustré par la figure 2.4. La clique $\mathcal{C}$ sépare l'arbre de jonction en plusieurs sous-arbres. L'un d'entre eux, représenté ici en traits pleins, contient tous les parents de X, par exemple X'.

Lemme 3 *Soit un sommet X appartenant à une clique $\mathcal{C}$. Si X admet un parent n'appartenant pas à $\mathcal{C}$ mais à une clique $\mathcal{C}'$, alors l'ensemble des cliques $\mathcal{C}''$ telles que $\mathcal{C}$ n'est pas sur le chemin entre $\mathcal{C}''$ et $\mathcal{C}'$ dans l'arbre de jonction contient tous les parents de X.*

Synopsis de la preuve.
Comme pour le lemme précédent, la preuve découle de la définition de l'arbre de jonction (propriété dite d'*intersection*).
Démonstration

On suppose que $X' \in \mathcal{C}' \setminus \mathcal{C}$ est un parent de X. Soit X'' un parent de X. Alors par moralité de $\mathcal{G}$, $X' \leftrightarrow X''$ et l'ensemble $\{X, X', X''\}$ engendre un graphe complet qui est un sous graphe d'une clique $\mathcal{C}''$ de l'arbre de jonction, différente de $\mathcal{C}$ car elle contient X'. Si $\mathcal{C}$ est situé sur le chemin entre $\mathcal{C}''$ et $\mathcal{C}'$, alors par propriété d'intersection de l'arbre de jonction, $\mathcal{C}'' \cap \mathcal{C}' \subset \mathcal{C}$ d'où $X' \in \mathcal{C}$, ce qui contredit notre hypothèse de départ. Donc X'' appartient à une clique $\mathcal{C}''$ telle que $\mathcal{C}$ n'est pas sur le chemin entre $\mathcal{C}''$ et $\mathcal{C}'$ dans l'arbre de jonction.

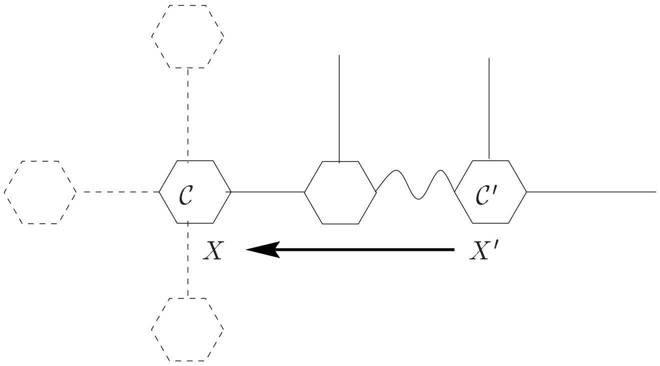


FIG. 2.4 – *Localisation des parents d'un sommet X dans l'arbre de jonction. Les cliques sont représentées par des hexagones, liés par les arêtes de l'arbre de jonction. La clique $\mathcal{C}$ sépare l'arbre de jonction en plusieurs sous-arbres. Celui représenté en traits pleins contient tous les parents de X, par exemple X'.*

Nous pouvons à présent montrer l'existence d'une clique $\mathcal{C}_0$ telle que $\nu_0^* = 0$, c'est-à-dire une clique sans parent extérieur à C_0 (ou encore *ancestrale*), par laquelle se termine l'algorithme arrière.

Proposition 5 *Il existe une clique $\mathcal{C}_0 = \{X_0^{(1)}, \ldots, X_0^{(N_{\mathcal{C}_0})}\}$ de $\mathcal{G}$ telle que $pa(X_0^{(1)}) = \emptyset$ (autrement dit, $X_0^{(1)} = S_0$) et $\forall l \in \{2, \ldots, N_{\mathcal{C}_0}\}, \quad pa(X_0^{(l)}) = \{X_0^{(1)}, \ldots, X_0^{(l-1)}\}$.*

Synopsis de la preuve.
Nous savons par la propriété 4 que le graphe $\mathcal{G}$ admet une unique source S_0. Dans le cas simple où S_0 appartient à une seule clique, nous montrons par l'absurde que cette clique vérifie la propriété désirée.
Dans le cas où S_0 appartient à plusieurs cliques, nous montrons que le graphe engendré par ces cliques contient aussi les parents de ses sommets (propriété illustrée par la figure 2.5). Puis nous montrons par récurrence que cette propriété reste vraie lorsqu'on supprime certaines cliques contenant S_0. La récurrence permet en fait de toutes les supprimer jusqu'à ce qu'il n'en reste qu'une, qui par définition, contiendra les parents de tous ses sommets. Cette clique est donc la clique $\mathcal{C}_0$ recherchée. On va en fait montrer à chaque pas de la récurrence qu'il existe une clique contenant S_0 pour laquelle tous les sommets, à partir d'un certain rang (donné par la proposition 3), n'ont aucun descendant dans les autres cliques contenant S_0. On prouve alors que supprimer ces sommets du graphe revient à supprimer l'une des cliques contenant $\mathcal{C}_0$ et que les cliques restantes contiennent encore les parents de leurs sommets. Pour montrer, à chaque étape de la récurrence, qu'il existe une clique $\mathcal{C}_i$ pour laquelle, à partir du rang ν_i^*, tous les sommets n'ont aucun descendant dans les autres cliques contenant S_0, on procède par l'absurde. Si pour toute clique $\mathcal{C}_i$, il existe un sommet de rang $\nu_i^* + k_i$ avec $k_i > 0$ ayant un descendant dans l'une des cliques pas encore supprimée, on construit un circuit dans $\mathcal{G}$ par récurrence (ce qui fait que la preuve comporte deux récurrences emboîtées).
Démonstration

<u>Cas 1</u> : S_0 appartient à une seule clique $\mathcal{C}_i = \{X_i^{(1)}, \ldots, X_i^{(N_{\mathcal{C}_i})}\}$. Alors par

définition de $X_i^{(1)}$ et comme S_0 est la source de $\mathcal{G}$, il vient $S_0 = X_i^{(1)}$. Supposons qu'un certain sommet $X_i^{(l)}$ de $\mathcal{C}_i$, pour $l \in \{2, \ldots, N_{\mathcal{C}_i}\}$, admette un parent Y dans le complémentaire $\mathcal{C}_i^c$ de $\mathcal{C}_i$. Alors $\exists j \neq i, \quad Y \in \mathcal{C}_j \backslash \mathcal{C}_i$. Ainsi, $Y \to X_i^{(l)}$ et $S_0 \to X_i^{(l)}$ donc par moralité de $\mathcal{G}$, et comme S_0 est la source, il vient $S_0 \to Y$. L'ensemble $\{S_0, Y, X_i^{(l)}\}$ engendre un graphe complet, qui est donc un sous-graphe d'un des éléments $\mathcal{C}$ de l'ensemble $V_{\mathcal{G}}$ des cliques de $\mathcal{G}$. Or $\mathcal{C} \neq \mathcal{C}_i$ car $Y \in \mathcal{C} \backslash \mathcal{C}_i$. De plus $S_0 \in \mathcal{C}$, ce qui est exclu dans le cas 1. Par conséquent, $\forall l \in \{2, \ldots, N_{\mathcal{C}_i}\}$, $\mathrm{pa}(X_i^{(l)}) \subset \{X_i^{(1)}, \ldots, X_i^{(N_{\mathcal{C}_i})}\}$. Comme, par absence de circuit, $X_i^{(l+l')} \notin \mathrm{pa}(X_i^{(l)})$ pour $l' \in \{1, \ldots, N_{\mathcal{C}_i} - l\}$, on obtient $\mathrm{pa}(X_i^{(l)}) = \{X_i^{(1)}, \ldots, X_i^{(l-1)}\}$ et bien sûr $\mathrm{pa}(X_i^{(1)}) = \emptyset$.

<u>Cas 2</u> : S_0 appartient à plusieurs cliques. On note $C_0, \ldots, C_r$ les cliques auxquelles S_0 appartient. Nous allons alors montrer que le graphe engendré par les sommets des cliques C_i, noté $\bigcup_j \mathcal{C}_j$, est stable pour la relation de parenté (c'est-à-dire que $\forall X \in \bigcup_j \mathcal{C}_j$, $\mathrm{pa}(X) \subset \bigcup_j \mathcal{C}_j$). Nous allons également montrer que cette propriété reste vraie quand on restreint l'union à une quantité de cliques strictement décroissante, jusqu'à arriver à la clique C_0 voulue.

<u>HR_k</u> : si $k \leq r$, $\exists (i_0, \ldots, i_{r-k}) \in \{1, \ldots, r\}^{r-k}$ tel que $\bigcup_{j=0}^{r-k} \mathcal{C}_{i_j}$ est stable pour la relation de parenté.

<u>HR_0</u> : montrons que $\bigcup_j \mathcal{C}_j$ est stable pour la relation de parenté : soit $i \leq r$, $X \in \mathcal{C}_i$ et $Y \in \mathrm{pa}(X)$. Nous souhaitons montrer que l'une des cliques $\mathcal{C}_j$ contient Y. Comme $X \in \mathcal{C}_i$ et $S_0 \in \mathcal{C}_i$, il vient $S_0 \leftrightarrow X$ et $S_0 \to X$ car S_0 est une source. De plus, $Y \to X$ donc par moralité de $\mathcal{G}$ et comme S_0 est une source, $S_0 \to Y$. Le graphe $(\{Y, S_0\}, \{\{Y, S_0\}\})$ est un graphe complet, qui est donc sous-graphe de l'une des cliques $\mathcal{C}$. La clique $\mathcal{C}$ contient S_0 donc $\mathcal{C} \in \{\mathcal{C}_0, \ldots, \mathcal{C}_r\}$ d'où $Y \in \mathcal{G}(\bigcup_j \mathcal{C}_j)$. Cette preuve est illustrée par la figure 2.5.

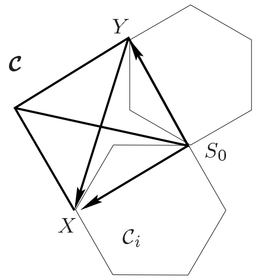


Fig. 2.5 – *Les sommets des cliques contenant la source S_0 ont leurs parents dans des cliques contenant S_0. Si $S_0 \in \mathcal{C}_i$ et si $X \in \mathcal{C}_i$ admet pour parent Y, alors Y est dans une clique $\mathcal{C}$ contenant $\mathcal{C}_0$, par moralité du graphe et par définition d'une source.*

<u>$\mathrm{HR}_k \Rightarrow \mathrm{HR}_{k+1}$</u> : par HR_k, il existe une union de $r-k+1$ cliques stable pour la relation de parenté. Quitte à renuméroter les cliques, nous supposons qu'il

s'agit de $\bigcup_{i=0}^{r-k} \mathcal{C}_i$. On va montrer (par l'absurde) que pour une certaine clique $\mathcal{C}_i$ de cette union, pour tout l, $X_i^{(\nu_i^*+l)}$ n'a aucun descendant dans les $\mathcal{C}_j \setminus \mathcal{C}_i$, faute de quoi on peut construire un circuit par récurrence.

Cette propriété est illustrée par la figure 2.6. Le graphe représenté sur cette figure admet pour cliques les graphes engendrés par $\{S_0, X_1, X_2\}$ (clique $\mathcal{C}_j$) et par $\{S_0, X_2, X_3, X_4, X_5\}$ (clique $\mathcal{C}_i$, pour laquelle $\nu_i^* = 3$). Supprimer les sommets X_4 et X_5 revient à conserver la clique $\mathcal{C}_j$, qui contient les parents de ses sommets. Ceci est en accord avec la propriété ci-dessus qui établit que les sommets au-delà de $\nu_i^* = 3$, c'est-à-dire X_4 et X_5, n'ont pas de descendant dans $\mathcal{C}_j$. En revanche, supprimer la clique $\mathcal{C}_j$ revient à garder la clique $\mathcal{C}_i$, qui ne contient pas les parents de ses sommets. En effet, X_1, qui est parent de X_2, n'est pas dans $\mathcal{C}_i$.

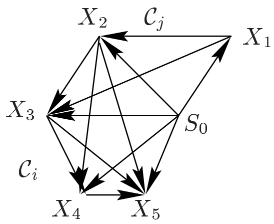


FIG. 2.6 – *Suppression d'une clique contenant S_0. Le graphe ci-dessus admet pour cliques les graphes engendrés par $\{S_0, X_1, X_2\}$ (clique $\mathcal{C}_j$) et par $\{S_0, X_2, X_3, X_4, X_5\}$ (clique $\mathcal{C}_i$). Supprimer les sommets X_4 et X_5 revient à conserver la clique $\mathcal{C}_j$, qui contient les parents de ses sommets.*

On utilise donc un raisonnement par l'absurde en supposant que pour tout i dans $\{0, \ldots, r-k\}$, il existe un k_i dans $\{1, \ldots, N_{\mathcal{C}_i} - \nu_i^*\}$ tel que $X_i^{(\nu_i^*+k_i)}$ possède un descendant dans l'un des ensembles $\mathcal{C}_j \setminus \mathcal{C}_i$. Nous notons $(*)$ cette hypothèse. La récurrence suivante permet alors de construire un circuit dans $\mathcal{G}$.

<u>HR$_t$</u> :

> il existe $(d_1, \ldots, d_t)$ dans $\{0, \ldots, r-k\}^t$ et un chemin passant par $X_{d_1}^{(\nu_{d_1}^*+k_{d_1})}, \ldots, X_{d_t}^{(\nu_{d_t}^*+k_{d_t})}$ et d'extrémités $X_{d_1}^{(\nu_{d_1}^*+k_{d_1})}$ et $X_{d_t}^{(\nu_{d_t}^*+k_{d_t})}$.

<u>HR$_2$</u> :

> Nous considérons, dans la clique $\mathcal{C}_1$, le sommet $X_1^{(\nu_1^*+k_1)}$. Par l'hypothèse $(*)$, $X_1^{(\nu_1^*+k_1)}$ admet un descendant noté $X_j^{(c_j)}$ dans une certaine clique $\mathcal{C}_j$ telle que $X_j^{(c_j)} \in \mathcal{C}_j \setminus \mathcal{C}_1$. On note alors $X_1^{(\nu_1^*+k_1)} \to X_j^{(c_j)}$.
>
> 1. Si $c_j < \nu_j^*+k_j$, alors $X_j^{(c_j)} \to X_j^{(\nu_j^*+k_j)}$ donc $(X_1^{(\nu_1^*+k_1)}, X_j^{(c_j)}, X_j^{(\nu_j^*+k_j)})$ est un chemin passant par (et d'extrémités) $X_1^{(\nu_1^*+k_1)}$ et $X_j^{(\nu_j^*+k_j)}$.

2. Si $c_j = \nu_j^* + k_j$, alors $X_1^{(\nu_1^*+k_1)} \to X_j^{(\nu_j^*+k_j)}$ donc $(X_1^{(\nu_1^*+k_1)}, X_j^{(\nu_j^*+k_j)})$ est un chemin passant par (et d'extrémités) $X_1^{(\nu_1^*+k_1)}$ et $X_j^{(\nu_j^*+k_j)}$.

3. Si $c_j > \nu_j^* + k_j$, alors $X_j^{(\nu_j^*+k_j)} \to X_j^{(c_j)}$. Par moralité de $\mathcal{G}$, on a $X_j^{(\nu_j^*+k_j)} \leftrightarrow X_1^{(\nu_1^*+k_1)}$. Alors soit $(X_1^{(\nu_1^*+k_1)}, X_j^{(\nu_j^*+k_j)})$ est un chemin passant par (et d'extrémités) $X_1^{(\nu_1^*+k_1)}$ et $X_j^{(\nu_j^*+k_j)}$, soit c'est le cas du chemin $(X_j^{(\nu_j^*+k_j)}, X_1^{(\nu_1^*+k_1)})$

<u>HR_t</u> $\Rightarrow$ <u>HR_{t+1}</u> :

par HR_t, il existe $(d_1, \ldots, d_t)$ dans $\{0, \ldots, r-k\}^t$ et un chemin passant par $X_{d_1}^{(\nu_{d_1}^*+k_{d_1})}, \ldots, X_{d_t}^{(\nu_{d_t}^*+k_{d_t})}$ et d'extrémités $X_{d_1}^{(\nu_{d_1}^*+k_{d_1})}$ et $X_{d_t}^{(\nu_{d_t}^*+k_{d_t})}$. Nous considérons, dans la clique $\mathcal{C}_{d_t}$, le sommet $X_{d_t}^{(\nu_{d_t}^*+k_{d_t})}$. Par l'hypothèse $(*)$, $X_{d_t}^{(\nu_{d_t}^*+k_{d_t})}$ admet un descendant noté $X_j^{(c_j)}$ dans une certaine clique $\mathcal{C}_j$ telle que $X_j^{(c_j)} \in \mathcal{C}_j \setminus \mathcal{C}_{d_t}$. On note alors $X_{d_t}^{(\nu_{d_t}^*+k_{d_t})} \to X_j^{(c_j)}$.

1. Si $c_j \leq \nu_j^* + k_j$, en appliquant le même raisonnement que pour HR_2, on peut construire un chemin entre $X_{d_t}^{(\nu_{d_t}^*+k_{d_t})}$ et $X_j^{(\nu_j^*+k_j)}$ passant ou non par $X_j^{(c_j)}$. On obtient alors un chemin passant par $X_{d_1}^{(\nu_{d_1}^*+k_{d_1})}, \ldots, X_{d_t}^{(\nu_{d_t}^*+k_{d_t})}, X_j^{(\nu_j^*+k_j)}$ et d'extrémités $X_{d_1}^{(\nu_{d_1}^*+k_{d_1})}$ et $X_j^{(\nu_j^*+k_j)}$.

2. Si $c_j > \nu_j^* + k_j$, alors dans le cas où $X_{d_t}^{(\nu_{d_t}^*+k_{d_t})} \to X_j^{(\nu_j^*+k_j)}$, on obtient le résultat directement, par le même raisonnement que dans le cas $t = 2$. Dans le cas où $X_j^{(\nu_j^*+k_j)} \to X_{d_t}^{(\nu_{d_t}^*+k_{d_t})}$, par moralité de $\mathcal{G}$, le prédécesseur Y de $X_{d_t}^{(\nu_{d_t}^*+k_{d_t})}$ dans le chemin construit à l'étape t de la récurrence est adjacent à $X_j^{(\nu_j^*+k_j)}$.
   Si $Y \to X_j^{(\nu_j^*+k_j)}$, alors on insère, dans ce chemin, le sommet $X_j^{(\nu_j^*+k_j)}$ entre Y et $X_{d_t}^{(\nu_{d_t}^*+k_{d_t})}$ (cas 2.7. a).
   Si $X_j^{(\nu_j^*+k_j)} \to Y$, on insère $X_j^{(\nu_j^*+k_j)}$ après le premier ancêtre de $X_{d_t}^{(\nu_{d_t}^*+k_{d_t})}$ (dans le chemin construit à l'étape t de la récurrence) parent de $X_j^{(\nu_j^*+k_j)}$ (cas 2.7. b) – ou avant $X_{d_1}^{(\nu_{d_1}^*+k_{d_1})}$ s'il n'en existe pas (cas 2.7. c).
   Ces trois cas sont illustrés par la figure 2.7. On obtient alors un chemin passant par tous les $X_{d_{t'}}^{(\nu_{d_{t'}}^*+k_{d_{t'}})}$, et d'extrémités deux des $X_{d_{t'}}^{(\nu_{d_{t'}}^*+k_{d_{t'}})}$, ce qui prouve l'hypothèse de récurrence au rang $t+1$.

<u>Conclusion</u> : pour $t = r - k + 1$, le nombre d'indices possibles pour les

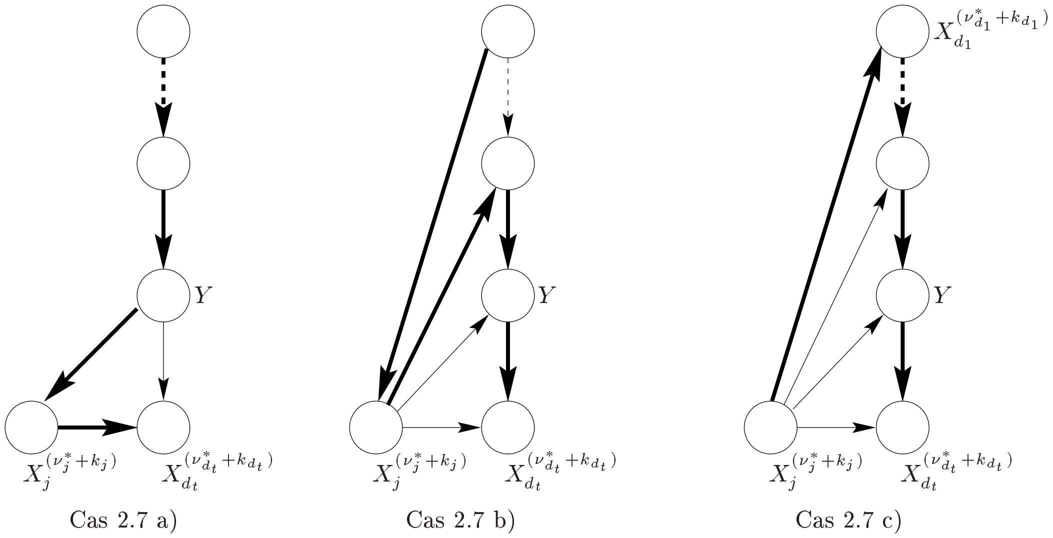


FIG. 2.7 – *Insertion d'un sommet dans un chemin passant par les sommets $X_{d_1}^{(\nu_{d_1}^*+k_{d_1})}, \ldots, X_{d_t}^{(\nu_{d_t}^*+k_{d_t})}$ adjacents à la source S_0. Dans le cas 2.7. a), le $t+1$ème sommet est inséré entre Y et le dernier sommet du chemin. Dans le cas 2.7. b), le $t+1$ème sommet est inséré après le premier ancêtre de $X_j^{(\nu_j^*+k_j)}$ (dans le chemin représenté) parent de $X_{d_t}^{(\nu_{d_t}^*+k_{d_t})}$. Dans le cas 2.7. c), le $t+1$ème sommet est inséré avant le premier sommet $X_{d_1}^{(\nu_{d_1}^*+k_{d_1})}$ du chemin. Le chemin obtenu est tracé en gras. Sa longueur est augmentée de 1.*

sommets $X_{d_{t'}}^{(\nu_{d_{t'}}^*+k_{d_{t'}})}$ étant fini de cardinal $r-k$, le chemin ainsi construit contient un circuit, ce qui contredit l'hypothèse que $\mathcal{G}$ est sans circuit. Ainsi, l'hypothèse $(*)$ est exclue ; autrement dit : il existe i dans $\{0, \ldots, r-k\}$ tel que pour tout l' dans $\{1, \ldots, N_{\mathcal{C}_i} - \nu_i^*\}$, $X_i^{(\nu_i^*+l')}$ n'a un aucun descendant dans l'un des ensembles $\mathcal{C}_j \setminus \mathcal{C}_i$ pour $j \in \{0, \ldots, r-k\}$ et $j \neq i$. Il nous reste à présent à supprimer l'une des cliques de $\bigcup_{j'=0}^{r-k} \mathcal{C}_{j'}$ de sorte que l'ensemble des sommets restants contienne encore les parents de ses sommets, ce qui conclura la deuxième récurrence.
Nous savons que pour tout l' dans $\{1, \ldots, N_{\mathcal{C}_i} - \nu_i^*\}$,

$$\mathrm{pa}(X_i^{(\nu_i^*+l')}) = \{X_i^{(1)}, \ldots, X_i^{(\nu_i^*+l'-1)}\},$$

ce qui exclut (par absence de circuit) que $X_i^{(\nu_i^*+l')}$ ait des sommets de la forme $X_i^{(\nu_i^*-l)}$ parmi ses descendants. D'autre part, comme

$$\mathrm{pa}(X_i^{(\nu_i^*)}) \subsetneq \{X_i^{(1)}, \ldots, X_i^{(\nu_i^*-1)}\},$$

il existe un parent Y de $X_i^{(\nu_i^*)}$ dans un certain $\mathcal{C}_j \setminus \mathcal{C}_i$, où $j \in \{0, \ldots, r-k\}$, puisque $\bigcup_{j'=0}^{r-k} \mathcal{C}_{j'}$ est stable pour la relation parentale (c'est notre hypothèse de récurrence au rang k). Puisque $X_i^{(l)} \to X_i^{(\nu_i^*)}$ pour $l < \nu_i^*$, alors par moralité de $\mathcal{G}$, $Y \leftrightarrow X_i^{(l)}$. De plus, comme les sommets $Y, X_i^{(1)}, \ldots, X_i^{(\nu_i^*)}$ sont dans des cliques $\mathcal{C}_{j'}$ avec $j \in \{0, \ldots, r-k\}$, ils admettent le sommet S_0 comme parent. Par conséquent, l'ensemble de sommets $\{S_0, Y, X_i^{(1)}, \ldots, X_i^{(\nu_i^*)}\}$ engendre un graphe complet, donc contenu dans une des cliques $\mathcal{C}_{j'}$ avec $j' \neq i$, car $Y \notin \mathcal{C}_i$. Alors chaque sommet de $\mathcal{C}_i \setminus \{X_i^{(\nu_i^*+1)}, \ldots X_i^{(N_{\mathcal{C}_i})}\}$ appartient à l'une des cliques $\mathcal{C}_{j'}$ avec $j' \neq i$ et $j' \leq r-k$. Donc $\bigcup_{j=0, j\neq i}^{r-k} \mathcal{C}_j$ est constitué des sommets de $\bigcup_{j=0}^{r-k} \mathcal{C}_j$ privé des $X_i^{(\nu_i^*+l)}$. Ceci est illustré par la figure 2.6 : nous avons en fait montré que les sommets $X_i^{(\nu_i^*+1)}, \ldots, X_i^{(N_{\mathcal{C}_i})}$ ne sont les parents d'aucun sommet de $\bigcup_{j=0, j\neq i}^{r-k} \mathcal{C}_j$ et d'aucun sommet de $\{X_i^{(1)}, \ldots, X_i^{(\nu_i^*)}\}$. Sur la figure 2.6, les sommets X_4 et X_5 ne sont parent d'aucun autre sommet. D'autre part, en supprimant totalement la clique $\mathcal{C}_i$, on ne supprime en fait que les sommets $X_i^{(\nu_i^*+1)}, \ldots, X_i^{(N_{\mathcal{C}_i})}$ car les autres sommets de $\mathcal{C}_i$ sont aussi dans d'autres cliques de $\bigcup_{j=0, j\neq i}^{r-k} \mathcal{C}_j$. Sur la figure 2.6, les sommets S_0, X_2 et X_3 sont aussi dans la clique $\mathcal{C}_j$, donc supprimer la clique $\mathcal{C}_i$ revient à supprimer les sommets X_4 et X_5 uniquement. Par conséquent, l'ensemble $\bigcup_{j=0, j\neq i}^{r-k} \mathcal{C}_j$ reste stable pour la relation parentale, ce qui découle de ce choix particulier de $\mathcal{C}_i$. Ceci prouve l'hypothèse de récurrence au rang k.

Conclusion : en appliquant le résultat de la récurrence pour $k = r$, on obtient une clique $\mathcal{C}_0$ stable pour la relation parentale, donc égale à son graphe ancestral.

Remarque 2.7
1. *Par la suite, la notation $\mathcal{C}_0$ fera référence à une clique vérifiant la proposition 5.*
2. *La démonstration de cette proposition donne une méthode pratique pour déterminer une telle clique dans un graphe.*
3. *Il peut exister plusieurs cliques vérifiant cette propriété, voir figure 2.8.*

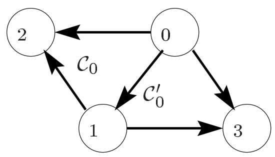


FIG. 2.8 – *Un graphe comportant plusieurs cliques stables pour la relation parentale.*

En choisissant $\mathcal{C}_0$ comme racine de l'arbre des cliques, on définit une orientation de cet arbre : soient $\mathcal{C}$ et $\mathcal{C}'$ deux cliques adjacentes dans l'arbre de jonction, distincte de

$\mathcal{C}_0$. Nous dirons que $\mathcal{C}$ est le parent de $\mathcal{C}'$ si et seulement si $\mathcal{C}$ est sur le chemin entre $\mathcal{C}'$ et $\mathcal{C}_0$. Dans le cas contraire, $\mathcal{C}'$ est le parent de $\mathcal{C}$. La clique $\mathcal{C}_0$ est parent de toutes ses cliques adjacentes. L'orientation de l'arbre de jonction est illustrée par la figure 2.9. Comme l'énonce la proposition 6 ci-dessous, il existe un chemin entre les sources des graphes $\mathcal{G}(\mathcal{C}_i)$, dont les arcs ont même sens que celles du chemin de l'arbre des cliques. Cette propriété est illustrée par la figure 2.9. Son intérêt est essentiellement de montrer qu'en général, au moins un sommet de chaque séparateur de cliques (qui est la source d'une des cliques) admet un parent dans la clique précédente. Nous en déduirons une manière inductive de calculer la loi des séparateurs de cliques.

Proposition 6 *Soit* $(\mathcal{C}_0, \ldots, \mathcal{C}_{i-1}, \mathcal{C}_i)$ *un chemin de longueur* i *de l'arbre de jonction. Alors pour tout* $j \leq i$, $X_j^{(1)} \in \mathcal{C}_{j-1} \cap \mathcal{C}_j$ *et* $[X_j^{(1)} = X_{j-1}^{(1)}$ *ou* $X_{j-1}^{(1)} \in pa(X_j^{(1)})]$.

Synopsis de la preuve.
On montre cette proposition par récurrence. La démonstration pour le cas $i = 1$ repose essentiellement sur la propriété de moralité de $\mathcal{G}$ et sur le lemme 2. La récurrence utilise une démonstration par l'absurde, en supposant que $X_{i+1}^{(1)} \notin \mathcal{C}_i \cap \mathcal{C}_{i+1}$. Toujours par l'absurde et en utilisant le lemme 2, on montre que $X_{i-1}^{(1)} = X_i^{(1)}$.

Démonstration

Ce résultat se montre par récurrence sur i :

HR$_i$: résultat énoncé ci-dessus.

HR$_1$: Soit $(\mathcal{C}_0, \mathcal{C}_1)$ un chemin de l'arbre de jonction.

Cas 1 : $S_0 = X_1^{(1)}$. Alors $X_1^{(1)} \in \mathcal{C}_0$ et le résultat est montré pour $i = 1$ (rappelons que $S_0 = X_0^{(1)}$).

Cas 2 : $S_0 \neq X_1^{(1)}$. Soit $X \in \mathcal{C}_0 \cap \mathcal{C}_1$. Comme $S_0 \in \mathcal{C}_0$, et par définition de $X_1^{(1)}$, il vient $S_0 \to X$ et $X_1^{(1)} \to X$. Par moralité de $\mathcal{G}$ et comme S_0 est la source, on a $S_0 \to X_1^{(1)}$. Le fait que pa$(S_0) = \emptyset$ et donc que $X_1^{(1)} \notin$ pa(S_0) exclut l'hypothèse que S_0 soit dans $\mathcal{C}_1$, par définition de $X_1^{(1)}$. Par le lemme 2, comme $\mathcal{C}_1$ et $\mathcal{C}_0$ sont adjacentes dans l'arbre de jonction, il vient : $X_1^{(1)} \in \mathcal{C}_0$. D'où $X_1^{(1)} \in \mathcal{C}_0 \cap \mathcal{C}_1$ et $X_1^{(0)} \to X_1^{(1)}$. Ceci prouve notre hypothèse de récurrence au rang $i = 1$.

HR$_i$ $\Rightarrow$ HR$_{i+1}$: soit $(\mathcal{C}_0, \ldots, \mathcal{C}_i, \mathcal{C}_{i+1})$ un chemin de longueur i de l'arbre de jonction. Supposons que $X_{i+1}^{(1)} \notin \mathcal{C}_i \cap \mathcal{C}_{i+1}$, ce qui revient à supposer que $X_{i+1}^{(1)} \in \mathcal{C}_{i+1} \setminus \mathcal{C}_i$. Du fait que $X_{i+1}^{(1)} \notin \mathcal{C}_i$, on a $X_{i+1}^{(1)} \neq X_i^{(1)}$. Soit $X \in \mathcal{C}_i \cap \mathcal{C}_{i+1}$: alors $X_i^{(1)} \to X$ et $X_{i+1}^{(1)} \to X$, donc par moralité de $\mathcal{G}$, $X_i^{(1)} \leftrightarrow X_{i+1}^{(1)}$. Par le lemme 2, comme $\mathcal{C}_i$ et $\mathcal{C}_{i+1}$ sont adjacentes dans l'arbre de jonction il vient $X_i^{(1)} \in \mathcal{C}_{i+1}$, donc $X_{i+1}^{(1)} \to X_i^{(1)}$. On va montrer que $X_i^{(1)} = X_{i-1}^{(1)}$ en prouvant que le cas $X_i^{(1)} \neq X_{i-1}^{(1)}$ est impossible :

Cas 1 : $X_i^{(1)} \neq X_{i-1}^{(1)}$. Par l'hypothèse HR$_i$, $X_{i-1}^{(1)} \to X_i^{(1)}$, donc par moralité de $\mathcal{G}$, $X_{i-1}^{(1)} \leftrightarrow X_{i+1}^{(1)}$. Par le lemme 2,

1. soit $X_{i+1}^{(1)} \in \mathcal{C}_{i-1}$. Alors par propriété d'intersection de l'arbre de jonction, $X_{i+1}^{(1)} \in \mathcal{C}_{i-1} \cap \mathcal{C}_{i+1} \Rightarrow X_{i+1}^{(1)} \in \mathcal{C}_i$, ce qui est contraire à notre hypothèse ;
2. soit $\{X_{i+1}^{(1)}, X_{i-1}^{(1)}\} \subset \mathcal{C}_i$ ce qui est encore contraire à notre hypothèse ;
3. soit $X_{i-1}^{(1)} \in \mathcal{C}_{i+1}$. Alors par propriété d'intersection de l'arbre de jonction, $X_{i-1}^{(1)} \in \mathcal{C}_{i-1} \cap \mathcal{C}_{i+1} \Rightarrow X_{i-1}^{(1)} \in \mathcal{C}_i$ donc $X_i^{(1)} \to X_{i-1}^{(1)}$ ce qui est contraire à l'hypothèse que $\mathcal{G}$ est sans circuit.

On en déduit que $X_i^{(1)} = X_{i-1}^{(1)}$, et nous sommes dans le

Cas 2 . Posons $i^* = \max\{j \leq i \,|\, X_j^{(1)} \neq X_i^{(1)}\}$. Cet ensemble est non vide car comme $X_{i+1}^{(1)} \to X_i^{(1)}$, il vient $S_0 \neq X_i^{(1)}$, par définition de S_0 et vu que $S_0 = X_0^{(1)}$. L'entier i^* est donc bien défini et $i^* < i - 1$. Par définition de i^*, pour tout j tel que $i > j > i^*$, $\quad X_j^{(1)} = X_i^{(1)}$ et par HR$_i$, $X_{i^*}^{(1)} \to X_i^{(1)}$. Par moralité de $\mathcal{G}$, $X_{i^*}^{(1)} \leftrightarrow X_{i+1}^{(1)}$. Par le lemme 2,

1. soit $X_{i+1}^{(1)} \in \mathcal{C}_{i^*}$. Alors par propriété d'intersection de l'arbre de jonction, $X_{i+1}^{(1)} \in \mathcal{C}_{i^*} \cap \mathcal{C}_{i+1} \Rightarrow X_{i+1}^{(1)} \in \mathcal{C}_i$, ce qui est contraire à notre hypothèse,
2. soit $\{X_{i+1}^{(1)}, X_{i^*}^{(1)}\} \subset \mathcal{C}_j$ pour $j \in \{i^* + 1, \ldots, i\}$. Alors $X_j^{(1)} \to X_{i+1}^{(1)}$ avec $X_j^{(1)} = X_i^{(1)}$, ce qui est exclu,
3. soit $X_{i^*}^{(1)} \in \mathcal{C}_{i+1}$. Alors par propriété d'intersection de l'arbre de jonction, $X_{i^*}^{(1)} \in \mathcal{C}_{i+1} \cap \mathcal{C}_{i^*} \Rightarrow X_{i^*}^{(1)} \in \mathcal{C}_i$, donc $X_i^{(1)} \to X_{i^*}^{(1)}$ ce qui contredit $X_{i^*}^{(1)} \to X_i^{(1)}$.

Conclusion : $X_{i+1}^{(1)} \in \mathcal{C}_i \cap \mathcal{C}_{i+1}$ donc soit $X_{i+1}^{(1)} = X_i^{(1)}$, soit $X_i^{(1)} \to X_{i+1}^{(1)}$ car $X_{i+1}^{(1)} \in \mathcal{C}_i$.

2.4.2 Phase arrière

La phase arrière est un parcours de graphe qui part des feuilles de l'arbre des cliques et remonte jusqu'à la racine $\mathcal{C}_0$, parcourant ainsi les arcs de l'arbre (ou, de manière équivalente, ses cliques) en sens contraire de leur orientation, d'où son nom. Elle permet le calcul de la vraisemblance et également le calcul de certaines probabilités notées $\tilde{\beta}_a(\boldsymbol{s}_{\mathbb{S}_a})$ et définies ci-dessous. Enfin, cette phase est requise par la phase avant, qui calcule, pour chaque clique $\mathcal{C}$, les probabilités $P(\boldsymbol{S}_\mathcal{C} = \boldsymbol{s}_\mathcal{C} | \boldsymbol{Y} = \boldsymbol{y})$ intervenant dans l'algorithme EM de la section 2.3 (voir la proposition 1 et la remarque qui la suit). Elle utilise les probabilités $\tilde{\beta}_a$ définies comme suit : soit a un arc de l'arbre de jonction $\mathcal{T}$, avec $a = (\mathcal{C}_i, \mathcal{C}_j)$, la clique $\mathcal{C}_i$ étant donc située sur le chemin entre $\mathcal{C}_0$ et $\mathcal{C}_j$ dans l'arbre de jonction. La suppression de a dans l'arbre des cliques le sépare en deux composantes, dont une seule ne contient pas $\mathcal{C}_0$. Nous notons $\mathcal{T}_a$ l'ensemble des cliques comprises dans cette partie, l'ensemble

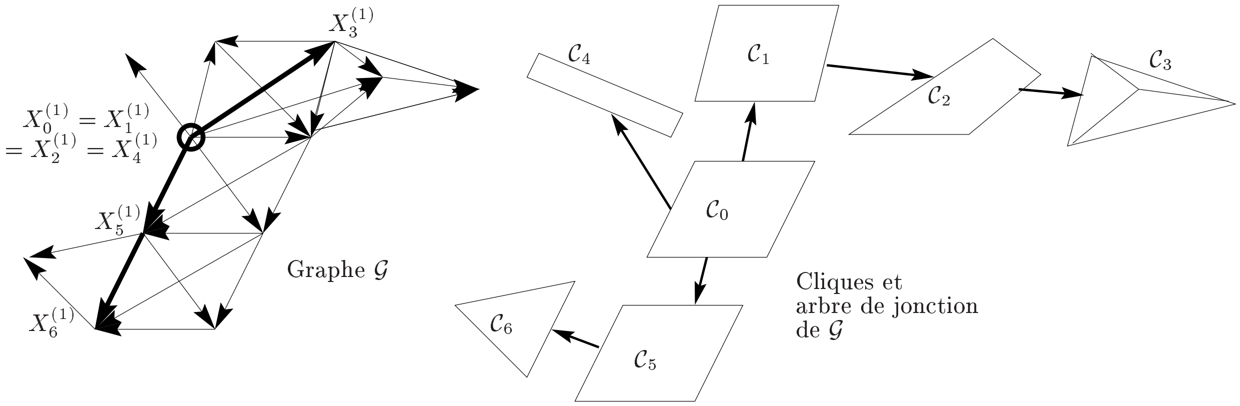


FIG. 2.9 – *Existence de chemins reliant les sources des cliques. Les arcs de l'arbre des cliques correspondent à des arcs entre les sources des cliques (exception faite des sommets qui sont source de plusieurs cliques, comme $X_0^{(1)}$).*

des autres cliques étant noté $\mathcal{T}_a^c$. Nous noterons $\mathcal{T}_a^-$ l'ensemble des cliques de $\mathcal{T}_a$ privé de la clique $\mathcal{C}_j$ et $\mathcal{T}_a^{c-}$ l'ensemble des cliques de $\mathcal{T}_a^c$ privé de la clique $\mathcal{C}_i$. Les sommets des cliques de $\mathcal{T}_a$ engendrent un graphe dont l'ensemble des sommets est noté $\bar{\mathcal{K}}_a$. Nous notons $\mathcal{K}_a = \bar{\mathcal{K}}_a \setminus \mathcal{S}_a$ (rappelons que $\mathcal{S}_a$ désigne le graphe engendré par l'ensemble des sommets de $\mathcal{C}_i \cap \mathcal{C}_j$, ou par raccourci ses sommets eux-mêmes). Les sommets des cliques de $\mathcal{T}_a^c$ engendrent un graphe dont l'ensemble des sommets est noté $\bar{\mathcal{K}}_a^c$. Nous notons $\mathcal{K}_a^c = \bar{\mathcal{K}}_a^c \setminus \mathcal{S}_a$ La fonction $\tilde{\beta}_a$ est alors définie par $\tilde{\beta}_a(\tilde{\boldsymbol{y}}, \tilde{\boldsymbol{x}}) = P(\boldsymbol{Y}_{\mathcal{K}_a} = \tilde{\boldsymbol{y}} | \boldsymbol{X}_{\mathcal{S}_a} = \tilde{\boldsymbol{x}})$, pour $\tilde{\boldsymbol{y}} \in \mathcal{Y}_{\mathcal{K}_a}$ et $\tilde{\boldsymbol{x}} \in \mathcal{X}_{\mathcal{S}_a}$. Ces notations sont illustrées par les figures 2.10 et 2.11. Notons qu'en pratique, la seule valeur de $\tilde{\boldsymbol{y}}$ qui nous intéresse, donc pour laquelle $\tilde{\beta}_a$ a besoin d'être calculée, est celle qui correspond à la valeur observée $\boldsymbol{y}_{\mathcal{K}_a}$. De même, $\boldsymbol{X}_{\mathcal{S}_a}$ comprend des variables aléatoires observées $\boldsymbol{Y}_{\mathcal{S}_a}$ et cachées $\boldsymbol{S}_{\mathcal{S}_a}$. Il suffit alors de calculer la quantité $\tilde{\beta}_a$ pour la valeur $\boldsymbol{Y}_{\mathcal{S}_a} = \boldsymbol{y}_{\mathcal{S}_a}$, cependant elle doit être calculée pour toute valeur $\boldsymbol{s}_{\mathcal{S}_a} \in \boldsymbol{\mathcal{S}}_{\mathcal{S}_a}$ des variables aléatoires cachées intervenant dans $\mathcal{S}_a$. On notera alors $\tilde{\beta}_a(\boldsymbol{s}_{\mathcal{S}_a})$ pour désigner $P(\boldsymbol{Y}_{\mathcal{K}_a} = \boldsymbol{y}_{\mathcal{K}_a} | \boldsymbol{Y}_{\mathcal{S}_a} = \boldsymbol{y}_{\mathcal{S}_a}, \boldsymbol{S}_{\mathcal{S}_a} = \boldsymbol{s}_{\mathcal{S}_a})$.

Propagation

Soit un arc $a = (\mathcal{C}_i, \mathcal{C}_j)$ de l'arbre de jonction. Par définition de la phase arrière, les quantités $\tilde{\beta}_{a_{j_l}}(\boldsymbol{s}_{\mathcal{S}_{a_{j_l}}})$ sont supposées connues pour tous les arcs a_{j_l} incidents à $\mathcal{C}_j$ autres que l'arc a (qui est le seul menant vers $\mathcal{C}_0$) et pour toutes les valeurs possibles de $\boldsymbol{s}_{\mathcal{S}_{a_{j_l}}} \in \boldsymbol{\mathcal{S}}_{\mathcal{S}_{a_{j_l}}}$. Nous rappelons que les valeurs de $\boldsymbol{Y}$ sont supposées connues, fixées à $\boldsymbol{y}$. Nous désirons alors calculer $\tilde{\beta}_a(\boldsymbol{s}_{\mathcal{S}_a})$ à partir des $\tilde{\beta}_{a_{j_l}}(\boldsymbol{s}_{\mathcal{S}_{a_{j_l}}})$. Pour cela, nous nous intéressons successivement aux séparateurs $\mathcal{S}_{a_{j_l}}$, en commençant par exemple par $\mathcal{S}_{a_{j_1}}$. Nous travaillons pour commencer sur la loi jointe de $\boldsymbol{Y}_{\mathcal{K}_a}, \boldsymbol{Y}_{\mathcal{S}_a}, \boldsymbol{S}_{\mathcal{S}_a}$ dont les variables se

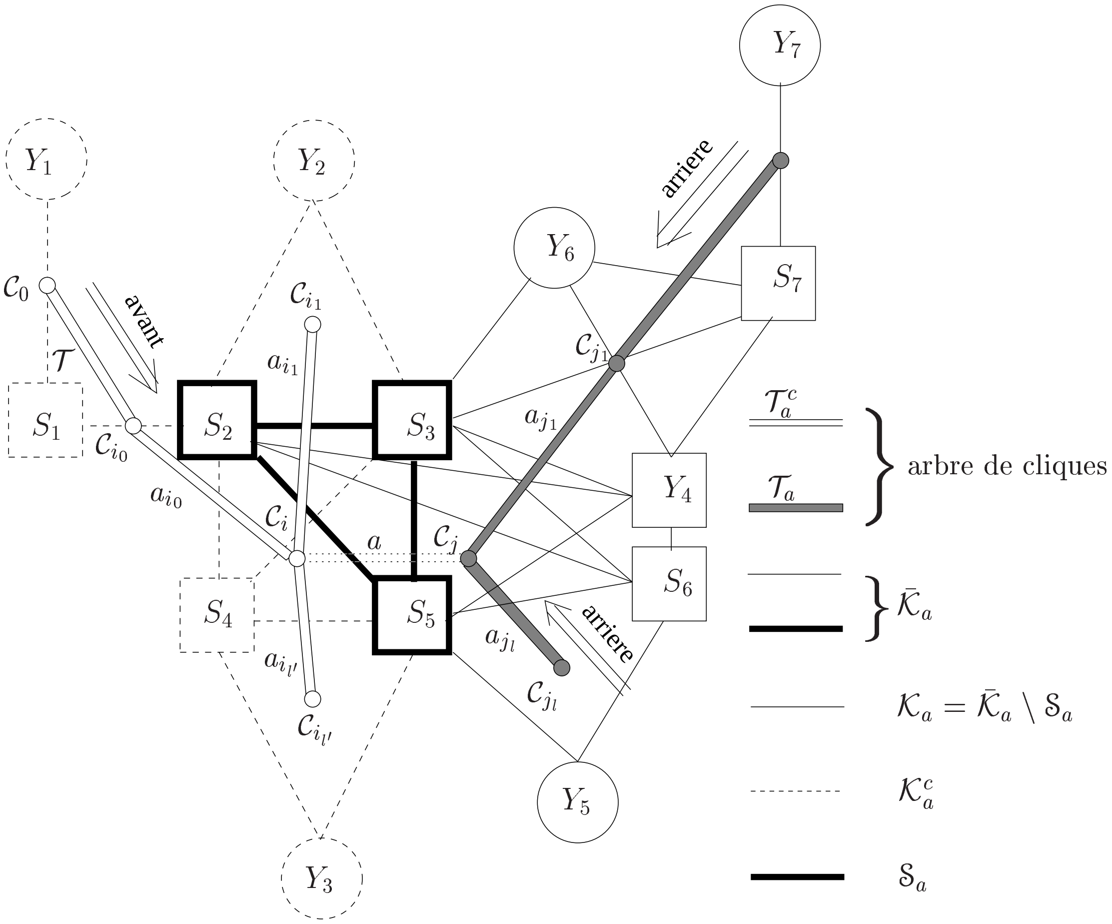


FIG. 2.10 – *Variables aléatoires intervenant dans l'algorithme arrière-avant et autres notations. L'arbre de jonction $\mathcal{T}$ est un arbre dont les sommets, représentés par des petits cercles, sont les cliques $\mathcal{C}_k$. Ses arcs sont représentés par des traits épais blancs ou gris. Les algorithmes avant et arrière parcourent les arcs a de l'arbre de jonction, avec $a = (\mathcal{C}_i, \mathcal{C}_j)$, la clique $\mathcal{C}_i$ étant située sur le chemin entre $\mathcal{C}_0$ et $\mathcal{C}_j$. La phase arrière débute aux feuilles de $\mathcal{T}$ et l'algorithme avant débute à la racine $\mathcal{C}_0$. La suppression, dans l'arbre des cliques, de l'arc a (dessiné en pointillés fins) le sépare en deux composantes, dont une seule ne contient pas $\mathcal{C}_0$: nous notons $\mathcal{T}_a$ l'ensemble des cliques comprises dans cette partie, l'ensemble des autres cliques étant noté $\mathcal{T}_a^c$. Les sommets des cliques de $\mathcal{T}_a$ engendrent un graphe dont l'ensemble des sommets est noté $\bar{\mathcal{K}}_a$ (partie du graphe en traits pleins). Le séparateur de cliques $\mathcal{S}_a = \mathcal{C}_i \cap \mathcal{C}_j$ est représenté en traits noirs épais. Nous notons $\mathcal{K}_a$ l'ensemble $\bar{\mathcal{K}}_a \setminus \mathcal{S}_a$, dessiné en traits noirs fins. Les sommets des cliques de $\mathcal{T}_a^c$ engendrent un graphe dont l'ensemble des sommets est noté $\bar{\mathcal{K}}_a^c$. Nous notons $\mathcal{K}_a^c = \bar{\mathcal{K}}_a^c \setminus \mathcal{S}_a$ (graphe en pointillés).*

scindent comme suit

$$\bar{\mathcal{K}}_a = \left(\bigcup_l \bar{\mathcal{K}}_{a_{j_l}}\right) \cup \mathcal{C}_j \text{ , d'où}$$

$$\bar{\mathcal{K}}_a \setminus \mathcal{S}_{a_{j_1}} = \mathcal{K}_{a_{j_1}} \cup \left(\bigcup_{l\neq 1} (\bar{\mathcal{K}}_{a_{j_l}} \setminus \mathcal{S}_{a_{j_1}})\right) \cup (\mathcal{C}_j \setminus \mathcal{S}_{a_{j_1}}). \tag{2.17}$$

Comme $\mathcal{S}_{a_{j_1}}$ est un séparateur de cliques, ses sommets séparent ceux de $\boldsymbol{Y}_{\mathcal{K}_{a_{j_1}}}$ et $\boldsymbol{S}_{\mathcal{S}_{a_{j_1}}\cap\mathcal{S}_a}$ de ceux des $\boldsymbol{Y}_{\bar{\mathcal{K}}_{a_{j_l}}\setminus\mathcal{S}_{a_{j_1}}}$, de $\boldsymbol{Y}_{\mathcal{C}_j\setminus\mathcal{S}_{a_{j_1}}}$ et de $\boldsymbol{S}_{\mathcal{S}_a\setminus\mathcal{S}_{a_{j_1}}}$. Or toutes les variables aléatoires de $\boldsymbol{S}_{\mathcal{S}_{a_{j_1}}}$ n'apparaissent pas dans les ensembles ci-dessus car elles ne sont pas toutes dans $\boldsymbol{S}_{\mathcal{S}_{a_{j_1}}\cap\mathcal{S}_a}$: il manque les variables aléatoires de $\boldsymbol{S}_{\mathcal{S}_{a_{j_1}}\setminus\mathcal{S}_a}$ pour faire intervenir $\boldsymbol{X}_{\mathcal{S}_{a_{j_1}}}$ tout entier. Or

$$\begin{aligned}
&P(\boldsymbol{Y}_{\mathcal{K}_a} = \boldsymbol{y}_{\mathcal{K}_a}, \boldsymbol{Y}_{\mathcal{S}_a} = \boldsymbol{y}_{\mathcal{S}_a}, \boldsymbol{S}_{\mathcal{S}_a} = \boldsymbol{s}_{\mathcal{S}_a}) \\
&\quad = \sum_{\boldsymbol{s}_{\mathcal{S}_{a_{j_1}}\setminus\mathcal{S}_a}} P(\boldsymbol{Y}_{\mathcal{K}_a} = \boldsymbol{y}_{\mathcal{K}_a}, \boldsymbol{Y}_{\mathcal{S}_a} = \boldsymbol{y}_{\mathcal{S}_a}, \boldsymbol{S}_{\mathcal{S}_a} = \boldsymbol{s}_{\mathcal{S}_a}, \boldsymbol{S}_{\mathcal{S}_{a_{j_1}}\setminus\mathcal{S}_a} = \boldsymbol{s}_{\mathcal{S}_{a_{j_1}}\setminus\mathcal{S}_a})
\end{aligned}$$

où, d'après la décomposition (2.17)

$$\begin{aligned}
&P(\boldsymbol{Y}_{\mathcal{K}_a} = \boldsymbol{y}_{\mathcal{K}_a}, \boldsymbol{Y}_{\mathcal{S}_a} = \boldsymbol{y}_{\mathcal{S}_a}, \boldsymbol{S}_{\mathcal{S}_a} = \boldsymbol{s}_{\mathcal{S}_a}, \boldsymbol{S}_{\mathcal{S}_{a_{j_1}}\setminus\mathcal{S}_a} = \boldsymbol{s}_{\mathcal{S}_{a_{j_1}}\setminus\mathcal{S}_a}) \\
&\quad = P(\{\boldsymbol{Y}_{\mathcal{K}_{a_{j_1}}} = \boldsymbol{y}_{\mathcal{K}_{a_{j_1}}}\} \bigcap_{l\neq 1} \{\boldsymbol{Y}_{\bar{\mathcal{K}}_{a_{j_l}}\setminus\mathcal{S}_{a_{j_1}}} = \boldsymbol{y}_{\bar{\mathcal{K}}_{a_{j_l}}\setminus\mathcal{S}_{a_{j_1}}}\} \cap \{\boldsymbol{Y}_{\mathcal{C}_j\setminus\mathcal{S}_{a_{j_1}}} = \boldsymbol{y}_{\mathcal{C}_j\setminus\mathcal{S}_{a_{j_1}}}\} \dots \\
&\quad \cap \{\boldsymbol{S}_{\mathcal{S}_a\setminus\mathcal{S}_{a_{j_1}}} = \boldsymbol{s}_{\mathcal{S}_a\setminus\mathcal{S}_{a_{j_1}}}\} \cap \{\boldsymbol{S}_{\mathcal{S}_{a_{j_1}}\cap\mathcal{S}_a} = \boldsymbol{s}_{\mathcal{S}_{a_{j_1}}\cap\mathcal{S}_a}\} \dots \\
&\quad \cap \{\boldsymbol{S}_{\mathcal{S}_{a_{j_1}}\setminus\mathcal{S}_a} = \boldsymbol{s}_{\mathcal{S}_{a_{j_1}}\setminus\mathcal{S}_a}\} \cap \{\boldsymbol{Y}_{\mathcal{S}_{a_{j_1}}} = \boldsymbol{y}_{\mathcal{S}_{a_{j_1}}}\}).
\end{aligned}$$

Notons que les éléments de $\boldsymbol{S}_{\mathcal{S}_{a_{j_1}}\cap\mathcal{S}_a}, \boldsymbol{S}_{\mathcal{S}_{a_{j_1}}\setminus\mathcal{S}_a}$ et $\boldsymbol{Y}_{\mathcal{S}_{a_{j_1}}}$ se regroupent en $\boldsymbol{X}_{\mathcal{S}_{a_{j_1}}}$. Rappelons que les sommets de $\mathcal{S}_{a_{j_1}}$ séparent ceux de $\boldsymbol{Y}_{\mathcal{K}_{a_{j_1}}}$ et $\boldsymbol{S}_{\mathcal{S}_{a_{j_1}}\cap\mathcal{S}_a}$ de ceux des $\boldsymbol{Y}_{\bar{\mathcal{K}}_{a_{j_l}}\setminus\mathcal{S}_{a_{j_1}}}$, de $\boldsymbol{Y}_{\mathcal{C}_j\setminus\mathcal{S}_{a_{j_1}}}$ et de $\boldsymbol{S}_{\mathcal{S}_a\setminus\mathcal{S}_{a_{j_1}}}$. Par équivalence entre la notion de séparation et celle d'indépendance conditionnelle, on obtient (en notant, de manière abrégée, $P(\boldsymbol{Y}_A)$ pour $P(\{\boldsymbol{Y}_A = \boldsymbol{y}_A\})$),

$$\begin{aligned}
&P(\boldsymbol{Y}_{\mathcal{K}_{a_{j_1}}}, \bigcap_{l\neq 1} \boldsymbol{Y}_{\bar{\mathcal{K}}_{a_{j_l}}\setminus\mathcal{S}_{a_{j_1}}}, \boldsymbol{Y}_{\mathcal{C}_j\setminus\mathcal{S}_{a_{j_1}}}, \boldsymbol{S}_{\mathcal{S}_a\setminus\mathcal{S}_{a_{j_1}}} | \boldsymbol{X}_{\mathcal{S}_{a_{j_1}}}) \\
&\quad = P(\boldsymbol{Y}_{\mathcal{K}_{a_{j_1}}} | \boldsymbol{X}_{\mathcal{S}_{a_{j_1}}}) P(\bigcap_{l\neq 1} \boldsymbol{Y}_{\bar{\mathcal{K}}_{a_{j_l}}\setminus\mathcal{S}_{a_{j_1}}}, \boldsymbol{Y}_{\mathcal{C}_j\setminus\mathcal{S}_{a_{j_1}}}, \boldsymbol{S}_{\mathcal{S}_a\setminus\mathcal{S}_{a_{j_1}}} | \boldsymbol{X}_{\mathcal{S}_{a_{j_1}}})
\end{aligned}$$

On en déduit, en multipliant l'équation ci-dessus par $P(\boldsymbol{X}_{\mathcal{S}_{a_{j_1}}} = \boldsymbol{x}_{\mathcal{S}_{a_{j_1}}})$ et en remarquant que $P(\boldsymbol{Y}_{\mathcal{K}_{a_{j_1}}} = \boldsymbol{y}_{\mathcal{K}_{a_{j_1}}} | \boldsymbol{X}_{\mathcal{S}_{a_{j_1}}} = \boldsymbol{x}_{\mathcal{S}_{a_{j_1}}}) = \tilde{\beta}_{a_{j_1}}(\boldsymbol{s}_{\mathcal{S}_{a_{j_1}}})$,

$$\begin{aligned}
&P(\boldsymbol{Y}_{\mathcal{K}_a} = \boldsymbol{y}_{\mathcal{K}_a}, \boldsymbol{Y}_{\mathcal{S}_a} = \boldsymbol{y}_{\mathcal{S}_a}, \boldsymbol{S}_{\mathcal{S}_a} = \boldsymbol{s}_{\mathcal{S}_a}) \\
&\quad = \sum_{\boldsymbol{s}_{\mathcal{S}_{a_{j_1}}\setminus\mathcal{S}_a} \in \mathcal{S}_{\mathcal{S}_{a_{j_1}}\setminus\mathcal{S}_a}} \Big[\tilde{\beta}_{a_{j_1}}(\boldsymbol{s}_{\mathcal{S}_{a_{j_1}}}) \\
&\quad \times P(\bigcap_{l\neq 1} \{\boldsymbol{Y}_{\bar{\mathcal{K}}_{a_{j_l}}} = \boldsymbol{y}_{\bar{\mathcal{K}}_{a_{j_l}}}\} \cap \{\boldsymbol{Y}_{\mathcal{C}_j} = \boldsymbol{y}_{\mathcal{C}_j}\} \cap \{\boldsymbol{S}_{\mathcal{S}_a} = \boldsymbol{s}_{\mathcal{S}_a}\} \cap \{\boldsymbol{S}_{\mathcal{S}_{a_{j_1}}\setminus\mathcal{S}_a} = \boldsymbol{s}_{\mathcal{S}_{a_{j_1}}\setminus\mathcal{S}_a}\})\Big]
\end{aligned} \tag{2.18}$$

En répétant ce raisonnement pour chaque séparateur $\mathcal{S}_{a_{j_l}}$ puis en conditionnant par $\{\boldsymbol{Y}_{\mathcal{S}_a} = \boldsymbol{y}_{\mathcal{S}_a}\} \cap \{\boldsymbol{S}_{\mathcal{S}_a} = \boldsymbol{s}_{\mathcal{S}_a}\}$, on obtient la formule de propagation

$$\tilde{\beta}_a(\boldsymbol{s}_{\mathcal{S}_a}) = \sum_{\boldsymbol{s}_{\mathcal{H}_a} \in \mathcal{S}_{\mathcal{H}_a}} P(\boldsymbol{Y}_{\mathcal{C}_j \setminus \mathcal{S}_a} = \boldsymbol{y}_{\mathcal{C}_j \setminus \mathcal{S}_a}, \boldsymbol{S}_{\mathcal{H}_a} = \boldsymbol{s}_{\mathcal{H}_a} | \boldsymbol{X}_{\mathcal{S}_a} = \boldsymbol{x}_{\mathcal{S}_a}) \prod_l \tilde{\beta}_{a_{j_l}}(\boldsymbol{s}_{\mathcal{S}_{a_{j_l}}}) \tag{2.19}$$

en notant $\mathcal{H}_a = \bigcup_l \mathcal{S}_{a_{j_l}} \setminus \mathcal{S}_a$ l'ensemble des variables aléatoires de $\mathcal{C}_j$ étant présentes dans au moins l'une des cliques $\mathcal{C}_{j_l}$ mais pas dans $\mathcal{C}_i$. Étant donné qu'a priori, seule la loi conditionnelle

$$P(\boldsymbol{X}_{\mathcal{C}_j \setminus \mathcal{S}_a} = \boldsymbol{x}_{\mathcal{C}_j \setminus \mathcal{S}_a} | \boldsymbol{X}_{\mathcal{S}_a} = \boldsymbol{x}_{\mathcal{S}_a})$$

est connue (car donnée par les expressions (2.22) ou (2.23)), on peut facilement faire intervenir les variables cachées de $\mathcal{C}_j$ non utilisées dans l'équation 2.19 pour arriver au résultat

$$\tilde{\beta}_a(\boldsymbol{s}_{\mathcal{S}_a}) = \sum_{\boldsymbol{s}_{\mathcal{C}_j \setminus \mathcal{S}_a} \in \mathcal{S}_{\mathcal{C}_j \setminus \mathcal{S}_a}} P(\boldsymbol{X}_{\mathcal{C}_j \setminus \mathcal{S}_a} = \boldsymbol{x}_{\mathcal{C}_j \setminus \mathcal{S}_a} | \boldsymbol{X}_{\mathcal{S}_a} = \boldsymbol{x}_{\mathcal{S}_a}) \prod_l \tilde{\beta}_{a_{j_l}}(\boldsymbol{s}_{\mathcal{S}_{a_{j_l}}}) \tag{2.20}$$

Notons que l'adaptation de la formule de Lucke, 1996 [87] conduit à l'équation ci-dessous. Le fait que le membre de gauche ne dépende que de la valeur de $\boldsymbol{s}_{\mathcal{S}_a}$ et que le membre de droite dépende en général de la valeur des variables cachées qui sont dans $\mathcal{C}_i$ mais pas dans $\mathcal{H}_a$ (ensemble contenant strictement $\mathcal{S}_a$), suggère une inexactitude dans cette équation.

$$\tilde{\beta}_a(\boldsymbol{s}_{\mathcal{S}_a}) = \sum_{\boldsymbol{s}_{\mathcal{H}_a} \in \mathcal{S}_{\mathcal{H}_a}} P(\boldsymbol{X}_{\mathcal{C}_j \setminus \mathcal{S}_a} = \boldsymbol{x}_{\mathcal{C}_j \setminus \mathcal{S}_a} | \boldsymbol{X}_{\mathcal{S}_a} = \boldsymbol{x}_{\mathcal{S}_a}) \prod_l \tilde{\beta}_{a_{j_l}}(\boldsymbol{s}_{\mathcal{S}_{a_{j_l}}})$$

Remarque 2.8 *On pourrait en fait appliquer un raisonnement analogue en utilisant un séparateur de sommets T_a plus grand que $\mathcal{S}_a$. D'après Lucke, 1996 [87] étant donné que l'on désire calculer $\tilde{\beta}_a(\boldsymbol{s}_{T_a})$ pour toute valeur de $\boldsymbol{s}_{T_a}$ dans $\mathcal{S}_{T_a}$, le nombre de valeurs à calculer, et donc le nombre de calculs, est d'autant plus réduit que $\mathcal{S}_{T_a}$ est de faible cardinal, c'est-à-dire que T_a comporte peu de sommets. C'est la raison pour laquelle on utilise les séparateurs de cliques $\mathcal{S}_a$, qui sont les séparateurs de sommets minimaux de $\mathcal{G}$. Autrement dit, l'intérêt de l'arbre de jonction (et des algorithmes basés sur son utilisation) est de permettre un parcours des séparateurs de sommets minimaux.*

Les seules difficultés d'implémentation se résument alors au calcul des probabilités $P(\boldsymbol{X}_{\mathcal{C}_j \setminus \mathcal{S}_a} = \boldsymbol{x}_{\mathcal{C}_j \setminus \mathcal{S}_a} | \boldsymbol{X}_{\mathcal{S}_a} = \boldsymbol{x}_{\mathcal{S}_a})$ et à l'initialisation des quantités $\tilde{\beta}_a$ lorsque a est un arc incident à une clique feuille de l'arbre de jonction. Nous apportons une solution basée sur les trois propositions suivantes (propositions 7 à 9). La première d'entre elles énonce qu'un sommet de $\mathcal{S}_a = \mathcal{C}_i \cap \mathcal{C}_j$ a ses parents soit tous dans $\mathcal{S}_a$, soit tous "avant $\mathcal{C}_j$" (par rapport au sens de parcours de la phase arrière), soit tous "après $\mathcal{C}_j$". La seconde proposition énonce que l'ensemble des sommets situés avant $\mathcal{C}_j$ contient tous les parents de tout sommet situé avant $\mathcal{C}_j$. La troisième proposition énonce que l'ensemble des sommets situés après $\mathcal{C}_j$ contient tous les parents de tout sommet situé après $\mathcal{C}_j$. Ces propositions prouvent que le calcul des probabilités $P(\boldsymbol{X}_{\mathcal{C}_j \setminus \mathcal{S}_a} = \boldsymbol{x}_{\mathcal{C}_j \setminus \mathcal{S}_a} | \boldsymbol{X}_{\mathcal{S}_a} = \boldsymbol{x}_{\mathcal{S}_a})$ ne requiert pas le parcours de tout le graphe $\mathcal{G}$. Seules les cliques déjà visitées par la phase arrière interviennent, ce qui rend possible le calcul de ces probabilités par une méthode inductive également.

Proposition 7 *Soit $a = (\mathcal{C}_i, \mathcal{C}_j)$ un arc de l'arbre des cliques et X un sommet de $\mathcal{S}_a = \mathcal{C}_i \cap \mathcal{C}_j$. Alors pa(X) est contenu soit dans $\mathcal{C}_i \cap \mathcal{C}_j$, soit dans $\mathcal{U}(\mathcal{T}_a^c)$, soit dans $\mathcal{U}(\mathcal{T}_a)$.*

Synopsis de la preuve.
Cette proposition est une conséquence du lemme 3. Elle est due essentiellement à la propriété d'intersection de l'arbre de jonction.
Démonstration

Nous distinguons les cas ci-dessous quant à la localisation d'un éventuel parent d'un sommet X de $\mathcal{S}_a$.

1. Si X admet un parent dans $\mathcal{C}_k \setminus (\mathcal{C}_i \cap \mathcal{C}_j)$, alors par le lemme 3,
   – si $\mathcal{C}_k \in \mathcal{T}_a$ avec $k \notin \{i, j\}$, alors $\mathcal{U}(\mathcal{T}_a^-)$ contient tous les parents de X (remarquons que $\mathcal{U}(\mathcal{T}_a^-)$ est contenu dans $\mathcal{U}(\mathcal{T}_a)$),
   – si $\mathcal{C}_k \in \mathcal{T}_a^c$ avec $k \notin \{i, j\}$, alors $\mathcal{U}(\mathcal{T}_a^{c-})$ contient pa(X), (remarquons que $\mathcal{U}(\mathcal{T}_a^{c-})$ est contenu dans $\mathcal{U}(\mathcal{T}_a^c)$),
   – si $k = i$, alors $\mathcal{U}(\mathcal{T}_a^c)$ contient pa(X),
   – si $k = j$, alors $\mathcal{U}(\mathcal{T}_a)$ contient pa(X).
2. Si X n'admet aucun parent dans les $\mathcal{C}_k \setminus (\mathcal{C}_i \cap \mathcal{C}_j)$, alors pa$(X) \subset (\mathcal{C}_i \cap \mathcal{C}_j)$.

En définitive, pa(X) est contenu dans $\mathcal{U}(\mathcal{T}_a^c)$, dans $\mathcal{U}(\mathcal{T}_a)$ ou dans $(\mathcal{C}_i \cap \mathcal{C}_j)$.

Remarque 2.9 *En réalité, $\mathcal{S}_a$ est à la fois inclus dans $\mathcal{U}(\mathcal{T}_a)$ et $\mathcal{U}(\mathcal{T}_a^c)$. On peut donc être moins précis que l'énoncé ci-dessus et dire que pa(X) est contenu soit dans $\mathcal{U}(\mathcal{T}_a^c)$, soit dans $\mathcal{U}(\mathcal{T}_a)$. D'autre part, cette proposition est illustrée par la figure 2.11 :*
– *$pa(\{S_2\}) = \{S_1\}$. Comme $S_1 \in \mathcal{U}(\mathcal{C}_0)$ et que $\mathcal{C}_0$ est un sommet de l'arbre $\mathcal{T}_a^c$, $pa(\{S_2\}) \subset \mathcal{U}(\mathcal{T}_a^c)$,*
– *$pa(\{S_3\}) = \{S_3\}$. Comme $S_2 \in \mathcal{S}_a$, $pa(\{S_3\}) \subset \mathcal{S}_a$,*
– *$pa(\{S_5\}) = \{S_2, S_3, Y_4\}$. Or $\{S_2, S_3, Y_4\} \subset \mathcal{U}(\mathcal{C}_j)$ et $\mathcal{C}_j$ est une des cliques de $\mathcal{T}_a$ donc $pa(\{S_5\}) \subset \mathcal{U}(\mathcal{T}_a)$.*

Proposition 8 *Soit $a = (\mathcal{C}_i, \mathcal{C}_j)$ un arc de l'arbre des cliques et X un sommet de $\mathcal{U}(\mathcal{T}_a) \setminus \mathcal{U}(\mathcal{S}_a)$. Alors $\mathcal{U}(\mathcal{T}_a)$ contient $pa(X)$.*

Synopsis de la preuve.
Comme la proposition précédente, cette proposition est due essentiellement à la propriété d'intersection de l'arbre de jonction. Elle est une conséquence du lemme 2.
Démonstration

Par définition de $\mathcal{T}_a$, $\exists k\ X \in \mathcal{C}_k \setminus \mathcal{S}_a$ avec $\mathcal{C}_k \in \mathcal{T}_a$. Soit $X' \in \mathcal{C}_{k'}$ un parent de X.

– soit $\mathcal{C}_{k'} \notin \mathcal{T}_a$. Alors $\mathcal{C}_i$ et $\mathcal{C}_j$ sont situés sur le chemin entre $\mathcal{C}_k$ et $\mathcal{C}_{k'}$ dans l'arbre de jonction. Par le lemme 2,
  – soit $X \in \mathcal{C}_{k'}$. Alors par propriété d'intersection de l'arbre de jonction, $X \in \mathcal{C}_i$ et $X \in \mathcal{C}_j$, ce qui contredit $X \notin \mathcal{U}(\mathcal{S}_a)$.
  – soit $X' \in \mathcal{C}_k$. Alors $X' \in \mathcal{U}(\mathcal{T}_a)$,
  – soit $\{X, X'\}$ est inclus dans une clique située entre $\mathcal{C}_k$ et $\mathcal{C}_{k'}$ dans l'arbre de jonction. Par propriété d'intersection, cette clique est située dans $\mathcal{T}_a$, faute de quoi $X \in \mathcal{C}_i$ et $X \in \mathcal{C}_j$. D'où $X' \in \mathcal{U}(\mathcal{T}_a)$.

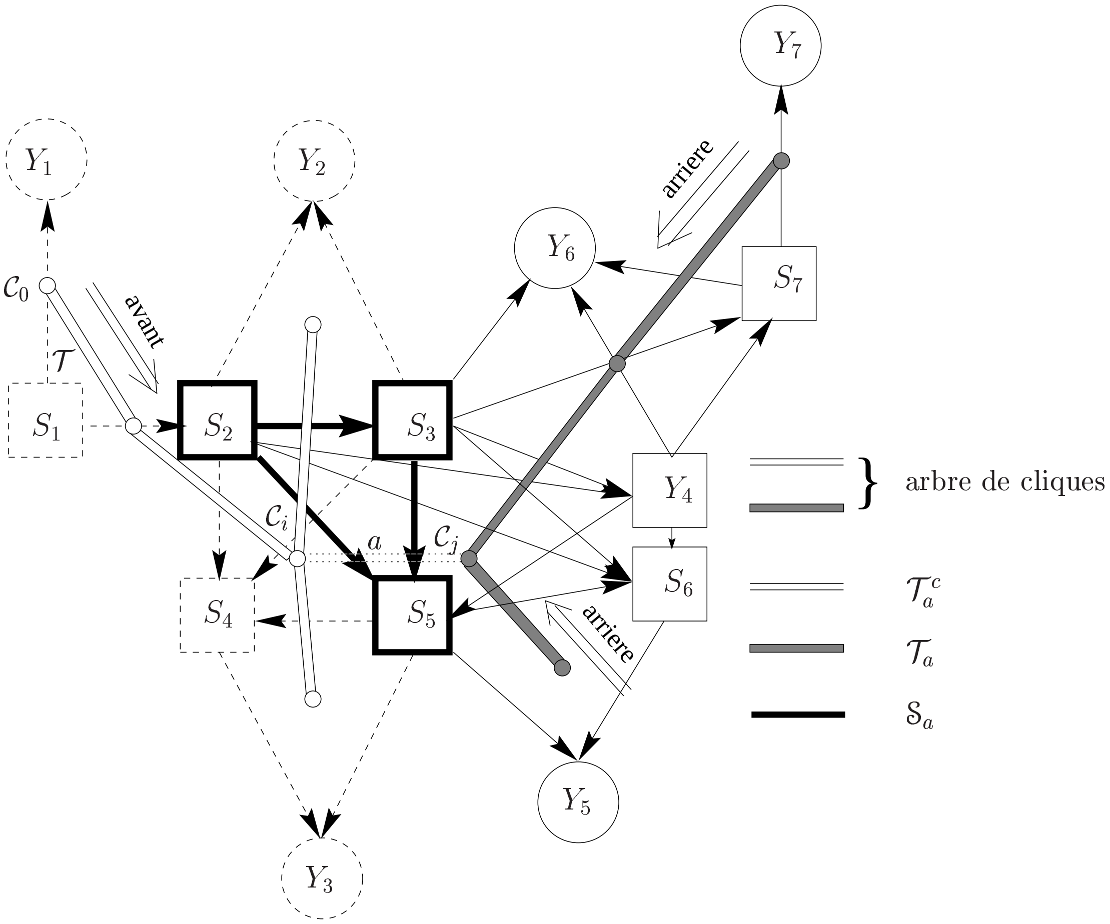


FIG. 2.11 – *Localisation des parents des sommets d'une clique. L'arbre de jonction $\mathcal{T}$ est un arbre dont les sommets, représentés par des petits cercles, sont les cliques $\mathcal{C}_k$. Ses arcs sont représentés par des traits épais blancs ou gris. La suppression, dans l'arbre des cliques, de l'arc a (dessiné en pointillés fins) le sépare en deux composantes, dont une seule ne contient pas $\mathcal{C}_0$: nous notons $\mathcal{T}_a$ l'ensemble des cliques comprises dans cette partie, l'ensemble des autres cliques étant noté $\mathcal{T}_a^c$. Les sommets des cliques de $\mathcal{T}_a$ engendrent un graphe dont l'ensemble des sommets est représenté en traits pleins. Les sommets des cliques de $\mathcal{T}_a^c$ engendrent un graphe dont l'ensemble des sommets est représenté en pointillés. Le séparateur de cliques $\mathcal{S}_a = \mathcal{C}_i \cap \mathcal{C}_j$ est représenté en traits noirs épais. La proposition 7 est illustrée par le fait que tous les parents de S_2 sont dans $\mathcal{T}_a^c$, tous les parents de S_3 dans $\mathcal{S}_a$ et tous les parents de S_5 dans $\mathcal{T}_a$. La proposition 8 est illustrée par le fait que les sommets de $\mathcal{T}_a \setminus \mathcal{S}_a$ (sommets en traits pleins et fins) ont leurs parents dans $\mathcal{T}_a$ (sommets en traits pleins, fins ou épais). La proposition 9 est illustrée par le fait que les sommets de $\mathcal{T}_a^c \setminus \mathcal{S}_a$ (sommets en traits pointillés) ont leurs parents dans $\mathcal{T}_a^c$ (sommets en traits pointillés ou traits noirs épais). Les notations $\mathcal{U}_a$ et $\mathcal{U}_a^c$ sont illustrées comme suit : le sommet S_3 a tous ses parents dans $\mathcal{S}_a$ (donc dans $\mathcal{C}_j$ également) et le sommet S_2 n'a pas tous ses parents dans $\mathcal{T}_a$ puisque $pa(\{S_2\}) = \{S_1\}$ et S_1 n'appartient qu'à des cliques de $\mathcal{T}_a^c$. On en déduit que $\mathcal{U}_a = \{S_3, S_5\}$ et $\mathcal{U}_a^c = \{S_2\}$.*

– soit $\mathcal{C}_{k'} \in \mathcal{T}_a$. Alors $X' \in \mathcal{U}(\mathcal{T}_a)$.

Dans tous les cas, $\mathcal{U}(\mathcal{T}_a)$ contient X'.

Remarque 2.10 *Cette proposition est illustrée par la figure 2.11 :*
- *$pa(\{Y_4\}) = \{S_2, S_3\}$. Or $\{S_2, S_3\} \subset \mathcal{U}(\mathcal{C}_j)$ et $\mathcal{C}_j$ est une des cliques de $\mathcal{T}_a$ donc $pa(\{Y_4\}) \subset \mathcal{U}(\mathcal{T}_a)$,*
- *$pa(\{Y_7\}) = \{S_7\}$. Or S_7 est un sommet du graphe engendré par l'ensemble $\{S_7, Y_7\}$, qui est une clique de $\mathcal{T}_a$ donc $pa(\{Y_7\}) \subset \mathcal{U}(\mathcal{T}_a)$.*

Proposition 9 *Soit $a = (\mathcal{C}_i, \mathcal{C}_j)$ un arc de l'arbre des cliques et X un sommet de $\mathcal{U}(\mathcal{T}_a^c) \setminus \mathcal{U}(\mathcal{S}_a)$. Alors $\mathcal{U}(\mathcal{T}_a^c)$ contient $pa(X)$.*

Démonstration

La démonstration est identique à celle de la proposition 8. Les rôles de $\mathcal{T}_a^c$ et de $\mathcal{T}_a$ sont en effet symétriques dans la démonstration, en particulier dans le lemme 2 utilisé.

Remarque 2.11 *Cette proposition est illustrée par la figure 2.11 :*
- *$pa(\{S_2\}) = \{S_1\}$. Or $\{S_1\} \subset \mathcal{U}(\mathcal{C}_0)$ et $\mathcal{C}_0$ est une des cliques de $\mathcal{T}_a^c$ donc $pa(\{S_2\}) \subset \mathcal{U}(\mathcal{T}_a^c)$,*
- *$pa(\{Y_2\}) = \{S_2, S_3\}$. Or $\{S_2, S_3\} \subset \mathcal{U}(\mathcal{C}_i)$ et $\mathcal{C}_i$ est une des cliques de $\mathcal{T}_a^c$ donc $pa(\{Y_2\}) \subset \mathcal{U}(\mathcal{T}_a^c)$.*

D'après la proposition 7 et la remarque qui suit cette proposition, pour un arc $a = (\mathcal{C}_i, \mathcal{C}_j)$ de l'arbre des cliques, on peut définir l'ensemble $\mathcal{U}_a$ des sommets X de $\mathcal{S}_a$ tels que pa(X) est entièrement contenu dans $\mathcal{U}(\mathcal{T}_a)$. Les sommets de $\mathcal{U}_a^c$, complémentaire de $\mathcal{U}_a$ dans $\mathcal{S}_a$, ont alors leurs parents contenus dans l'ensemble $\mathcal{U}(\mathcal{T}_a^c)$ et admettent au moins un parent dans $\mathcal{U}(\mathcal{T}_a^c \setminus \mathcal{S}_a)$. D'après la proposition 6, $\mathcal{U}_a^c$ contient en général $X_i^{(1)}$ qui admet un parent dans une clique précédent $\mathcal{C}_i$ sur le chemin entre $\mathcal{C}_0$ et $\mathcal{C}_i$ dans l'arbre de jonction. Les seules exceptions sont les cliques contenant S_0 puisque pa$(\{S_0\}) = \emptyset$. Par convention, on dira que $S_0 \in \mathcal{U}_a^c$ pour de telles cliques, ce qui fait que $\mathcal{U}_a^c$ contient toujours au moins un sommet. En revanche, $\mathcal{U}_a$ peut être vide dans certains cas, comme pour les arbres de Markov cachés, voir section 2.4.4.

La figure 2.11 permet d'illustrer ces notations pour l'arc a : S_5 a tous ses parents dans $\mathcal{C}_j$ (graphe engendré par l'ensemble de sommets $\{S_2, S_3, S_5, Y_4, Y_6\}$) qui est une clique de $\mathcal{T}_a$. Le sommet S_3 a tous ses parents dans $\mathcal{S}_a$ (donc dans $\mathcal{C}_j$ également) et le sommet S_2 n'a pas tous ses parents dans $\mathcal{T}_a$ puisque pa$(\{S_2\}) = \{S_1\}$ et S_1 n'appartient qu'à des cliques de $\mathcal{T}_a^c$. On en déduit que $\mathcal{U}_a = \{S_3, S_5\}$ et $\mathcal{U}_a^c = \{S_2\}$.

On déduit des propositions 7 à 9 un moyen de calculer les probabilités $P(\boldsymbol{X}_{\mathcal{C}_j \setminus \mathcal{S}_a} = \boldsymbol{x}_{\mathcal{C}_j \setminus \mathcal{S}_a} | \boldsymbol{X}_{\mathcal{S}_a} = \boldsymbol{x}_{\mathcal{S}_a})$ (que nous notons de manière abrégée $P(\boldsymbol{X}_{\mathcal{C}_j \setminus \mathcal{S}_a} | \boldsymbol{X}_{\mathcal{S}_a})$) par un algorithme qui ne nécessite pas un parcours complet de $\mathcal{G}$. Les calculs se basent sur l'équation

$$P(\boldsymbol{X}_{\mathcal{C}_j}) = P(\boldsymbol{X}_{\mathcal{C}_j \setminus \mathcal{S}_a}) P(\boldsymbol{X}_{\mathcal{S}_a \setminus \mathcal{U}_a^c}) P(\boldsymbol{X}_{\mathcal{U}_a^c})$$

Suivant les différents cas ci-dessous, nous donnons, en fonction des paramètres, une expression des probabilités $P(\boldsymbol{X}_{\mathcal{C}_j \setminus \mathcal{S}_a})$ et $P(\boldsymbol{X}_{\mathcal{S}_a \setminus \mathcal{U}_a^c})$ ou directement une expression de $P(\boldsymbol{X}_{\mathcal{C}_j \setminus \mathcal{S}_a} = \boldsymbol{x}_{\mathcal{C}_j \setminus \mathcal{S}_a})$.

1. Si un sommet de $\mathcal{S}_a$ admet un parent dans $\mathcal{C}_j \setminus \mathcal{S}_a$, alors l'ensemble $\mathrm{An}(\mathcal{C}_j)$ engendre un graphe stable pour la relation parentale, pour lequel la factorisation (1.5) des modèles graphiques à structure orientée acyclique s'applique. Ainsi,

$$P(\boldsymbol{X}_{\mathcal{C}_j}) = \sum_{\boldsymbol{X}_{\mathrm{An}(\mathcal{C}_j)\setminus\mathcal{C}_j}} P(\boldsymbol{X}_{\mathrm{An}(\mathcal{C}_j)}, \boldsymbol{X}_{\mathcal{C}_j}) = \sum_{\boldsymbol{X}_{\mathrm{An}(\mathcal{C}_j)\setminus\mathcal{C}_j}} \prod_{u \in \mathrm{An}(\mathcal{C}_j)\cup\mathcal{C}_j} P(X_u | \boldsymbol{X}_{\mathrm{pa}(u)})$$

où les probabilités $P(X_u|\boldsymbol{X}_{\mathrm{pa}(u)})$ sont données par les paramètres du modèle, notés $\lambda_{\boldsymbol{X}_{\mathrm{pa}(u)},X_u}$ (où $\lambda_{\boldsymbol{x}_{\mathrm{pa}(u)},x_u}$ quand la notation complète $P(X_u = x_u|\boldsymbol{X}_{\mathrm{pa}(u)} = \boldsymbol{x}_{\mathrm{pa}(u)})$ est employée). Nous opérons alors une partition des sommets de $\mathrm{An}(\mathcal{C}_j)$ suivant la localisation de leurs parents : les sommets de $\mathrm{An}(\mathcal{U}_a^c)$ ont tous leurs parents dans $\mathcal{T}_a^c$ (et au moins un de leurs parents dans $\mathcal{T}_a^c \setminus \mathcal{S}_a$) tandis que les sommets de l'ensemble $\mathrm{An}(\mathcal{C}_j)\setminus \mathrm{An}(\mathcal{U}_a^c)$, noté $\tilde{\mathrm{An}}(a)$, ont tous leurs parents dans $\mathcal{T}_a$. Le but est de factoriser la probabilité $P(\boldsymbol{X}_{\mathcal{C}_j} = \boldsymbol{x}_{\mathcal{C}_j})$ en un produit dont un facteur est $P(\boldsymbol{X}_{\mathcal{U}_a^c} = \boldsymbol{x}_{\mathcal{U}_a^c})$ (quantité inconnue) et l'autre facteur une quantité qui se déduit de paramètres du modèle associés à des cliques déjà parcourues (quantité calculable). On saura alors "conditionner par la quantité inconnue" – c'est-à-dire calculer $P(\boldsymbol{X}_{\mathcal{S}_a\setminus\mathcal{U}_a^c} = \boldsymbol{x}_{\mathcal{S}_a\setminus\mathcal{U}_a^c})$. On obtient dans un premier temps

$$\begin{aligned}P(\boldsymbol{X}_{\mathcal{C}_j}) &= \sum_{\boldsymbol{X}_{\mathrm{An}(\mathcal{U}_a^c)\setminus\mathcal{C}_j}} \sum_{\boldsymbol{X}_{\tilde{\mathrm{An}}(a)\setminus\mathcal{C}_j}} \prod_{u\in\mathrm{An}(\mathcal{U}_a^c)} \prod_{u\in\tilde{\mathrm{An}}(a)} \lambda_{\boldsymbol{X}_{\mathrm{pa}(u)},X_u} \\ &= \left[\sum_{\boldsymbol{X}_{\mathrm{An}(\mathcal{U}_a^c)\setminus\mathcal{C}_j}} \prod_{u\in\mathrm{An}(\mathcal{U}_a^c)} \lambda_{\boldsymbol{X}_{\mathrm{pa}(u)},X_u}\right] \sum_{\boldsymbol{X}_{\tilde{\mathrm{An}}(a)\setminus\mathcal{C}_j}} \prod_{u\in\tilde{\mathrm{An}}(a)} \lambda_{\boldsymbol{X}_{\mathrm{pa}(u)},X_u}\end{aligned}$$

car pour $u \in \mathrm{An}(\mathcal{U}_a^c)$, u a tous ses parents dans $\mathrm{An}(\mathcal{U}_a^c)$ ou dans $\mathcal{C}_j$, par définition de $\mathcal{U}_a^c$ et par la propriété 9. Par conséquent, $\lambda_{\boldsymbol{X}_{\mathrm{pa}(u)},X_u}$ ne dépend pas des $\boldsymbol{X}_{\tilde{\mathrm{An}}(a)\setminus\mathcal{C}_j}$. D'autre part, pour la même raison, la factorisation (1.5) des modèles graphiques à structure orientée acyclique s'applique dans le graphe engendré par $\mathrm{An}(\mathcal{U}_a^c)$ donc

$$\begin{aligned}\sum_{\boldsymbol{X}_{\mathrm{An}(\mathcal{U}_a^c)\setminus\mathcal{C}_j}} \prod_{u\in\mathrm{An}(\mathcal{U}_a^c)} \lambda_{\boldsymbol{X}_{\mathrm{pa}(u)},X_u} &= \sum_{\boldsymbol{X}_{\mathrm{An}(\mathcal{U}_a^c)\setminus\mathcal{C}_j}} \prod_{u\in\mathrm{An}(\mathcal{U}_a^c)} P(X_u|\boldsymbol{X}_{\mathrm{pa}(u)}) \\ &= \sum_{\boldsymbol{X}_{\mathrm{An}(\mathcal{U}_a^c)\setminus\mathcal{C}_j}} P(\boldsymbol{X}_{\mathrm{An}(\mathcal{U}_a^c)}) = P(\boldsymbol{X}_{\mathcal{U}_a^c}).\end{aligned}$$

On en déduit

$$P(\boldsymbol{X}_{\mathcal{C}_j}) = P(\boldsymbol{X}_{\mathcal{U}_a^c}) \sum_{\boldsymbol{X}_{\tilde{\mathrm{An}}(a)\setminus\mathcal{C}_j}} \prod_{u\in\tilde{\mathrm{An}}(a)} \lambda_{\boldsymbol{X}_{\mathrm{pa}(u)},X_u}. \tag{2.21}$$

Remarque 2.12 *Les calculs qui précèdent requièrent des sommations sur toutes les valeurs possibles des ancêtres de certaines variables aléatoires. Notons que parmi ces ancêtres ne peut figurer aucune variable aléatoire à valeurs continues, puisque nous avons fait l'hypothèse que celles-ci étaient sans descendants. Dans le cas contraire, il faudrait être en mesure d'intégrer des probabilités sur toutes les valeurs possibles des variables aléatoires à valeurs continues.*

Il s'ensuit, d'autre part,

$$P(\boldsymbol{X}_{\mathcal{S}_a}) = \sum_{\boldsymbol{X}_{\mathcal{C}_j \setminus \mathcal{S}_a}} P(\boldsymbol{X}_{\mathcal{C}_j}) = \sum_{\boldsymbol{X}_{\mathcal{C}_j \setminus \mathcal{S}_a}} P(\boldsymbol{X}_{\mathcal{U}_a^c}) \sum_{\boldsymbol{X}_{\tilde{\text{An}}(a) \setminus \mathcal{C}_j}} \prod_{u \in \tilde{\text{An}}(a)} \lambda_{\boldsymbol{X}_{\text{pa}(u)}, X_u}$$
$$= P(\boldsymbol{X}_{\mathcal{U}_a^c}) \sum_{\boldsymbol{X}_{\mathcal{C}_j \setminus \mathcal{S}_a}} \sum_{\boldsymbol{X}_{\tilde{\text{An}}(a) \setminus \mathcal{C}_j}} \prod_{u \in \tilde{\text{An}}(a)} \lambda_{\boldsymbol{X}_{\text{pa}(u)}, X_u}.$$

Remarque 2.13 *L'équation ci-dessus fait intervenir une sommation sur toutes les valeurs possibles de* $\boldsymbol{X}_{\mathcal{C}_j \setminus \mathcal{S}_a}$ *dans* $\mathcal{X}_{\mathcal{C}_j \setminus \mathcal{S}_a}$. *En général, certaines variables aléatoires observées* Y_u *appartenant à* $\boldsymbol{X}_{\mathcal{C}_j \setminus \mathcal{S}_a}$ *sont à valeurs continues et la notation* $\sum_{y_u}$ *désigne en fait l'intégrale* $\int_{\mathcal{Y}_u} dy_u$. *Nous montrons ci-dessous que cette intégrale se calcule simplement en remplaçant* $P_{\theta_{\boldsymbol{x}_{pa(u)}}}(y_u)$ *par la valeur un.*

Ce résultat provient essentiellement d'une propriété des modèles de la famille $\mathcal{D}$, pour lesquels tous les sommets u du graphe d'indépendance conditionnelle appartenant à $\mathcal{U}_\theta$ (c'est-à-dire les variables aléatoires observées à valeurs continues) sont des puits. Ils ne sont donc parent d'aucun sommet dans le graphe. Cette propriété découle de la définition de la famille $\mathcal{D}$.

Si $\mathcal{C}_j$ ne contient aucune variable aléatoire à valeurs continues, le problème ne se pose pas, mais dans le cas contraire, dans l'expression

$$\prod_{v \in \tilde{\text{An}}(a)} \lambda_{\boldsymbol{x}_{\text{pa}(v)}, x_v},$$

une variable aléatoire Y_u à valeurs continues ne pourra intervenir que dans $\lambda_{\boldsymbol{x}_{\text{pa}(u)}, y_u}$ qui vaut $P_{\theta_{\boldsymbol{x}_{\text{pa}(u)}}}(y_u)$ par définition. Par conséquent,

$$\int_{\mathcal{Y}_u} \prod_{v \in \tilde{\text{An}}(a)} \lambda_{\boldsymbol{x}_{\text{pa}(v)}, x_v} \, dy_u = \prod_{\substack{v \in \tilde{\text{An}}(a) \\ v \neq u}} \lambda_{\boldsymbol{x}_{\text{pa}(v)}, x_v} \int_{\mathcal{Y}_u} P_{\theta_{\boldsymbol{x}_{\text{pa}(u)}}}(y_u) \, dy_u = \prod_{\substack{v \in \tilde{\text{An}}(a) \\ v \neq u}} \lambda_{\boldsymbol{x}_{\text{pa}(v)}, x_v}$$

car $P_{\theta_{\boldsymbol{x}_{\text{pa}(u)}}}$ représente une densité.

On obtient en définitive

$$P(\boldsymbol{X}_{\mathcal{C}_j \setminus \mathcal{S}_a} = \boldsymbol{x}_{\mathcal{C}_j \setminus \mathcal{S}_a} | \boldsymbol{X}_{\mathcal{S}_a} = \boldsymbol{x}_{\mathcal{S}_a}) = \frac{\sum_{\boldsymbol{x}_{\tilde{\text{An}}(a) \setminus \mathcal{C}_j}} \prod_{u \in \tilde{\text{An}}(a)} \lambda_{\boldsymbol{x}_{\text{pa}(u)}, x_u}}{\sum_{\tilde{\boldsymbol{x}}_{\mathcal{C}_j \setminus \mathcal{S}_a}} \sum_{\boldsymbol{x}_{\tilde{\text{An}}(a) \setminus \mathcal{C}_j}} \prod_{u \in \tilde{\text{An}}(a)} \lambda_{\boldsymbol{x}_{\text{pa}(u)}, x_u}}. \tag{2.22}$$

2. Si aucun sommet de $\mathcal{S}_a$ n'admet de parent dans $\mathcal{C}_j \setminus \mathcal{S}_a$, alors le principe est le même que ci-dessus mais les calculs sont simplifiés par le fait que $\mathcal{U}_a = \emptyset$ donc $\mathcal{U}_a^c = \mathcal{S}_a$. On en déduit $P(\boldsymbol{X}_{\mathcal{S}_a}) = P(\boldsymbol{X}_{\mathcal{U}_a^c})$ et

$$P(\boldsymbol{X}_{\mathcal{C}_j \setminus \mathcal{S}_a} = \boldsymbol{x}_{\mathcal{C}_j \setminus \mathcal{S}_a} | \boldsymbol{X}_{\mathcal{S}_a} = \boldsymbol{x}_{\mathcal{S}_a}) = \sum_{\boldsymbol{x}_{\tilde{\text{An}}(a) \setminus \mathcal{C}_j}} \prod_{u \in \tilde{\text{An}}(a)} \lambda_{\boldsymbol{x}_{\text{pa}(u)}, x_u}. \tag{2.23}$$

Initialisation de la phase arrière

Soit $\mathcal{C}_j$ une clique feuille de l'arbre de jonction, donc reliée à une unique clique $\mathcal{C}_i$ par un arc a de l'arbre de jonction. Alors par définition des quantités $\tilde{\beta}_a(\boldsymbol{s}_{\mathbb{S}_a})$, l'algorithme est initialisé en calculant

$$P(\boldsymbol{Y}_{\mathcal{C}_j\setminus\mathbb{S}_a} = \boldsymbol{y}_{\mathcal{C}_j\setminus\mathbb{S}_a}|\boldsymbol{X}_{\mathbb{S}_a} = \boldsymbol{x}_{\mathbb{S}_a}) = \sum_{\boldsymbol{s}_{\mathcal{C}_j\setminus\mathbb{S}_a}} P(\boldsymbol{Y}_{\mathcal{C}_j\setminus\mathbb{S}_a} = \boldsymbol{y}_{\mathcal{C}_j\setminus\mathbb{S}_a}, \boldsymbol{S}_{\mathcal{C}_j\setminus\mathbb{S}_a} = \boldsymbol{s}_{\mathcal{C}_j\setminus\mathbb{S}_a}|\boldsymbol{X}_{\mathbb{S}_a} = \boldsymbol{x}_{\mathbb{S}_a})$$

où $P(\boldsymbol{Y}_{\mathcal{C}_j\setminus\mathbb{S}_a} = \boldsymbol{y}_{\mathcal{C}_j\setminus\mathbb{S}_a}, \boldsymbol{S}_{\mathcal{C}_j\setminus\mathbb{S}_a} = \boldsymbol{s}_{\mathcal{C}_j\setminus\mathbb{S}_a}|\boldsymbol{X}_{\mathbb{S}_a} = \boldsymbol{x}_{\mathbb{S}_a})$ est donné par la formule (2.22) ou (2.23) suivant que $\mathcal{U}_a = \emptyset$ ou non. Les équations sont cependant modifiées par le fait que $(\mathrm{An}(\mathcal{C}_j)\setminus \mathrm{An}(\mathcal{U}_a^c)) \subset \mathcal{C}_j$ par définition de $\mathcal{U}_a^c$. D'où $\tilde{\mathrm{An}}(a) \setminus \mathcal{C}_j = \emptyset$ et

$$\tilde{\beta}_a(\boldsymbol{s}_{\mathbb{S}_a}) = \begin{cases} \displaystyle\sum_{\boldsymbol{s}_{\mathcal{C}_j\setminus\mathbb{S}_a}} \frac{\prod_{u\in\tilde{\mathrm{An}}(a)} \lambda_{\boldsymbol{x}_{\mathrm{pa}(u)},x_u}}{\sum_{\boldsymbol{x}_{\mathcal{C}_j\setminus\mathbb{S}_a}} \prod_{u\in\tilde{\mathrm{An}}(a)} \lambda_{\boldsymbol{x}_{\mathrm{pa}(u)},x_u}} & \text{si } \mathcal{U}_a \neq \emptyset \\ \displaystyle\sum_{\boldsymbol{s}_{\mathcal{C}_j\setminus\mathbb{S}_a}} \prod_{u\in\tilde{\mathrm{An}}(a)} \lambda_{\boldsymbol{x}_{\mathrm{pa}(u)},x_u} & \text{si } \mathcal{U}_a = \emptyset \end{cases} \tag{2.24}$$

Commentaires sur la propagation

La propagation est essentiellement basée sur la formule (2.20) où les probabilités $P(\boldsymbol{X}_{\mathcal{C}_j\setminus\mathbb{S}_a} = \boldsymbol{x}_{\mathcal{C}_j\setminus\mathbb{S}_a}|\boldsymbol{X}_{\mathbb{S}_a} = \boldsymbol{x}_{\mathbb{S}_a})$ sont calculées en utilisant les formules (2.22) ou (2.23).

Remarque 2.14 *Les quantités*

$$\sum_{\boldsymbol{x}_{\tilde{An}(a)\setminus\mathcal{C}_j}} \prod_{u\in\tilde{An}(a)} \lambda_{\boldsymbol{x}_{pa(u)},x_u}$$

peuvent dans de nombreux cas être calculées de manière inductive, comme les quantités $\tilde{\beta}_a(\boldsymbol{s}_{\mathbb{S}_a})$ elles-mêmes, vu que par les propriétés 7 à 9 et par définition de $\mathcal{U}_a^c$, les sommets de $\tilde{An}(a)$ sont dans $\mathcal{T}_a$ (rappelons que $\tilde{An}(a) = An(\mathcal{C}_j)\setminus An(\mathcal{U}_a^c)$). Ainsi, les quantités

$$\sum_{\boldsymbol{x}_{\tilde{An}(a_{j_l})\setminus\mathcal{C}_{j_l}}} \prod_{u\in\tilde{An}(a_{j_l})} \lambda_{\boldsymbol{x}_{pa(u)},x_u}$$

ont été calculées pour tous les arcs $a_{j_l} = (\mathcal{C}_j, \mathcal{C}_{j_l})$ incidents à $\mathcal{C}_j$ autres que a avec $\mathcal{C}_{j_l} \in \mathcal{T}_a$. De plus, dans de nombreux cas, le nombre de cliques nécessaires pour contenir tous les parents des sommets de $\mathcal{C}_j \setminus \mathcal{U}_a^c$ est borné par une constante. C'est en particulier le cas des modèles de Markov cachés dynamiques ou de tout modèle où le nombre de variables aléatoires contenu dans chaque clique, fonction du nombre de variables aléatoires observées n, est borné par une constante L. Le nombre de sommets appartenant à $\tilde{An}(a) \setminus \bigcup_l \tilde{An}(a_{j_l})$ reste alors borné également.

Remarque 2.15 *Le dénominateur de l'expression (2.22) peut être vu comme la probabilité $P(\boldsymbol{X}_{\mathbb{S}_a\setminus\mathcal{U}_a^c} = \boldsymbol{x}_{\mathbb{S}_a\setminus\mathcal{U}_a^c}|\boldsymbol{X}_{\mathcal{U}_a^c} = \boldsymbol{x}_{\mathcal{U}_a^c})$. D'autre part, les probabilités $P(\boldsymbol{X}_{\mathcal{C}_j\setminus\mathbb{S}_a} = \boldsymbol{x}_{\mathcal{C}_j\setminus\mathbb{S}_a}|\boldsymbol{X}_{\mathbb{S}_a} = \boldsymbol{x}_{\mathbb{S}_a})$ doivent être mémorisées si la phase avant est exécutée, voir section 2.4.3.*

Terminaison de l'algorithme

La vraisemblance de λ est obtenue lorsque tous les arcs de l'arbre de jonction ont été parcourus. Nous notons $a_1, \ldots, a_l$ les arcs incidents à $\mathcal{C}_0$ dans l'arbre de jonction. Le raisonnement ayant conduit aux formules de propagation (2.20) est alors adapté pour obtenir :

$$P(\boldsymbol{Y} = \boldsymbol{y}) = \sum_{\boldsymbol{s}_{\mathcal{C}_0} \in \mathcal{S}_{\mathcal{C}_0}} P(\boldsymbol{X}_{\mathcal{C}_0} = \boldsymbol{x}_{\mathcal{C}_0}) \prod_l \tilde{\beta}_{a_l}(\boldsymbol{s}_{\mathbb{S}_{a_l}}) \tag{2.25}$$

où

$$P(\boldsymbol{X}_{\mathcal{C}_0} = \boldsymbol{x}_{\mathcal{C}_0}) = \prod_i P(X_0^{(i)} | X_0^{(1)}, \ldots, X_0^{(i-1)})$$

et où pour tout i, les probabilités $P(X_0^{(i)} = x_0^{(i)} | X_0^{(1)} = x_0^{(1)}, \ldots, X_0^{(i-1)} = x_0^{(i-1)})$ sont donnés par les paramètres du modèle, vu que $\mathrm{pa}(X_0^{(i)}) = \{X_0^{(1)}, \ldots, X_0^{(i-1)}\}$.

2.4.3 Phase avant

La phase avant est un parcours de graphe qui part de $\mathcal{C}_0$ et descend jusqu'aux feuilles de l'arbre des cliques, parcourant ainsi les arcs de l'arbre (ou, de manière équivalente, ses cliques) dans le même sens que leur orientation. Son but est le calcul, pour chaque clique $\mathcal{C}$, des probabilités $P(\boldsymbol{S}_{\mathcal{C}} = \boldsymbol{s}_{\mathcal{C}} | \boldsymbol{Y} = \boldsymbol{y})$ intervenant dans l'algorithme EM de la section 2.3 (voir la propriété 1 et la remarque qui la suit). Elle utilise les probabilités $\tilde{\alpha}_a$ définies comme suit : soit a un arc de l'arbre de jonction $\mathcal{T}$, avec $a = (\mathcal{C}_i, \mathcal{C}_j)$, la clique $\mathcal{C}_i$ étant donc située sur le chemin entre $\mathcal{C}_0$ et $\mathcal{C}_j$ dans l'arbre de jonction. La fonction $\tilde{\alpha}_a$ est alors définie par $\tilde{\alpha}_a(\tilde{\boldsymbol{y}}, \tilde{\boldsymbol{x}}) = P(\boldsymbol{Y}_{\mathcal{K}_a^c} = \tilde{\boldsymbol{y}}, \boldsymbol{X}_{\mathbb{S}_a} = \tilde{\boldsymbol{x}})$, pour $\tilde{\boldsymbol{y}} \in \mathcal{Y}_{\mathcal{K}_a^c}$ et $\tilde{\boldsymbol{x}} \in \mathcal{X}_{\mathbb{S}_a}$. Comme pour les quantités $\tilde{\beta}$, lorsque les observations $\boldsymbol{y}$ sont fixées, on notera $\tilde{\alpha}_a(\boldsymbol{s}_{\mathbb{S}_a})$ pour $P(\boldsymbol{Y}_{\mathcal{K}_a^c} = \boldsymbol{y}_{\mathcal{K}_a^c}, \boldsymbol{Y}_{\mathbb{S}_a} = \boldsymbol{y}_{\mathbb{S}_a}, \boldsymbol{S}_{\mathbb{S}_a} = \boldsymbol{s}_{\mathbb{S}_a})$. Comme nous le verrons dans l'équation de propagation avant (2.27), la phase avant utilise des quantités calculées dans la phase arrière et doit donc être effectuée après celle-ci.

Initialisation

Soit $a = (\mathcal{C}_0, \mathcal{C}_j)$ un arc incident à $\mathcal{C}_0$. On note $(a_{j_l})_l$ les autres arcs incidents à $\mathcal{C}_0$. Par définition, $\tilde{\alpha}_a(\boldsymbol{s}_{\mathbb{S}_a})$ est initialisé en calculant la probabilité $P(\boldsymbol{Y}_{\mathcal{K}_a^c} = \boldsymbol{y}_{\mathcal{K}_a^c}, \boldsymbol{X}_{\mathbb{S}_a} = \boldsymbol{x}_{\mathbb{S}_a})$. Le calcul s'effectue à partir des $P(\boldsymbol{Y}_{\mathcal{K}_{a_{j_l}}} = \boldsymbol{y}_{\mathcal{K}_{a_{j_l}}} | \boldsymbol{X}_{\mathbb{S}_{a_{j_l}}} = \boldsymbol{x}_{\mathbb{S}_{a_{j_l}}})$, et suit le même raisonnement que celui aboutissant aux équations de propagation arrière (2.19). Ainsi, on obtient, en utilisant le séparateur $\mathbb{S}_{a_{j_1}}$:

$$\begin{aligned}
&P(\boldsymbol{Y}_{\mathcal{K}_a^c} = \boldsymbol{y}_{\mathcal{K}_a^c}, \boldsymbol{Y}_{\mathbb{S}_a} = \boldsymbol{y}_{\mathbb{S}_a}, \boldsymbol{S}_{\mathbb{S}_a} = \boldsymbol{s}_{\mathbb{S}_a}) \\
&\quad = \sum_{\boldsymbol{s}_{\mathbb{S}_{a_{j_1}} \setminus \mathbb{S}_a} \in \mathcal{S}_{\mathbb{S}_{a_{j_1}} \setminus \mathbb{S}_a}} \Bigg[\tilde{\beta}_{a_{j_1}}(\boldsymbol{s}_{\mathbb{S}_{a_{j_1}}}) \\
&\quad \times P\Big(\bigcap_{l \neq 1} \{\boldsymbol{Y}_{\bar{\mathcal{K}}_{a_{j_l}}} = \boldsymbol{y}_{\bar{\mathcal{K}}_{a_{j_l}}} \cap \{\boldsymbol{Y}_{\mathcal{C}_0} = \boldsymbol{y}_{\mathcal{C}_0}\} \cap \{\boldsymbol{S}_{\mathbb{S}_a} = \boldsymbol{s}_{\mathbb{S}_a}\} \cap \{\boldsymbol{S}_{\mathbb{S}_{a_{j_1}} \setminus \mathbb{S}_a} = \boldsymbol{s}_{\mathbb{S}_{a_{j_1}} \setminus \mathbb{S}_a}\}\Big) \Bigg].
\end{aligned}$$

En répétant ce raisonnement pour chaque séparateur $\mathcal{S}_{a_{j_l}}$, on obtient la formule d'initialisation

$$\tilde{\alpha}_a(\boldsymbol{s}_{\mathcal{S}_a}) = \sum_{\boldsymbol{s}_{\mathcal{H}_0} \in \mathcal{S}_{\mathcal{H}_0}} P(\boldsymbol{Y}_{\mathcal{C}_0} = \boldsymbol{y}_{\mathcal{C}_0}, \boldsymbol{S}_{\mathcal{H}_0} = \boldsymbol{s}_{\mathcal{H}_0}, \boldsymbol{S}_{\mathcal{S}_a} = \boldsymbol{s}_{\mathcal{S}_a}) \prod_l \tilde{\beta}_{a_{j_l}}(\boldsymbol{s}_{\mathcal{S}_{a_{j_l}}})$$

en notant $\mathcal{H}_0 = \bigcup_l \mathcal{S}_{a_{j_l}} \setminus \mathcal{S}_a$ l'ensemble des variables aléatoires de $\mathcal{C}_0$ étant présentes dans au moins l'une des cliques $\mathcal{C}_{j_l}$ mais pas dans $\mathcal{C}_j$. On peut aussi faire intervenir les variables cachées de $\mathcal{C}_0$ non utilisées dans l'équation précédente pour arriver au résultat

$$\tilde{\alpha}_a(\boldsymbol{s}_{\mathcal{S}_a}) = \sum_{\boldsymbol{s}_{\mathcal{C}_0 \setminus \mathcal{S}_a} \in \mathcal{S}_{\mathcal{C}_0 \setminus \mathcal{S}_a}} P(\boldsymbol{X}_{\mathcal{C}_0} = \boldsymbol{x}_{\mathcal{C}_0}) \prod_l \tilde{\beta}_{a_{j_l}}(\boldsymbol{s}_{\mathcal{S}_{a_{j_l}}}) \tag{2.26}$$

où $P(\boldsymbol{X}_{\mathcal{C}_0} = \boldsymbol{x}_{\mathcal{C}_0})$ a déjà été calculé dans l'étape terminale de la phase arrière.

Propagation

Soit $a = (\mathcal{C}_i, \mathcal{C}_j)$ un arc de l'arbre des cliques. On note a_{i_0} l'arc incident à $\mathcal{C}_i$ qui intervient dans le chemin entre $\mathcal{C}_0$ et $\mathcal{C}_i$ dans l'arbre de jonction et $(a_{i_l})_l$ les autres arcs, différents de a et incidents à $\mathcal{C}_i$. L'équation (2.19) est aisément adaptée pour arriver à l'équation de propagation

$$\tilde{\alpha}_a(\boldsymbol{s}_{\mathcal{S}_a}) = \sum_{\boldsymbol{s}_{\mathcal{C}_i \setminus \mathcal{S}_a} \in \mathcal{S}_{\mathcal{C}_i \setminus \mathcal{S}_a}} P(\boldsymbol{X}_{\mathcal{C}_i \setminus \mathcal{S}_{a_{i_0}}} = \boldsymbol{x}_{\mathcal{C}_i \setminus \mathcal{S}_{a_{i_0}}} | \boldsymbol{X}_{\mathcal{S}_{a_{i_0}}} = \boldsymbol{x}_{\mathcal{S}_{a_{i_0}}}) \tilde{\alpha}_{i_0}(\boldsymbol{s}_{\mathcal{S}_{a_{i_0}}}) \prod_l \tilde{\beta}_{a_{i_l}}(\boldsymbol{s}_{\mathcal{S}_{a_{i_l}}}) \tag{2.27}$$

où $\tilde{\alpha}_{i_0}(\boldsymbol{s}_{\mathcal{S}_{a_{i_0}}})$ a été calculé à l'étape précédente et où les autres quantités ont été calculées lors de la phase arrière.

Remarque 2.16 *Il y a une différence fondamentale entre la propagation dans la phase avant et dans la phase arrière, illustrée par la figure 2.10. Dans la phase arrière, basée sur les quantités $\tilde{\beta}_a(\boldsymbol{s}_{\mathcal{S}_a})$, c'est une arborescence complète de variables de ce type qui intervient à chaque propagation, à savoir toutes les quantités $\tilde{\beta}_{a_{j_l}}(\boldsymbol{s}_{\mathcal{S}_{a_{j_l}}})$ pour les arcs a_{j_l} incidentes à $\mathcal{C}_j$ mais pas à $\mathcal{C}_i$.*

Dans la phase avant, basée sur les quantités $\tilde{\alpha}_a(\boldsymbol{s}_{\mathcal{S}_a})$, une seule quantité $\tilde{\alpha}_{a_{i_0}}(\boldsymbol{s}_{\mathcal{S}_{a_{i_0}}})$ intervient dans la propagation : c'est celle associée à l'arc a_{i_0} compris dans le chemin allant de $\mathcal{C}_0$ à $\mathcal{C}_i$. Les autres arcs incidents à $\mathcal{C}_i$ mais pas à $\mathcal{C}_j$, désignés par $a_{i_1}, \ldots, a_{i_l}$, font intervenir les quantités $\tilde{\beta}_{a_{i_l}}(\boldsymbol{s}_{\mathcal{S}_{a_{i_l}}})$

Le même calcul que celui de la terminaison de la phase arrière donne les probabilités nécessaires à l'implémentation de l'algorithme EM (au coefficient multiplicatif près $P(\boldsymbol{Y} = \boldsymbol{y})$, donné par la phase arrière) :

$$P(\boldsymbol{S}_{\mathcal{C}_i} = \boldsymbol{s}_{\mathcal{C}_i}, \boldsymbol{Y} = \boldsymbol{y}) = P(\boldsymbol{Y}_{\mathcal{C}_i} = \boldsymbol{y}_{\mathcal{C}_i}, \boldsymbol{S}_{\mathcal{C}_i} = \boldsymbol{s}_{\mathcal{C}_i}) \frac{\tilde{\alpha}_{a_{i_0}}(\boldsymbol{s}_{\mathcal{S}_{a_{i_0}}})}{P(\boldsymbol{X}_{\mathcal{S}_{a_{i_0}}} = \boldsymbol{x}_{\mathcal{S}_{a_{i_0}}})} \tilde{\beta}_a(\boldsymbol{s}_{\mathcal{S}_a}) \prod_l \tilde{\beta}_{a_{i_l}}(\boldsymbol{s}_{\mathcal{S}_{a_{i_l}}})$$

où le quotient

$$\frac{P(\boldsymbol{X}_{\mathcal{C}_i} = \boldsymbol{x}_{\mathcal{C}_i})}{P(\boldsymbol{X}_{\mathcal{S}_{a_{i_0}}} = \boldsymbol{x}_{\mathcal{S}_{a_{i_0}}})} = P(\boldsymbol{X}_{\mathcal{C}_i \backslash \mathcal{S}_{a_{i_0}}} = \boldsymbol{x}_{\mathcal{C}_i \backslash \mathcal{S}_{a_{i_0}}} | \boldsymbol{X}_{\mathcal{S}_{a_{i_0}}} = \boldsymbol{x}_{\mathcal{S}_{a_{i_0}}})$$

est donné par la phase arrière. D'où, en définitive,

$$\begin{aligned} &P(\boldsymbol{S}_{\mathcal{C}_i} = \boldsymbol{s}_{\mathcal{C}_i}, \boldsymbol{Y} = \boldsymbol{y}) = \\ &\quad P(\boldsymbol{X}_{\mathcal{C}_i \backslash \mathcal{S}_{a_{i_0}}} = \boldsymbol{x}_{\mathcal{C}_i \backslash \mathcal{S}_{a_{i_0}}} | \boldsymbol{X}_{\mathcal{S}_{a_{i_0}}} = \boldsymbol{x}_{\mathcal{S}_{a_{i_0}}}) \tilde{\alpha}_{a_{i_0}}(\boldsymbol{s}_{\mathcal{S}_{a_{i_0}}}) \tilde{\beta}_a(\boldsymbol{s}_{\mathcal{S}_a}) \prod_l \tilde{\beta}_{a_{i_l}}(\boldsymbol{s}_{\mathcal{S}_{a_{i_l}}}). \end{aligned} \tag{2.28}$$

L'algorithme arrière-avant peut être implémenté directement à l'aide des équations ci-dessus. Le programme prend alors en entrée un graphe correspondant à la structure du modèle et retourne en sortie les quantités nécessaires à l'algorithme EM. Sa complexité est moindre que celle de l'algorithme de Lucke, 1996 [87] (ou de l'algorithme d'arbre de jonction), à savoir d'ordre $\mathcal{O}(\sum_{C \in \mathcal{V}_\mathcal{G}} \text{taille}(C))$, ou au pire égale, dans la mesure où la phase d'initialisation de l'algorithme de Lucke consiste à calculer les $P(\boldsymbol{X}_{\mathcal{C}_j} = \boldsymbol{x}_{\mathcal{C}_j})$. Or dans notre algorithme arrière-avant, le calcul de ces probabilités n'est pas nécessaire : il suffit de calculer les $P(\boldsymbol{X}_{\mathcal{C}_j \backslash \mathcal{S}_a} = \boldsymbol{x}_{\mathcal{C}_j \backslash \mathcal{S}_a} | \boldsymbol{X}_{\mathcal{S}_a} = \boldsymbol{x}_{\mathcal{S}_a})$, ce qui est de complexité moindre. Les calculs de propagation de l'algorithme d'arbre de jonction sont de même complexité que ceux de notre algorithme, d'après la formule (1.4). Il s'en suit que notre algorithme reste de complexité (au pire) $\mathcal{O}(\sum_{C \in \mathcal{V}_\mathcal{G}} \text{taille}(C))$. Dans le cas classique, par exemple de modèles de Markov cachés dynamiques, où chaque variable aléatoire du modèle peut prendre au plus K valeurs, où chaque clique a un nombre de variables aléatoires N_C borné par une constante L et où le nombre de cliques est une fonction polynomiale de degré m de N, la complexité de calcul de notre algorithme arrière-avant est dans $\mathcal{O}(K^L N^m)$.

Variante de la phase avant

D'après les équations (2.27) et (2.28), il est facile de construire une phase avant calculant directement les probabilités $P(\boldsymbol{S}_{\mathcal{C}_j} = \boldsymbol{s}_{\mathcal{C}_j}, \boldsymbol{Y} = \boldsymbol{y})$ de manière inductive. Nous reprenons les notations du paragraphe précédent concernant $\mathcal{C}_i$ et notons $a_{j_1}, \ldots, a_{j_l}$ les arcs distincts de a et incidents à $\mathcal{C}_j$. On obtient alors

$$\begin{aligned} P(\boldsymbol{S}_{\mathcal{C}_j} = \boldsymbol{s}_{\mathcal{C}_j}, \boldsymbol{Y} = \boldsymbol{y}) = &\frac{P(\boldsymbol{X}_{\mathcal{C}_j \backslash \mathcal{S}_a} = \boldsymbol{x}_{\mathcal{C}_j \backslash \mathcal{S}_a} | \boldsymbol{X}_{\mathcal{S}_a} = \boldsymbol{x}_{\mathcal{S}_a})}{\tilde{\beta}_a(\boldsymbol{s}_{\mathcal{S}_a})} \prod_l \tilde{\beta}_{a_{j_l}}(\boldsymbol{s}_{\mathcal{S}_{a_{j_l}}}) \\ &\times \sum_{\boldsymbol{s}_{\mathcal{C}_i \backslash \mathcal{S}_a} \in \mathcal{S}_{\mathcal{C}_i \backslash \mathcal{S}_a}} P(\boldsymbol{S}_{\mathcal{C}_i} = \boldsymbol{s}_{\mathcal{C}_i}, \boldsymbol{Y} = \boldsymbol{y}). \end{aligned}$$

2.4.4 Application aux modèles d'arbres de Markov cachés

Nous montrons à présent comment l'algorithme arrière-avant présenté ci-dessus s'applique aux arbres de Markov cachés. En réalité, ce travail ne sert qu'à l'interprétation des formules arrière-avant et à l'explicitation du rôle des paramètres, puisqu'en pratique cet algorithme peut être programmé de manière générique. Notons qu'aucun nouveau calcul analytique n'est nécessaire ; il s'agit juste d'identifier les cliques du graphe d'indépendance conditionnelle et les arcs de l'arbre de jonction. Nous montrerons que notre algorithme arrière-avant est équivalent, dans ce cas, à l'*algorithme ascendant-descendant* de Crouse, Nowak et Baraniuk, 1998 [32] qui ont proposé le modèle des arbres de Markov cachés. Dans ce qui suit, nous considérons un modèle homogène gaussien, identique à celui défini dans la section 2.3.6.

Soit u un indice du processus observé. Nous introduisons les notations suivantes pour les cliques du graphe d'indépendance conditionnelle :
- la clique engendrée par $\{Y_u, S_u\}$ est notée $\mathcal{C}_u$;
- la clique engendrée par $\{S_{\rho(u)}, S_u\}$ est notée $\mathcal{D}_u$.

L'arbre de jonction $\mathcal{T}$ ainsi que les notations précédentes sont représentées figure 2.12. Les arcs de l'arbre de jonction sont de deux types :
- les arcs $(\mathcal{C}_{\rho(u)}, \mathcal{D}_u)$ notés b_u ;
- les arcs $(\mathcal{D}_u, \mathcal{C}_u)$ notés a_u.

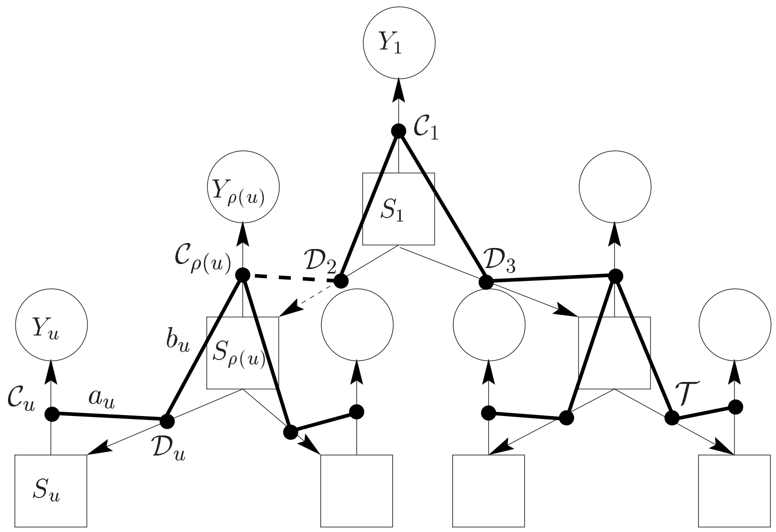


FIG. 2.12 – *Arbre de jonction des arbres de Markov cachés*

Les séparateurs de cliques sont donc les ensembles $\mathcal{S}_{b_u} = \{S_{\rho(u)}\}$ et $\mathcal{S}_{a_u} = \{S_u\}$.

Initialisation de la phase arrière

La première étape consiste à choisir une clique de référence parmi celles stables pour la relation parentale, en l'occurrence $\mathcal{C}_1$ ou l'une des cliques adjacentes à $\mathcal{C}_1$, à savoir l'une des cliques $\mathcal{D}_v$ où v est un enfant de la racine. Par symétrie du graphe, il est plus commode de choisir $\mathcal{C}_1$ comme référence (c'est la clique notée $\mathcal{C}_0$ dans la section 2.4.1).

La phase arrière est initialisée en les cliques feuilles de l'arbre de jonction $\mathcal{T}$, qui sont les cliques $\mathcal{C}_u$ associées aux feuilles u de l'arbre des indices. Soit $\mathcal{C}_u$ l'une d'entre elles : $\mathcal{C}_u$ est incidente à l'arc a_u. La quantité $\tilde{\beta}_{a_u}(j)$ est par définition égale à $P(Y_u = y_u|S_u = j)$. Il est évident que cette quantité est donnée par les lois d'émission du modèle mais nous pouvons vérifier la cohérence de cette constatation avec la formule d'initialisation (2.24). Nous avons $\mathcal{U}_{a_u} = \emptyset$ car $\mathrm{pa}(S_u) = \{S_{\rho(u)}\} \subset \mathcal{T}^c_{a_u}$. Donc $\mathcal{U}^c_{a_u} = \mathcal{S}_{a_u} = \{S_u\}$ et $\tilde{\beta}_{a_u}(j)$ est initialisé par

$$\sum_{\boldsymbol{s}_{\mathcal{C}_u \setminus \mathcal{S}_{a_u}}} \prod_{v \in \tilde{\mathrm{An}}(a_u)} \lambda_{\boldsymbol{x}_{\mathrm{pa}(v)}, x_v}.$$

Or $\mathrm{An}(\mathcal{C}_u) \setminus \mathrm{An}(\mathcal{U}^c_{a_u}) = \{Y_u\}$ donc $\tilde{\mathrm{An}}(a_u)$ se restreint à $\{Y_u\}$. D'autre part, $\mathcal{C}_u \setminus \mathcal{S}_{a_u}$ ne contient aucune variable aléatoire cachée. Donc $\tilde{\beta}_{a_u}(j) = \lambda_{j,y_u}$ qui vaut par définition $f_{\theta_j}(y_u)$.

Propagation dans la phase arrière

On note $\boldsymbol{X}_u$ l'ensemble des sommets du sous-arbre enraciné en X_u. Le graphe entier est alors désigné par $\boldsymbol{X}_1$. On note alors $\bar{\boldsymbol{X}}_u$ l'ensemble des sommets dans $\boldsymbol{X}_1 \setminus \boldsymbol{X}_u$ et $\boldsymbol{X}_{\boldsymbol{c}(u)}$ l'ensemble des enfants de X_u. Ces notations sont illustrées figure 2.13. On

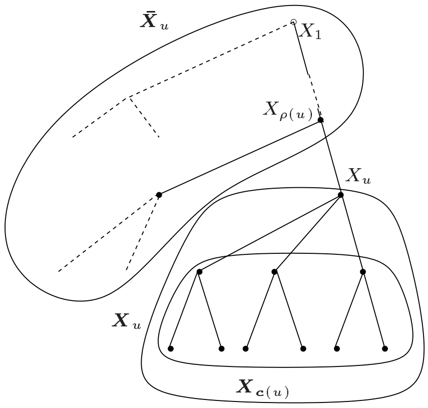


FIG. 2.13 – *Notations utilisées pour désigner des sous-arbres.*

considère l'arc b_u : son extrémité $\mathcal{D}_u$ est dans $\mathcal{T}_{b_u}$. La quantité $\tilde{\beta}_{b_u}(j)$ correspond alors, par définition, à $P(\boldsymbol{Y}_u = \boldsymbol{y}_u | S_{\rho(u)} = j)$. La formule de propagation (2.20) donne, compte tenu de $\mathcal{D}_u \setminus \mathcal{S}_{b_u} = \{S_u\}$:

$$\tilde{\beta}_{b_u}(j) = \sum_k P(S_u = k | S_{\rho(u)} = j) \tilde{\beta}_{a_u}(k).$$

La quantité $P(S_u = k | S_{\rho(u)} = j)$ est directement donnée par le paramètre $p_{j,k}$ du modèle, ce qui est cohérent avec l'équation (2.23) vu que $\mathcal{U}_{b_u} = \emptyset$, $\mathcal{U}^c_{b_u} = \mathcal{S}_{b_u} = \{S_{\rho(u)}\}$ et que, du fait que $\mathrm{An}(\mathcal{D}_u) = \mathrm{An}(\mathcal{U}^c_{b_u}) \cup \{S_u\}$, on a $\tilde{\mathrm{An}}(b_u) = \{S_u\}$ et $\tilde{\mathrm{An}}(b_u) \setminus \mathcal{D}_u = \emptyset$. D'autre part, $\mathrm{pa}(S_u) = \{S_{\rho(u)}\}$ donc

$$\sum_{\boldsymbol{s}_{\tilde{\mathrm{An}}(b_u) \setminus \mathcal{D}_u}} \prod_{v \in \tilde{\mathrm{An}}(b_u)} \lambda_{\boldsymbol{x}_{\mathrm{pa}(v)}, x_v} = \lambda_{s_{\rho(u)}, s_u} = p_{s_{\rho(u)}, s_u}.$$

En définitive,

$$\tilde{\beta}_{b_u}(j) = \sum_k p_{j,k} \tilde{\beta}_{a_u}(k) \tag{2.29}$$

De plus, considérons l'arc a_u dont l'extrémité $\mathcal{C}_u$ est dans $\mathcal{T}_{a_u}$. La quantité $\tilde{\beta}_{a_u}(k)$ correspond alors, par définition, à $P(\boldsymbol{Y}_u = \boldsymbol{y}_u|S_u = k)$. La formule de propagation (2.20) donne, compte tenu de $\mathcal{C}_u \setminus \mathcal{S}_{a_u} = \{Y_u\}$ et vu que $\{b_v\}_{v\in \boldsymbol{c}(u)}$ est l'ensemble des autres arcs incidents à $\mathcal{C}_u$:

$$\tilde{\beta}_{a_u}(k) = \sum_{\emptyset} P(Y_u = y_u|S_u = k) \prod_{v\in \boldsymbol{c}(u)} \tilde{\beta}_{b_v}(k).$$

De même que dans la phase d'initialisation, la quantité $P(Y_u = y_u|S_u = k)$ est directement donnée par $f_{\theta_k}(y_u)$, autrement dit par le paramètre θ_k du modèle. Ceci se déduit de l'équation (2.23) dans la mesure où $\mathcal{U}_{a_u} = \emptyset$, $\mathcal{U}^c_{a_u} = \mathcal{S}_{a_u} = \{S_u\}$ et que, du fait que $\mathrm{An}(\mathcal{C}_u) = \mathrm{An}(\mathcal{U}^c_{a_u}) \cup \{Y_u\}$, on a $\tilde{\mathrm{An}}(a_u) = \{Y_u\}$ et $\tilde{\mathrm{An}}(a_u) \setminus \mathcal{C}_u = \emptyset$. On obtient alors

$$\tilde{\beta}_{a_u}(k) = f_{\theta_k}(y_u) \prod_{v\in \boldsymbol{c}(u)} \tilde{\beta}_{b_v}(k). \tag{2.30}$$

Terminaison de la phase arrière

Le calcul de la vraisemblance est donné par l'équation (2.25) où $P(Y_1 = y_1, s_1 = j) = P(Y_1 = y_1|s_1 = j)P(s_1 = j) = f_{\theta_j}(y_1)\pi_j$, conformément à ce qui était annoncé en section 2.4.2 dans le paragraphe concernant la terminaison, vu que $\mathrm{pa}(Y_1) = \{S_1\}$ et que S_1 est la source du graphe. On obtient alors

$$\mathcal{L}_{\boldsymbol{y}}(\lambda) = \sum_j \pi_j f_{\theta_j}(y_1) \prod_{v\in \boldsymbol{c}(1)} \tilde{\beta}_{b_v}(j).$$

En conclusion, la phase arrière est un parcours de l'arbre $\boldsymbol{Y}$ qui commence par les feuilles et finit par la racine. Lors du parcours, on atteint le sommet Y_u : la loi du sous-arbre $\boldsymbol{Y}_u$, sachant l'état caché S_u, puis sachant son parent, est alors calculée. On parcourt ensuite le sommet parent de Y_u. On obtient ainsi la loi conditionnelle de sous-arbres comportant de plus en plus de sommets jusqu'à arriver à l'arbre entier $\boldsymbol{Y}$.

Notons que les formules d'initialisation, de propagation (2.29) et (2.30) et de terminaison coïncident avec celles de Crouse *et al.*, 1998 [32]. Le fait que les enfants d'un sommet u doivent être traités avant lui conduit à une récursion qualifiée d'*ascendante* par les auteurs[1].

Initialisation de la phase avant

La phase avant est initialisée en les arcs b_v pour chaque enfant v du sommet racine. La quantité $\tilde{\alpha}_{b_v}(j)$ est par définition égale à $P(\bar{\boldsymbol{Y}}_v = \bar{\boldsymbol{y}}_v, S_1 = j)$. Compte tenu de l'égalité $P(Y_1 = y_1, s_1 = j) = f_{\theta_j}(y_1)\pi_j$ et $\mathcal{C}_1 \setminus \mathcal{S}_{b_v} = \{Y_1\}$, il vient d'après l'équation (2.26)

$$\tilde{\alpha}_{b_v}(j) = \pi_j f_{\theta_j}(y_1) \prod_{\substack{v'\in \boldsymbol{c}(1)\\ v'\neq v}} \tilde{\beta}_{b_{v'}}(j).$$

[1] *upward recursion* en anglais.

Propagation dans la phase avant

On considère l'arc a_u dont l'extrémité $\mathcal{D}_u$ est dans $\mathcal{T}^c_{a_u}$. La quantité $\tilde{\alpha}_{a_u}(j)$ correspond alors, par définition, à $P(\bar{\boldsymbol{Y}}_u = \bar{\boldsymbol{y}}_u, S_u = j)$. La formule de propagation (2.27) donne, compte tenu de $\mathcal{D}_u \setminus \mathcal{S}_{a_u} = \{S_{\rho(u)}\}$ et du fait que l'autre arc incident à $\mathcal{D}_u$ est b_u,

$$\tilde{\alpha}_{a_u}(j) = \sum_k P(S_u = j | S_{\rho(u)} = k)\tilde{\alpha}_{b_u}(k) = \sum_k p_{k,j}\tilde{\alpha}_{b_u}(k) \tag{2.31}$$

De même que dans la phase arrière, $P(S_u = j|S_{\rho(u)} = k)$ est donné de manière immédiate par le paramètre $p_{k,j}$. Ceci peut être retrouvé grâce à l'équation (2.23) en déterminant le plus petit graphe ancestral de $\mathcal{D}_u$ et celui de $\{S_u\}$.

En considérant l'arc b_u dont l'extrémité $\mathcal{C}_{\rho(u)}$ est dans $\mathcal{T}^c_{b_u}$, on obtient la quantité $\tilde{\alpha}_{b_u}(j)$ définie par $P(\bar{\boldsymbol{Y}}_u = \bar{\boldsymbol{y}}_u, S_{\rho(u)} = j)$ et calculée par la formule de propagation (2.27), en remarquant que $\mathcal{C}_{\rho(u)} \setminus \mathcal{S}_{b_u} = \{Y_{\rho(u)}\}$, que $a_{\rho(u)}$ est l'arc incident à $\mathcal{C}_{\rho(u)}$ situé sur le chemin de $\mathcal{C}_1$ à $\mathcal{C}_{\rho(u)}$ et que l'ensemble des autres arcs incidents à $\mathcal{C}_{\rho(u)}$ est $\{b_v\}_{v \in \boldsymbol{c}(\rho(u)) \setminus \{u\}}$; d'où

$$\tilde{\alpha}_{b_u}(j) = \sum_{\emptyset} P(Y_{\rho(u)} = y_{\rho(u)} | S_{\rho(u)} = j)\tilde{\alpha}_{a_{\rho(u)}}(j) \prod_{\substack{v \in \boldsymbol{c}(\rho(u)) \\ v \neq u}} \tilde{\beta}_{b_v}(j),$$

soit

$$\tilde{\alpha}_{b_u}(j) = f_{\theta_j}(y_{\rho(u)})\tilde{\alpha}_{a_{\rho(u)}}(j) \prod_{\substack{v \in \boldsymbol{c}(\rho(u)) \\ v \neq u}} \tilde{\beta}_{b_v}(j). \tag{2.32}$$

En conclusion, la phase avant est un parcours de l'arbre $\boldsymbol{Y}$ qui commence par la racine et finit par les feuilles. Lors du parcours, on atteint le sommet Y_u. On considère alors le sous-arbre $\bar{\boldsymbol{Y}}_u$ qui est l'arbre entier privé du sous-arbre $\boldsymbol{Y}_u$, dont on calcule la loi jointe avec l'état caché parent $S_{\rho(u)}$ puis avec l'état caché S_u. On parcourt ensuite les descendants de Y_u. On obtient ainsi la loi jointe de l'arbre entier $\boldsymbol{Y}$, privé de sous-arbres comportant de moins en moins de sommets, jusqu'à arriver à l'arbre $\boldsymbol{Y}$.

Les formules de propagation (2.31) et (2.32) peuvent être combinées pour donner la récursion

$$\tilde{\alpha}_{a_u}(j) = \sum_k p_{k,j} f_{\theta_k}(y_{\rho(u)})\tilde{\alpha}_{a_{\rho(u)}}(k) \prod_{\substack{v \in \boldsymbol{c}(\rho(u)) \\ v \neq u}} \tilde{\beta}_{b_v}(k).$$

Or d'après l'équation (2.30), on a

$$f_{\theta_k}(y_{\rho(u)}) \prod_{\substack{v \in \boldsymbol{c}(\rho(u)) \\ v \neq u}} \tilde{\beta}_{b_v}(k) = \frac{\tilde{\beta}_{a_{\rho(u)}}(k)}{\tilde{\beta}_{b_u}(k)}. \tag{2.33}$$

On en déduit

$$\tilde{\alpha}_{a_u}(j) = \sum_k p_{k,j}\tilde{\alpha}_{a_{\rho(u)}}(k)\frac{\tilde{\beta}_{a_{\rho(u)}}(k)}{\tilde{\beta}_{b_u}(k)},$$

ce qui coïncide avec les formules d'initialisation et de propagation de Crouse *et al.*, 1998 [32]. Le fait que le parent d'un sommet u doive être traité avant lui conduit à une récursion qualifiée de *descendante* par les auteurs[2].

Enfin, il nous reste à déterminer les lois conditionnelles des états cachés des cliques sachant les variables observées, utilisées dans l'étape E de l'algorithme EM (voir section 2.3.6). Nous obtenons directement l'égalité ci-dessous à partir de l'équation (2.28) en considérant la clique $\mathcal{D}_u$, en remarquant que b_u est l'arc incident à $\mathcal{D}_u$ située sur le chemin de $\mathcal{C}_1$ à $\mathcal{D}_u$, que a_u est l'autre arc incident à $\mathcal{D}_u$ et que $\mathcal{D}_u \setminus \mathcal{S}_{b_u} = \{S_u\}$:

$$P(S_{\rho(u)} = k, S_u = j, \boldsymbol{Y} = \boldsymbol{y}) = P(S_u = j | S_{\rho(u)} = k)\tilde{\alpha}_{b_u}(k)\tilde{\beta}_{a_u}(j).$$

On obtient donc

$$P(S_{\rho(u)} = k, S_u = j | \boldsymbol{Y} = \boldsymbol{y}) = \frac{p_{k,j}\tilde{\alpha}_{b_u}(k)\tilde{\beta}_{a_u}(j)}{P(\boldsymbol{Y} = \boldsymbol{y})}. \tag{2.34}$$

En considérant la clique $\mathcal{C}_u$ et en remarquant que a_u est l'arc incident à $\mathcal{C}_u$ situé sur le chemin de $\mathcal{C}_1$ à $\mathcal{C}_u$, l'ensemble des autres arcs incidents à $\mathcal{C}_u$ étant $\{b_v\}_{v \in \boldsymbol{c}(u)}$, et en remarquant que $\mathcal{C}_u \setminus \mathcal{S}_{a_u} = \{Y_u\}$, on obtient

$$P(S_u = j | \boldsymbol{Y} = \boldsymbol{y}) = \frac{P(Y_u = y_u | S_u = j)\tilde{\alpha}_{a_u}(j) \prod_{v \in \boldsymbol{c}(u)} \tilde{\beta}_{b_v}(j)}{P(\boldsymbol{Y} = \boldsymbol{y})}.$$

D'où

$$P(S_u = j | \boldsymbol{Y} = \boldsymbol{y}) = \frac{f_{\theta_j}(y_u)\tilde{\alpha}_{a_u}(j) \prod_{v \in \boldsymbol{c}(u)} \tilde{\beta}_{b_v}(j)}{P(\boldsymbol{Y} = \boldsymbol{y})}. \tag{2.35}$$

On peut écrire différemment ces résultats en remarquant que $S_{\rho(u)}$ sépare les différentes composantes de $\bar{\boldsymbol{Y}}_u$, à savoir $Y_{\rho(u)}$, $\bar{\boldsymbol{Y}}_{\rho(u)}$ et les sous-arbres $\{\boldsymbol{Y}_v\}_{v \in \boldsymbol{c}(\rho(u)) \setminus \{u\}}$ dans le graphe d'indépendance conditionnelle. Par conséquent,

$$\begin{aligned} &P(\bar{\boldsymbol{Y}}_u = \bar{\boldsymbol{y}}_u | S_{\rho(u)} = k) \\ &\quad = P(Y_{\rho(u)} = y_{\rho(u)} | S_{\rho(u)} = k) P(\bar{\boldsymbol{Y}}_{\rho(u)} = \bar{\boldsymbol{y}}_{\rho(u)} | S_{\rho(u)} = k) \prod_{\substack{v \in \boldsymbol{c}(\rho(u)) \\ v \neq u}} P(\boldsymbol{Y}_v = \boldsymbol{y}_v | S_{\rho(u)} = k) \end{aligned}$$

soit encore, en multipliant l'équation ci-dessus par $P(S_{\rho(u)} = k)$,

$$\tilde{\alpha}_{b_u}(k) = f_{\theta_k}(y_{\rho(u)})\tilde{\alpha}_{b_{\rho(u)}}(k) \prod_{\substack{v \in \boldsymbol{c}(\rho(u)) \\ v \neq u}} \tilde{\beta}_{b_v}(k).$$

On déduit de l'équation ci-dessus et des équations (2.33) et (2.34) que

$$P(S_{\rho(u)} = k, S_u = j | \boldsymbol{Y} = \boldsymbol{y}) = \frac{p_{k,j}\tilde{\alpha}_{b_{\rho(u)}}(k)\tilde{\beta}_{a_{\rho(u)}}(k)\tilde{\beta}_{a_u}(j)}{P(\boldsymbol{Y} = \boldsymbol{y})\tilde{\beta}_{a_u}(k)},$$

[2] *downward recursion* en anglais.

ce qui est la formule de Crouse, Nowak et Baraniuk, 1998 [32]. De plus, d'après les équations (2.30) et (2.35), on retrouve leur formule

$$P(S_u = j | \boldsymbol{Y} = \boldsymbol{y}) = \frac{\tilde{\alpha}_{a_u}(j)\tilde{\beta}_{a_u}(j)}{P(\boldsymbol{Y} = \boldsymbol{y})}.$$

Remarque 2.17 *Les chaînes de Markov cachées sont des arbres de Markov cachés particuliers (arbres linéaires). Or les formules avant-arrière de Baum et al., 1970 [7] sont données par les équations suivantes, où $\boldsymbol{Y}_{t'}^{t}$ désigne la séquence $(Y_{t'}, \ldots, Y_t)$. Rappelons que S_1 est la source du graphe d'indépendance conditionnelle (voir figure 1.5).*

$$\tilde{\beta}_t(j) = P(\boldsymbol{Y}_{t+1}^{n} = \boldsymbol{y}_{t+1}^{n} | S_t = j) = \sum_k p_{j,k}\tilde{\beta}_{t+1}(k) f_{\theta_k}(y_{t+1}) \tag{2.36}$$

$$\tilde{\alpha}_t(j) = P(\boldsymbol{Y}_{1}^{t} = \boldsymbol{y}_{1}^{t}, S_t = j) = \sum_i p_{i,j}\tilde{\alpha}_{t-1}(i) f_{\theta_j}(y_t) \tag{2.37}$$

On constate que la phase avant ne fait pas intervenir les quantités $\tilde{\beta}_t(j)$. Ceci s'explique par le fait que chaque clique est incidente à au plus deux arcs. Cela a pour conséquence que la phase avant peut être exécutée avant la phase arrière, ce qui n'est pas le cas en général dans les modèles de la classe $\mathcal{D}$. En outre, on peut constater la coïncidence de l'algorithme de Baum et al. avec le nôtre, mis à part les quantités $\tilde{\beta}_t$ qui interviennent une fois de moins dans chaque équation de propagation, par rapport aux arbres de Markov cachés. Ceci est dû au fait que chaque sommet n'a qu'un seul successeur dans le cas des chaînes.

2.4.5 Algorithme arrière-avant avec des probabilités de lissage

L'algorithme arrière-avant présenté dans les sections 2.4.2 et 2.4.3 a l'inconvénient de se baser sur les probabilités $\tilde{\beta}_a(\boldsymbol{s}_{\mathcal{S}_a}) = P(\boldsymbol{Y}_{\mathcal{K}_a} = \boldsymbol{y}_{\mathcal{K}_a} | \boldsymbol{Y}_{\mathcal{S}_a} = \boldsymbol{y}_{\mathcal{S}_a}, \boldsymbol{S}_{\mathcal{S}_a} = \boldsymbol{s}_{\mathcal{S}_a})$ et $\tilde{\alpha}_a(\boldsymbol{s}_{\mathcal{S}_a}) = P(\boldsymbol{Y}_{\mathcal{K}_a^c} = \boldsymbol{y}_{\mathcal{K}_a^c}, \boldsymbol{Y}_{\mathcal{S}_a} = \boldsymbol{y}_{\mathcal{S}_a}, \boldsymbol{S}_{\mathcal{S}_a} = \boldsymbol{s}_{\mathcal{S}_a})$ qui font intervenir la loi jointe de $\boldsymbol{Y}$. Le but est de décomposer la probabilité $P(\boldsymbol{S}_{\mathcal{C}_i} = \boldsymbol{s}_{\mathcal{C}_i}, \boldsymbol{Y} = \boldsymbol{y})$ en utilisant la factorisation (2.28), afin d'obtenir les quantités intervenant dans l'algorithme EM. Or ces probabilités sont en substance des sommes de produits de probabilités, d'après les équations de propagation (2.20) et (2.27). Ces quantités tendent donc exponentiellement vite vers zéro quand le nombre de données observées n tend vers $+\infty$ (c'est-à-dire en réalité quand l'arc a s'approche de la clique $\mathcal{C}_0$ alors que le nombre total d'arcs est fonction croissante de n dans l'arbre de jonction, dans le cadre de modèles de Markov cachés dynamiques). C'est pourquoi les implémentations numériques de l'algorithme arrière-avant directement basées sur les quantités $\tilde{\beta}_a$ et $\tilde{\alpha}_a$, sont vouées à des problèmes de limitation due à la représentation des réels proches de zéro[3]. Ce problème d'instabilité numérique est abordé, dans le cas des chaînes de Markov cachées, dans Levinson *et al.*, 1983 [83]. Les auteurs proposent en fait d'utiliser des facteurs d'échelle pour renormaliser les quantités "avant", de manière à ce que leur somme vaille un, puis d'utiliser *les mêmes facteurs d'échelle* dans la phase arrière. Devijver, 1985 [37] a en fait montré que cet

[3] *underflow* en anglais.

algorithme *ad-hoc* pouvait être justifié par la méthodologie des modèles à espace d'états (voir également Künsch, 2001 [74]).

Cette section montre qu'on peut en fait décomposer directement les probabilités $P(\boldsymbol{S}_{\mathcal{C}_i} = \boldsymbol{s}_{\mathcal{C}_i}|\boldsymbol{Y} = \boldsymbol{y})$, apparentées aux *probabilités de lissage* $P(S_u = j|\boldsymbol{Y} = \boldsymbol{y})$, en produits de probabilités qui restent bornées quand n augmente (sous l'hypothèse que la taille des cliques est une fonction bornée de n). Ainsi on obtient un algorithme permettant d'utiliser les formules de EM avec un nombre de données observées arbitrairement grand. En outre, cette section illustre l'avantage de disposer d'algorithmes interprétables en termes probabilistes quand il s'agit de les améliorer, ce qui n'est pas le cas de l'algorithme d'arbre de jonction par exemple.

Introduction : cas des chaînes de Markov cachées

Le problème évoqué ci-dessus a été traité dans le cas de chaînes de Markov cachées par Devijver, 1985 [37]. Nous avons vu dans la remarque 2.17 que les chaînes de Markov cachées possèdent des spécificités par rapport aux arbres de Markov cachés et plus généralement aux modèles de la classe $\mathcal{D}$, qui rendent les algorithmes de calcul de probabilité plus simples. En particulier, la phase avant ne fait pas intervenir les résultats de la phase arrière. Nous présentons cependant le principe de l'algorithme de Devijver en guise d'introduction.

Nous reprenons les notations de la remarque 2.17. L'algorithme de Devijver est basé sur la décomposition :

$$\forall t \in \{1,\dots,n\}, \quad P(S_t = j|\boldsymbol{Y}_1^n = \boldsymbol{y}_1^n) = \frac{P(\boldsymbol{Y}_{t+1}^n = \boldsymbol{y}_{t+1}^n|S_t = j)}{P(\boldsymbol{Y}_{t+1}^n = \boldsymbol{y}_{t+1}^n|\boldsymbol{Y}_1^t = \boldsymbol{y}_1^t)} P(S_t = j|\boldsymbol{Y}_1^t = \boldsymbol{y}_1^t)$$

qui se substitue à la décomposition habituelle

$$P(S_t = j, \boldsymbol{Y}_1^n = \boldsymbol{y}_1^n) = P(\boldsymbol{Y}_{t+1}^n = \boldsymbol{y}_{t+1}^n|S_t = j) P(S_t = j, \boldsymbol{Y}_1^t = \boldsymbol{y}_1^t).$$

Devijver définit donc les quantités arrière et avant ainsi :

$$\beta_t(j) = \frac{P(\boldsymbol{Y}_{t+1}^n = \boldsymbol{y}_{t+1}^n|S_t = j)}{P(\boldsymbol{Y}_{t+1}^n = \boldsymbol{y}_{t+1}^n|\boldsymbol{Y}_1^t = \boldsymbol{y}_1^t)}, \qquad \alpha_t(j) = P(S_t = j|\boldsymbol{Y}_1^t = \boldsymbol{y}_1^t),$$

puis établit des formules de propagation pour ces quantités. Les quantités avant sont des *probabilités de filtrage*, au sens des modèles à espace d'états. Du fait de cette définition, la phase avant doit être exécutée en premier car la phase arrière en utilise les résultats. Une adaptation naturelle des travaux de Devijver à nos modèles consisterait à définir, pour un arc $a = (\mathcal{C}_i, \mathcal{C}_j)$ de l'arbre de jonction (la clique $\mathcal{C}_i$ étant donc située sur le chemin entre $\mathcal{C}_0$ et $\mathcal{C}_j$ dans l'arbre),

$$\beta_a(\boldsymbol{x}_{\mathcal{S}_a}) = \frac{P(\boldsymbol{Y}_{\mathcal{K}_a} = \boldsymbol{y}_{\mathcal{K}_a}|\boldsymbol{X}_{\mathcal{S}_a} = \boldsymbol{x}_{\mathcal{S}_a})}{P(\boldsymbol{Y}_{\mathcal{K}_a} = \boldsymbol{y}_{\mathcal{K}_a}|\boldsymbol{Y}_{\mathcal{K}_a^c} = \boldsymbol{y}_{\mathcal{K}_a^c}, \boldsymbol{Y}_{\mathcal{C}_j} = \boldsymbol{y}_{\mathcal{C}_j})}$$

et

$$\alpha_a(\boldsymbol{x}_{\mathcal{S}_a}) = P(\boldsymbol{X}_{\mathcal{S}_a} = \boldsymbol{x}_{\mathcal{S}_a}|\boldsymbol{Y}_{\mathcal{K}_a^c} = \boldsymbol{y}_{\mathcal{K}_a^c}).$$

Or nous savons, d'après la section 2.4.3, qu'en dehors du contexte simplifié des chaînes de Markov cachées, la phase avant utilise les résultats de la phase arrière et qu'un algorithme où la phase arrière reposerait sur des quantités associées à la phase avant (en l'occurrence $P(\boldsymbol{Y}_{\mathcal{K}_a^c} = \boldsymbol{y}_{\mathcal{K}_a^c})$) est peu susceptible d'exister. Par un argument similaire, la méthode des facteurs d'échelle de Levinson *et al.*, 1983 [83] ne peut s'appliquer. D'autre part, même en tenant compte du fait que dans les modèles de Markov cachés généraux, la phase "arrière" est exécutée avant la phase "avant" et en définissant les facteurs d'échelle comme une quantité de normalisation des quantités arrière, il serait impossible d'utiliser les mêmes quantités de normalisation lors de la phase avant.

C'est pourquoi, en nous basant également sur les travaux de Durand, Gonçalvès et Guédon, 2002 [46] concernant les arbres de Markov cachés, nous proposons les définitions suivantes :

$$\beta_a(\boldsymbol{x}_{\mathcal{S}_a}) = \frac{P(\boldsymbol{Y}_{\mathcal{K}_a} = \boldsymbol{y}_{\mathcal{K}_a} | \boldsymbol{X}_{\mathcal{S}_a} = \boldsymbol{x}_{\mathcal{S}_a})}{P(\boldsymbol{Y}_{\mathcal{K}_a} = \boldsymbol{y}_{\mathcal{K}_a})} \tag{2.38}$$

$$\alpha_a(\boldsymbol{x}_{\mathcal{S}_a}) = \frac{P(\boldsymbol{Y}_{\mathcal{K}_a^c} = \boldsymbol{y}_{\mathcal{K}_a^c} | \boldsymbol{X}_{\mathcal{S}_a} = \boldsymbol{x}_{\mathcal{S}_a})}{P(\boldsymbol{Y}_{\mathcal{K}_a^c} = \boldsymbol{y}_{\mathcal{K}_a^c}, \boldsymbol{Y}_{\mathcal{C}_j} = \boldsymbol{y}_{\mathcal{C}_j} | \boldsymbol{Y}_{\mathcal{K}_a} = \boldsymbol{y}_{\mathcal{K}_a})}. \tag{2.39}$$

Il découle de la définition de β_a que

$$\beta_a(\boldsymbol{x}_{\mathcal{S}_a}) = \frac{P(\boldsymbol{X}_{\mathcal{S}_a} = \boldsymbol{x}_{\mathcal{S}_a} | \boldsymbol{Y}_{\mathcal{K}_a} = \boldsymbol{y}_{\mathcal{K}_a})}{P(\boldsymbol{X}_{\mathcal{S}_a} = \boldsymbol{x}_{\mathcal{S}_a})}.$$

La quantité $P(\boldsymbol{X}_{\mathcal{S}_a} = \boldsymbol{x}_{\mathcal{S}_a})$ intervenant au dénominateur n'a pour but que de simplifier les calculs et d'éviter les redondances, comme nous le verrons ci-dessous.

Remarquons que ces définitions, utilisées avec les chaînes de Markov cachées, conduiraient à la décomposition suivante de $P(S_t = j | \boldsymbol{Y}_1^n = \boldsymbol{y}_1^n)$, alternative à celle de Devijver

$$P(S_t = j | \boldsymbol{Y}_1^n = \boldsymbol{y}_1^n) = \frac{P(\boldsymbol{Y}_1^t = \boldsymbol{y}_1^t | S_t = j)}{P(\boldsymbol{Y}_t^1 = \boldsymbol{y}_1^t | \boldsymbol{Y}_{t+1}^n = \boldsymbol{y}_{t+1}^n)} P(S_t = j | \boldsymbol{Y}_{t+1}^n = \boldsymbol{y}_{t+1}^n).$$

Phase préliminaire

Nous proposons un algorithme en trois phases, généralisant l'algorithme de Durand, Gonçalvès et Guédon, 2002 [46] pour les arbres de Markov cachés. Les auteurs proposent une adaptation "à la Devijver" de l'algorithme *ascendant-descendant* de la section 2.4.4. D'après la définition de $\beta_a(\boldsymbol{x}_{\mathcal{S}_a})$, on s'attend à ce que les probabilités $P(\boldsymbol{X}_{\mathcal{S}_a} = \boldsymbol{x}_{\mathcal{S}_a})$ interviennent dans l'algorithme. Nous verrons qu'en réalité, il suffit de connaître ces probabilités pour les feuilles de l'arbre de jonction uniquement. D'autre part, nous verrons que, comme dans la section 2.4.3, il est possible de baser la phase avant sur les quantités $\alpha_a(\boldsymbol{x}_a)$ où directement sur les probabilités $P(\boldsymbol{X}_{\mathcal{C}_j} = \boldsymbol{x}_{\mathcal{C}_j} | \boldsymbol{Y} = \boldsymbol{y})$. Pour l'algorithme basé sur les quantités $\alpha_a(\boldsymbol{x}_a)$, le calcul de la loi des cliques $P(\boldsymbol{X}_{\mathcal{C}_j} = \boldsymbol{x}_{\mathcal{C}_j})$ est nécessaire. Ce calcul est inutile pour l'algorithme basé sur les probabilités $P(\boldsymbol{X}_{\mathcal{C}_j} = \boldsymbol{x}_{\mathcal{C}_j} | \boldsymbol{Y} = \boldsymbol{y})$, ce qui rend ce dernier préférable lorsque le but est uniquement l'implémentation de l'étape E de l'algorithme EM. La phase préliminaire consiste à calculer les probabilités $P(\boldsymbol{X}_{\mathcal{U}_a^c} = \boldsymbol{x}_{\mathcal{U}_a^c})$.

En effet, le calcul des probabilités $P(\boldsymbol{X}_{\mathcal{C}_j} = \boldsymbol{x}_{\mathcal{C}_j})$ s'effectue en se basant sur les remarques 2.14 et 2.15 et sur l'égalité

$$\begin{aligned} &P(\boldsymbol{X}_{\mathcal{C}_j} = \boldsymbol{x}_{\mathcal{C}_j}) \\ &\quad = P(\boldsymbol{X}_{\mathcal{C}_j \setminus \mathbb{S}_a} = \boldsymbol{x}_{\mathcal{C}_j \setminus \mathbb{S}_a}) P(\boldsymbol{X}_{\mathbb{S}_a \setminus \mathcal{U}_a^c} = \boldsymbol{x}_{\mathbb{S}_a \setminus \mathcal{U}_a^c} | \boldsymbol{X}_{\mathcal{U}_a^c} = \boldsymbol{x}_{\mathcal{U}_a^c}) P(\boldsymbol{X}_{\mathcal{U}_a^c} = \boldsymbol{x}_{\mathcal{U}_a^c}). \quad (2.40) \end{aligned}$$

De même, le calcul des $P(\boldsymbol{X}_{\mathbb{S}_a} = \boldsymbol{x}_{\mathbb{S}_a})$ est basé sur l'égalité

$$P(\boldsymbol{X}_{\mathbb{S}_a} = \boldsymbol{x}_{\mathbb{S}_a}) = P(\boldsymbol{X}_{\mathbb{S}_a \setminus \mathcal{U}_a^c} = \boldsymbol{x}_{\mathbb{S}_a \setminus \mathcal{U}_a^c} | \boldsymbol{X}_{\mathcal{U}_a^c} = \boldsymbol{x}_{\mathcal{U}_a^c}) P(\boldsymbol{X}_{\mathcal{U}_a^c} = \boldsymbol{x}_{\mathcal{U}_a^c}). \quad (2.41)$$

D'après l'équation (2.21), seul le calcul de la probabilité $P(\boldsymbol{X}_{\mathcal{U}_a^c} = \boldsymbol{x}_{\mathcal{U}_a^c})$ n'a pas été traité précédemment. Deux méthodes sont possibles pour ce calcul :

1. on peut calculer $P(\boldsymbol{X}_{\mathcal{C}_j} = \boldsymbol{x}_{\mathcal{C}_j}) = P(\boldsymbol{Y}_{\mathcal{C}_j} = \boldsymbol{y}_{\mathcal{C}_j}, \boldsymbol{S}_{\mathcal{C}_j} = \boldsymbol{s}_{\mathcal{C}_j})$ par un algorithme inductif pour toute valeur $\boldsymbol{s}_{\mathcal{C}_j} \in \mathcal{S}_{\mathcal{C}_j}$ (rappelons que $\boldsymbol{y}_{\mathcal{C}_j}$ est fixé ici). En effet, nous avons vu que $P(\boldsymbol{X}_{\mathcal{C}_0} = \boldsymbol{x}_{\mathcal{C}_0})$ se déduit facilement des paramètres du modèle (voir la phase de terminaison de l'algorithme arrière de la section 2.4.2). La récurrence se fait comme suit : soit $a = (\mathcal{C}_i, \mathcal{C}_j)$ un arc de l'arbre de jonction. Supposons que $P(\boldsymbol{X}_{\mathcal{C}_i} = \boldsymbol{x}_{\mathcal{C}_i})$ soit connue. Comme $\mathcal{U}_a^c \subset \mathcal{C}_i$, on en déduit la loi de $\boldsymbol{X}_{\mathcal{U}_a^c}$. La loi de $\boldsymbol{X}_{\mathcal{C}_j}$ est alors obtenue par les égalités (2.40) et (2.41), où le calcul des probabilités $P(\boldsymbol{X}_{\mathcal{C}_j \setminus \mathbb{S}_a} = \boldsymbol{x}_{\mathcal{C}_j \setminus \mathbb{S}_a})$ et $P(\boldsymbol{X}_{\mathbb{S}_a \setminus \mathcal{U}_a^c} = \boldsymbol{x}_{\mathbb{S}_a \setminus \mathcal{U}_a^c} | \boldsymbol{X}_{\mathcal{U}_a^c} = \boldsymbol{x}_{\mathcal{U}_a^c})$ est basé sur les équations (2.22) et (2.23) et sur les remarques 2.15 et 2.14. Cette phase constitue la phase d'initialisation de l'algorithme de Lucke et de celui de l'arbre de jonction : sa complexité est en $\mathcal{O}(\sum_{C \in \mathcal{V}_\mathcal{G}} \text{taille}(C))$. Cependant, nous verrons dans la section 2.4.6, qui présente une application aux arbres de Markov cachés de notre algorithme utilisant les quantités $\beta_a(\boldsymbol{s}_{\mathbb{S}_a})$, que cette méthode peut conduire à des calculs inutiles. En effet, il n'est pas nécessaire de connaître la loi de $\boldsymbol{X}_{\mathcal{C}_i}$ pour connaître celle de $\boldsymbol{X}_{\mathcal{U}_a^c}$: il suffit de connaître la loi jointe de ces sommets et de leurs parents. On en déduit la méthode alternative suivante :
2. on détermine un sommet $X_{j'}^{(1)}$ de $\mathrm{An}(\mathcal{U}_a^c)$ tel que, si l'on note l'ensemble des ancêtres des sommets de $\mathcal{U}_a^c$ qui ne sont pas ancêtre de $X_{j'}^{(1)}$ par

$$A_{j'} = \mathcal{U}(\mathrm{An}(\mathcal{U}_a^c)) \setminus \mathcal{U}(\mathrm{An}(\{X_{j'}^{(1)}\})),$$

alors tout sommet de $A_{j'}$ a ses parents dans $A_{j'} \cup \{X_{j'}^{(1)}\}$. Un tel sommet existe toujours, puisque la source S_0 vérifie cette propriété. On choisira alors un sommet $X_{j'}^{(1)}$ tel que le graphe engendré par $A_{j'} \cup \{X_{j'}^{(1)}\}$ comporte un nombre de sommets minimal. La loi de $\boldsymbol{X}_{\mathcal{U}_a^c}$ est alors donnée par

$$P(\boldsymbol{X}_{\mathcal{U}_a^c} = \boldsymbol{x}_{\mathcal{U}_a^c}) = \sum_{\boldsymbol{x}_{(A_{j'} \cup \{X_{j'}^{(1)}\}) \setminus \mathcal{U}_a^c}} \prod_{u \in A_{j'}} \lambda_{\boldsymbol{x}_{\mathrm{pa}(u)}, x_u} P(X_{j'}^{(1)} = x_{j'}^{(1)}),$$

d'après la factorisation (1.5). Il suffit alors de parcourir les sommets $X_{j'}^{(1)}$ pour calculer les probabilités $P(X_{j'}^{(1)} = x_{j'}^{(1)})$ de manière inductive. On en déduit $P(\boldsymbol{X}_{\mathcal{U}_a^c} = \boldsymbol{x}_{\mathcal{U}_a^c})$ puis $P(X_j^{(1)} = x_j^{(1)})$, puisque $X_j^{(1)} \in \mathcal{U}_a^c$. Le problème de cette méthode est qu'il est difficile d'en évaluer la complexité dans le cas général. En particulier, la détermination d'un sommet $X_{j'}^{(1)}$ conduisant à un graphe $A_{j'}$ minimal est d'une complexité difficile à évaluer.

Remarque 2.18 *Remarquons que dans les deux méthodes ci-dessus interviennent des formules comportant des sommations sur toutes les valeurs possibles des ancêtres de certaines variables aléatoires. Notons que parmi ces ancêtres ne peut figurer aucune variable aléatoire à valeurs continues, puisque nous avons fait l'hypothèse que celles-ci étaient sans descendants. Dans le cas contraire, il faudrait être en mesure d'intégrer des probabilités sur toutes les valeurs possibles des variables aléatoires à valeurs continues.*

Phase arrière

Nous reprenons les notations de la section 2.4.2. De manière immédiate, d'après la définition (2.38), il vient

$$\beta_a(\boldsymbol{x}_{\mathcal{S}_a}) = \frac{\tilde{\beta}_a(\boldsymbol{x}_{\mathcal{S}_a})}{P(\boldsymbol{Y}_{\mathcal{K}_a} = \boldsymbol{y}_{\mathcal{K}_a})} = \frac{P(\boldsymbol{X}_{\mathcal{S}_a} = \boldsymbol{x}_{\mathcal{S}_a} | \boldsymbol{Y}_{\mathcal{K}_a} = \boldsymbol{y}_{\mathcal{K}_a})}{P(\boldsymbol{X}_{\mathcal{S}_a} = \boldsymbol{x}_{\mathcal{S}_a})}.$$

On déduit donc de la formule de propagation (2.20) pour $\tilde{\beta}_a(\boldsymbol{x}_{\mathcal{S}_a})$

$$\begin{aligned} P(\boldsymbol{X}_{\mathcal{S}_a} = \boldsymbol{x}_{\mathcal{S}_a} | \boldsymbol{Y}_{\mathcal{K}_a} = \boldsymbol{y}_{\mathcal{K}_a}) &= \frac{P(\boldsymbol{X}_{\mathcal{S}_a} = \boldsymbol{x}_{\mathcal{S}_a})}{P(\boldsymbol{Y}_{\mathcal{K}_a} = \boldsymbol{y}_{\mathcal{K}_a})} \tilde{\beta}_a(\boldsymbol{x}_{\mathcal{S}_a}) \\ &= \frac{P(\boldsymbol{X}_{\mathcal{S}_a} = \boldsymbol{x}_{\mathcal{S}_a})}{P(\boldsymbol{Y}_{\mathcal{K}_a} = \boldsymbol{y}_{\mathcal{K}_a})} \sum_{\boldsymbol{s}_{\mathcal{C}_j \setminus \mathcal{S}_a} \in \mathcal{S}_{\mathcal{C}_j \setminus \mathcal{S}_a}} \Bigg\{ P(\boldsymbol{X}_{\mathcal{C}_j \setminus \mathcal{S}_a} = \boldsymbol{x}_{\mathcal{C}_j \setminus \mathcal{S}_a} | \boldsymbol{X}_{\mathcal{S}_a} = \boldsymbol{x}_{\mathcal{S}_a}) \\ &\quad \times \prod_l [P(\boldsymbol{Y}_{\mathcal{K}_{a_{j_l}}} = \boldsymbol{y}_{\mathcal{K}_{a_{j_l}}}) \beta_{a_{j_l}}(\boldsymbol{x}_{\mathcal{S}_{a_{j_l}}})] \Bigg\}, \end{aligned}$$

d'où, comme les $P(\boldsymbol{Y}_{\mathcal{K}_{a_{j_l}}} = \boldsymbol{y}_{\mathcal{K}_{a_{j_l}}})$ ne dépendent pas des $\boldsymbol{s}_{\mathcal{C}_j \setminus \mathcal{S}_a}$,

$$\begin{aligned} P(\boldsymbol{X}_{\mathcal{S}_a} = \boldsymbol{x}_{\mathcal{S}_a} | \boldsymbol{Y}_{\mathcal{K}_a} = \boldsymbol{y}_{\mathcal{K}_a}) &= P(\boldsymbol{X}_{\mathcal{S}_a} = \boldsymbol{x}_{\mathcal{S}_a}) \frac{\prod_l P(\boldsymbol{Y}_{\mathcal{K}_{a_{j_l}}} = \boldsymbol{y}_{\mathcal{K}_{a_{j_l}}})}{P(\boldsymbol{Y}_{\mathcal{K}_a} = \boldsymbol{y}_{\mathcal{K}_a})} \\ &\times \sum_{\boldsymbol{s}_{\mathcal{C}_j \setminus \mathcal{S}_a} \in \mathcal{S}_{\mathcal{C}_j \setminus \mathcal{S}_a}} P(\boldsymbol{X}_{\mathcal{C}_j \setminus \mathcal{S}_a} = \boldsymbol{x}_{\mathcal{C}_j \setminus \mathcal{S}_a} | \boldsymbol{X}_{\mathcal{S}_a} = \boldsymbol{x}_{\mathcal{S}_a}) \prod_l \beta_{a_{j_l}}(\boldsymbol{x}_{\mathcal{S}_{a_{j_l}}}) \end{aligned}$$

où $P(\boldsymbol{X}_{\mathcal{U}_a^c} = \boldsymbol{x}_{\mathcal{U}_a^c})$ est donné par la phase préliminaire. Les probabilités

$$P(\boldsymbol{X}_{\mathcal{C}_j \setminus \mathcal{S}_a} = \boldsymbol{x}_{\mathcal{C}_j \setminus \mathcal{S}_a} | \boldsymbol{X}_{\mathcal{S}_a} = \boldsymbol{x}_{\mathcal{S}_a}) \text{ et } P(\boldsymbol{X}_{\mathcal{S}_a \setminus \mathcal{U}_a^c} = \boldsymbol{x}_{\mathcal{S}_a \setminus \mathcal{U}_a^c} | \boldsymbol{X}_{\mathcal{U}_a^c} = \boldsymbol{x}_{\mathcal{U}_a^c})$$

sont déterminées de la même manière que dans la section 2.4.2, voir remarques 2.14 et 2.15. Ceci permet de calculer

$$P(\boldsymbol{X}_{\mathcal{S}_a} = \boldsymbol{x}_{\mathcal{S}_a}) = P(\boldsymbol{X}_{\mathcal{S}_a \setminus \mathcal{U}_a^c} = \boldsymbol{x}_{\mathcal{S}_a \setminus \mathcal{U}_a^c} | \boldsymbol{X}_{\mathcal{U}_a^c} = \boldsymbol{x}_{\mathcal{U}_a^c}) P(\boldsymbol{X}_{\mathcal{U}_a^c} = \boldsymbol{x}_{\mathcal{U}_a^c}).$$

En pratique, on calculera ensuite les quantités intermédiaires

$$\gamma_a(\boldsymbol{x}_{\mathbb{S}_a}) = \sum_{\boldsymbol{s}_{\mathcal{C}_j \setminus \mathbb{S}_a} \in \mathcal{S}_{\mathcal{C}_j \setminus \mathbb{S}_a}} P(\boldsymbol{X}_{\mathcal{C}_j \setminus \mathbb{S}_a} = \boldsymbol{x}_{\mathcal{C}_j \setminus \mathbb{S}_a} | \boldsymbol{X}_{\mathbb{S}_a} = \boldsymbol{x}_{\mathbb{S}_a}) \prod_l \beta_{a_{j_l}}(\boldsymbol{x}_{\mathbb{S}_{a_{j_l}}}) \tag{2.42}$$

$$= P(\boldsymbol{X}_{\mathbb{S}_a} = \boldsymbol{x}_{\mathbb{S}_a} | \boldsymbol{Y}_{\mathcal{K}_a} = \boldsymbol{y}_{\mathcal{K}_a}) \frac{P(\boldsymbol{Y}_{\mathcal{K}_a} = \boldsymbol{y}_{\mathcal{K}_a})}{P(\boldsymbol{X}_{\mathbb{S}_a} = \boldsymbol{x}_{\mathbb{S}_a}) \prod_l P(\boldsymbol{Y}_{\mathcal{K}_{a_{j_l}}} = \boldsymbol{y}_{\mathcal{K}_{a_{j_l}}})}$$

puis, en remarquant que

$$\sum_{\boldsymbol{x}_{\mathbb{S}_a} \in \mathcal{X}_{\mathbb{S}_a}} P(\boldsymbol{X}_{\mathbb{S}_a} = \boldsymbol{x}_{\mathbb{S}_a} | \boldsymbol{Y}_{\mathcal{K}_a} = \boldsymbol{y}_{\mathcal{K}_a}) = 1$$

et en notant

$$\mathcal{N}_a = \sum_{\tilde{\boldsymbol{x}}_{\mathbb{S}_a}} P(\boldsymbol{X}_{\mathbb{S}_a} = \tilde{\boldsymbol{x}}_{\mathbb{S}_a}) \gamma_a(\tilde{\boldsymbol{x}}_{\mathbb{S}_a}),$$

on obtient

$$\mathcal{N}_a = \sum_{\tilde{\boldsymbol{x}}_{\mathbb{S}_a}} P(\boldsymbol{X}_{\mathbb{S}_a} = \tilde{\boldsymbol{x}}_{\mathbb{S}_a}) \gamma_a(\tilde{\boldsymbol{x}}_{\mathbb{S}_a}) = \frac{P(\boldsymbol{Y}_{\mathcal{K}_a} = \boldsymbol{y}_{\mathcal{K}_a})}{\prod_l P(\boldsymbol{Y}_{\mathcal{K}_{a_{j_l}}} = \boldsymbol{y}_{\mathcal{K}_{a_{j_l}}})}, \tag{2.43}$$

d'où

$$P(\boldsymbol{X}_{\mathbb{S}_a} = \boldsymbol{x}_{\mathbb{S}_a} | \boldsymbol{Y}_{\mathcal{K}_a} = \boldsymbol{y}_{\mathcal{K}_a}) = \frac{P(\boldsymbol{X}_{\mathbb{S}_a} = \boldsymbol{x}_{\mathbb{S}_a}) \gamma_a(\boldsymbol{x}_{\mathbb{S}_a})}{\mathcal{N}_a}$$

et

$$\beta_a(\boldsymbol{x}_{\mathbb{S}_a}) = \frac{\gamma_a(\boldsymbol{x}_{\mathbb{S}_a})}{\mathcal{N}_a}. \tag{2.44}$$

Initialisation de la phase arrière

La phase arrière est initialisée par le calcul des probabilités

$$P(\boldsymbol{X}_{\mathbb{S}_a} = \boldsymbol{x}_{\mathbb{S}_a}) = P(\boldsymbol{X}_{\mathbb{S}_a \setminus \mathcal{U}_a^c} = \boldsymbol{x}_{\mathbb{S}_a \setminus \mathcal{U}_a^c} | \boldsymbol{X}_{\mathcal{U}_a^c} = \boldsymbol{x}_{\mathcal{U}_a^c}) P(\boldsymbol{X}_{\mathcal{U}_a^c} = \boldsymbol{x}_{\mathcal{U}_a^c}),$$

où $P(\boldsymbol{X}_{\mathcal{U}_a^c} = \boldsymbol{x}_{\mathcal{U}_a^c})$ est donnée par la phase préliminaire et où $P(\boldsymbol{X}_{\mathbb{S}_a \setminus \mathcal{U}_a^c} = \boldsymbol{x}_{\mathbb{S}_a \setminus \mathcal{U}_a^c} | \boldsymbol{X}_{\mathcal{U}_a^c} = \boldsymbol{x}_{\mathcal{U}_a^c})$ est déterminée de la même manière que dans la section 2.4.2, c'est-à-dire par le dénominateur de l'équation (2.22) ou directement par $P(\boldsymbol{X}_{\mathcal{U}_a^c} = \boldsymbol{x}_{\mathcal{U}_a^c})$, voir également la remarque 2.15 et la section 2.4.2, équation (2.23).

Ensuite, la quantité $\tilde{\beta}_a(\boldsymbol{x}_{\mathbb{S}_a})$ est initialisée par l'équation (2.24). Puis de même que dans les équations (2.42) à (2.44), la quantité $\beta_a(\boldsymbol{x}_{\mathbb{S}_a})$ est initialisée par

$$\gamma_a(\boldsymbol{x}_{\mathbb{S}_a}) = \tilde{\beta}_a(\boldsymbol{x}_{\mathbb{S}_a}),$$
$$\mathcal{N}_a = \sum_{\tilde{\boldsymbol{x}}_{\mathbb{S}_a} \in \mathcal{X}_{\mathbb{S}_a}} P(\boldsymbol{X}_{\mathbb{S}_a} = \tilde{\boldsymbol{x}}_{\mathbb{S}_a}) \gamma_a(\tilde{\boldsymbol{x}}_{\mathbb{S}_a}) = P(\boldsymbol{Y}_{\mathcal{K}_a} = \boldsymbol{y}_{\mathcal{K}_a})$$
$$\text{et} \quad \beta_a(\boldsymbol{x}_{\mathbb{S}_a}) = \tfrac{\gamma_a(\boldsymbol{x}_{\mathbb{S}_a})}{\mathcal{N}_a}.$$

Notons que pour les arcs a incidents aux feuilles de l'arbre de jonction, la quantité $\mathcal{N}_a$ vaut $P(\boldsymbol{Y}_{\mathcal{K}_a} = \boldsymbol{y}_{\mathcal{K}_a})$ et non pas $\dfrac{P(\boldsymbol{Y}_{\mathcal{K}_a} = \boldsymbol{y}_{\mathcal{K}_a})}{\prod_l P(\boldsymbol{Y}_{\mathcal{K}_{a_{j_l}}} = \boldsymbol{y}_{\mathcal{K}_{a_{j_l}}})}$ comme pour les autres arcs.

Commentaires sur la propagation dans la phase arrière

Soit un arc $a = (\mathcal{C}_i, \mathcal{C}_j)$ de l'arbre de jonction. Par définition de la phase arrière, les quantités $\beta_{a_{j_l}}(\boldsymbol{x}_{\mathbb{S}_{a_{j_l}}})$ sont supposées connues pour tous les arcs a_{j_l} incidents à $\mathcal{C}_j$ autres que l'arc a (qui est le seul menant vers $\mathcal{C}_0$) et pour toutes les valeurs possibles de $\boldsymbol{x}_{\mathbb{S}_{a_{j_l}}} \in \mathcal{X}_{\mathbb{S}_{a_{j_l}}}$. Nous désirons calculer $\beta_a(\boldsymbol{x}_{\mathbb{S}_a})$ à partir des $\beta_{a_{j_l}}(\boldsymbol{x}_{\mathbb{S}_{a_{j_l}}})$. Les équations de propagation sont données par les formules (2.42) à (2.44). Les probabilités $P(\boldsymbol{X}_{\mathcal{C}_j} = \boldsymbol{x}_{\mathcal{C}_j})$ sont calculées par l'équation (2.40), puis stockées en mémoire si l'algorithme avant utilisant les quantités $\alpha_a(\boldsymbol{x}_{\mathbb{S}_a})$ doit être exécuté après. Cela n'est pas nécessaire si c'est l'algorithme avant utilisant les probabilités $P(\boldsymbol{S}_{\mathcal{C}_j} = \boldsymbol{s}_{\mathcal{C}_j} | \boldsymbol{Y} = \boldsymbol{y})$ qui est exécuté après. Les quantités de normalisation $\mathcal{N}_a$ doivent être stockées dans tous les cas où une phase avant est exécutée ensuite. Les produits partiels $\prod_{\tilde{a} \in \mathcal{T}_a} \mathcal{N}_{\tilde{a}}$, ou plutôt les sommes partielles $\sum_{\tilde{a} \in \mathcal{T}_a} \ln(\mathcal{N}_{\tilde{a}})$, doivent être également stockées pour le calcul de la vraisemblance dans la mesure où une récurrence montre facilement que $P(\boldsymbol{Y}_{\mathcal{K}_a} = \boldsymbol{y}_{\mathcal{K}_a})$ est égal au produit des quantités $\mathcal{N}_{\tilde{a}}$ pour tous les arcs $\tilde{a}$ parcourus, y compris l'arc a (c'est-à-dire pour tous les arcs de $\mathcal{T}_a$, y compris l'arc a). En résumé,

$$P(\boldsymbol{Y}_{\mathcal{K}_a} = \boldsymbol{y}_{\mathcal{K}_a}) = \mathcal{N}_a \prod_{\tilde{a} \in \mathcal{T}_a} \mathcal{N}_{\tilde{a}}. \tag{2.45}$$

Calcul de la vraisemblance

La vraisemblance de λ est obtenue lorsque le parcours de tous les arcs de l'arbre de jonction est achevé. Nous notons $a_1, \ldots, a_l$ les arcs incidents à $\mathcal{C}_0$ dans l'arbre de jonction. L'équation (2.25) donne

$$\begin{aligned} P(\boldsymbol{Y} = \boldsymbol{y}) &= \sum_{\boldsymbol{s}_{\mathcal{C}_0} \in \mathcal{S}_{\mathcal{C}_0}} P(\boldsymbol{X}_{\mathcal{C}_0} = \boldsymbol{x}_{\mathcal{C}_0}) \prod_l [P(\boldsymbol{Y}_{\mathcal{K}_{a_l}} = \boldsymbol{y}_{\mathcal{K}_{a_l}}) \beta_{a_l}(\boldsymbol{x}_{\mathbb{S}_{a_l}})] \\ &= \left[\prod_l P(\boldsymbol{Y}_{\mathcal{K}_{a_l}} = \boldsymbol{y}_{\mathcal{K}_{a_l}})\right] \sum_{\boldsymbol{s}_{\mathcal{C}_0} \in \mathcal{S}_{\mathcal{C}_0}} [P(\boldsymbol{X}_{\mathcal{C}_0} = \boldsymbol{x}_{\mathcal{C}_0}) \prod_l \beta_{a_l}(\boldsymbol{x}_{\mathbb{S}_{a_l}})] \end{aligned}$$

avec $\prod_l P(\boldsymbol{Y}_{\mathcal{K}_{a_l}} = \boldsymbol{y}_{\mathcal{K}_{a_l}}) = \prod_{a \in \mathcal{T}} \mathcal{N}_a$ d'après l'équation (2.45). Par conséquent,

$$\ln(P(\boldsymbol{Y} = \boldsymbol{y})) = \ln\left(\sum_{\boldsymbol{s}_{\mathcal{C}_0} \in \mathcal{S}_{\mathcal{C}_0}} P(\boldsymbol{X}_{\mathcal{C}_0} = \boldsymbol{x}_{\mathcal{C}_0}) \prod_l \beta_{a_l}(\boldsymbol{x}_{\mathbb{S}_{a_l}})\right) + \sum_{a \in \mathcal{T}} \ln(\mathcal{N}_a)$$

où les sommes partielles $\sum_{a \in \mathcal{T}_{a_l}} \ln(\mathcal{N}_a)$ ont été stockées dans la phase arrière et où $P(\boldsymbol{X}_{\mathcal{C}_0} = \boldsymbol{x}_{\mathcal{C}_0})$ est un produit de paramètres du modèle - voir la phase de terminaison de l'algorithme arrière utilisant les $\tilde{\beta}_a$ section 2.4.2. Si la phase avant doit être exécutée, on gardera en mémoire la quantité $\mathcal{N} = \dfrac{P(\boldsymbol{Y} = \boldsymbol{y})}{\prod_l P(\boldsymbol{Y}_{\mathcal{K}_{a_l}} = \boldsymbol{y}_{\mathcal{K}_{a_l}})}$.

Phase avant

Nous reprenons les notations de la section 2.4.3. D'après la définition (2.39) de $\alpha_a(\boldsymbol{x}_{\mathcal{S}_a})$, il est clair que

$$\alpha_a(\boldsymbol{x}_{\mathcal{S}_a}) = \frac{\tilde{\alpha}_a(\boldsymbol{x}_{\mathcal{S}_a}) P(\boldsymbol{Y}_{\mathcal{K}_a} = \boldsymbol{y}_{\mathcal{K}_a})}{P(\boldsymbol{X}_{\mathcal{S}_a} = \boldsymbol{x}_{\mathcal{S}_a}) P(\boldsymbol{Y} = \boldsymbol{y})}. \tag{2.46}$$

Il découle alors de l'équation de propagation avant (2.27) que

$$\begin{aligned}
&\frac{P(\boldsymbol{X}_{\mathcal{S}_a} = \boldsymbol{x}_{\mathcal{S}_a}) P(\boldsymbol{Y} = \boldsymbol{y})}{P(\boldsymbol{Y}_{\mathcal{K}_a} = \boldsymbol{y}_{\mathcal{K}_a})} \alpha_a(\boldsymbol{x}_{\mathcal{S}_a}) \\
&= \sum_{\boldsymbol{s}_{\mathcal{C}_i \setminus \mathcal{S}_a} \in \mathcal{S}_{\mathcal{C}_i \setminus \mathcal{S}_a}} \Bigg[P(\boldsymbol{X}_{\mathcal{C}_i \setminus \mathcal{S}_{a_{i_0}}} = \boldsymbol{x}_{\mathcal{C}_i \setminus \mathcal{S}_{a_{i_0}}} | \boldsymbol{X}_{\mathcal{S}_{a_{i_0}}} = \boldsymbol{x}_{\mathcal{S}_{a_{i_0}}}) \frac{P(\boldsymbol{X}_{\mathcal{S}_{a_{i_0}}} = \boldsymbol{x}_{\mathcal{S}_{a_{i_0}}}) P(\boldsymbol{Y} = \boldsymbol{y})}{P(\boldsymbol{Y}_{\mathcal{K}_{a_{i_0}}} = \boldsymbol{y}_{\mathcal{K}_{a_{i_0}}})} \\
&\qquad \times \alpha_{a_{i_0}}(\boldsymbol{x}_{a_{i_0}}) \prod_l [P(\boldsymbol{Y}_{\mathcal{K}_{a_{i_l}}} = \boldsymbol{y}_{\mathcal{K}_{a_{i_l}}}) \beta_{a_{i_l}}(\boldsymbol{x}_{\mathcal{S}_{a_{i_l}}})] \Bigg]
\end{aligned}$$

Or

$$\mathcal{N}_{a_{i_0}} = \frac{P(\boldsymbol{Y}_{\mathcal{K}_{a_{i_0}}} = \boldsymbol{y}_{\mathcal{K}_{a_{i_0}}})}{P(\boldsymbol{Y}_{\mathcal{K}_a} = \boldsymbol{y}_{\mathcal{K}_a}) \prod_l P(\boldsymbol{Y}_{\mathcal{K}_{a_{i_l}}} = \boldsymbol{y}_{\mathcal{K}_{a_{i_l}}})},$$

donc

$$\begin{aligned}
&\alpha_a(\boldsymbol{x}_{\mathcal{S}_a}) = \\
&\quad \frac{1}{\mathcal{N}_{a_{i_0}}} \sum_{\boldsymbol{s}_{\mathcal{C}_i \setminus \mathcal{S}_a} \in \mathcal{S}_{\mathcal{C}_i \setminus \mathcal{S}_a}} P(\boldsymbol{X}_{\mathcal{C}_i \setminus \mathcal{S}_a} = \boldsymbol{x}_{\mathcal{C}_i \setminus \mathcal{S}_a} | \boldsymbol{X}_{\mathcal{S}_a} = \boldsymbol{x}_{\mathcal{S}_a}) \alpha_{a_{i_0}}(\boldsymbol{x}_{a_{i_0}}) \prod_l \beta_{a_{i_l}}(\boldsymbol{x}_{\mathcal{S}_{a_{i_l}}}).
\end{aligned} \tag{2.47}$$

Initialisation de la phase avant

Soit $a = (\mathcal{C}_0, \mathcal{C}_j)$ un arc incident à $\mathcal{C}_0$. On note $a_{j_1}, \ldots, a_{j_l}$ les autres arcs incidents à $\mathcal{C}_0$. À partir de l'égalité (2.46) et de l'équation (2.26), on obtient

$$\begin{aligned}
&\frac{P(\boldsymbol{X}_{\mathcal{S}_a} = \boldsymbol{x}_{\mathcal{S}_a}) P(\boldsymbol{Y} = \boldsymbol{y})}{P(\boldsymbol{Y}_{\mathcal{K}_a} = \boldsymbol{y}_{\mathcal{K}_a})} \alpha_a(\boldsymbol{x}_{\mathcal{S}_a}) \\
&\quad = \sum_{\boldsymbol{s}_{\mathcal{C}_0 \setminus \mathcal{S}_a} \in \mathcal{S}_{\mathcal{C}_0 \setminus \mathcal{S}_a}} P(\boldsymbol{X}_{\mathcal{C}_0} = \boldsymbol{x}_{\mathcal{C}_0}) \prod_l [P(\boldsymbol{Y}_{\mathcal{K}_{a_{j_l}}} = \boldsymbol{y}_{\mathcal{K}_{a_{j_l}}}) \beta_{a_{j_l}}(\boldsymbol{x}_{\mathcal{S}_{a_{j_l}}})],
\end{aligned}$$

soit

$$\alpha_a(\boldsymbol{x}_{\mathcal{S}_a}) = \frac{1}{\mathcal{N}} \sum_{\boldsymbol{s}_{\mathcal{C}_0 \setminus \mathcal{S}_a} \in \mathcal{S}_{\mathcal{C}_0 \setminus \mathcal{S}_a}} P(\boldsymbol{X}_{\mathcal{C}_0 \setminus \mathcal{S}_a} = \boldsymbol{X}_{\mathcal{C}_0 \setminus \mathcal{S}_a} | \boldsymbol{X}_{\mathcal{S}_a} = \boldsymbol{x}_{\mathcal{S}_a}) \prod_l \beta_{a_{j_l}}(\boldsymbol{x}_{\mathcal{S}_{a_{j_l}}}) \tag{2.48}$$

où $P(\boldsymbol{X}_{\mathcal{C}_0 \setminus \mathcal{S}_a} = \boldsymbol{X}_{\mathcal{C}_0 \setminus \mathcal{S}_a} | \boldsymbol{X}_{\mathcal{S}_a} = \boldsymbol{x}_{\mathcal{S}_a})$ a déjà été calculé à partir des paramètres lors des dernières étapes de la phase arrière.

Commentaires sur la propagation dans la phase avant

Soit $a = (\mathcal{C}_i, \mathcal{C}_j)$ un arc de l'arbre des cliques. On note a_{i_0} l'arc incident à $\mathcal{C}_i$ qui intervient dans le chemin entre $\mathcal{C}_0$ et $\mathcal{C}_i$ dans l'arbre de jonction et $a_{i_1}, \ldots, a_{i_l}$ les autres arcs, différents de a et incidents à $\mathcal{C}_i$. La propagation consiste simplement en l'application de la formule (2.47) où les probabilités $P(\boldsymbol{X}_{\mathcal{C}_i \setminus \mathcal{S}_a} = \boldsymbol{x}_{\mathcal{C}_i \setminus \mathcal{S}_a} | \boldsymbol{X}_{\mathcal{S}_a} = \boldsymbol{x}_{\mathcal{S}_a})$ ont été calculées puis gardées en mémoire lors de la phase arrière.

Les probabilités nécessaires à l'implémentation de l'algorithme EM sont calculées en adaptant l'équation (2.28), qui donne

$$\begin{aligned}
&P(\boldsymbol{S}_{\mathcal{C}_i} = \boldsymbol{s}_{\mathcal{C}_i}, \boldsymbol{Y} = \boldsymbol{y}) = \\
&\quad P(\boldsymbol{X}_{\mathcal{C}_i \setminus \mathcal{S}_{a_{i_0}}} = \boldsymbol{x}_{\mathcal{C}_i \setminus \mathcal{S}_{a_{i_0}}} | \boldsymbol{X}_{\mathcal{S}_{a_{i_0}}} = \boldsymbol{x}_{\mathcal{S}_{a_{i_0}}}) \frac{P(\boldsymbol{X}_{\mathcal{S}_{a_{i_0}}} = \boldsymbol{x}_{\mathcal{S}_{a_{i_0}}}) P(\boldsymbol{Y} = \boldsymbol{y})}{P(\boldsymbol{Y}_{\mathcal{K}_{a_{i_0}}} = \boldsymbol{y}_{\mathcal{K}_{a_{i_0}}})} \alpha_{a_{i_0}}(\boldsymbol{x}_{\mathcal{S}_{a_{i_0}}}) \\
&\quad \times P(\boldsymbol{Y}_{\mathcal{K}_a} = \boldsymbol{y}_{\mathcal{K}_a}) \beta_a(\boldsymbol{x}_{\mathcal{S}_a}) \prod_l [P(\boldsymbol{Y}_{\mathcal{K}_{a_{i_l}}} = \boldsymbol{y}_{\mathcal{K}_{a_{i_l}}}) \beta_{a_{i_l}}(\boldsymbol{x}_{\mathcal{S}_{a_{i_l}}})].
\end{aligned}$$

Compte tenu de l'équation

$$\mathcal{N}_{a_{i_0}} = \frac{P(\boldsymbol{Y}_{\mathcal{K}_{a_{i_0}}} = \boldsymbol{y}_{\mathcal{K}_{a_{i_0}}})}{P(\boldsymbol{Y}_{\mathcal{K}_a} = \boldsymbol{y}_{\mathcal{K}_a}) \prod_l P(\boldsymbol{Y}_{\mathcal{K}_{a_{i_l}}} = \boldsymbol{y}_{\mathcal{K}_{a_{i_l}}})},$$

on obtient en définitive

$$P(\boldsymbol{S}_{\mathcal{C}_i} = \boldsymbol{s}_{\mathcal{C}_i} | \boldsymbol{Y} = \boldsymbol{y}) = P(\boldsymbol{X}_{\mathcal{C}_i} = \boldsymbol{x}_{\mathcal{C}_i}) \frac{\alpha_{a_{i_0}}(\boldsymbol{x}_{\mathcal{S}_{a_{i_0}}})}{\mathcal{N}_{a_{i_0}}} \beta_a(\boldsymbol{x}_{\mathcal{S}_a}) \prod_l \beta_{a_{i_l}}(\boldsymbol{x}_{\mathcal{S}_{a_{i_l}}}) \tag{2.49}$$

où les probabilités $P(\boldsymbol{X}_{\mathcal{C}_i} = \boldsymbol{x}_{\mathcal{C}_i})$ ont été calculées lors de la phase préliminaire, ou se déduisent de l'équation (2.40).

L'algorithme arrière-avant utilisant les probabilités de lissage est de complexité égale à celle de l'algorithme de Lucke, 1996 [87] (ou celui d'arbre de jonction) dans la mesure où la phase d'initialisation de ce dernier consiste à calculer les $P(\boldsymbol{X}_{\mathcal{C}_j} = \boldsymbol{x}_{\mathcal{C}_j})$, qui sont également calculés dans notre algorithme. Les calculs de propagation de l'algorithme d'arbre de jonction sont de même complexité que ceux de notre algorithme, d'après la formule (1.4). Il s'ensuit que notre algorithme est de complexité (au pire) d'ordre $\mathcal{O}(\sum_{C \in \mathcal{V}_\mathcal{G}} \text{taille}(C))$.

Dans le cas classique, par exemple de modèles de Markov cachés dynamiques (chaque variable aléatoire du modèle pouvant alors prendre au plus K valeurs), où chaque clique a un nombre de variables aléatoires N_C borné par L et où le nombre de cliques est une fonction polynomiale de degré m du nombre total N de variables aléatoires du modèle, la complexité de calcul de notre algorithme arrière-avant utilisant les probabilités de lissage reste dans $\mathcal{O}(K^L N^m)$. Par rapport à l'algorithme arrière-avant utilisant la loi jointe de $\boldsymbol{Y}$ et les quantités $\tilde{\beta}_a$ et $\tilde{\alpha}_a$ (sections 2.4.2 et 2.4.3), l'algorithme ci-dessus requiert une phase préliminaire supplémentaire dont la complexité est encore dans $\mathcal{O}(K^L N^m)$.

La phase arrière de notre algorithme arrière-avant utilisant les probabilités de lissage est basée sur les quantités

$$\beta_a(\boldsymbol{x}_{\mathcal{S}_a}) = \frac{P(\boldsymbol{Y}_{\mathcal{K}_a} = \boldsymbol{y}_{\mathcal{K}_a} | \boldsymbol{X}_{\mathcal{S}_a} = \boldsymbol{x}_{\mathcal{S}_a})}{P(\boldsymbol{Y}_{\mathcal{K}_a} = \boldsymbol{y}_{\mathcal{K}_a})},$$

c'est-à-dire essentiellement sur les probabilités $P(\boldsymbol{X}_{\mathcal{S}_a} = \boldsymbol{x}_{\mathcal{S}_a} | \boldsymbol{Y}_{\mathcal{K}_a} = \boldsymbol{y}_{\mathcal{K}_a})$ dont le maximum sur $\boldsymbol{x}_{\mathcal{S}_a} \in \mathcal{X}_{\mathcal{S}_a}$ est une fonction bornée du nombre de données observées n, quand a est proche de $\mathcal{C}_0$ – du moins sous l'hypothèse que le nombre de variables aléatoires des cliques ne croît pas avec n – puisque

$$\sum_{\tilde{\boldsymbol{x}} \in \mathcal{X}_{\mathcal{S}_a}} P(\boldsymbol{X}_{\mathcal{S}_a} = \tilde{\boldsymbol{x}} | \boldsymbol{Y}_{\mathcal{K}_a} = \boldsymbol{y}_{\mathcal{K}_a}) = 1.$$

Ainsi, les quantités $\beta_a(\boldsymbol{x}_{\mathcal{S}_a})$ ne tendent pas vers 0 lorsque a se rapproche de $\mathcal{C}_0$. De même, les probabilités $P(\boldsymbol{S}_{\mathcal{C}_j} = \boldsymbol{s}_{\mathcal{C}_j} | \boldsymbol{Y} = \boldsymbol{y})$ ont un maximum sur $\boldsymbol{s}_{\mathcal{C}_j} \in \mathcal{S}_{\mathcal{C}_j}$ qui reste borné. On en déduit, d'après l'équation (2.49), que les quantités $\dfrac{\alpha_a(\boldsymbol{x}_a)}{\mathcal{N}_a}$, soit en général les $\alpha_a(\boldsymbol{x}_a)$, restent elles aussi bornées. Ceci explique que notre algorithme ne soit pas sujet au problème des limitations liées à la représentation des réels proches de zéro (*underflow*). Cette propriété est rendue plus évidente par la variante ci-dessous.

Variante de la phase avant

D'après les équations (2.27) et (2.47), il est facile de construire une phase avant calculant directement les probabilités $P(\boldsymbol{S}_{\mathcal{C}_j} | \boldsymbol{Y} = \boldsymbol{y})$ de manière inductive. On obtient alors l'équation d'initialisation

$$P(\boldsymbol{S}_{\mathcal{C}_0} = \boldsymbol{s}_{\mathcal{C}_0} | \boldsymbol{Y} = \boldsymbol{y}) = \frac{1}{\mathcal{N}} P(\boldsymbol{X}_{\mathcal{C}_0} = \boldsymbol{x}_{\mathcal{C}_0}) \prod_l \beta_{a_l}(\boldsymbol{x}_{\mathcal{S}_{a_l}}) \tag{2.50}$$

où $\{a_1, \ldots, a_l\}$ désigne l'ensemble des arcs incidents à $\mathcal{C}_0$ et où la quantité $\prod_l P(\boldsymbol{Y}_{\mathcal{K}_{a_l}} = \boldsymbol{y}_{\mathcal{K}_{a_l}})$ est donnée par $\prod_{a \in \mathcal{T}} \mathcal{N}_a$, comme dans le calcul de la vraisemblance. L'équation de propagation permettant de déterminer $P(\boldsymbol{S}_{\mathcal{C}_j} = \boldsymbol{s}_{\mathcal{C}_j} | \boldsymbol{Y} = \boldsymbol{y})$ à partir de $P(\boldsymbol{S}_{\mathcal{C}_i} = \boldsymbol{s}_{\mathcal{C}_i} | \boldsymbol{Y} = \boldsymbol{y})$, où $(\mathcal{C}_i, \mathcal{C}_j) = a$ est un arc de l'arbre de jonction est

$$\begin{aligned} &P(\boldsymbol{S}_{\mathcal{C}_j} = \boldsymbol{s}_{\mathcal{C}_j} | \boldsymbol{Y} = \boldsymbol{y}) \\ &= \left[\frac{P(\boldsymbol{X}_{\mathcal{C}_j \setminus \mathcal{S}_a} = \boldsymbol{x}_{\mathcal{C}_j \setminus \mathcal{S}_a} | \boldsymbol{X}_{\mathcal{S}_a} = \boldsymbol{x}_{\mathcal{S}_a})}{\mathcal{N}_a \beta_a(\boldsymbol{s}_{\mathcal{S}_a})} \prod_l \beta_{a_{j_l}}(\boldsymbol{s}_{\mathcal{S}_{a_{j_l}}}) \right] \sum_{\boldsymbol{s}_{\mathcal{C}_i \setminus \mathcal{S}_a} \in \mathcal{S}_{\mathcal{C}_i \setminus \mathcal{S}_a}} P(\boldsymbol{S}_{\mathcal{C}_i} = \boldsymbol{s}_{\mathcal{C}_i} | \boldsymbol{Y} = \boldsymbol{y}) \end{aligned} \tag{2.51}$$

Les arcs distincts de a et incidents à $\mathcal{C}_j$ sont désignés par $a_{j_1}, \ldots, a_{j_l}$, de la même manière que dans les paragraphes précédents. L'équation ci-dessus montre qu'un algorithme avant basé directement sur les probabilités $P(\boldsymbol{S}_{\mathcal{C}_j} = \boldsymbol{s}_{\mathcal{C}_j} | \boldsymbol{Y} = \boldsymbol{y})$ rend inutile le calcul des quantités $P(\boldsymbol{X}_{\mathcal{C}_j} = \boldsymbol{x}_{\mathcal{C}_j})$ lors de la phase arrière puisqu'elles n'interviennent pas dans l'équation ci-dessus. Le calcul des probabilités $P(\boldsymbol{X}_{\mathcal{U}_a^c} = \boldsymbol{x}_{\mathcal{U}_a^c})$ est suffisant ; il est effectué dans la phase préliminaire de l'algorithme.

2.4.6 Application aux arbres de Markov cachés des algorithmes arrière-avant avec probabilités de lissage

Dans cette section, nous appliquons l'algorithme arrière-avant utilisant des probabilités de lissage de la section 2.4.5 au modèle d'arbre de Markov caché de la section 2.4.4. Nous rappelons ci-dessous les notations concernant les cliques de la structure de ce modèle et l'arbre de jonction (figure 2.12).

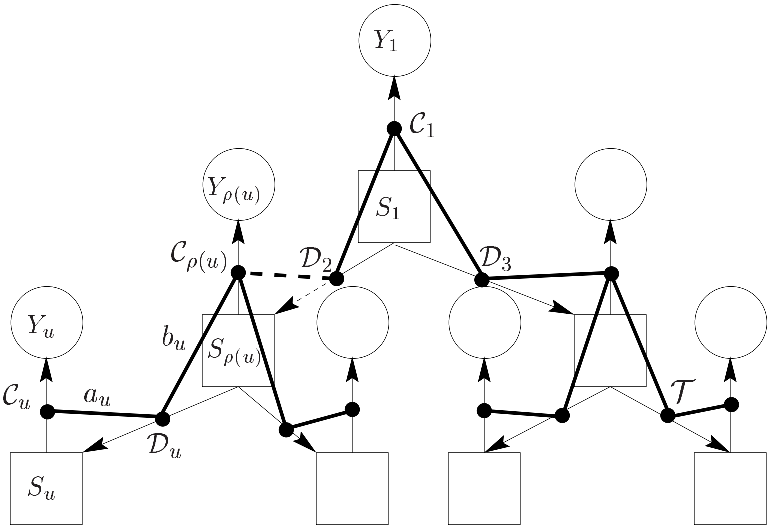


FIG. 2.12 – *Arbre de jonction des arbres de Markov cachés*

Phase préliminaire

Rappelons que la phase préliminaire consiste à calculer, de manière inductive, les probabilités $P(\boldsymbol{X}_{\mathcal{U}_a^c} = \boldsymbol{x}_{\mathcal{U}_a^c})$ pour tous les arcs a de l'arbre de jonction. Or nous avons vu dans la section 2.4.4 que $\mathcal{U}_a^c = \mathcal{S}_a$ pour tout $a \in \mathcal{T}$, c'est-à-dire que $\mathcal{U}_a^c$ est l'un des ensembles $\{S_u\}$. D'après la partie de la section 2.4.5 dédiée à la phase préliminaire, une première méthode de calcul peut être envisagée. Elle consiste, à partir de la loi de S_1 donnée par le paramètre π, à calculer la loi de chaque clique par les équations

$$P(S_u = j, S_{\rho(u)} = k) = p_{k,j} P(S_{\rho(u)} = k)$$

et

$$P(Y_u = y_u, S_u = j) = f_{\theta_j}(y_u) P(S_u = j),$$

puis à déduire la loi de S_u de la loi de $(S_u, S_{\rho(u)})$ par exemple. Cependant, le calcul de la loi du couple (Y_u, S_u) est en fait inutile : seule la loi de S_u nous intéresse. C'est pourquoi il est préférable de recourir à la deuxième méthode, à savoir : déterminer un sommet $S_{u'}$ de $\mathrm{An}(\{S_u\})$ tel que tout sommet de l'ensemble A_u défini par

$$A_u = \mathrm{An}(\{S_u\}) \setminus \mathrm{An}(\{S_{u'}\})$$

ait ses parents dans $A_u \cup \{S_{u'}\}$. On doit choisir $S_{u'}$ de sorte que A_u soit le plus petit possible. On trouve alors $S_{u'} = S_{\rho(u)}$: c'est le parent de S_u. Dans ce cas, $A_u = \{S_u\}$. On obtient alors la récursion

$$\begin{aligned} & P(S_1 = j) = \pi_j \\ \text{et } \forall u, \quad & P(S_u = j) = \sum_k p_{k,j} P(S_{\rho(u)} = k), \end{aligned}$$

qui suffit à calculer la loi marginale de chaque état caché S_u mais dont la complexité est moindre que celle de la première méthode, consistant à calculer la loi de toutes les cliques.

Phase arrière

D'après les sections 2.4.5 et 2.4.4, la phase arrière est initialisée en les cliques feuilles $\mathcal{C}_u$ de l'arbre de jonction, l'indice u étant lui-même une feuille de l'arbre des indices. La clique $\mathcal{C}_u$ est incidente à l'arc a_u, donc la phase arrière est initialisée par le calcul des quantités

$$\begin{aligned} \gamma_{a_u}(j) = \tilde{\beta}_{a_u}(j) = f_{\theta_j}(y_u), \\ \mathcal{N}_{a_u} = \sum_j P(S_u = j)\gamma_{a_u}(j) = P(Y_u = y_u) \\ \text{et} \quad \beta_{a_u}(j) = \frac{\gamma_{a_u}(j)}{\mathcal{N}_{a_u}}. \end{aligned}$$

La phase de propagation consiste en les équations (2.42) à (2.44). Leur application aux arbres de Markov cachés donne

$$\begin{cases} \gamma_{b_u}(j) & = \sum_k p_{j,k}\beta_{a_u}(k) \\ \mathcal{N}_{b_u} & = \sum_j P(S_{\rho(u)} = j)\gamma_{b_u}(j) \\ \beta_{b_u}(j) & = \dfrac{\gamma_{b_u}(j)}{\mathcal{N}_{b_u}}, \end{cases} \tag{2.52}$$

ce qui correspond à l'équation (2.29) avec l'introduction d'une quantité de normalisation. De même on obtient

$$\begin{cases} \gamma_{a_u}(k) & = f_{\theta_k}(y_u) \prod_{v \in \mathbf{c}(u)} \beta_{b_v}(k) \\ \mathcal{N}_{a_u} & = \sum_k P(S_u = k)\gamma_{a_u}(k) \\ \beta_{a_u}(k) & = \dfrac{\gamma_{a_u}(k)}{\mathcal{N}_{a_u}}, \end{cases} \tag{2.53}$$

analogue à l'équation (2.30). Par définition,

$$\beta_{b_u}(j) = \frac{P(S_{\rho(u)} = j | \boldsymbol{Y}_u = \boldsymbol{y}_u)}{P(S_{\rho(u)} = j)}$$

et

$$\beta_{a_u}(k) = \frac{P(S_u = k | \boldsymbol{Y}_u = \boldsymbol{y}_u)}{P(S_u = k)}.$$

Le calcul de la quantité de normalisation $\mathcal{N}_{b_u}$ mérite d'être commenté dans la mesure où $\mathcal{K}_{b_u} = \mathcal{K}_{a_u}$, et donc

$$\mathcal{N}_{b_u} = \frac{P(\boldsymbol{Y}_{\mathcal{K}_{b_u}} = \boldsymbol{y}_{\mathcal{K}_{b_u}})}{P(\boldsymbol{Y}_{\mathcal{K}_{a_u}} = \boldsymbol{y}_{\mathcal{K}_{a_u}})} = 1$$

ce qui s'explique, entre autres, par le fait que a_u est le seul arc distinct de b_u incident à $\mathcal{D}_u$. En revanche, comme $\mathcal{C}_u$ admet les arcs incidents a_u et $\{b_v\}_{v\in\boldsymbol{c}(u)}$ – donc au moins trois arcs incidents en général – il vient

$$\mathcal{N}_{a_u} = \frac{P(\boldsymbol{Y}_{\mathcal{K}_{a_u}} = \boldsymbol{y}_{\mathcal{K}_{a_u}})}{\prod\limits_{v\in\boldsymbol{c}(u)} P(\boldsymbol{Y}_{\mathcal{K}_{b_v}} = \boldsymbol{y}_{\mathcal{K}_{b_v}})} = \frac{P(\boldsymbol{Y}_{\mathcal{K}_{a_u}} = \boldsymbol{y}_{\mathcal{K}_{a_u}})}{\prod\limits_{v\in\boldsymbol{c}(u)} P(\boldsymbol{Y}_v = \boldsymbol{y}_v)} \neq 1 \text{ en général.}$$

Ceci montre d'une part que cet algorithme avant-arrière, du fait de sa généricité, peut conduire à faire des calculs inutiles. Le caractère explicite (vis-à-vis du rôle des paramètres, et en termes probabilistes) de ses formules de transition est toutefois un atout puisqu'il permet une détection aisée des calculs inutiles et une optimisation rapide de l'algorithme pour un modèle particulier. D'autre part, un argument semblable explique l'absence de phase préliminaire dans l'algorithme de Devijver dans les chaînes de Markov cachées. En effet, dans ces modèles, chaque clique est incidente à au plus deux arcs de l'arbre de jonction et le dénominateur $P(S_t = j)$, dans les quantités arrière, se simplifie avec les quantités de normalisation et les $P(S_t = j)$ intervenant dans la définition des $\gamma_t(j)$ – voir équations (2.42) à (2.44).

Enfin, le calcul précédent, les équations de propagation (2.52) et (2.53) et la définition de $\beta_{b_u}(j)$ et $\beta_{a_u}(k)$ permettent d'établir les formules

$$\frac{P(S_{\rho(u)} = j|\boldsymbol{Y}_u = \boldsymbol{y}_u)}{P(S_{\rho(u)} = j)} = \sum_k p_{j,k} \frac{P(S_u = k|\boldsymbol{Y}_u = \boldsymbol{y}_u)}{P(S_u = k)}$$

et

$$P(S_u = k|\boldsymbol{Y}_u = \boldsymbol{y}_u) = \frac{f_{\theta_k}(y_u)\left\{\prod\limits_{v\in\boldsymbol{c}(u)} \beta_{a_u}(k)\right\} P(S_u = k)}{\mathcal{N}_{a_u}}.$$

On obtient ainsi les formules de propagation de la phase ascendante (autrement dit, arrière) de l'algorithme ascendant-descendant utilisant les probabilités de lissage proposé par Durand, Gonçalvès et Guédon, 2002 [46]. Le calcul de la log-vraisemblance coïncide également puisqu'il est donné dans les deux cas par $\ln(\mathcal{L}_{\boldsymbol{y}}(\lambda)) = \prod\limits_{a\in\mathcal{T}} \ln(\mathcal{N}_a)$ où la définition de la quantité $\mathcal{N}_0$ est étendue à la clique $\mathcal{C}_0$ par

$$\mathcal{N}_0 = \sum_j \pi_j f_{\theta_j}(y_1) \prod_{u\in\boldsymbol{c}(1)} \beta_{b_u}(j).$$

Phase avant

Nous présentons ici la phase avant basée sur la loi jointe des variables cachées de chaque clique conditionnellement au processus observé. D'après l'équation (2.50), cette phase est initialisée par

$$P(S_1 = k|\boldsymbol{Y} = \boldsymbol{y}) = \frac{1}{\mathcal{N}} \pi_k f_{\theta_k}(y_1) \prod_{u\in\boldsymbol{c}(1)} \beta_{b_u}(k)$$

où la quantité de normalisation

$$\mathcal{N} = \frac{P(\boldsymbol{Y} = \boldsymbol{y})}{\prod\limits_{u \in \boldsymbol{c}(1)} P(\boldsymbol{Y}_u = \boldsymbol{y}_u)}$$

a été utilisée lors du calcul de la vraisemblance.

En considérant l'arc a_u, on obtient d'après la formule (2.51) la première équation de propagation

$$P(S_u = k|\boldsymbol{Y} = \boldsymbol{y}) = \frac{f_{\theta_k}(y_u)}{\mathcal{N}_{a_u}\beta_{a_u}(k)} \prod_{v \in \boldsymbol{c}(u)} \beta_{b_v}(k) \sum_j P(S_{\rho(u)} = j, S_u = k|\boldsymbol{Y} = \boldsymbol{y})$$

En remarquant que d'après les équations (2.52) on a

$$\frac{f_{\theta_k}(y_u) \prod\limits_{v \in \boldsymbol{c}(u)} \beta_{b_v}(k)}{\mathcal{N}_{a_u}\beta_{a_u}(k)} = 1,$$

on obtient un équivalent trivial de l'équation ci-dessus

$$P(S_u = k|\boldsymbol{Y} = \boldsymbol{y}) = \sum_j P(S_{\rho(u)} = j, S_u = k|\boldsymbol{Y} = \boldsymbol{y}). \tag{2.54}$$

En considérant l'arc b_u, on obtient

$$P(S_{\rho(u)} = j, S_u = k|\boldsymbol{Y} = \boldsymbol{y}) = \frac{p_{j,k}}{\mathcal{N}_{b_u}\beta_{b_u}(j)}\beta_{a_u}(k)P(S_{\rho(u)} = j|\boldsymbol{Y} = \boldsymbol{y})$$

On en déduit, vu que $\mathcal{N}_{b_u} = 1$,

$$\begin{aligned} &P(S_{\rho(u)} = j, S_u = k|\boldsymbol{Y} = \boldsymbol{y}) \\ &\quad = \frac{p_{j,k}P(S_{\rho(u)} = j)}{P(S_{\rho(u)} = j|\boldsymbol{Y}_u = \boldsymbol{y}_u)P(S_u = k)}P(S_u = k|\boldsymbol{Y}_u = \boldsymbol{y}_u)P(S_{\rho(u)} = j|\boldsymbol{Y} = \boldsymbol{y}) \end{aligned}$$

ce qui est exactement l'équation de propagation avant de Durand, Gonçalvès et Guédon, 2002 [46]. On déduit également, en combinant l'équation ci-dessus et l'équation (2.54), la récursion

$$P(S_u = k|\boldsymbol{Y} = \boldsymbol{y}) = \frac{P(S_u = k|\boldsymbol{Y}_u = \boldsymbol{y}_u)}{P(S_u = k)} \sum_j \frac{p_{j,k}P(S_{\rho(u)} = j)}{P(S_{\rho(u)} = j|\boldsymbol{Y}_u = \boldsymbol{y}_u)}P(S_{\rho(u)} = j|\boldsymbol{Y} = \boldsymbol{y})$$

qui correspond également à l'algorithme de lissage de Durand, Gonçalvès et Guédon. L'algorithme présenté dans la section 2.4.5 en est donc une généralisation.

Exemple numérique

Nous considérons une réalisation du modèle d'arbre de Markov caché de la section 1.3 du chapitre 1 dont nous rappelons ci-dessous les paramètres. Le paramètre π désigne la loi de la racine S_1, qui est un état caché. Les lois d'émission sont les lois gaussiennes de paramètres (μ_k, σ_k^2). La matrice de transition entre un état caché père et ses descendants est notée P. Nous utilisons donc les valeurs :

$$\pi_1 = 1 \quad \pi_2 = 0 \quad \mu_1 = -3 \quad \sigma_1^2 = 1 \quad \mu_2 = 3 \quad \sigma_2^2 = 2 \quad P = \begin{bmatrix} 0,9 & 0,1 \\ 0 & 1 \end{bmatrix}.$$

L'arbre considéré est binaire et équilibré, de profondeur 8 et comporte donc $2^8 - 1 = 255$ variables aléatoires observées. Dans un premier temps, nous calculons la vraisemblance du paramètre ayant servi à générer les données par l'*algorithme ascendant* de Crouse, Nowak et Baraniuk, 1998 [32] à l'aide d'un programme écrit en MATLAB (version 6.0.0.88 (R12) pour Linux). Les quantités intermédiaires $\tilde{\beta}$ utilisées dans l'algorithme ont une valeur inférieure au plus petit réel positif représentable en MATLAB, c'est-à-dire environ $2,2251.10^{-308}$ et la vraisemblance ne peut donc être calculée par cet algorithme. Puis nous calculons la vraisemblance par notre algorithme ascendant utilisant des probabilités de lissage. La vraisemblance obtenue est d'environ 10^{-1071}. Il est possible de simuler des arbres de profondeur 13 avec les paramètres ci-dessus, soit des arbres à 8191 sommets. Il s'agit du plus grand arbre binaire pouvant être stocké par MATLAB, vu la mémoire disponible (256 Mo). La vraisemblance peut encore être calculée pour de tels arbres, et vaut environ 10^{-33349}. De tels ordres de grandeur pour les probabilités mises en jeu montrent l'importance d'algorithmes calculant directement la log-vraisemblance (comme les algorithmes de lissage, section 2.4.5) et non la vraisemblance (comme les algorithmes de factorisation de la loi jointe, section 2.4.2). Bien sûr, l'estimation des paramètres à partir de processus arborescents simulés de profondeur supérieure à 8 n'est possible par la méthodologie EM qu'en utilisant l'algorithme ci-dessus, et non pas l'algorithme de la section 2.4.4.

2.4.7 Calcul de probabilités dans les modèles non orientés triangulés

Nous avons annoncé, dans le chapitre 1 section 1.5, qu'il est facile d'appliquer les algorithmes précédents aux modèles graphiques non orientés triangulés. D'après Lucke, 1996 [87], on peut paramétrer de tels modèles par les $\lambda_{\boldsymbol{x}_\mathcal{C}} = P(\boldsymbol{X}_\mathcal{C} = \boldsymbol{x}_\mathcal{C})$ où $\boldsymbol{x}_\mathcal{C}$ décrit toutes les valeurs possibles de $\mathcal{X}_\mathcal{C}$ – si toutes les variables aléatoires intervenant dans le modèle sont à valeurs discrètes. Les formules de réestimation de l'algorithme EM à l'itération η sont alors

$$\hat{\lambda}_{\boldsymbol{x}_\mathcal{C}} = P_{\lambda^{(\eta-1)}}(\boldsymbol{S}_\mathcal{C} = \boldsymbol{s}_\mathcal{C} | \boldsymbol{Y} = \boldsymbol{y}) \mathbb{I}_{\{\boldsymbol{Y}_\mathcal{C} = \boldsymbol{y}_\mathcal{C}\}}.$$

Les probabilités $P_{\lambda^{(\eta-1)}}(\boldsymbol{S}_\mathcal{C} = \boldsymbol{s}_\mathcal{C} | \boldsymbol{Y} = \boldsymbol{y})$ peuvent être calculées par les formules arrière-avant de la section 2.4.5. Les calculs sont grandement simplifiés par le fait que $P(\boldsymbol{X}_\mathcal{C} = \boldsymbol{x}_\mathcal{C})$ est donné par le paramètre $\lambda_{\boldsymbol{x}_\mathcal{C}}$ et par conséquent,

- n'importe quelle clique du graphe peut faire office de clique $\mathcal{C}_0$ pour terminer l'algorithme ;
- le calcul de $P(\boldsymbol{X}_\mathcal{C} = \boldsymbol{x}_\mathcal{C})$ ne demande ni calcul inductif, ni phase préliminaire.

On obtient ainsi un algorithme efficace et numériquement stable pour le calcul des probabilités dans les modèles non orientés triangulés.

2.4.8 Modèles homogènes avec des paramètres nuls

Dans la section 2.3.2, nous avons fait l'hypothèse que les probabilités $P(X_u = x_u|\boldsymbol{X}_{\mathrm{pa}(u)} = \boldsymbol{x}_{\mathrm{pa}(u)})$ étaient strictement positives pour chaque sommet u, pour chaque valeur de x_u dans $\mathcal{X}_u$ et de $\boldsymbol{x}_{\mathrm{pa}(u)}$ dans $\mathcal{X}_{\mathrm{pa}(u)}$, afin que la log-vraisemblance complétée soit définie et qu'on puisse estimer les paramètres par l'algorithme EM. Ceci revient à supposer que tous les paramètres π_i et $p_{\boldsymbol{a},i}$ sont strictement positifs. Quant aux probabilités $P_{\theta_{\boldsymbol{a}}}(y_u)$, on fait en général l'hypothèse que le support de P_θ ne dépend pas de θ, auquel cas la vraisemblance est nulle dès que l'un des y_u n'est pas dans ce support. Dans plusieurs phénomènes réels d'intérêt, on peut observer des états transitoires où des états absorbants, comme par exemple en reconnaissance de la parole (voir Rabiner, 1989 [102]) ou encore en fiabilité de logiciels (voir l'application du chapitre 3). La modélisation de tels phénomènes requiert l'hypothèse de nullité de certains paramètres.

Ainsi, nous considérons un modèle de Markov caché homogène de la famille $\mathcal{D}$ tel que $p^{(\bar{u})}_{\boldsymbol{a},i} = 0$ pour un certain couple $(\boldsymbol{a}, i)$. Il est en fait possible d'estimer le paramètre $p^{(\bar{u})}$ en tenant compte de cette contrainte : pour chaque sommet u de la classe d'équivalence $\bar{u}$ associée au paramètre $p^{(\bar{u})}$, d'après la propriété 1, l'ensemble $\{u\}\cup$ pa(u) est inclus dans une clique $\mathcal{C}_j$ de l'arbre de jonction. D'après la section 2.3.3, la formule de réestimation de $p^{(\bar{u})}_{\boldsymbol{a},i}$ est une fraction faisant intervenir $P_{\hat{\lambda}^{(\eta-1)}}(\boldsymbol{S}_{\mathrm{pa}(u)} = \boldsymbol{s}_{\mathrm{pa}(u)}|\boldsymbol{Y} = \boldsymbol{y})$ ou $P_{\hat{\lambda}^{(\eta-1)}}(S_u = i, \boldsymbol{S}_{\mathrm{pa}(u)} = \boldsymbol{s}_{\mathrm{pa}(u)}|\boldsymbol{Y} = \boldsymbol{y})$ au numérateur, suivant que X_u est observée ou cachée. Cette probabilité s'obtient à partir de

$$P_{\hat{\lambda}^{(\eta-1)}}(\boldsymbol{S}_{\mathcal{C}_j\backslash(\mathcal{U}_{\boldsymbol{a}}\cup\{i\})} = \boldsymbol{s}_{\mathcal{C}_j\backslash(\mathcal{U}_{\boldsymbol{a}}\cup\{i\})}, \boldsymbol{S}_{\mathcal{U}_{\boldsymbol{a}}} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}}, S_u = i|\boldsymbol{Y} = \boldsymbol{y}),$$

calculée par l'algorithme arrière-avant des sections 2.4.2 et 2.4.3 (formule (2.28) par exemple, ou sa variante). Cette formule montre que la probabilité $P_{\hat{\lambda}^{(\eta-1)}}(\boldsymbol{X}_{\mathcal{C}_j} = \boldsymbol{x}_{\mathcal{C}_j}|\boldsymbol{Y} = \boldsymbol{y})$ est proportionnelle à $P_{\hat{\lambda}^{(\eta-1)}}(\boldsymbol{X}_{\mathcal{C}_j} = \boldsymbol{x}_{\mathcal{C}_j})$, qui peut elle-même s'écrire

$$\begin{aligned} &P_{\hat{\lambda}^{(\eta-1)}}(\boldsymbol{S}_{\mathcal{C}_j} = \boldsymbol{s}_{\mathcal{C}_j}) \\ &\quad = \prod_{l>1} P_{\hat{\lambda}^{(\eta-1)}}(X_j^{(l)} = x_j^{(l)}|X_j^{(1)} = x_j^{(1)}, \ldots, X_j^{(l-1)} = x_j^{(l-1)}) P_{\hat{\lambda}^{(\eta-1)}}(X_j^{(1)} = x_j^{(1)}), \end{aligned}$$

avec les notations de la figure 2.1. Or la variable aléatoire X_u est l'un des $X_j^{(l)}$ par définition de $\mathcal{C}_j$ donc pa$(X_u) = \{X_j^{(1)}, \ldots, X_j^{(l-1)}\}$ et

$$P_{\hat{\lambda}^{(\eta-1)}}(X_j^{(l)} = x_j^{(l)}|X_j^{(1)} = x_j^{(1)}, \ldots, X_j^{(l-1)} = x_j^{(l-1)}) = 0,$$

dès lors que $p^{(\bar{u})}_{\boldsymbol{a},i} = 0$ pour le paramètre $\hat{\lambda}^{(\eta-1)}$ à l'itération précédente de l'algorithme EM. Donc $P_{\hat{\lambda}^{(\eta-1)}}(\boldsymbol{S}_{\mathcal{C}_j\backslash(\mathcal{U}_{\boldsymbol{a}}\cup\{i\})} = \boldsymbol{s}_{\mathcal{C}_j\backslash(\mathcal{U}_{\boldsymbol{a}}\cup\{i\})}, \boldsymbol{S}_{\mathcal{U}_{\boldsymbol{a}}} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}}, S_u = i|\boldsymbol{Y} = \boldsymbol{y}) = 0$ pour toute valeur de $\boldsymbol{s}_{\mathcal{C}_j\backslash(\mathcal{U}_{\boldsymbol{a}}\cup\{i\})}$ et le numérateur de la formule de réestimation de $p^{(\bar{u})}_{\boldsymbol{a},i}$ est nul.

En conclusion, comme il a été remarqué dans Rabiner, 1989 [102] dans le cadre des chaînes de Markov cachées, l'imposition de contraintes de type $p^{(\bar{u})}_{\boldsymbol{a},i} = 0$ ne modifie pas l'étape M de l'algorithme EM, en réalité. Ceci provient du fait que toute valeur des paramètres de transition (ou des paramètres initiaux) valant 0 à l'initialisation de l'algorithme reste nulle à chaque itération η, comme nous venons de le montrer. En conclusion, l'algorithme EM de la section 2.3 reste valable même quand la log-vraisemblance complétée est nulle pour certaines valeurs de $\boldsymbol{s}$, autrement dit quand dans le modèle, certaines probabilités de transition $P(X_u = x_u | \boldsymbol{X}_{\mathrm{pa}(u)} = \boldsymbol{x}_{\mathrm{pa}(u)})$ sont nulles. En revanche, nous conserverons l'hypothèse que pour tout ensemble $\bar{u}$ de sommets dans $\bar{\mathcal{U}}_\theta$, le support de $P_{\theta(\bar{u})}$ ne dépend pas de la valeur de $\theta^{(\bar{u})}$.

2.4.9 Simulation des états cachés dans les variantes stochastiques de EM

Dans cette section, nous montrons comment simuler les états cachés de manière à implémenter les algorithmes SEM et EM à la Gibbs. Rappelons qu'il s'agit d'algorithmes de restauration-maximisation qui consistent à remplacer les données manquantes $\boldsymbol{s}$ par un tirage aléatoire $\hat{\boldsymbol{s}}$, dans l'étape de restauration.

Algorithme SEM

Dans le cas de l'algorithme SEM, les données manquantes sont simulées suivant la loi

$$P_{\hat{\lambda}^{(\eta)}}(\boldsymbol{S} = \boldsymbol{s} | \boldsymbol{Y} = \boldsymbol{y}).$$

Nous supposons donc, dans cette section, la loi P fixée à $P_{\hat{\lambda}^{(\eta)}}$. Pour implémenter l'algorithme SEM, il suffit de savoir simuler $\hat{s}_1$ suivant la loi $P(S_1 = s_1 | \boldsymbol{Y} = \boldsymbol{y})$, puis en supposant que l'on ait simulé la séquence $(\hat{s}_1, \ldots, \hat{s}_t)$, notée $\hat{\boldsymbol{s}}_1^t$, de savoir simuler $\hat{s}_{t+1}$ suivant la loi

$$P(S_{t+1} = s_{t+1} | \boldsymbol{S}_1^t = \hat{\boldsymbol{s}}_1^t, \boldsymbol{Y} = \boldsymbol{y}). \tag{2.55}$$

Ceci provient de la factorisation suivante de la loi $P(\boldsymbol{S} = \boldsymbol{s} | \boldsymbol{Y} = \boldsymbol{y})$:

$$P(\boldsymbol{S} = \boldsymbol{s} | \boldsymbol{Y} = \boldsymbol{y}) = P(S_1 = s_1 | \boldsymbol{Y} = \boldsymbol{y}) \prod_t P(S_{t+1} = s_{t+1} | \boldsymbol{S}_1^t = \boldsymbol{s}_1^t, \boldsymbol{Y} = \boldsymbol{y}).$$

Nous proposons, pour simuler ainsi les états cachés, une méthode dont la première étape est l'exécution de l'algorithme arrière-avant utilisant les probabilités de lissage de la section 2.4.5. On obtient ainsi les quantités $\beta_a(\boldsymbol{s}_{\mathbb{S}_a})$ pour chaque arc a de l'arbre de jonction et les probabilités $P(\boldsymbol{S}_{\mathcal{C}_j} = \boldsymbol{s}_{\mathcal{C}_j} | \boldsymbol{Y} = \boldsymbol{y})$ pour chaque clique $\mathcal{C}_j$ de l'arbre de jonction. La loi conditionnelle $P(S_0^{(1)} = j | \boldsymbol{Y} = \boldsymbol{y})$ de la source $S_0 = S_0^{(1)}$ se déduit de la loi $P(\boldsymbol{S}_{\mathcal{C}_0} = \boldsymbol{s}_{\mathcal{C}_0} | \boldsymbol{Y} = \boldsymbol{y})$ par sommation sur toutes les valeurs possibles de $\boldsymbol{S}_{\mathcal{C}_0 \backslash \{S_0^{(1)}\}}$. Les états cachés $(S_0^{(l)})_l$ de $\mathcal{C}_0$ sont alors simulés de manière séquentielle. Supposons que $(\hat{s}_0^{(1)}, \ldots, \hat{s}_0^{(l-1)})$ soit une réalisation du processus $(S_0^{(1)}, \ldots, S_0^{(l-1)})$, suivant sa loi conditionnelle sachant $\boldsymbol{Y} = \boldsymbol{y}$. Alors

$$\begin{aligned} & P(S_0^{(l)} = j | S_0^{(1)} = \hat{s}_0^{(1)}, \ldots, S_0^{(l-1)} = \hat{s}_0^{l-(1)}, \boldsymbol{Y} = \boldsymbol{y}) \\ = & \frac{P(S_0^{(l)} = j, S_0^{(1)} = \hat{s}_0^{(1)}, \ldots, S_0^{(l-1)} = \hat{s}_0^{(l-1)} | \boldsymbol{Y} = \boldsymbol{y})}{\sum_{j'} P(S_0^{(l)} = j', S_0^{(1)} = \hat{s}_0^{(1)}, \ldots, S_0^{(l-1)} = \hat{s}_0^{(l-1)} | \boldsymbol{Y} = \boldsymbol{y})} \end{aligned}$$

où $P(S_0^{(l)} = j, S_0^{(1)} = \hat{s}_0^{(1)}, \ldots, S_0^{(l-1)} = \hat{s}_0^{l-(1)} | \boldsymbol{Y} = \boldsymbol{y})$ se déduit de $P(\boldsymbol{S}_{\mathcal{C}_0} = \boldsymbol{s}_{\mathcal{C}_0} | \boldsymbol{Y} = \boldsymbol{y})$.

Les états cachés des autres cliques sont alors simulés par un parcours de l'arbre de jonction de type avant : soit $a = (\mathcal{C}_i, \mathcal{C}_j)$ un arc de l'arbre de jonction $\mathcal{T}$, la clique $\mathcal{C}_i$ étant donc située sur le chemin entre $\mathcal{C}_0$ et $\mathcal{C}_j$ dans $\mathcal{T}$. On suppose avoir simulé un certain nombre d'états cachés appartenant à un ensemble de sommets A inclus dans l'ensemble $\mathcal{U}(\mathcal{T}_a^c)$ de la partie de l'arbre jonction (séparé par l'arc a) contenant $\mathcal{C}_0$. On suppose de plus que A contient les sommets cachés $\mathcal{U}(\mathcal{C}_i) \cap \mathcal{U}_{\boldsymbol{S}}$ de la clique $\mathcal{C}_i$. Ainsi, $\hat{\boldsymbol{s}}_A$ est une réalisation du processus $\boldsymbol{S}_A$ tirée suivant sa loi conditionnelle sachant $\boldsymbol{Y} = \boldsymbol{y}$. La simulation des états cachés de $\mathcal{C}_j \setminus A$ suivant la loi $P(\boldsymbol{S}_{\mathcal{C}_j \setminus A} | \boldsymbol{S}_A = \hat{\boldsymbol{s}}_A, \boldsymbol{Y} = \boldsymbol{y})$ se fait comme suit. Rappelons que $\mathcal{K}_a^c$ est l'ensemble des sommets des cliques de $\mathcal{T}_a^c$ (voir figure 2.1). Alors par hypothèse, $A \subset \mathcal{K}_a^c$. Par conséquent,

$$\begin{aligned}
&P(\boldsymbol{S}_{\mathcal{C}_j \setminus A} = \boldsymbol{s}_{\mathcal{C}_j \setminus A} | \boldsymbol{S}_A = \hat{\boldsymbol{s}}_A, \boldsymbol{Y} = \boldsymbol{y}) \\
&\quad = P(\boldsymbol{S}_{\mathcal{C}_j \setminus A} = \boldsymbol{s}_{\mathcal{C}_j \setminus A} | \boldsymbol{S}_{A \setminus \mathbb{S}_a} = \hat{\boldsymbol{s}}_{A \setminus \mathbb{S}_a}, \boldsymbol{S}_{\mathbb{S}_a} = \hat{\boldsymbol{s}}_{\mathbb{S}_a}, \boldsymbol{Y}_{\mathbb{S}_a} = \boldsymbol{y}_{\mathbb{S}_a}, \boldsymbol{Y}_{\mathcal{U}_{\boldsymbol{Y}} \setminus \mathbb{S}_a} = \boldsymbol{y}_{\mathcal{U}_{\boldsymbol{Y}} \setminus \mathbb{S}_a}) \\
&\quad = P(\boldsymbol{S}_{\mathcal{C}_j \setminus A} = \boldsymbol{s}_{\mathcal{C}_j \setminus A} | \boldsymbol{S}_{\mathbb{S}_a} = \hat{\boldsymbol{s}}_{\mathbb{S}_a}, \boldsymbol{Y}_{\mathbb{S}_a} = \boldsymbol{y}_{\mathbb{S}_a}, \boldsymbol{Y}_{\mathcal{U}_{\boldsymbol{Y}} \setminus \mathbb{S}_a} = \boldsymbol{y}_{\mathcal{U}_{\boldsymbol{Y}} \setminus \mathbb{S}_a}) \qquad (2.56) \\
&\quad = P(\boldsymbol{S}_{\mathcal{C}_j \setminus A} = \boldsymbol{s}_{\mathcal{C}_j \setminus A} | \boldsymbol{S}_{\mathbb{S}_a} = \hat{\boldsymbol{s}}_{\mathbb{S}_a}, \boldsymbol{Y} = \boldsymbol{y}) \\
&\quad = \frac{P(\boldsymbol{S}_{\mathcal{C}_j \setminus A} = \boldsymbol{s}_{\mathcal{C}_j \setminus A}, \boldsymbol{S}_{\mathbb{S}_a} = \hat{\boldsymbol{s}}_{\mathbb{S}_a} | \boldsymbol{Y} = \boldsymbol{y})}{\sum\limits_{\tilde{\boldsymbol{s}}_{\mathcal{C}_j \setminus A}} P(\boldsymbol{S}_{\mathcal{C}_j \setminus A} = \tilde{\boldsymbol{s}}_{\mathcal{C}_j \setminus A}, \boldsymbol{S}_{\mathbb{S}_a} = \hat{\boldsymbol{s}}_{\mathbb{S}_a} | \boldsymbol{Y} = \boldsymbol{y})}.
\end{aligned}$$

L'équation (2.56) provient du fait que $\mathbb{S}_a$ est un séparateur de cliques, donc par définition de $\mathcal{K}_a^c$, ses sommets séparent ceux de $\mathcal{C}_j \setminus \mathbb{S}_a$ de ceux de $\mathcal{K}_a^c$. Par conséquent, les sommets de $\boldsymbol{S}_{\mathbb{S}_a}$ et $\boldsymbol{Y}$ séparent également ceux de $\mathcal{C}_j \setminus \mathbb{S}_a$ de ceux de $\mathcal{K}_a^c$. Il s'ensuit que $\boldsymbol{S}_{\mathcal{C}_j \setminus A}$ est indépendant de $\boldsymbol{S}_{A \setminus \mathbb{S}_a}$ sachant $\{\boldsymbol{S}_{\mathbb{S}_a} = \hat{\boldsymbol{s}}_{\mathbb{S}_a}\} \cap \{\boldsymbol{Y} = \boldsymbol{y}\}$. Comme la loi $P(\boldsymbol{S}_{\mathcal{C}_j} = \boldsymbol{s}_{\mathcal{C}_j} | \boldsymbol{Y} = \boldsymbol{y})$ est donnée, pour tout $\boldsymbol{s}_{\mathcal{C}_j} \in \mathcal{S}_{\mathcal{C}_j}$, par l'algorithme arrière-avant, on en déduit $P(\boldsymbol{S}_{\mathcal{C}_j \setminus A} = \boldsymbol{s}_{\mathcal{C}_j \setminus A} | \boldsymbol{S}_A = \hat{\boldsymbol{s}}_A, \boldsymbol{Y} = \boldsymbol{y})$ pour tout $\boldsymbol{s}_{\mathcal{C}_j \setminus A} \in \mathcal{S}_{\mathcal{C}_j \setminus A}$. Soit $\{S_1, \ldots, S_{l_j}\}$ l'ensemble des variables cachées de $\mathcal{C}_j \setminus A$. Ces variables aléatoires sont simulées suivant la même procédure que pour celles de la clique $\mathcal{C}_0$, c'est-à-dire que

– l'on simule une réalisation $\hat{s}_1$ de la variable S_1, tirée suivant la loi

$$\begin{aligned}
&P(S_1 = j | \boldsymbol{S}_A = \hat{\boldsymbol{s}}_A, \boldsymbol{Y} = \boldsymbol{y}) \\
&\quad = \sum_{\tilde{s}_2, \ldots, \tilde{s}_{l_j}} P(S_1 = j, S_2 = \tilde{s}_2, \ldots, S_{l_j} = \tilde{s}_{l_j} | \boldsymbol{S}_A = \hat{\boldsymbol{s}}_A, \boldsymbol{Y} = \boldsymbol{y}).
\end{aligned}$$

– les autres variables S_l sont simulées séquentiellement : supposons que $(\hat{s}_1, \ldots, \hat{s}_{l-1})$ soit une réalisation du processus $(S_1, \ldots, S_{l-1})$, suivant sa loi conditionnelle sachant $\boldsymbol{Y} = \boldsymbol{y}$. Alors S_l est simulée suivant la loi

$$\begin{aligned}
&P(S_l = j | S_1 = \hat{s}_1, \ldots, S_{l-1} = \hat{s}_{l-1}, \boldsymbol{S}_A = \hat{\boldsymbol{s}}_A, \boldsymbol{Y} = \boldsymbol{y}) \\
&= \frac{P(S_l = j, S_1 = \hat{s}_1, \ldots, S_{l-1} = \hat{s}_{l-1} | \boldsymbol{S}_A = \hat{\boldsymbol{s}}_A, \boldsymbol{Y} = \boldsymbol{y})}{\sum\limits_{j'} P(S_l = j', S_1 = \hat{s}_1, \ldots, S_{l-1} = \hat{s}_{l-1} | \boldsymbol{S}_A = \hat{\boldsymbol{s}}_A, \boldsymbol{Y} = \boldsymbol{y})}
\end{aligned}$$

où $P(S_l = j', S_1 = \hat{s}_1, \ldots, S_{l-1} = \hat{s}_{l-1} | \boldsymbol{S}_A = \hat{\boldsymbol{s}}_A, \boldsymbol{Y} = \boldsymbol{y})$ se déduit de $P(\boldsymbol{S}_{\mathcal{C}_j \setminus A} = \boldsymbol{s}_{\mathcal{C}_j \setminus A} | \boldsymbol{S}_A = \hat{\boldsymbol{s}}_A, \boldsymbol{Y} = \boldsymbol{y})$.

En conclusion, seuls les sommets de $\hat{\boldsymbol{s}}$ appartenant à $\mathcal{C}_i$ (et même, plus précisément, seuls ceux appartenant à $\mathcal{C}_i \cap \mathcal{C}_j$) interviennent dans le tirage des variables aléatoires de $\mathcal{C}_j$ quand un algorithme de type avant est utilisé (parcours de l'arbre de jonction en partant de $\mathcal{C}_0$ et en allant vers les feuilles de l'arbre). La formule (2.55) laisserait penser que l'ordre dans lequel les états cachés sont simulés est crucial. En réalité, nous avons une grande latitude pour le choix de cet ordre : d'après ce qui précède, il suffit, pour simuler les états cachés d'une clique $\mathcal{C}_j$, que tous les états cachés des cliques situées sur le chemin entre $\mathcal{C}_0$ et $\mathcal{C}_j$ dans l'arbre de jonction aient été simulés. Enfin, cette étape est d'une complexité $\mathcal{O}(\sum_{C \in \mathcal{V}_\mathcal{G}} \text{taille}(C))$. Cette complexité est due à l'exécution préalable de l'algorithme arrière-avant. Rappelons que dans le cas de modèles de Markov cachés dynamiques, où chaque variable aléatoire du modèle peut prendre au plus K valeurs, où chaque clique a un nombre de variables aléatoires N_C borné par une constante L et où le nombre de cliques est une fonction polynomiale de degré m de N, la complexité de calcul de l'algorithme arrière-avant est $\mathcal{O}(K^L N^m)$. À titre d'exemple, dans les chaînes de Markov cachées, la complexité de cet algorithme est $\mathcal{O}(K^2 n)$.

Algorithme EM à la Gibbs

Dans le cas de l'algorithme EM à la Gibbs, on suppose qu'on a, à l'itération η de l'algorithme, une réalisation $\hat{\boldsymbol{s}}^{(\eta-1)}$ du processus caché. Comme dans l'algorithme SEM, les $\hat{z}^{(\eta)}$ sont simulés de manière séquentielle ; cependant, la manière de simuler les états cachés diffère de celle de l'algorithme SEM. On simule tout d'abord S_1 suivant la loi

$$P(S_1 | \boldsymbol{Y} = \boldsymbol{y}, \bigcap_{v \neq 1} \{S_v = \hat{s}_v^{(\eta-1)}\}), \tag{2.57}$$

ce qui fournit une réalisation $\hat{s}_1^{(\eta)}$ puis on simule S_2 suivant la loi

$$P(S_2 | \boldsymbol{Y} = \boldsymbol{y}, S_1 = \hat{s}_1^{(\eta)}, \bigcap_{v \neq 2} \{S_v = \hat{s}_v^{(\eta-1)}),$$

et ainsi de suite. On suppose donc avoir simulé les états cachés de l'ensemble $A \subset \mathcal{U}_{\boldsymbol{S}}$ et on désire simuler S_u, pour $u \in \mathcal{U} \setminus A$, suivant la loi

$$P(S_u = j | \boldsymbol{Y} = \boldsymbol{y}, \boldsymbol{S}_A = \hat{\boldsymbol{s}}_A^{(\eta)}, \bigcap_{v \notin A, v \neq u} \{S_v = \hat{s}_v^{(\eta-1)}).$$

Notons en(u) l'ensemble des enfants du sommet u de $\mathcal{U}$, c'est-à-dire :

$$\text{en}(u) = \{u' \in \mathcal{U} | (u, u') \in \mathcal{E}\}$$

où $\mathcal{E}$ est l'ensemble des arcs de $\mathcal{G}$. D'après Spiegelhalter *et al.*, 1996 [112] la probabilité ci-dessus s'écrit

$$\begin{aligned} & P(S_u = j | \boldsymbol{Y} = \boldsymbol{y}, \bigcap_{v \neq u} \{S_v = s_v\}) \qquad (2.58) \\ &= \frac{P(S_u = j | \boldsymbol{X}_{\text{pa}(u)} = \boldsymbol{x}_{\text{pa}(u)}) \prod_{v \in \text{en}(u)} P(X_v = x_v | \boldsymbol{X}_{\text{pa}(v) \setminus \{u\}} = \boldsymbol{x}_{\text{pa}(v) \setminus \{u\}}, S_u = j)}{\sum_j P(S_u = j | \boldsymbol{X}_{\text{pa}(u)} = \boldsymbol{x}_{\text{pa}(u)}) \prod_{v \in \text{en}(u)} P(X_v = x_v | \boldsymbol{X}_{\text{pa}(v) \setminus \{u\}} = \boldsymbol{x}_{\text{pa}(v) \setminus \{u\}}, S_u = j)}. \end{aligned}$$

Or ces probabilités sont directement données par les paramètres du modèle, puisque $P(X_v = j|\boldsymbol{X}_{\mathrm{pa}(v)} = \boldsymbol{a})$ est égal au paramètre $\lambda_{\boldsymbol{a},j}$. Ceci permet de calculer la probabilité (2.57) sans avoir recours à l'algorithme arrière-avant.

Notons que les états cachés peuvent être simulés en principe dans un ordre quelconque, contrairement à l'algorithme SEM, bien que l'on choisisse en général un ordre de simulation respectant la hiérarchie entre sommets définie par le graphe d'indépendance conditionnelle. Rappelons que $N_{\boldsymbol{S}}$ représente le nombre d'états cachés du modèle : d'après l'équation (2.58) et du fait que chaque état caché doive être simulé, l'étape de restauration des états cachés par simulation dans EM à la Gibbs est de complexité $\mathcal{O}(N_{\boldsymbol{S}})$, sous réserve que le nombre de descendants de chaque sommet soit une fonction bornée du nombre de sommets du graphe – ce qui est le cas en général. À titre d'exemple, dans les chaînes de Markov cachées, la complexité de cet algorithme est $\mathcal{O}(Kn)$.

Utilisation de ces algorithmes pour l'estimation bayésienne

Dans l'approche bayésienne, l'estimation du paramètre λ est basée sur la loi a posteriori $\pi(\boldsymbol{\lambda} = \lambda|\boldsymbol{Y} = \boldsymbol{y})$, déduite de la vraisemblance $\mathcal{L}(\boldsymbol{Y} = \boldsymbol{y}|\boldsymbol{\lambda} = \lambda)$ et de la loi a priori $\pi(\boldsymbol{\lambda} = \lambda)$. L'existence de données manquantes $\boldsymbol{s}$ rendant impossible l'obtention d'une valeur numérique exacte des probabilités a posteriori, on a recours à une méthode d'augmentation de données pour générer une chaîne de Markov $(\hat{\lambda}^{(m)}, \hat{\boldsymbol{s}}^{(m)})$ de loi stationnaire $\pi(\boldsymbol{\lambda} = \lambda, \boldsymbol{S} = \boldsymbol{s}|\boldsymbol{Y} = \boldsymbol{y})$ dont la marginale en $\boldsymbol{\lambda}$ donne $\pi(\boldsymbol{\lambda} = \lambda|\boldsymbol{Y} = \boldsymbol{y})$.

On simule alors, à chaque itération de l'algorithme, $\hat{\lambda}$ suivant la loi $\pi(\boldsymbol{\lambda} = \lambda|\boldsymbol{S} = \boldsymbol{s}, \boldsymbol{Y} = \boldsymbol{y})$ puis $\hat{\boldsymbol{s}}$ suivant la loi $P(\boldsymbol{S} = \boldsymbol{s}|\boldsymbol{Y} = \boldsymbol{y}, \boldsymbol{\lambda} = \lambda)$. On peut envisager de simuler $\hat{\boldsymbol{s}}$ suivant cette loi jointe, mais aussi, de façon alternative, composante par composante suivant la loi $P(S_u = j|\boldsymbol{Y} = \boldsymbol{y}, \bigcap_{v \neq 1} \{S_v = s_v\})$. Dans l'article de Scott, 2002 [108], traitant de l'estimation bayésienne dans les chaînes de Markov cachées, l'auteur nomme *échantillonnage de Gibbs avant-arrière* la méthode consistant à simuler $\hat{\boldsymbol{s}}$ suivant la loi jointe et *échantillonnage de Gibbs direct* la méthode consistant à simuler $\hat{\boldsymbol{s}}$ composante par composante, pour la raison que nous verrons ci-dessous.

Dans les deux cas, on part d'une valeur initiale $\hat{\lambda}^{(0)}$ du paramètre. Pour l'échantillonnage de Gibbs direct, on dispose également d'une valeur initiale $\hat{\boldsymbol{s}}^{(0)}$ du processus caché. À l'itération m, l'échantillonnage de Gibbs avant-arrière procède comme suit :

1. simuler $\hat{\boldsymbol{s}}^{(m+1)}$ suivant la loi $P(\boldsymbol{S} = \boldsymbol{s}|\boldsymbol{Y} = \boldsymbol{y}, \boldsymbol{\lambda} = \hat{\lambda}^{(m)})$,
2. simuler $\hat{\lambda}^{(m+1)}$ suivant la loi $\pi(\boldsymbol{\lambda} = \lambda|\boldsymbol{S} = \hat{\boldsymbol{s}}^{(m+1)}, \boldsymbol{Y} = \boldsymbol{y})$,

à la différence de l'échantillonnage de Gibbs direct, qui procède ainsi

1. pour tout $u \in \{1, \ldots, N_{\mathcal{S}}\}$, simuler $\hat{s}_u^{(m+1)}$ suivant la loi
$$P(S_u|\boldsymbol{Y} = \boldsymbol{y}, \boldsymbol{\lambda} = \hat{\lambda}^{(m)}, \bigcap_{v<u} \{S_v = \hat{s}_v^{(m+1)}\}, \bigcap_{v>u} \{S_v = \hat{s}_v^{(m)}\}),$$
2. simuler $\hat{\lambda}^{(m+1)}$ suivant la loi $\pi(\boldsymbol{\lambda} = \lambda|\boldsymbol{S} = \hat{\boldsymbol{s}}^{(m+1)}, \boldsymbol{Y} = \boldsymbol{y})$.

Ainsi, on remarque que l'étape 1 de l'échantillonnage de Gibbs avant-arrière repose sur le même principe que la simulation du processus caché dans l'algorithme SEM, tandis que la simulation dans l'échantillonnage de Gibbs direct est similaire à la simulation dans EM à la Gibbs. Ceci montre l'intérêt des deux algorithmes présentés dans cette section, dans le cadre de l'estimation bayésienne. D'autre part, cette constatation met en évidence la différence de complexité calculatoire entre les deux algorithmes : le premier nécessite une récursion avant-arrière de complexité polynomiale en général (d'où le nom d'*échantillonnage de Gibbs avant-arrière*) alors que le second est d'une complexité moindre (linéaire, typiquement). C'est sans doute l'une des raisons pour laquelle l'échantillonnage de Gibbs direct est plus souvent utilisé. Notons que la dénomination *avant-arrière* est adaptée aux chaînes de Markov cachées mais devrait plutôt devenir *arrière-avant* pour les autres modèles de la famille $\mathcal{D}$.

L'article de Scott, 2002 [108] préconise l'usage de l'échantillonnage de Gibbs avant-arrière et montre que son usage accélère la mélangeance vis-à-vis de $\hat{\boldsymbol{\lambda}}^{(m)}$. Sa preuve s'appuie sur le fait qu'une mélangeance plus rapide pour $\hat{\boldsymbol{S}}^{(m)}$ se traduit par une mélangeance plus rapide pour $\hat{\boldsymbol{\lambda}}^{(m)}$, suivant un principe analogue à celui de la dualité au sens de Diebolt et Robert, 1994 [39], introduite dans le cadre des modèles de mélanges indépendants. La preuve de Scott est basée sur la comparaison de l'autocovariance de statistiques exhaustives des données complètes, qui est plus élevée dans le cas de l'échantillonnage de Gibbs direct, ce qui explique la mélangeance plus lente. Cependant, la comparaison globale des deux méthodes d'échantillonnage est complexe puisqu'il s'agit de savoir si le surcoût engendré par la récursion *arrière-avant* est réellement compensé par l'accélération de la mélangeance. Scott cite également d'autres raisons de disposer d'une méthode permettant de simuler $\boldsymbol{S}$ plutôt que de moyenner une quantité sur toutes les valeurs de $\boldsymbol{s} \in \mathcal{S}$ par l'algorithme de Metropolis-Hastings. Ainsi, l'auteur relève que pour contrôler la convergence de l'algorithme MCMC, on peut utiliser une certaine fonction de $\boldsymbol{s}$. La simulation du processus caché peut également être importante, suivant l'application, ou pour l'évaluation de l'adéquation du modèle.

2.4.10 Conclusion sur le calcul de probabilités

Cette section a mis en évidence l'existence d'algorithmes efficaces, d'interprétation plus aisée que l'algorithme d'arbre de jonction de Jensen, Lauritzen et Olesen, 1990 [67] et numériquement stables, grâce à la factorisation des probabilités de lissage au lieu de celle des probabilités jointes. Ces algorithmes permettent de calculer la vraisemblance ainsi que les probabilités $P(\boldsymbol{S}_\mathcal{C} = \boldsymbol{s}_\mathcal{C} | \boldsymbol{Y} = \boldsymbol{y})$ qui interviennent dans l'algorithme EM.

Dans de nombreux cas, il peut être utile de connaître la loi conditionnelle $P(\boldsymbol{X}_A = \boldsymbol{x}_A | \boldsymbol{X}_B = \boldsymbol{x}_B)$ pour deux ensembles de sommets A et B quelconques. Par exemple, dans le cas introduit dans la section 1.5, où le modèle graphique implémente un système expert d'aide au diagnostic médical (figure 1.14), il peut être souhaitable, suivant les cas, de calculer la probabilité de souffrir d'une dyspnée sachant qu'on a une bronchite et pas de cancer du poumon. Dans d'autres contextes on désirera calculer la probabilité d'avoir un cancer du poumon sachant qu'on a une bronchite et une dyspnée, en l'absence de toute information sur les autres variables aléatoires.

Ce calcul peut alors être réalisé en déterminant $P(\boldsymbol{X}_A = \boldsymbol{x}_A, \boldsymbol{X}_B = \boldsymbol{x}_B)$ et $P(\boldsymbol{X}_B = \boldsymbol{x}_B)$ par l'algorithme arrière-avant, en considérant pour le calcul de $P(\boldsymbol{X}_A = \boldsymbol{x}_A, \boldsymbol{X}_B = \boldsymbol{x}_B)$ que les variables de $A \cup B$ sont observées (fixées à $\boldsymbol{x}_A$ et à $\boldsymbol{x}_B$, respectivement), toutes les autres variables aléatoires étant inconnues, donc cachées. La probabilité $P(\boldsymbol{X}_B = \boldsymbol{x}_B)$ peut se déduire du calcul précédent par sommation sur toutes les valeurs de $\boldsymbol{x}_A$ si le nombre de valeurs possibles reste raisonnable. Dans le cas contraire, $P(\boldsymbol{X}_B = \boldsymbol{x}_B)$ est calculé par une nouvelle exécution de l'algorithme arrière-avant, en considérant cette fois que toutes les variables aléatoires n'appartenant pas à $\boldsymbol{X}_B$ sont cachées. Ce raisonnement ne s'applique, a priori, que si ces dernières sont à valeurs discrètes. À ce point de l'exposé, par conséquent, le problème du calcul de probabilités n'est pas entièrement résolu. En particulier, si A est un sous-ensemble des variables aléatoires observées à valeurs continues, nous n'avons pas abordé le problème du calcul de $P(\boldsymbol{Y}_A = \boldsymbol{y}_A)$. Ce problème ayant un lien avec l'implémentation de la validation croisée, il sera traité au chapitre 3 dédié à la sélection de modèle.

L'intérêt de nos algorithmes, du point de vue de l'interprétation, est illustré par la section 2.4.9. Dans ce contexte, on souhaite simuler $\boldsymbol{S}$ suivant la loi $P(\boldsymbol{S} = \boldsymbol{s}|\boldsymbol{Y} = \boldsymbol{y})$. Si l'on souhaite utiliser l'algorithme de Jensen, Lauritzen et Olesen, ce sera souvent à la manière d'une *boîte noire*. Autrement dit on fournit, en entrée d'un logiciel implémentant cet algorithme, la valeur de variables aléatoires et on obtient en sortie la probabilité que ces variables aléatoires prennent les valeurs d'entrée – sans s'occuper de la manière dont ces probabilités sont calculées. Ceci conduit, pour le problème de simulation ci-dessus, à considérer un état caché S_1 puis, par exécution de l'algorithme d'arbre de jonction, à calculer la probabilité $P(S_1 = j|\boldsymbol{Y} = \boldsymbol{y})$ ce qui permet d'obtenir une valeur $\hat{s}_1$ par simulation. On choisit ensuite un état caché S_2 dont on calcule, *par une exécution supplémentaire de l'algorithme d'arbre de jonction*, la loi conditionnelle $P(S_2 = j|\boldsymbol{Y} = \boldsymbol{y}, S_1 = \hat{s}_1)$ de manière à obtenir une valeur simulée $\hat{s}_2$. On procède ainsi autant de fois qu'il y a d'états cachés dans le modèle, ce qui conduit à exécuter $N_{\boldsymbol{S}}$ fois l'algorithme de Jensen, Lauritzen et Olesen. Or nous avons montré dans la section 2.4.9 qu'une *unique* exécution de notre algorithme arrière-avant suffit pour simuler $\boldsymbol{S}$ suivant la loi $P(\boldsymbol{S} = \boldsymbol{s}|\boldsymbol{Y} = \boldsymbol{y})$. Ceci met en évidence le fait que dans le cas où plusieurs probabilités conditionnelles doivent être successivement calculées, l'usage d'algorithmes de type boîte noire conduit éventuellement à calculer maintes et maintes fois la même quantité sans que l'on puisse le prévoir d'avance, alors qu'un algorithme dont les calculs intermédiaires sont interprétables permet de planifier les calculs de manière à éviter les redondances.

Enfin, les algorithmes que nous présentons peuvent être directement implémentés de manière à pouvoir traiter, de manière générique, tous les modèles de la famille $\mathcal{D}$. Nous avons détaillé, dans les sections 2.4.4 et 2.4.6, les calculs effectués par les algorithmes arrière-avant dans le cas d'arbres de Markov cachés. De même, dans la section 2.5.3, nous détaillerons, pour le même modèle, les calculs effectués par l'algorithme de Viterbi pour la restauration des états cachés par le principe du MAP. Enfin, concernant l'estimation des paramètres de ce modèle par l'algorithme EM, nous avons donné dans la section 2.3.6 les formules de réestimation à chaque itération de l'algorithme. Mais il est important de comprendre que ces sections ont uniquement une vocation pédagogique. Une personne qui souhaite implémenter ces algorithmes n'aura aucun besoin de développer les calculs de ces quatre sections ; il lui suffira d'implémenter l'algorithme générique, de dessiner le graphe d'indépendance conditionnelle et d'entrer la valeur des données observées (comme cela se fait déjà avec certains logiciels d'identification dans les modèles graphiques, comme la boîte à outils de MATLAB intitulée *réseaux bayésiens* – en anglais *Bayes Net Toolbox*[4]). En sortie d'un tel logiciel fondé sur nos calculs, il obtiendra une paramétrisation du modèle ayant interprétation probabiliste (probabilités de transition ou d'émission), un estimateur des paramètres obtenu par l'algorithme EM ou ses variantes et, s'il le souhaite, le détail de tous les calculs analytiques effectués, le tracé de l'arbre des cliques et l'interprétation probabiliste et graphique des quantités intermédiaires de l'algorithme. Autrement dit, les formules analytiques arrière-avant et l'interprétation des quantités intermédiaires pour un modèle particulier de la famille $\mathcal{D}$ *ne sont pas un calcul préalable* nécessaire à la programmation d'un logiciel fondé sur les résultats de cette section : *ce sont au contraire des quantités fournies par ce logiciel.*

Il est cependant important de remarquer que notre algorithme traite certains problèmes par des méthodes sous-optimales en termes de nombre de calculs effectués. Le premier problème est, dans l'algorithme arrière-avant utilisant les probabilités de lissage, le calcul préliminaire de la loi marginale des processus $\boldsymbol{X}_{\mathcal{S}_a}$ ou même seulement la loi marginale des processus $\boldsymbol{X}_{\mathcal{U}_a^c}$, par une phase de type avant. Nous avons donné, dans la section 2.4.5, une première méthode pour effectuer ce calcul, dont la complexité est la même que celle de l'algorithme arrière-avant. Cette méthode consiste à calculer la loi marginale de chaque clique. Cependant, nous avons décrit une méthode alternative permettant de calculer uniquement la loi marginale de chaque processus $\boldsymbol{X}_{\mathcal{U}_a^c}$. Dans certains cas, cette procédure est de complexité moindre, ce qui est illustré par l'exemple des arbres de Markov cachés, dans la section 2.4.6. La difficulté est d'implémenter la deuxième méthode dans le cas général et d'en comparer la complexité avec celle du calcul de la loi marginale des cliques. Du fait que l'algorithme de calcul de la loi marginale de chaque clique est de complexité raisonnable, ce problème n'est pas un défaut majeur de notre algorithme. Le second problème est lié au calcul des quantités de normalisation dans ce même algorithme arrière-avant utilisant les probabilités de lissage. En effet, on peut parfois montrer qu'elles valent un, comme pour certains arcs de l'arbre de jonction des arbres de Markov cachés, voir section 2.4.6, et dans ce cas leur calcul est inutile. Nous ne disposons pas, pour l'instant, de manière automatique pour repérer les cas où la quantité de normalisation n'a pas besoin d'être calculée. Il est possible que cela soit le cas pour tous les arcs n'ayant qu'un seul arc adjacent dans le sens de parcours arrière, et seulement pour ces arcs, mais cela resterait à vérifier. Notons que le nombre de calculs en jeu est négligeable, dans ce cas, par rapport à la complexité globale. Toujours est-il que ces deux remarques peuvent inciter à réécrire les algorithmes génériques de la section précédente pour des modèles particuliers, afin de les optimiser, en se basant sur les calculs analytiques de cette section – qui, rappelons-le, peuvent être effectués par un logiciel implémentant nos algorithmes génériques. Dans ce cas uniquement, les formules analytiques arrière-avant et l'interprétation des quantités intermédiaires sont préalables à la programmation d'un logiciel optimisé pour un modèle particulier. Cette pratique consistant à établir les formules de calculs de probabilités pour un modèle particulier, afin par exemple d'écrire un logiciel implémentant l'algorithme EM pour ce modèle, a jusqu'ici été la plus courante. Hormis les deux raisons ci-dessus, ce procédé n'a plus lieu d'être pour les modèles de la famille $\mathcal{D}$.

[4]disponible sur `http ://www.mathtools.net/Applications/Statistics/MATLAB/index.html` et sur `http ://www.cs.berkeley.edu/ murphyk/Bayes/bnt.html`

2.5 Algorithme du MAP

L'algorithme du MAP (Maximum A Posteriori) résout le problème de la restauration des états cachés : étant donné une réalisation du processus observé $\boldsymbol{y}$, il s'agit de trouver la réalisation du processus caché $\hat{\boldsymbol{s}}$ la plus probable correspondant à $\boldsymbol{y}$, c'est-à-dire réalisant le maximum en $\boldsymbol{s}$ de la fonction $P(\boldsymbol{S} = \boldsymbol{s}|\boldsymbol{Y} = \boldsymbol{y})$. Cet algorithme calcule également la valeur P^* du maximum. De manière équivalente, $\hat{\boldsymbol{s}}$ maximise la quantité $P(\boldsymbol{S} = \boldsymbol{s}, \boldsymbol{Y} = \boldsymbol{y})$. Dans cette section, les paramètres λ du modèle sont fixés et connus ; nous notons alors $P = P_\lambda$. Parmi les applications de la restauration des états cachés figurent la classification automatique, l'interprétation du modèle (par exemple pour relier les états cachés aux différentes parties du mot prononcé en reconnaissance de parole, voir Jelinek *et al.*, 1975 [65]) et la variante CEM de EM présentée en section 2.3.5.

Une méthode de restauration du processus caché alternative au MAP consiste également à restaurer les états cachés à partir de leur valeur la plus probable, mais sur la base d'un critère local, c'est-à-dire en déterminant individuellement chaque état le plus probable

$$\hat{s}_u = \arg\max_j P(S_u = j|\boldsymbol{Y} = \boldsymbol{y}).$$

Ces probabilités (dites *probabilités de lissage*) sont obtenues par l'algorithme arrière-avant de la section 2.4, à partir des probabilités $P(\boldsymbol{S}_{\mathcal{C}} = \boldsymbol{s}_{\mathcal{C}}|\boldsymbol{Y} = \boldsymbol{y})$. Une discussion sur ces méthodes dans le cadre des chaînes de Markov cachées est disponible dans Ephraim et Mehrav, 2002 [47] et dans Fredkin et Rice, 1992 [50]. Bien que cette méthode maximise l'espérance du nombre d'états corrects, le processus caché ainsi restauré n'est pas toujours optimal au sens du MAP. Il est même possible que, dans le cas où certaines transitions sont interdites (c'est-à-dire dans le cas où $p_{\boldsymbol{a},i} = 0$ pour certains couples $(\boldsymbol{a}, i)$), le processus caché obtenu en maximisant séparément la probabilité conditionnelle de chaque état puisse être un processus invalide, *i.e.* de probabilité nulle. C'est pourquoi nous proposons la solution suivante pour le problème de restauration globale.

De même que le calcul de la vraisemblance par sommation sur toutes les valeurs possibles des états cachés est généralement infaisable, vu le nombre d'opérations en jeu, le calcul du MAP par maximisation sur toutes les réalisations possibles du processus caché est infaisable. L'algorithme de Dawid (voir Smyth, Heckerman et Jordan, 1997 [110]) permet le calcul numérique mais non analytique de ces probabilités, de manière inductive, par un algorithme de complexité polynomiale. Il s'agit essentiellement de l'algorithme d'arbre de jonction où l'équation (1.4) est remplacée par l'équation

$$\psi_S^*(\boldsymbol{x}_S) = \max_{\boldsymbol{x}_{C\setminus S} \in C\setminus S} \psi_C(\boldsymbol{x}_C).$$

De même que l'algorithme de jonction, l'algorithme de Dawid n'est pas adapté à la paramétrisation naturelle des modèles graphiques probabilistes orientés acycliques et souffre d'un manque d'interprétation de ses calculs intermédiaires ainsi que d'instabilité numérique. Notons qu'en général, ces défauts ne sont pas aussi pénalisant que dans le cadre du calcul de probabilités, car en général les calculs intermédiaires n'ont pas d'utilité. Les problèmes d'instabilité numérique sont simplement résolus par le calcul de $\log(\psi_S^*)$ au lieu de ψ_S^*, rendu possible par l'absence de sommation dans les formules de propagation.

Dans cette section, nous rappelons le principe de l'algorithme de Viterbi, qui est l'algorithme du MAP pour les chaînes de Markov cachées. Cet algorithme ne peut être directement adapté aux modèles de la famille $\mathcal{D}$, c'est pourquoi nous en proposons une variante, construite à partir d'une récursion *arrière*. Celle-ci sert ensuite de base à l'algorithme du MAP pour les modèles plus généraux de la famille $\mathcal{D}$. Bien que l'algorithme de Dawid soit utilisable, notre algorithme a l'intérêt d'être plus facile d'interprétation car ses quantités intermédiaires ont une signification probabiliste et car il explicite le rôle des paramètres du modèle. De plus, il admet une justification claire et intuitive, donnée ci-après. Il est analogue à l'algorithme arrière de la section 2.4.2. Enfin, nous montrons comment notre algorithme générique s'applique aux arbres de Markov cachés, ce qui permet de retrouver l'algorithme de Viterbi développé par Durand, Gonçalvès et Guédon, 2002 [46].

2.5.1 Introduction : l'algorithme de Viterbi dans les chaînes de Markov cachées

Le premier algorithme de restauration globale dans le cas de processus à structure cachée est dû à Viterbi, 1967 [117]. À l'origine, l'algorithme de Viterbi est dédié à l'analyse de processus de Markov bruités, à bruit sans mémoire. La justification de cet algorithme est disponible dans Forney, 1973 [49] et s'appuie sur la théorie des graphes. L'auteur prouve que la maximisation de $P(\boldsymbol{S} = \boldsymbol{s}|\boldsymbol{Y} = \boldsymbol{y})$ est équivalente à la recherche du plus court chemin dans un graphe à arêtes pondérées. Sans doute cette preuve pourrait-elle être adaptée au contexte des modèles de Markov cachés de la famille $\mathcal{D}$ mais cela ne contribuerait pas à fournir une interprétation aux quantités intermédiaires de l'algorithme. L'algorithme de Viterbi est un algorithme de programmation dynamique, c'est-à-dire une méthode de résolution de problèmes d'optimisation qui repose sur une propriété de décomposabilité de la fonction à optimiser.

Dans le cas d'une chaîne de Markov cachée $(Y_1, \dots, Y_n) = \boldsymbol{Y}_1^n$, du fait que le processus caché $(S_1, \dots, S_n) = \boldsymbol{S}_1^n$ est une chaîne de Markov, on a pour tout t (Jelinek, 1997 [64]) la factorisation

$$\begin{aligned} &\max_{s_1,\dots,s_n} P(\boldsymbol{S}_1^n = \boldsymbol{s}_1^n, \boldsymbol{Y}_1^n = \boldsymbol{y}_1^n) \\ &\quad = \max_{s_t}\{ \max_{s_{t+1},\dots,s_n} (\boldsymbol{Y}_{t+1}^n = \boldsymbol{y}_{t+1}^n, \boldsymbol{S}_{t+1}^n = \boldsymbol{s}_{t+1}^n | S_t = s_t) \\ &\quad\quad \times \max_{s_1,\dots,s_{t-1}} P(\boldsymbol{S}_1^t = \boldsymbol{s}_1^t, \boldsymbol{Y}_1^t = \boldsymbol{y}_1^t)\}. \end{aligned} \tag{2.59}$$

Cette factorisation permet de construire la récursion usuelle de type avant due à Viterbi, basée sur les probabilités $P(\boldsymbol{S}_1^t = \boldsymbol{s}_1^t, \boldsymbol{Y}_1^t = \boldsymbol{y}_1^t)$. La factorisation alternative suivante est également valide

$$\begin{aligned} &\max_{s_1,\dots,s_n} P(\boldsymbol{S}_1^n = \boldsymbol{s}_1^n, \boldsymbol{Y}_1^n = \boldsymbol{y}_1^n) \\ &\quad = \max_{s_t}\{ \max_{s_{t+1},\dots,s_n} P(\boldsymbol{Y}_t^n = \boldsymbol{y}_t^n, \boldsymbol{S}_{t+1}^n = \boldsymbol{s}_{t+1}^n | S_t = s_t) \\ &\quad\quad \times \max_{s_1,\dots,s_{t-1}} P(\boldsymbol{S}_1^t = \boldsymbol{s}_1^t, \boldsymbol{Y}_1^{t-1} = \boldsymbol{y}_1^{t-1})\} \ ; \end{aligned} \tag{2.60}$$

elle permettra une récursion de type arrière basée sur les probabilités $P(\boldsymbol{Y}_t^n = \boldsymbol{y}_t^n, \boldsymbol{S}_{t+1}^n = \boldsymbol{s}_{t+1}^n | S_t = s_t)$. Posons alors, pour tout $t \in \{1, \ldots, n\}$ et tout $j \in \{1, \ldots, K\}$ où K est le nombre d'états cachés :

$$\begin{aligned} \tilde{\delta}_t(j) &= \max_{s_1,\ldots,s_{t-1}} P(S_t = j, \boldsymbol{S}_1^{t-1} = \boldsymbol{s}_1^{t-1}, \boldsymbol{Y}_1^t = \boldsymbol{y}_1^t) \\ \text{et} \quad \delta_t(j) &= \max_{s_{t+1},\ldots,s_n} P(\boldsymbol{Y}_t^n = \boldsymbol{y}_t^n, \boldsymbol{S}_{t+1}^n = \boldsymbol{s}_{t+1}^n | S_t = j), \end{aligned}$$

avec la convention $\delta_n(j) = P(Y_n = y_n | S_n = j)$. La décomposition (2.59) se réécrit donc

$$\begin{aligned} &\max_{s_1,\ldots,s_n} P(\boldsymbol{S}_1^n = \boldsymbol{s}_1^n, \boldsymbol{Y}_1^n = \boldsymbol{y}_1^n) \\ &= \max_j \{ \max_{s_{t+1},\ldots,s_n} P(\boldsymbol{Y}_{t+1}^n = \boldsymbol{y}_{t+1}^n, \boldsymbol{S}_{t+1}^n = \boldsymbol{s}_{t+1}^n | S_t = j) \tilde{\delta}_t(j) \}, \end{aligned} \tag{2.61}$$

tandis que la décomposition (2.60) se réécrit

$$\begin{aligned} &\max_{s_1,\ldots,s_n} P(\boldsymbol{S}_1^n = \boldsymbol{s}_1^n, \boldsymbol{Y}_1^n = \boldsymbol{y}_1^n) \\ &= \max_j \{ \delta_t(j) \max_{s_1,\ldots,s_{t-1}} P(S_t = j, \boldsymbol{S}_1^{t-1} = \boldsymbol{s}_1^{t-1}, \boldsymbol{Y}_1^{t-1} = \boldsymbol{y}_1^{t-1}) \}. \end{aligned}$$

Ainsi, il est possible d'utiliser les quantités $\tilde{\delta}_t(j)$ pour développer la récursion avant classique de l'algorithme de Viterbi pour les chaînes de Markov cachées, initialisée pour $t = 1$ par

$$\begin{aligned} \tilde{\delta}_1(j) &= P(Y_1 = y_1 | S_1 = j) P(S_1 = j) \\ &= P_{\theta_j}(y_1) \pi_j. \end{aligned}$$

Puis, pour chacune des valeurs de t dans $\{2, \ldots, n\}$, prises dans l'ordre croissant, la récursion s'écrit comme suit

$$\begin{aligned} \tilde{\delta}_t(j) &= \max_{s_1,\ldots,s_{t-1}} P(S_t = j, \boldsymbol{S}_1^{t-1} = \boldsymbol{s}_1^{t-1}, \boldsymbol{Y}_1^t = \boldsymbol{y}_1^t) \\ &= P(Y_t = y_t | S_t = j) \max_i \{ P(S_t = j | S_{t-1} = i) \\ &\qquad \times \max_{s_1,\ldots,s_{t-2}} P(S_{t-1} = i, \boldsymbol{S}_1^{t-2} = \boldsymbol{s}_1^{t-2}, \boldsymbol{Y}_1^{t-1} = \boldsymbol{y}_1^{t-1}) \} \\ &= P_{\theta_j}(y_t) \max_i \{ p_{ij} \tilde{\delta}_{t-1}(i) \}. \end{aligned} \tag{2.62}$$

De plus, pour pouvoir restaurer les états cachés lors d'une phase ultérieure de *recherche rétrograde*[5], on est amené à stocker la quantité

$$\tilde{\psi}_t(j) = \arg\max_i \{ P_{\theta_j}(y_t) p_{ij} \tilde{\delta}_{t-1}(i) \} = \arg\max_i \{ p_{ij} \tilde{\delta}_{t-1}(i) \},$$

qui correspond à la valeur optimale de l'état caché à l'instant $t-1$ si l'état optimal à t est $\hat{s}_t = j$ (voir équation (2.63) ci-dessous.)

L'état final est donné par $\hat{s}_n = \arg\max_j \tilde{\delta}_n(j)$ et la probabilité jointe de la séquence observée $\boldsymbol{y}_1^n$ et de la séquence cachée optimale associée est $P^* = \max_j \tilde{\delta}_n(j)$. On retrouve

[5] *backtracking* en anglais.

l'état optimal $\hat{s}_t$, quand la séquence optimale $\hat{\boldsymbol{s}}_{t+1}^n$ est connue, en utilisant la factorisation (2.61) de la probabilité jointe maximale et la formule de récursion (2.62), qui donnent :

$$\begin{aligned}
\hat{s}_t &= \arg\max_j \{P(\boldsymbol{Y}_{t+2}^n = \boldsymbol{y}_{t+2}^n, \boldsymbol{S}_{t+2}^n = \hat{\boldsymbol{s}}_{t+2}^n | S_{t+1} = \hat{s}_{t+1}) P_{\theta_{\hat{s}_{t+1}}}(y_t) p_{j\hat{s}_{t+1}} \tilde{\delta}_t(j)\} \\
&= \arg\max_j \{P_{\theta_{\hat{s}_{t+1}}}(y_t) p_{j\hat{s}_{t+1}} \tilde{\delta}_t(j)\} \qquad (2.63) \\
&= \tilde{\psi}_{t+1}(\hat{s}_{t+1}).
\end{aligned}$$

Les équations ci-dessus définissent l'algorithme de recherche rétrograde pour la restauration des états cachés.

Le problème de la récursion de Viterbi usuelle est qu'elle n'est pas adaptable aux modèles de Markov cachés de la famille $\mathcal{D}$. Pour en comprendre la raison, rappelons que le calcul de la vraisemblance, dans les chaînes de Markov cachées, se fait soit par une récursion avant, qui part de la source à $t = 1$ et finit à $t = n$, soit par une récursion arrière, qui part de $t = n$ et finit à $t = 1$. Dans les modèles de la famille $\mathcal{D}$, la récursion avant fait appel aux résultats de la récursion arrière, c'est pourquoi le calcul de la vraisemblance ne peut être réalisée qu'à partir de la récursion arrière. De la même manière, l'algorithme de Viterbi ne peut se faire, dans ces modèles, que sur la base d'une récursion de type arrière.

C'est pourquoi nous montrons ci-dessous comment les quantités $\delta_t(j)$ peuvent être utilisées pour développer un nouvel algorithme de Viterbi pour les chaînes de Markov cachées basé sur une récursion arrière. C'est cette variante qui pourra être adaptée aux modèles plus généraux de la famille $\mathcal{D}$. Cet algorithme alternatif est initialisé à l'instant $t = n$ par

$$\begin{aligned}
\delta_n(j) &= P(Y_n = y_n | S_n = j) \\
&= P_{\theta_j}(y_n).
\end{aligned}$$

La récursion arrière est donnée, pour chacune des valeurs de t dans $\{2, \ldots, n\}$ prises dans l'ordre décroissant, par

$$\begin{aligned}
\delta_t(j) &= \max_{s_{t+1},\ldots,s_n} P(\boldsymbol{Y}_t^n = \boldsymbol{y}_t^n, \boldsymbol{S}_{t+1}^n = \boldsymbol{s}_{t+1}^n | S_t = j) \\
&= \max_k \{ \max_{s_{t+2},\ldots,s_n} P(\boldsymbol{Y}_{t+1}^n = \boldsymbol{y}_{t+1}^n, \boldsymbol{S}_{t+2}^n = \boldsymbol{s}_{t+2}^n | S_{t+1} = k) \\
&\qquad \times P(S_{t+1} = k | S_t = j)\} P(Y_t = y_t | S_t = j) \\
&= \max_k \{\delta_{t+1}(k) p_{jk}\} P_{\theta_j}(y_t). \qquad (2.64)
\end{aligned}$$

La restauration ultérieure des états cachés nécessite le calcul de la quantité

$$\psi_t(j) = \arg\max_k \{\delta_{t+1}(k) p_{jk}\}$$

qui correspond à la valeur optimale de l'état à l'instant $t + 1$ si l'état optimal à t est $\hat{s}_t = j$.

On obtient, pour $t = 1$,

$$\delta_1(j) = \max_{s_2,\ldots,s_n} P(\boldsymbol{Y}_1^n = \boldsymbol{y}_1^n, \boldsymbol{S}_2^n = \boldsymbol{s}_2^n | S_1 = j).$$

Par conséquent, la probabilité de la séquence cachée optimale associée à la séquence observée $\boldsymbol{y}_1^n$ est

$$\begin{aligned} P^* &= \max_j \{ \max_{s_2,\ldots,s_n} P(\boldsymbol{Y}_1^n = \boldsymbol{y}_1^n, \boldsymbol{S}_2^n = \boldsymbol{s}_2^n | S_1 = j) P(S_1 = j) \} \\ &= \max_j \{ \delta_1(j) \pi_j \}. \end{aligned}$$

L'état optimal à $t = 1$ est donné par $\hat{s}_1 = \arg\max_j \{\delta_1(j)\pi_j\}$. La séquence d'états optimale est alors extraite par la procédure de recherche avant suivante : pour tout $t \in \{2, \ldots, n\}$,

$$\hat{s}_t = \psi_{t-1}(\hat{s}_{t-1}).$$

2.5.2 Algorithme du MAP dans le cas général

Nous supposons dans ce qui suit que le processus observé $\boldsymbol{y}$ est fixé. De même que la phase arrière de la section 2.4.2, l'algorithme du MAP est un parcours de graphe qui part des feuilles de l'arbre des cliques et remonte jusqu'à la racine $\mathcal{C}_0$, parcourant ainsi les arcs de l'arbre en sens contraire de leur orientation. Elle utilise les quantités δ_a définies pour tout arc $a = (\mathcal{C}_i, \mathcal{C}_j)$ de l'arbre de jonction $\mathcal{T}$ (la clique $\mathcal{C}_i$ étant donc située sur le chemin entre $\mathcal{C}_0$ et $\mathcal{C}_j$ dans $\mathcal{T}$) par

$$\delta_a(\boldsymbol{s}_{\mathcal{S}_a}) = \max_{\boldsymbol{s}_{\mathcal{K}_a}} P(\boldsymbol{Y}_{\mathcal{K}_a} = \boldsymbol{y}_{\mathcal{K}_a}, \boldsymbol{S}_{\mathcal{K}_a} = \boldsymbol{s}_{\mathcal{K}_a} | \boldsymbol{Y}_{\mathcal{S}_a} = \boldsymbol{y}_{\mathcal{S}_a}, \boldsymbol{S}_{\mathcal{S}_a} = \boldsymbol{s}_{\mathcal{S}_a}).$$

Initialisation

Soit $\mathcal{C}_j$ une clique feuille de l'arbre de jonction, donc reliée à une unique clique $\mathcal{C}_i$ par un arc a de l'arbre de jonction. Par définition des quantités $\delta_a(\boldsymbol{s}_{\mathcal{S}_a})$, l'algorithme est initialisé en calculant $\max_{\boldsymbol{s}_{\mathcal{C}_j \setminus \mathcal{S}_a}} P(\boldsymbol{Y}_{\mathcal{C}_j \setminus \mathcal{S}_a} = \boldsymbol{y}_{\mathcal{C}_j \setminus \mathcal{S}_a}, \boldsymbol{S}_{\mathcal{C}_j \setminus \mathcal{S}_a} = \boldsymbol{s}_{\mathcal{C}_j \setminus \mathcal{S}_a} | \boldsymbol{Y}_{\mathcal{S}_a} = \boldsymbol{y}_{\mathcal{S}_a}, \boldsymbol{S}_{\mathcal{S}_a} = \boldsymbol{s}_{\mathcal{S}_a})$ ce qui est donné, comme pour l'initialisation des quantités $\tilde{\beta}_a(\boldsymbol{s}_{\mathcal{S}_a})$, par la formule (2.22) ou (2.23) suivant que $\mathcal{U}_a = \emptyset$ ou non. On obtient donc

$$\delta_a(\boldsymbol{s}_{\mathcal{S}_a}) = \begin{cases} \max_{\boldsymbol{s}_{\mathcal{C}_j \setminus \mathcal{S}_a}} \dfrac{\prod_{u \in \tilde{\mathrm{An}}(\mathcal{U}_a)} \lambda_{\boldsymbol{x}_{\mathrm{pa}(u)}, x_u}}{\sum_{\boldsymbol{x}_{\mathcal{C}_j \setminus \mathcal{S}_a}} \prod_{u \in \tilde{\mathrm{An}}(\mathcal{U}_a)} \lambda_{\boldsymbol{x}_{\mathrm{pa}(u)}, x_u}} & \text{si } \mathcal{U}_a \neq \emptyset \,; \\ \max_{\boldsymbol{s}_{\mathcal{C}_j \setminus \mathcal{S}_a}} \prod_{u \in \tilde{\mathrm{An}}(\mathcal{U}_a)} \lambda_{\boldsymbol{x}_{\mathrm{pa}(u)}, x_u} & \text{si } \mathcal{U}_a = \emptyset. \end{cases} \tag{2.65}$$

De même que dans le cas des chaînes de Markov cachées, la restauration ultérieure des états cachés nécessite le stockage de la quantité, notée $\psi_a(\boldsymbol{s}_{\mathcal{S}_a})$, réalisant le maximum en $\boldsymbol{s}_{\mathcal{C}_j \setminus \mathcal{S}_a}$ de l'expression (2.65).

Propagation

Soit un arc $a = (\mathcal{C}_i, \mathcal{C}_j)$ de l'arbre de jonction. De même que dans la phase arrière vue en section 2.4.2, les quantités $\delta_{a_{jl}}(\boldsymbol{s}_{\mathcal{S}_{a_{jl}}})$ sont supposées connues pour tous les arcs

a_{j_l} incidents à $\mathcal{C}_j$ autres que l'arc a (qui est le seul menant vers $\mathcal{C}_0$) et pour toutes les valeurs possibles de $\boldsymbol{s}_{\mathcal{S}_{a_{j_l}}} \in \mathcal{S}_{\mathcal{S}_{a_{j_l}}}$. Nous rappelons que les valeurs de $\boldsymbol{Y}$ sont supposées connues, fixées à $\boldsymbol{y}$. Nous désirons alors calculer $\delta_a(\boldsymbol{s}_{\mathcal{S}_a})$ à partir des $\delta_{a_{j_l}}(\boldsymbol{s}_{\mathcal{S}_{a_{j_l}}})$. Pour cela, nous utilisons successivement chaque séparateur $\mathcal{S}_{a_{j_l}}$, à commencer par $\mathcal{S}_{a_{j_1}}$ par exemple, pour factoriser la probabilité jointe $P(\boldsymbol{Y}_{\mathcal{K}_a} = \boldsymbol{y}_{\mathcal{K}_a}, \boldsymbol{S}_{\mathcal{K}_a} = \boldsymbol{s}_{\mathcal{K}_a}, \boldsymbol{Y}_{\mathcal{S}_a} = \boldsymbol{y}_{\mathcal{S}_a}, \boldsymbol{S}_{\mathcal{S}_a} = \boldsymbol{s}_{\mathcal{S}_a})$ de manière à obtenir une équation analogue à (2.18)

$$\begin{aligned}
&P(\boldsymbol{Y}_{\mathcal{K}_a} = \boldsymbol{y}_{\mathcal{K}_a}, \boldsymbol{Y}_{\mathcal{S}_a} = \boldsymbol{y}_{\mathcal{S}_a}, \boldsymbol{S}_{\mathcal{K}_a} = \boldsymbol{s}_{\mathcal{K}_a}, \boldsymbol{S}_{\mathcal{S}_a} = \boldsymbol{s}_{\mathcal{S}_a}) \\
&\quad = P(\boldsymbol{Y}_{\mathcal{K}_{a_{j_1}}} = \boldsymbol{y}_{\mathcal{K}_{a_{j_1}}}, \boldsymbol{S}_{\mathcal{K}_{a_{j_1}}} = \boldsymbol{s}_{\mathcal{K}_{a_{j_1}}} | \boldsymbol{Y}_{\mathcal{S}_{a_{j_1}}} = \boldsymbol{y}_{\mathcal{S}_{a_{j_1}}}, \boldsymbol{S}_{\mathcal{S}_{a_{j_1}}} = \boldsymbol{s}_{\mathcal{S}_{a_{j_1}}}) \\
&\quad \times P(\bigcap_{l \neq 1} \{\boldsymbol{Y}_{\bar{\mathcal{K}}_{a_{j_l}}} = \boldsymbol{y}_{\bar{\mathcal{K}}_{a_{j_l}}}\} \bigcap_{l \neq 1} \{\boldsymbol{S}_{\bar{\mathcal{K}}_{a_{j_l}}} = \boldsymbol{s}_{\bar{\mathcal{K}}_{a_{j_l}}}\} \cap \{\boldsymbol{Y}_{\mathcal{C}_j} = \boldsymbol{y}_{\mathcal{C}_j}\} \cap \{\boldsymbol{S}_{\mathcal{C}_j} = \boldsymbol{s}_{\mathcal{C}_j}\}).
\end{aligned}$$

En répétant le raisonnement ci-dessus pour chacun des séparateurs $\mathcal{S}_{a_{j_l}}$ puis en conditionnant par $\{\boldsymbol{Y}_{\mathcal{S}_a} = \boldsymbol{y}_{\mathcal{S}_a}\} \cap \{\boldsymbol{S}_{\mathcal{S}_a} = \boldsymbol{s}_{\mathcal{S}_a}\}$, on obtient la factorisation

$$\begin{aligned}
&P(\boldsymbol{X}_{\mathcal{K}_a} = \boldsymbol{x}_{\mathcal{K}_a} | \boldsymbol{X}_{\mathcal{S}_a} = \boldsymbol{x}_{\mathcal{S}_a}) \\
&\quad = P(\boldsymbol{X}_{\mathcal{C}_j \setminus \mathcal{S}_a} = \boldsymbol{x}_{\mathcal{C}_j \setminus \mathcal{S}_a} | \boldsymbol{X}_{\mathcal{S}_a} = \boldsymbol{x}_{\mathcal{S}_a}) \prod_l P(\boldsymbol{X}_{\mathcal{K}_{a_{j_l}}} = \boldsymbol{x}_{\mathcal{K}_{a_{j_l}}} | \boldsymbol{X}_{\mathcal{S}_{a_{j_l}}} = \boldsymbol{x}_{\mathcal{S}_{a_{j_l}}})
\end{aligned} \tag{2.66}$$

En remarquant que $\{\{\mathcal{K}_{a_{j_l}}\}_{l=1,2,\ldots}, (\mathcal{C}_j \setminus \mathcal{S}_a)\}$ est une partition de $\mathcal{K}_a$, on obtient

$$\begin{aligned}
\delta_a(\boldsymbol{s}_a) &= \max_{\boldsymbol{s}_{\mathcal{K}_a}} P(\boldsymbol{Y}_{\mathcal{K}_a} = \boldsymbol{y}_{\mathcal{K}_a}, \boldsymbol{S}_{\mathcal{K}_a} = \boldsymbol{s}_{\mathcal{K}_a} | \boldsymbol{Y}_{\mathcal{S}_a} = \boldsymbol{y}_{\mathcal{S}_a}, \boldsymbol{S}_{\mathcal{S}_a} = \boldsymbol{s}_{\mathcal{S}_a}) \\
&= \max_{\boldsymbol{s}_{\mathcal{C}_j \setminus \mathcal{S}_a}} \max_{\boldsymbol{s}_{\mathcal{K}_{a_{j_1}}}} \ldots \max_{\boldsymbol{s}_{\mathcal{K}_{a_{j_l}}}} \Bigg[P(\boldsymbol{X}_{\mathcal{C}_j \setminus \mathcal{S}_a} = \boldsymbol{x}_{\mathcal{C}_j \setminus \mathcal{S}_a} | \boldsymbol{X}_{\mathcal{S}_a} = \boldsymbol{x}_{\mathcal{S}_a}) \\
&\qquad \times \prod_l P(\boldsymbol{X}_{\mathcal{K}_{a_{j_l}}} = \boldsymbol{x}_{\mathcal{K}_{a_{j_l}}} | \boldsymbol{X}_{\mathcal{S}_{a_{j_l}}} = \boldsymbol{x}_{\mathcal{S}_{a_{j_l}}}) \Bigg] \\
&= \max_{\boldsymbol{s}_{\mathcal{C}_j \setminus \mathcal{S}_a}} \Bigg[P(\boldsymbol{X}_{\mathcal{C}_j \setminus \mathcal{S}_a} = \boldsymbol{x}_{\mathcal{C}_j \setminus \mathcal{S}_a} | \boldsymbol{X}_{\mathcal{S}_a} = \boldsymbol{x}_{\mathcal{S}_a}) \prod_l \max_{\boldsymbol{s}_{\mathcal{K}_{a_{j_l}}}} P(\boldsymbol{X}_{\mathcal{K}_{a_{j_l}}} = \boldsymbol{x}_{\mathcal{K}_{a_{j_l}}} | \boldsymbol{X}_{\mathcal{S}_{a_{j_l}}} = \boldsymbol{x}_{\mathcal{S}_{a_{j_l}}}) \Bigg] \\
&= \max_{\boldsymbol{s}_{\mathcal{C}_j \setminus \mathcal{S}_a}} \Bigg[P(\boldsymbol{Y}_{\mathcal{C}_j \setminus \mathcal{S}_a} = \boldsymbol{y}_{\mathcal{C}_j \setminus \mathcal{S}_a}, \boldsymbol{S}_{\mathcal{C}_j \setminus \mathcal{S}_a} = \boldsymbol{s}_{\mathcal{C}_j \setminus \mathcal{S}_a} | \boldsymbol{Y}_{\mathcal{S}_a} = \boldsymbol{y}_{\mathcal{S}_a}, \boldsymbol{S}_{\mathcal{S}_a} = \boldsymbol{s}_{\mathcal{S}_a}) \\
&\qquad \times \prod_l \delta_{a_{j_l}}(\boldsymbol{s}_{\mathcal{S}_{a_{j_l}}}) \Bigg]
\end{aligned} \tag{2.67}$$

où la quantité $P(\boldsymbol{Y}_{\mathcal{C}_j \setminus \mathcal{S}_a} = \boldsymbol{y}_{\mathcal{C}_j \setminus \mathcal{S}_a}, \boldsymbol{S}_{\mathcal{C}_j \setminus \mathcal{S}_a} = \boldsymbol{s}_{\mathcal{C}_j \setminus \mathcal{S}_a} | \boldsymbol{Y}_{\mathcal{S}_a} = \boldsymbol{y}_{\mathcal{S}_a}, \boldsymbol{S}_{\mathcal{S}_a} = \boldsymbol{s}_{\mathcal{S}_a})$ est donnée en fonction des paramètres du modèle par l'équation (2.22) ou l'équation (2.23), suivant que $\mathcal{U}_a = \emptyset$ ou non, et se calcule de manière inductive, comme dans l'algorithme arrière de la section 2.4.2.

De même que dans notre algorithme de Viterbi pour les chaînes de Markov cachées (défini par l'équation de propagation (2.64)), la restauration ultérieure des états cachés se fait par une procédure de recherche avant. Celle-ci nécessite le stockage de la quantité

$$\begin{aligned}\psi_a(\boldsymbol{s}_a) \quad = \quad \arg \max_{\boldsymbol{s}_{\mathcal{C}_j \setminus \mathbb{S}_a}} \Big[& P(\boldsymbol{Y}_{\mathcal{C}_j \setminus \mathbb{S}_a} = \boldsymbol{y}_{\mathcal{C}_j \setminus \mathbb{S}_a}, \boldsymbol{S}_{\mathcal{C}_j \setminus \mathbb{S}_a} = \boldsymbol{s}_{\mathcal{C}_j \setminus \mathbb{S}_a} | \boldsymbol{Y}_{\mathbb{S}_a} = \boldsymbol{y}_{\mathbb{S}_a}, \boldsymbol{S}_{\mathbb{S}_a} = \boldsymbol{s}_{\mathbb{S}_a}) \\ & \times \prod_l \delta_{a_{j_l}}(\boldsymbol{s}_{\mathbb{S}_{a_{j_l}}}) \Big],\end{aligned}$$

qui correspond à la valeur optimale de $\boldsymbol{s}_{\mathcal{C}_j \setminus \mathbb{S}_a}$ quand le séparateur de clique optimal est $\hat{\boldsymbol{s}}_a = \boldsymbol{s}_a$.

Terminaison

La valeur P^* est obtenue lorsque tous les arcs de l'arbre de jonction ont été parcourus. Nous notons $a_1, \ldots, a_l$ les arcs incidents à $\mathcal{C}_0$ dans l'arbre de jonction. Le raisonnement ayant conduit à la factorisation (2.66) est alors adapté pour obtenir :

$$P^* = \max_{\boldsymbol{s} \in \mathcal{S}} P(\boldsymbol{Y} = \boldsymbol{y}, \boldsymbol{S} = \boldsymbol{s}) = \max_{\boldsymbol{s}_{\mathcal{C}_0} \in \mathcal{S}_{\mathcal{C}_0}} \left\{ P(\boldsymbol{Y}_{\mathcal{C}_0} = \boldsymbol{y}_{\mathcal{C}_0}, \boldsymbol{S}_{\mathcal{C}_0} = \boldsymbol{s}_{\mathcal{C}_0}) \prod_l \delta_{a_l}(\boldsymbol{s}_{\mathbb{S}_{a_l}}) \right\} \quad (2.68)$$

où

$$P(\boldsymbol{X}_{\mathcal{C}_0} = \boldsymbol{x}_{\mathcal{C}_0}) = P(X_0^{(1)}) \prod_{i>1} P(X_0^{(i)} | X_0^{(1)}, \ldots, X_0^{(i-1)})$$

et où les probabilités $P(X_0^{(i)} = x_0^{(i)} | X_0^{(1)} = x_0^{(1)}, \ldots, X_0^{(i-1)} = x_0^{(i-1)})$ et $P(X_0^{(1)})$ sont données par les paramètres du modèle, vu que $\mathrm{pa}(X_0^{(i)}) = \{X_0^{(1)}, \ldots, X_0^{(i-1)}\}$ pour $i > 1$.

Restauration des états cachés

La restauration des états cachés se fait de manière classique, comme dans l'algorithme de Viterbi (voir section 2.5.1), c'est-à-dire par un parcours de type avant de l'arbre des cliques. La phase de restauration est initialisée par

$$\hat{\boldsymbol{s}}_{\mathcal{C}_0} = \arg \max_{\boldsymbol{s}_{\mathcal{C}_0} \in \mathcal{S}_{\mathcal{C}_0}} \left\{ P(\boldsymbol{Y}_{\mathcal{C}_0} = \boldsymbol{y}_{\mathcal{C}_0}, \boldsymbol{S}_{\mathcal{C}_0} = \boldsymbol{s}_{\mathcal{C}_0}) \prod_l \delta_{a_l}(\boldsymbol{s}_{\mathbb{S}_{a_l}}) \right\}.$$

Les autres états cachés optimaux sont donnés, pour tous les arcs $a = (\mathcal{C}_i, \mathcal{C}_j)$ de l'arbre de jonction, parcourus dans le sens *avant*, par la valeur de $\boldsymbol{s}_{\mathcal{C}_j \setminus \mathbb{S}_a}$ maximisant l'expression (2.67). Ainsi, connaissant la valeur optimale $\hat{\boldsymbol{s}}_{\mathbb{S}_a}$, on en déduit la valeur optimale de $\boldsymbol{s}_{\mathcal{C}_j \setminus \mathbb{S}_a}$ par

$$\hat{\boldsymbol{s}}_{\mathcal{C}_j \setminus \mathbb{S}_a} = \psi_a(\hat{\boldsymbol{s}}_{\mathbb{S}_a}).$$

La complexité totale de l'algorithme est identique à celle de la phase arrière de l'algorithme de la section 2.4.2, c'est-à-dire d'ordre $\mathcal{O}(\sum_{C \in \mathcal{V}_\mathcal{G}} \mathrm{taille}(C))$.

Interprétation a posteriori

La formule d'initialisation (2.65) de notre algorithme de Viterbi est analogue à la formule d'initialisation (2.24) de notre algorithme arrière (voir section 2.4.2) où les sommes sont remplacées par des maximisations. De même, la formule de propagation de l'algorithme de Viterbi (2.67) est similaire à celle de la phase arrière (2.20) à ceci près que les sommes sont remplacées par des maximisations. La même comparaison peut être établie pour les formules de terminaison (2.68) et (2.25), respectivement pour l'algorithme de Viterbi et la phase arrière de la section 2.4.2.

Il s'ensuit que du point de vue de la programmation, l'algorithme de Viterbi est obtenu à partir de l'algorithme arrière de la section 2.4.2 en remplaçant les sommes par des maximisations dans les formules d'initialisation, de propagation et de terminaison.

Stabilité numérique

De par sa nature même, l'algorithme du MAP calcule une probabilité jointe du type $P(\boldsymbol{Y} = \boldsymbol{y}, \boldsymbol{S} = \boldsymbol{s})$ qui tend vers 0 exponentiellement vite quand le nombre de variables aléatoires observées n tend vers $+\infty$. Ceci rend impossible son implémentation directe pour des raisons de limitations liées à la représentation des réels proches de zéro. La construction d'un algorithme basé sur des probabilités conditionnelles (c'est-à-dire sur le même principe que l'algorithme arrière-avant utilisant les probabilités de lissage de la section 2.4.5) est aisé. Cet algorithme calcule directement $P(\boldsymbol{S} = \boldsymbol{s}|\boldsymbol{Y} = \boldsymbol{y})$ au lieu de $P(\boldsymbol{Y} = \boldsymbol{y}, \boldsymbol{S} = \boldsymbol{s})$; cependant, en général, le nombre de variables aléatoires cachées d'un modèle de Markov caché est fonction croissante du nombre de variables aléatoires observées. Par conséquent, $P(\boldsymbol{S} = \boldsymbol{s}|\boldsymbol{Y} = \boldsymbol{y})$ tend également vers 0 exponentiellement vite quand n tend vers $+\infty$ et un tel algorithme ne résout pas les problèmes de stabilité numérique. La solution consiste en fait à calculer $\ln(\delta_a(\boldsymbol{s}_{\mathbb{S}_a}))$ au lieu de $\delta_a(\boldsymbol{s}_{\mathbb{S}_a})$ dans l'algorithme ci-dessus, ce qui est rendu possible par le fait que seuls des produits et des maximisations interviennent dans l'algorithme, et par croissance de la fonction ln. On obtient ainsi un algorithme numériquement stable, de complexité en générale polynomiale, d'interprétation facile grâce à la définition probabiliste de $\delta_a(\boldsymbol{s}_{\mathbb{S}_a})$, où le rôle des paramètres du modèle est explicite.

2.5.3 Application : algorithme du MAP dans les arbres de Markov cachés

Dans cette section, nous appliquons l'algorithme du MAP de la section 2.5.2 au modèle d'arbre de Markov caché de la section 2.4.4. Nous rappelons ci-dessous les notations concernant les cliques de la structure de ce modèle et l'arbre de jonction (figure 2.12). D'après la remarque de la section 2.5.2 concernant l'interprétation a posteriori de l'algorithme de Viterbi, les formules qui suivent s'obtiennent directement à partir de celles de la section 2.4.4 en remplaçant les sommes par des maximisations dans les phases d'initialisation, de propagation et de terminaison. Cependant, les calculs sont brièvement détaillés ci-dessus pour une meilleure interprétation de l'algorithme de Viterbi.

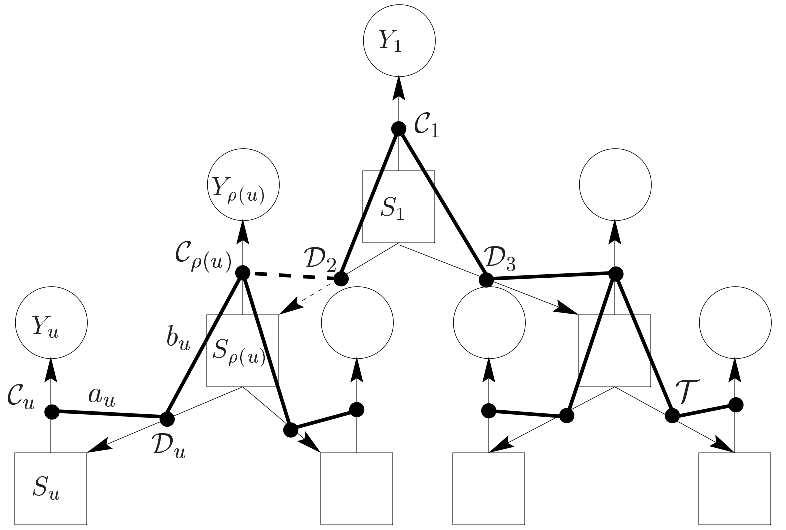


FIG. 2.12 – *Arbre de jonction des arbres de Markov cachés*

Initialisation

Rappelons que d'après la section 2.4.4, il est commode de choisir la clique $\mathcal{C}_1$ comme clique de référence de l'arbre de jonction. L'algorithme du MAP est initialisé en les cliques feuilles $\mathcal{C}_u$ de l'arbre de jonction $\mathcal{T}$, l'indice u étant lui-même une feuille de l'arbre des indices. La clique $\mathcal{C}_u$ est incidente à l'arc a_u. La quantité $\delta_{a_u}(j)$ est par définition égale à $P(Y_u = y_u|S_u = j)$, d'après les formules (2.65) et (2.24). En effet, l'ensemble $\mathcal{C}_u \setminus \mathcal{S}_{a_u}$ ne contient aucun état caché. D'après la section 2.4.4, on a donc $\delta_{a_u}(j) = f_{\theta_j}(y_u)$.

Propagation

D'après la formule (2.67), la propagation dans l'algorithme du MAP est donné, pour l'arc b_u de l'arbre de jonction, par la formule

$$\delta_{b_u}(j) = \max_k P(S_u = k|S_{\rho(u)} = j)\delta_{a_u}(k)$$

car a_u est l'unique arc de $\mathcal{T}_{b_u}$ incident à b_u (voir également la formule de récursion avant (2.29) pour les arbres de Markov cachés). La quantité $P(S_u = k|S_{\rho(u)} = j)$ est donnée par le paramètre $p_{j,k}$ du modèle, de même que dans la section 2.4.4. Donc en définitive,

$$\delta_{b_u}(j) = \max_k p_{j,k}\delta_{a_u}(k). \tag{2.69}$$

On considère ensuite l'arc a_u, incident aux arcs $\{b_v\}_{v\in\boldsymbol{c}(u)}$ dans $\mathcal{T}_{a_u}$. La formule de propagation (2.67) donne alors

$$\delta_{a_u}(k) = \max_{\emptyset} P(Y_u = y_u|S_u = k) \prod_{v\in\boldsymbol{c}(u)} \delta_{b_v}(k)$$

car l'ensemble $\mathcal{C}_u \setminus \mathcal{S}_{a_u} = \{Y_u\}$ ne contient pas d'état caché. Encore une fois, la quantité $P(Y_u = y_u|S_u = k)$ est directement donnée par le paramètre θ_k du modèle – voir section

2.4.4 et équation (2.30) – et on obtient

$$\delta_{a_u}(k) = f_{\theta_k}(y_u) \prod_{v \in \boldsymbol{c}(u)} \delta_{b_v}(k).$$

Remarquons que notre algorithme du MAP fournit une interprétation probabiliste pour les quantités $\delta_{b_u}(j)$ et $\delta_{a_u}(k)$, à savoir

$$\begin{aligned} \delta_{b_u}(j) &= \max_{\boldsymbol{s}_u} P(\boldsymbol{Y}_u = \boldsymbol{y}_u, \boldsymbol{S}_u = \boldsymbol{s}_u | S_{\rho(u)} = j) \quad \text{et} \\ \delta_{a_u}(k) &= \max_{\boldsymbol{s}_{\boldsymbol{c}(u)}} P(\boldsymbol{Y}_u = \boldsymbol{y}_u, \boldsymbol{S}_{\boldsymbol{c}(u)} = \boldsymbol{s}_{\boldsymbol{c}(u)} | S_u = k). \end{aligned}$$

Ainsi, le calcul de la quantité $\delta_{b_u}(j)$ permet de déterminer le processus caché le plus probable associé au sous-arbre $\boldsymbol{Y}_u$ enraciné en Y_u, sachant la valeur de l'état caché parent $S_{\rho(u)}$. Le calcul de la quantité $\delta_{a_u}(k)$ permet de déterminer les sous-arbres cachés enfants les plus probables associés à l'arbre $\boldsymbol{Y}_u$, sachant la valeur de l'état caché S_u. Rappelons en effet que $\boldsymbol{S}_{\boldsymbol{c}(u)}$ désigne le sous-arbre caché $\boldsymbol{S}_u$ privé de sa racine S_u.

Terminaison et restauration

Le calcul de $P^* = \max_{\boldsymbol{s}} P(\boldsymbol{Y} = \boldsymbol{y}, \boldsymbol{S} = \boldsymbol{s})$ s'effectue grâce à l'équation (2.68) où $P(Y_1 = y_1, s_1 = j) = f_{\theta_j}(y_1)\pi_j$, de même qu'en section 2.4.4. On obtient alors

$$P^* = \max_j \pi_j f_{\theta_j}(y_1) \prod_{v \in \boldsymbol{c}(1)} \delta_{d_v}(j).$$

En posant

$$\delta_0(j) = f_{\theta_j}(y_1) \prod_{v \in \boldsymbol{c}(1)} \delta_{d_v}(j),$$

on retombe exactement sur l'algorithme de Viterbi pour les arbres de Markov cachés développé par Durand, Gonçalvès et Guédon, 2002 [46], avec en particulier la phase de terminaison

$$P^* = \max_j \pi_j \delta_0(j).$$

La restauration des états cachés s'effectue à partir de la racine, où

$$\hat{s}_1 = \arg\max_j \pi_j \delta_0(j)$$

puis par un parcours d'arbre descendant, qui permet de déduire $\hat{s}_u$ de la valeur $\hat{s}_{\rho(u)}$, suivant la formule (2.69).

2.6 Propriétés de l'algorithme EM et de ses variantes

L'algorithme EM et ses variantes produisent une suite $(\hat{\lambda}_n)_n$ dont nous rappelons ci-dessous les propriétés de convergence (on parle en général des *propriétés de convergence de l'algorithme EM*). Notons qu'il est inutile, pour étudier ces propriétés, de faire l'hypothèse que les données $\boldsymbol{y}$ utilisées pour l'estimation sont la réalisation d'un modèle de Markov caché.

Algorithme EM

L'étude de la convergence des suites $(\ln(\mathcal{L}_{\boldsymbol{y}}(\hat{\lambda}_n)))_{n\in\mathbb{N}}$ et $(\hat{\lambda}_n)_{n\in\mathbb{N}}$ a été réalisée par Wu, 1983 [124]. La suite $(\ln(\mathcal{L}_{\boldsymbol{y}}(\hat{\lambda}_n)))_{n\in\mathbb{N}}$ est croissante, ce qui découle de l'inégalité de Jensen (autrement dit de la concavité de la fonction ln). Sous les hypothèses suivantes :

(i) pour tout λ_0 tel que $\mathcal{L}_{\boldsymbol{y}}(\lambda_0) > 0$, l'ensemble $\Lambda_{\lambda_0} = \{\lambda \in \Lambda | \mathcal{L}_{\boldsymbol{y}}(\lambda) \geq \mathcal{L}_{\boldsymbol{y}}(\lambda_0)\}$ est compact ;

(ii) $\mathcal{L}_{\boldsymbol{y}}$ est continue sur Λ et différentiable sur l'intérieur de Λ ;

(iii) pour tout $(\lambda, \lambda') \in \Lambda^2$, les applications partielles $\lambda \longrightarrow Q(\lambda, \lambda')$ et $\lambda' \longrightarrow Q(\lambda, \lambda')$ sont continues sur Λ ;

(iv) $\forall n \in \mathbb{N}, \quad \hat{\lambda}_n$ appartient à l'intérieur de Λ ;

toute valeur d'adhérence λ^* de la suite $(\hat{\lambda}_n)_{n\in\mathbb{N}}$ est un point stationnaire de la log-vraisemblance. De plus, $(\ln(\mathcal{L}_{\boldsymbol{y}}(\hat{\lambda}_n)))_{n\in\mathbb{N}}$ converge vers $\ln(\mathcal{L}_{\boldsymbol{y}}(\lambda^*))$. Les conditions (i) et (ii), du fait que $\Lambda \subset \mathbb{R}^d$, entraînent que la log-vraisemblance est majorée, ce qui est parfois restrictif. Par exemple, dans le cas de mélanges gaussiens, la log-vraisemblance n'est pas bornée en général et la condition (i) ne peut être satisfaite. Les conditions (ii) et (iii) sont en principe vérifiées dans cette étude, pour peu que l'application $\theta \longrightarrow P_\theta$ (lois d'émission) soit continue sur Θ et différentiable à l'intérieur de Θ.

La convergence de la suite $(\hat{\lambda}_n)_{n\in\mathbb{N}}$ nécessite des hypothèses plus restrictives que celles ci-dessus. Même sous ces hypothèses, sa limite n'est pas toujours un maximum local de la vraisemblance. Sa convergence vers un point stationnaire, voire vers un maximum local de la log-vraisemblance, est assurée sous des conditions données dans Wu, 1983 [124], qui sont assez fortes et difficiles à vérifier dans le contexte des modèles de Markov cachées (voire fausses, comme l'unimodalité de la vraisemblance).

D'après Celeux et Clairambault, 1991 [18], les propriétés de l'algorithme EM, étudiées principalement dans le contexte des modèles de mélanges indépendants, se généralisent à des mélanges non indépendants. L'algorithme EM fournit de bons résultats en pratique. Cependant, lorsque les composants du mélange ne sont pas très séparés, il fournit des estimations très dépendantes de la position initiale et converge souvent avec une lenteur insupportable (voir à ce sujet la section 3.8.1).

Algorithme CEM

Pour parer à la lenteur de l'algorithme EM, l'algorithme CEM est parfois utilisé, en particulier en reconnaissance de parole (voir Ephraim et Mehrav, 2002 [47]). D'après les auteurs, cet algorithme est utile lorsque les données sont des vecteurs de dimension élevée.

L'algorithme CEM converge en un nombre fini d'itérations et toujours rapidement. Son inconvénient est de fournir des estimations biaisées, même avec des échantillons de grande taille, surtout si les composants du mélange sont peu séparés ou si les probabilités marginales des états cachés sont assez différentes, dans le cas de modèles stationnaires (voir Celeux et Clairambault, 1991 [18]).

Versions stochastiques de EM

D'après Celeux et Clairambault, 1992 [19], l'algorithme SEM pallie l'influence de la valeur initiale du paramètre. Il parvient à identifier des mélanges assez imbriqués et permet de déceler le nombre de composants si on en connaît un majorant, pour peu que l'échantillon des données observées ne soit pas trop petit. De plus, l'introduction de perturbations aléatoires dans l'algorithme EM permet en principe d'éviter une stabilisation vers des valeurs stationnaires de $\mathcal{L}_{\boldsymbol{y}}$ autres que des maxima locaux. En général, la suite aléatoire $(\hat{\boldsymbol{\lambda}}_n)_{n\in I\!N}$ est une chaîne de Markov uniformément géométriquement ergodique et géométriquement φ-mélangeante, qui converge en loi vers son unique loi stationnaire. Ces propriétés sont également vérifiées par l'algorithme EM *à la Gibbs*, bien que la chaîne $(\hat{\boldsymbol{\lambda}}_n)_{n\in I\!N}$ mélange moins rapidement dans ce cas (voir section 2.4.9). Sous des conditions plus fortes, la loi limite de la suite $(\hat{\boldsymbol{\lambda}}_n)_{n\in I\!N}$ résultant de SEM est une loi normale centrée sur l'estimateur consistant du maximum de vraisemblance (voir Diebolt et Celeux, 1993 [38]).

Notons également l'existence de l'algorithme SAEM (*Simulated Annealing EM*), de Celeux et Diebolt, 1990 [21], intermédiaire entre EM et SEM. À l'étape M de l'algorithme, les paramètres sont obtenus par combinaison convexe des estimateurs obtenus respectivement par EM et SEM, où le poids donné à SEM tend vers zéro quand le nombre d'itérations tend vers l'infini. L'algorithme SAEM répond aux limitations bien connues de l'algorithme EM ; mais de plus il se comporte mieux pour traiter de petits échantillons. Par ailleurs, il est plus simple à appréhender que l'algorithme SEM dans la mesure où il converge presque sûrement vers un maximum local de la vraisemblance, tandis que l'algorithme SEM converge en loi.

2.7 Application : un modèle de changement de régularité locale

Dans cette section, nous proposons une application en traitement du signal illustrant l'intérêt des modèles d'arbres de Markov cachés. Cette application met en œuvre les algorithmes d'inférence vus dans les sections 2.3.6, 2.4.6 et 2.5.3. Il s'agit également d'un exemple d'application pouvant être formulée comme un problème de restauration des états cachés, ce qui illustre le fait que cette étape peut, dans certains cas, être la plus importante. Il s'agit en l'occurrence de détecter un changement de régularité locale d'un processus, tout en estimant les différentes valeurs de cette régularité.

Soit $\boldsymbol{x}_1^T = (x_1, \ldots, x_T)$ la réalisation d'un processus échantillonné à régularité (de Hölder) constante par morceaux, par exemple un mouvement brownien fractionnaire homogène par morceaux (H-mbf). La régularité locale d'une fonction (ou d'une trajectoire d'un processus stochastique) est définie comme suit (voir par exemple Mallat, 1998 [89]) : la fonction f a pour régularité locale h $(k < h < k+1)$ à l'instant t si et seulement s'il existe deux constantes C et t_0 avec $0 < C < \infty$ et $0 < t_0$, et s'il existe un polynôme P_k de degré k tel que pour tout $t - t_0 < t' < t + t_0$ et pour tout $h' \leq h$,

$$|f(t') - P_k(t')| \leq C|t' - t|^{h'}. \tag{2.70}$$

Dans la simulation qui suit, nous considérons le modèle de mbf composite suivant : on suppose que $T = 2^U$ et qu'entre $t = 1$ et $t = T_0$ avec $1 \leq T_0 < T$, la régularité locale est $H = H_0$ et qu'entre $t = T_0 + 1$ et $t = T$, sa régularité locale est $H = H_1$ (propriété (*)). Ce travail est motivé, par exemple, par les travaux d'Abry et Veitch, 1998 [1], où les auteurs montrent l'importance cruciale de la détection de changements de régularité locale dans le contexte de télétraffic réseau.

Notre méthode est basée sur l'analyse multirésolution de $\boldsymbol{x}_1^T$. Dans un premier temps, on calcule une transformée en ondelette orthonormée discrète $(y_v^u)_{1 \leq u \leq J_0,\ 0 \leq\ v\ \leq\ 2^u - 1}$ de $\boldsymbol{x}_1^T$ par le produit scalaire suivant

$$y_v^u \ = \ \sum_{k=1}^{2^U} x_k 2^{u/2} \psi(2^u k - v)$$

où J_0 correspond à l'échelle la plus fine. L'ondelette mère ψ est une fonction oscillante, de valeur moyenne nulle, à décroissance rapide (éventuellement à support compact). Comme dans Nowak et Baraniuk, 1998 [32], nous combinons une approche statistique et une analyse en ondelette du signal. Ceci signifie que le signal $\boldsymbol{x}_1^T$ est représenté par ses coefficients d'ondelette $(y_v^u)_{u,v}$, considérés comme des réalisations des variables aléatoires $(Y_v^u)_{u,v}$. Les auteurs justifient un modèle d'arbre binaire de Markov caché pour la transformée en ondelette, plutôt qu'un modèle gaussien indépendant, par deux observations :

– des dépendances résiduelles subsistent entre coefficients d'ondelette ;
– les coefficients d'ondelette sont en général non gaussiens.

Rappelons qu'un H-mbf est un processus gaussien, centré, à incréments stationnaires, et que la régularité de sa trajectoire vaut H presque sûrement et presque partout. Par conséquent, d'après Jaffard, 1991 [63], Flandrin, 1992 [48] et enfin Wornell et Oppenheim, 1992 [122]), les variables aléatoires Y_v^u de sa décomposition en ondelette sont normales, identiquement distribuées à échelle donnée et centrées, de variance

$$\mathrm{var}(Y_v^u) = \sigma^2 2^{u(2H+1)}.$$

Pour le signal de notre exemple test simplifié, la régularité locale étant H_0 pour $1 \leq t \leq T_0$ et H_1 pour $T_0 + 1 \leq t \leq T$, nous considérons un modèle à deux états avec les lois conditionnelles

$$(Y_v^u | S_v^u = j) \sim \mathcal{N}(0, \sigma_j^2 2^{u(2H_j+1)}).$$

Par conséquent, nous modélisons la loi de $(Y_v^u)_{u,v}$ par l'arbre de Markov caché suivant :

– la loi de Y_v^u est une loi de mélange de densité

$$f(Y_v^u = y_v^u) = \sum_{j=0}^{1} P(S_v^u = j) f_{\theta_j}(y_v^u)$$

où S_v^u est une variable aléatoire à deux états notés $\{0;1\}$ et $f_{\theta_j}(y_v^u)$ est la densité de la loi gaussienne centrée, de variance $\sigma_j^2 2^{u(2H_j+1)}$;
- $(S_v^u)_{u,v}$ est un arbre de Markov binaire (c'est-à-dire que chaque sommet, sauf les feuilles, a exactement deux descendants) de loi initiale $(\pi_j)_j$ et de matrice de transition P ;
- les coefficients d'ondelette sont conditionnellement indépendants sachant les états cachés.

Nous reprenons les notations de la section 2.4.4, en particulier la notation $\boldsymbol{Y}_1^1$ désigne le processus aléatoire observé et la notation $\boldsymbol{S}_1^1$ désigne le processus caché.

Dans le cas d'un changement abrupt de régularité à l'instant T_0, le modèle d'arbre de Markov caché $(\boldsymbol{Y}_1^1, \boldsymbol{S}_1^1)$ vérifie les deux propriétés suivantes, qui découlent directement de (*) :
- (P_I) pour chaque sous-arbre $\boldsymbol{S}_v^u$ de $\boldsymbol{S}_1^1$, il existe j dans $\{0;1\}$ tel que le sous-arbre gauche de $\boldsymbol{S}_v^u$ est entièrement dans l'état j ou son sous-arbre droit est entièrement dans l'état j ;
- (P_{II}) si $S_{t_1}^{J_0}$ et $S_{t_2}^{J_0}$ sont deux feuilles telles que $t_1 < t_2$ et que $S_{t_1}^{J_0} = S_{t_2}^{J_0} = j$ alors pour tout t entre t_1 et t_2, $S_t^{J_0} = j$.

Pour détecter le changement de régularité locale, nous calculons la transformée en ondelette discrète y_v^u du signal en utilisant une ondelette de Daubechies à support compact, de régularité un (aussi appelée ondelette de Haar), jusqu'à l'échelle $J_0 = U$, qui est maximale. Les paramètres du modèle sont alors estimés par l'algorithme EM, basé sur les formules de réestimation de la section 2.3.6 et l'*algorithme ascendant-descendant* de la section 2.4.6. Notons que le modèle considéré ici est légèrement différent du modèle des deux sections sus-citées, dans la mesure où les lois d'émission dépendent de la position (u, v) dans l'arbre (en réalité, uniquement du niveau u de résolution). Il s'agit donc d'un modèle homogène quant aux états cachés et non homogène quant aux lois d'émission. Les paramètres H_j et σ_j sont estimés à l'étape M par une adaptation du calcul, dû à Wornell et Oppenheim, 1992 [122], de l'estimateur de maximum de vraisemblance. Ces travaux concernent l'estimation des paramètres dans un modèle gaussien indépendant, tel que $Y_v^u \sim \mathcal{N}(0, \sigma^2 2^{u(2H+1)})$. La présence d'états cachés impose non seulement de recourir à l'algorithme EM mais aussi, à l'étape E, de maximiser la quantité (2.71), qui diffère de la log-vraisemblance essentiellement par la présence des coefficients $\gamma_v^u(j)$, d'où l'adaptation ci-dessous.

Supposons que sachant $S_v^u = j$, Y_v^u suive la loi $\mathcal{N}\left(0, \dfrac{\sigma_j^2}{\beta_j^u}\right)$ où $\beta_j = 2^{-2H_j-1}$. La fonction $\lambda \to Q(\lambda, \hat{\lambda}^{(\eta-1)})$, à l'itération η de l'algorithme EM, comporte un terme qui dépend uniquement des σ_j^2 et des β_j, et s'écrit

$$-\frac{1}{2}\sum_{j=0}^{1}\sum_{u=1}^{J_0}\sum_{v=0}^{2^{J_0-u}-1}\left\{\frac{(y_v^u)^2}{\sigma_{j,u}^2} + \ln(2\pi\sigma_{j,u}^2)\right\}\gamma_v^u(j) \qquad (2.71)$$

où $\sigma_{j,u}^2 = \sigma_j^2 \beta_j^{-u}$ et $\gamma_v^u(j) = P(S_v^u = j | \boldsymbol{Y}_1^1 = \boldsymbol{y}_1^1)$. Cette expression peut être maximisée séparément en (σ_0^2, β_0) et en (σ_1^2, β_1) ; nous isolons par exemple le terme correspondant à $j = 0$. Posons alors

$$\hat{R}_u^2 = \sum_{v=0}^{2^{J_0-u}-1} (y_v^u)^2 \gamma_v^u(0) \text{ et } \Gamma_u = \sum_v \gamma_v^u(0).$$

Nous sommes ramenés à la maximisation de la fonction G définie par

$$\begin{aligned} G(\sigma^2, \beta) &= \sum_{u=1}^{J_0} \sum_{v=0}^{2^{J_0-u}-1} \left\{ \frac{(y_v^u)^2}{\sigma_u^2} + \ln(2\pi\sigma_u^2) \right\} \gamma_v^u(0) \\ &= \sum_u \left\{ \frac{1}{\sigma_u^2} \sum_v (y_v^u)^2 \gamma_v^u(0) + \ln(2\pi\sigma_u^2) \sum_v \gamma_v^u(0) \right\} \\ &= \sum_u \left\{ \frac{\hat{R}_u^2}{\sigma_u^2} + \ln(2\pi\sigma_u^2)\Gamma_u \right\} \end{aligned}$$

avec $\sigma_u^2 = \sigma^2\beta^{-u}$. L'annulation du gradient de G aboutit aux équations suivantes :

$$\begin{aligned} \nabla_\beta G(\sigma^2, \beta) &= \tfrac{1}{\beta} \sum_u u \left\{ \tfrac{\hat{R}_u^2}{\sigma^2} \beta^u - \Gamma_u \right\} = 0 \Leftrightarrow \sum_u u \hat{R}_u^2 \beta^u = \sigma^2 \sum_u u \Gamma_u \\ \nabla_{\sigma^2} G(\sigma^2, \beta) &= \tfrac{1}{\sigma^4} \sum_u u \left\{ \sigma^2 \Gamma_u - \hat{R}_u^2 \beta^u \right\} = 0 \Leftrightarrow \sum_u \hat{R}_u^2 \beta^u = \sigma^2 \sum_u \Gamma_u. \end{aligned}$$

La valeur de σ^2 maximisant G est donc donnée par

$$\hat{\sigma}^2 = \frac{\sum_{u'} \hat{R}_{u'}^2 \beta^{u'}}{\sum_{u'} \Gamma_{u'}}$$

et β est solution de l'équation

$$\sum_u u \hat{R}_u^2 \beta^u = \frac{\sum_{u'} u' \Gamma_{u'}}{\sum_{u'} \Gamma_{u'}} \sum_u u \Gamma_u$$

$$\text{ce qui équivaut à} \quad \sum_u \hat{R}_u^2 \beta^u \Big(u - \frac{\sum_{u'} u' \Gamma_{u'}}{\sum_{u'} \Gamma_{u'}}\Big) = 0,$$

autrement dit β est racine du polynôme

$$f(X) = \sum_u \hat{R}_u^2 X^u \Big(\frac{u}{\sum_{u'} u' \Gamma_{u'}} - \frac{1}{\sum_{u'} \Gamma_{u'}} \Big) = \sum_u C_u \hat{R}_u^2 X^u$$

avec $C_u = \dfrac{u}{\sum_{u'} u' \Gamma_{u'}} - \dfrac{1}{\sum_{u'} \Gamma_{u'}}$. D'après le lemme de Wornell et Oppenheim, 1992 [122], ce polynôme admet une unique racine positive $\hat{\beta}$. Ce résultat s'obtient en montrant que la dérivée de $X^{-u^*}f(X)$ est strictement positive, où $u^* = \frac{\sum_{u'} u'\Gamma_{u'}}{\sum_{u'} \Gamma_{u'}}$. On obtient la valeur $\hat{\beta}_j$ à l'étape M de l'algorithme EM par dichotomie, sachant que β est compris entre $\frac{1}{8}$ et $\frac{1}{2}$. On en déduit $\hat{H}_j = -\frac{1}{2}(1 + \log_2(\hat{\beta}_j))$ pour $j = 0$ et $j = 1$.

En définitive, on obtient les estimateurs $\hat{P}$, $\hat{\pi}$, $\hat{\sigma}_0$, $\hat{\sigma}_1$, $\hat{H}_0$ et $\hat{H}_1$. La détection du saut est effectuée par une restauration des états cachés sous les contraintes (P_I) et (P_{II}), en utilisant l'algorithme de Viterbi de la section 2.5.3. On obtient ainsi une valeur $\hat{\boldsymbol{s}}$ pour l'arbre caché $\boldsymbol{S}$, telle qu'exactement l'un (maximal) des sous-arbres de $\hat{\boldsymbol{s}}$, noté $\hat{\boldsymbol{s}}_v^u$, est dans l'état j, toutes les autres variables aléatoires cachées étant dans l'état $1-j$. Par conséquent, il existe une unique feuille $S_{t^*}^{J_0}$ de l'arbre telle que $S_{t^*}^{J_0} \neq S_{t^*+1}^{J_0}$. L'instant de saut T_0 est estimé par

$$\hat{T}_0 = 2.t^*$$

En pratique, pour éviter une trop grande discontinuité de la trajectoire à l'instant de transition T_0 et pour assurer une prescription exacte de la régularité locale $H(t)$ en chaque point t, nous simulons un mouvement brownien multifractionnaire tel qu'il est défini dans Levy-Vehel et Peltier, 1995 [99] par la méthode proposée par les auteurs, avec une régularité de Hölder transitoire continue (Figure 2.14) :

$$\forall t \in \{1, \ldots, 1024\} \qquad H(t) = 0,1 \tanh\left(-20 + \frac{40(t-1)}{1023}\right) + 0,5. \tag{2.72}$$

Nous posons $H_0 = 0,4$ et $H_1 = 0,6$. Nous construisons alors le processus $\boldsymbol{x}_1^{1024} = (x(t))_{t=1,\ldots,1024}$ de régularité locale donnée par (2.72). Une trajectoire ainsi simulée est représentée figure 2.15 a).

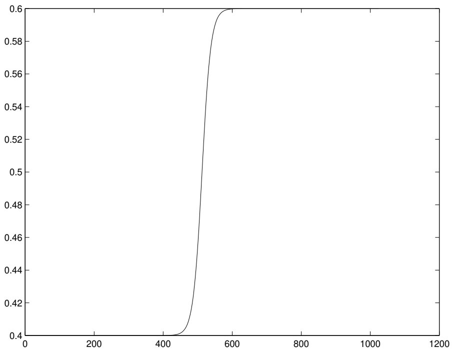


FIG. 2.14 – *Évolution temporelle du paramètre de régularité locale.*

La figure 2.15 b) montre la carte des probabilités lissées $P(S_v^u = 1|\boldsymbol{Y} = \boldsymbol{y})$. L'axe des ordonnées du schéma représente la profondeur de l'arbre, dont la racine est située sur la ligne inférieure. Par convention, la profondeur de l'arbre est J_0 et la racine de l'arbre est située à la profondeur un. La figure 2.15 c) montre le résultat de la restauration des états cachés sous contraintes. La frontière entre les deux états est utilisée pour localiser l'instant T_0 de transition dans $H(t)$. Les estimateurs des paramètres sont $\hat{H}_0 = 0,3009$, $\hat{H}_1 = 0,6649$ et $\hat{T}_0 = 520$.

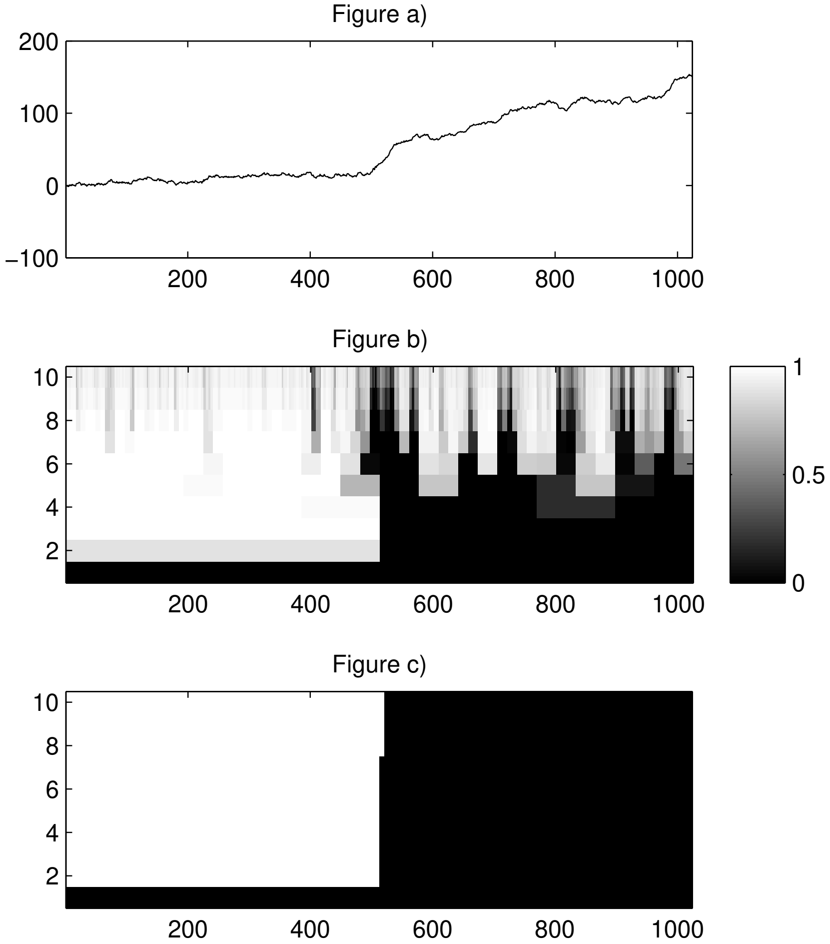


FIG. 2.15 – *Arbre caché associé à la transformée en ondelette du signal – a) Trajectoire d'un mbf à régularité H constante par morceaux, valant $H_0 = 0,4$ (correspondant à l'état 0) pour $t = 1, \ldots, 512$ et $H_1 = 0,6$ (correspondant à l'état 1) pour $t = 513, \ldots, 1024$; b) Carte des probabilités lissées. Le niveau de gris indique la valeur de la probabilité conditionnelle d'être dans l'état 1 pour un coefficient donné ; c) Arbre caché restauré.*

Les résultats ci-dessus appellent plusieurs remarques. Tout d'abord, les estimateurs de H_0 et H_1 sont imprécis, ce qui est dû aux faibles tailles d'échantillon disponibles pour chacun des états. Ils sont néanmoins cohérents avec les performances annoncées dans Wornell et Oppenheim, 1992 [122]. En particulier, la méthode utilisée pour l'estimation de H_j et σ_j connaît les mêmes limitations que l'algorithme décrit dans Wornell et Oppenheim. Malgré les faibles performances de l'estimation, du point de vue de la classification, la séparation des composants du mélange par notre méthode est totalement satisfaisante. L'imprécision sur l'estimation des H_i n'affecte pas l'estimation de l'instant de transition $\hat{T}_0$.

Remarque 2.19 *Dans notre exemple très simplifié, la carte des probabilités de lissage (figure 2.15-b) est une étape complémentaire de la restauration. En soi, cette carte n'apporte pas beaucoup d'information, vu l'objet de cette application. Cependant, l'incertitude* *apparente sur les états cachés situés après l'instant de transition T_0 mérite d'être commentée. Rappelons que par définition de la régularité locale de Hölder d'une trajectoire, h est le supremum sur tous les h' vérifiant l'inégalité de l'expression (2.70). Ceci signifie que ponctuellement, des régularités plus faibles sont susceptibles d'être estimées. Lors de l'analyse de la partie la plus régulière de la trace, le modèle à deux états utilisé permet effectivement d'estimer des régularités plus faibles que la régularité effective, d'où ces changements d'états. Encore une fois, conformément à la définition (2.70), ceci ne se produit évidemment pas dans la partie gauche de la trajectoire (partie la moins régulière).*

2.8 Conclusion sur l'inférence dans les modèles de Markov cachés

Dans ce chapitre, nous avons montré comment l'étude des propriétés du graphe d'indépendance conditionnelle de modèles de Markov cachés permet d'élaborer un algorithme d'estimation des paramètres basé sur la méthodologie EM. La difficulté réside essentiellement dans l'étape E ; elle est liée au calcul de probabilités. Notre démarche a consisté à présenter les algorithmes de calcul de probabilités dans le contexte de l'implémentation de l'étape E, puis à présenter certaines de leurs autres applications. Le problème du calcul de probabilités est poursuivi dans la section 3.4.3.

Les résultats obtenus mettent en évidence, d'une part, l'intérêt de disposer d'algorithmes interprétables, afin de pouvoir les utiliser en évitant les calculs redondants, de réaliser du calcul analytique, par exemple d'espérances ou de fonctions de répartition conditionnelles, et afin de modifier ou d'améliorer ces algorithmes, au besoin. La principale amélioration réalisée concerne la stabilité numérique ; celle-ci est obtenue par l'adoption d'un point de vue issu des modèles à espace d'états, qui conduit à des algorithmes analogues à ceux de lissage.

Pour différentes raisons, nous avons été amenés à faire des hypothèses sur la structure du modèle : hypothèse d'un graphe orienté sans cycle pour faciliter son interprétation et sa paramétrisation, celle de moralité de ce graphe, pour obtenir de manière immédiate les propriétés d'indépendance conditionnelle entre variables aléatoires et pour pouvoir développer des algorithmes récursifs génériques, et enfin l'hypothèse d'absence d'arc entre variables aléatoires observées, qui permet une paramétrisation générique dans le cas de variables observées à valeurs continues. Cette contrainte a également des conséquences sur les algorithmes de calcul de probabilités utilisés en sélection de modèles (voir section 3.4.3). Ces derniers permettent l'inférence dans des modèles à observations Y_t supprimées ou manquantes. Le chapitre 5 propose une discussion sur l'extension de ces méthodes lorsque les contraintes ci-dessus sont relâchées.

Chapitre 3

Sélectionner un modèle de Markov caché

3.1 Introduction

Dans la section 2.4 du chapitre 2, nous avons vu comment calculer des probabilités dans les modèles de Markov cachés de la famille $\mathcal{D}$ quand les paramètres λ sont connus. De plus, dans la section 2.3, nous avons vu comment estimer ces paramètres quand la structure $\mathcal{G}$ du modèle, l'ensemble $\mathcal{S}$ des valeurs prises par le processus caché et la famille $(P_\theta)_{\theta\in\Theta}$ des lois d'émission sont connus. Ces techniques permettent de traiter un certain nombre de problèmes, comme l'application présentée en section 2.7. Cependant, dans la plupart des problèmes de modélisation de phénomènes réels, la structure $\mathcal{G}$ du modèle, la nature de l'ensemble $\mathcal{S}$ et la famille $(P_\theta)_{\theta\in\Theta}$ sont également des paramètres inconnus, qui doivent être déterminés.

Ce chapitre présente des outils permettant la sélection de modèles dans le contexte des modèles de Markov cachés. Nous commençons le chapitre par un exposé de la problématique de la sélection de modèles, tout en insistant sur les liens entre l'utilisation du modèle (lié à l'objectif du modélisateur) et les critères ou les méthodes de sélection. Ces différents objectifs expliquent la diversité des points de vue retenus pour aborder la sélection de modèles. La présentation des méthodes qui en découlent se fait, dans la suite du chapitre, en les regroupant suivant l'approche qui est à leur origine. Ainsi, nous étudions les méthodes basées sur les tests, sur la théorie de l'information, sur une approche bayésienne, sur la vraisemblance marginale pénalisée et une méthode issue de la classification.

Nous proposons ensuite une comparaison expérimentale de la plupart de ces critères pour le choix du nombre d'états cachés d'une chaîne de Markov cachée. Dans un premier temps, les séquences sont effectivement simulées suivant cette loi. Puis nous présentons une application des chaînes de Markov cachées à la modélisation du processus des défaillances et des corrections de logiciels, mettant en œuvre plusieurs aspects de la sélection de modèles, avec principalement le choix du nombre d'états cachés de la chaîne de Markov et le choix de son type de matrice de transition. Ces comparaisons expérimentales mettent en évidence le bon comportement des critères BIC et celui de validation croisée basée sur le demi-échantillonnage.

3.2 Problématique de la sélection de modèles

Dans cette section, nous commençons par définir ce qu'est un modèle, puis nous donnons plusieurs exemples de problèmes relevant de la sélection de modèles : choix du nombre d'états cachés, du graphe d'indépendance conditionnelle, de la famille des lois d'émission, du type de matrice de transition entre états et le choix d'un modèle gaussien. Enfin, nous présentons certains enjeux de la sélection de modèles en énonçant divers points de vue qui donnent lieu à autant de critères de sélection. Ce sont ces critères qui seront passés en revue dans les sections 3.3 à 3.8.

3.2.1 Définition d'un modèle

Nous considérons des modèles de Markov cachés dynamiques (*i.e.* dont la structure est basée sur la répétition d'un motif) et identifiables. Dans ce cadre, une loi $P_{\lambda,\boldsymbol{Y}}$ de la famille $\mathcal{D}$ est caractérisée par sa structure $\mathcal{G}$, par l'ensemble $\mathcal{S}$ des valeurs des états cachés, par la famille $\mathcal{P} = (P_\theta)_{\theta\in\Theta}$ des lois d'émission et par la valeur λ des paramètres, qui appartient à un espace $\Lambda \subset \mathbb{R}^d$ dépendant de $\mathcal{G}$, de $\mathcal{S}$ et de $\mathcal{P}$. On appellera *modèle* $\mathcal{M}(\mathcal{G},\mathcal{S},\mathcal{P})$ la famille de lois de probabilités

$$\mathcal{M}(\mathcal{G},\mathcal{S},\mathcal{P}) = \{P_{\lambda,\boldsymbol{Y}} | \lambda \in \Lambda(\mathcal{G},\mathcal{S},\mathcal{P})\}.$$

Exemple : modèles de chaînes de Markov cachées à famille d'émission gaussienne.
Nous considérons la structure $\mathcal{G}$ définie par le graphe d'indépendance conditionnelle de la figure 1.5.

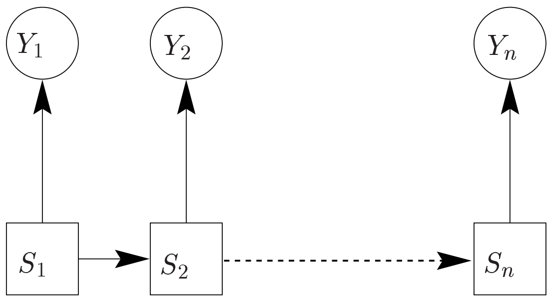


FIG. 1.5 – *La structure des modèles de chaîne de Markov cachée.*

On suppose que la famille des lois d'émission est la famille gaussienne, de sorte que

$$\mathcal{P} = \{f_{\mu,\sigma^2} | \mu \in \mathbb{R},\ \sigma^2 \in \mathbb{R}_+^*\},$$

où f_{μ,σ^2} désigne la densité de la loi gaussienne de paramètres (μ, σ^2). Le nombre de valeurs possibles pour les états cachés, noté K, n'est pas connu. Les ensembles $\mathcal{G}$ et $\mathcal{P}$ étant ainsi fixés, on peut considérer, pour chaque valeur de K, le *modèle* $\mathcal{M}(K)$ défini comme l'ensemble des chaînes de Markov cachées gaussiennes à K états cachés.

3.2.2 Modèles en compétition

Nous avons vu dans la section précédente qu'un modèle dépendait de la structure $\mathcal{G}$, de la famille $\mathcal{P}$ des lois d'émission et du cardinal de l'ensemble des réalisations $\mathcal{S}$

du processus caché. L'ensemble de tous les modèles de Markov cachés possibles est clairement infini puisqu'il existe une infinité de graphes d'indépendance conditionnelle ; les possibilités de choix des variables cachées du modèle étant elles-mêmes infinies. En outre, même à graphe d'indépendance conditionnelle fixé, il existe une infinité de choix pour $\mathcal{S}$. Les méthodes présentées dans les sections suivantes ne s'appliquent que dans les cas où le nombre de modèles en compétition est fini. Plus précisément, nous verrons que ces méthodes requièrent l'estimation des paramètres au moins une fois pour chaque modèle en compétition. Vu le temps de calcul nécessaire à l'estimation de ces paramètres, nous considérerons uniquement les cas où les modèles en compétition sont en nombre restreint. Le principe consistant à considérer a priori un nombre restreint de modèles au lieu d'envisager autant de modèles que possible est préconisé, dans le contexte de la modélisation statistique en biologie, par Burnham et Anderson, 1998 [17]. En particulier, les auteurs défendent le point de vue suivant : *si un modèle n'a aucun sens par rapport au phénomène étudié, il ne devrait pas faire partie de l'ensemble des modèles en compétition.* Cette exigence d'un petit nombre de modèles en compétition exclut par exemple le problème du choix de la structure $\mathcal{G}$ parmi toutes celles possédant un nombre fixé N de sommets, étant donné que le nombre de modèles en compétition est alors une fonction exponentielle de N.

En revanche, les problèmes suivants de sélection de modèles rentrent dans notre cadre d'étude. Par la suite, nous considérerons essentiellement le problème 1. Nous ne ferons qu'évoquer les difficultés liées au problème 4. Les autres problèmes de sélection de modèles ne sont présentés que pour information mais les méthodes étudiées dans ce chapitre permettent de les traiter également.

1. Choix du nombre d'états cachés, sous l'hypothèse que tous les états cachés sont à valeurs dans un même ensemble fini.

2. Choix du graphe d'indépendance conditionnelle, quand le nombre de graphes considérés est restreint. Ceci permet de choisir par exemple entre un modèle de mélange indépendant et un modèle de chaîne de Markov cachée.

3. Choix de la famille des lois d'émission. On pourra par exemple choisir entre la famille log-normale et la famille gamma pour les lois d'émission de chaînes de Markov cachées.

4. Choix de contraintes sur la matrice de transition (interdiction de certaines transitions).

5. Choix du type de modèle gaussien. Nous développons ici, de manière plutôt détaillée, ce que nous entendons par "modèle gaussien", bien que d'une part, ces modèles soient plus utilisés dans le cas de modèles de mélanges indépendants que dans les modèles de Markov cachés généraux, souvent monodimensionnels, et bien que nous ne traitions pas ce problème de sélection en particulier. En revanche, la description qui suit permet de comprendre le principe des logiciels d'identification présentés en annexe A et utilisés à la fin de ce chapitre.

   Lorsque l'on choisit pour $\mathcal{P}$ la famille gaussienne, on peut restreindre cette famille en imposant des contraintes sur la matrice de variance-covariance. En effet, d'après Banfield et Raftery, 1993 [5], il est intéressant, du point de vue de l'interprétation des composants d'un modèle de mélange gaussien, d'utiliser la décomposition spectrale de la matrice Σ_k de l'état caché k. Nous présentons en réalité la décomposition unique suivante, due à Celeux et Govaert, 1995 [24] et légèrement différente de celle de Banfield et Raftery :

$$\Sigma_k = \lambda_k D_k A_k D_k$$

où

- $\lambda_k = |\Sigma_k|^{\frac{1}{d}} > 0$;
- D_k est la matrice orthogonale des vecteurs propres de Σ_k ;
- A_k est la matrice diagonalisée, normalisée de D_k. Ses coefficients sont les valeurs propres normalisées de D_k, classées par ordre décroissant sur la diagonale. La normalisation consiste à diviser les valeurs propres par λ_k ; ainsi $|A_k| = 1$.

Les différents facteurs de cette décomposition ont l'avantage d'avoir une interprétation géométrique (voir Biernacki, 1997 [12]) : λ_k s'interprète comme le volume associé à l'état caché k, D_k comme son orientation et A_k comme sa forme. Dans le cas particulier d'observations dans $I\!R^2$, ces quantités sont illustrées par la figure 3.1. Dans ce cas, D_k est une matrice de rotation, d'un certain angle α_k, et A_k est définie par son premier coefficient diagonal a_k, d'où

$$A_k = \begin{bmatrix} a_k & 0 \\ 0 & \dfrac{1}{a_k} \end{bmatrix}, \quad D_k \begin{bmatrix} \cos(\alpha_k) & -\sin(\alpha_k) \\ \sin(\alpha_k) & \cos(\alpha_k) \end{bmatrix}.$$

L'*ellipse de concentration* (ou de variance) associée à la matrice de variance de l'état k, représentée figure 3.1, a pour centre μ_k et contient $\xi\%$ des points que génère la loi gaussienne de paramètres (μ_k, Σ_k). Tracer l'ellipse de concentration pour l'état k revient à tracer l'ensemble des points $x = (x_1, x_2)$ vérifiant l'équation

$$\sqrt{{}^t(x - \mu_k)\Sigma_k^{-1}(x - \mu_k)} = \frac{\xi}{100}.$$

Cette courbe est composée des points équidistants du centre μ_k au sens de la distance de Mahalanobis.

En contraignant certaines des quantités λ_k, A_k et α_k à être égales pour toutes les valeurs de k, on obtient diverses familles aisément interprétables de modèles gaussiens. Les différentes hypothèses sur ces quantités conduisent à huit modèles généraux. Par exemple, on peut supposer des volumes différents d'un état caché à l'autre mais la même forme et la même orientation des matrices Σ_k en imposant que pour tout k, $A_k = A$ et $D_k = D$, les matrices A et D étant en général à estimer. Nous notons ce modèle en utilisant la même notation que Biernacki, 1997 [12], à savoir : $[\lambda_k DAD]$. Avec la même convention, la notation $[\lambda D_k A D_k]$ désignerait un modèle à volumes égaux et formes égales mais à orientations différentes.

D'autres types de situation offrent un intérêt, le premier consistant à supposer que les matrices Σ_k sont diagonales. Ceci revient à supposer, vu la reparamétrisation, que les matrices D_k sont des matrices de permutation. Dans ce cas, les changements d'orientation de Σ_k n'ont pas d'intérêt et nous notons $\Sigma_k = \lambda_k B_k$, où B_k est une matrice diagonale de déterminant égal à 1. Cette hypothèse conduit aux quatre

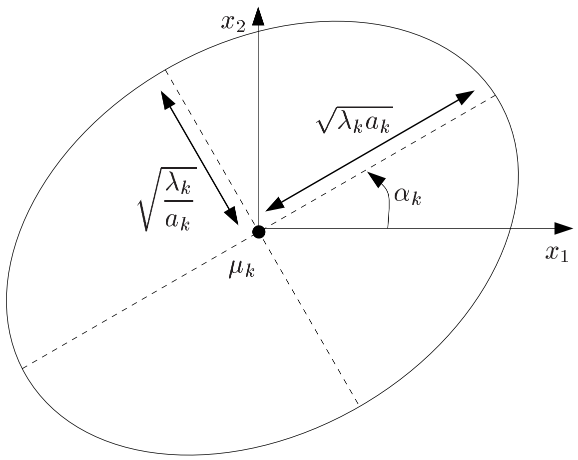


FIG. 3.1 – *Interprétation géométrique de la décomposition spectrale. La quantité λ_k représente le volume de l'ellipse de concentration, $a_k = A_k(1,1)$ détermine sa forme et l'angle α_k de la matrice de rotation D_k détermine son orientation.*

modèles supplémentaires $[\lambda B]$, $[\lambda_k B]$, $[\lambda B_k]$ et $[\lambda_k B_k]$. Le second type de situation consiste à supposer les modèles sphériques, c'est-à-dire que $B_k = I$ où I désigne la matrice identité (dans $I\!R^d$). Nous obtenons les modèles $[\lambda I]$ et $[\lambda_k I]$, ce qui porte à 14 le nombre de modèles gaussiens. Ces modèles sont représentés figure 3.2.

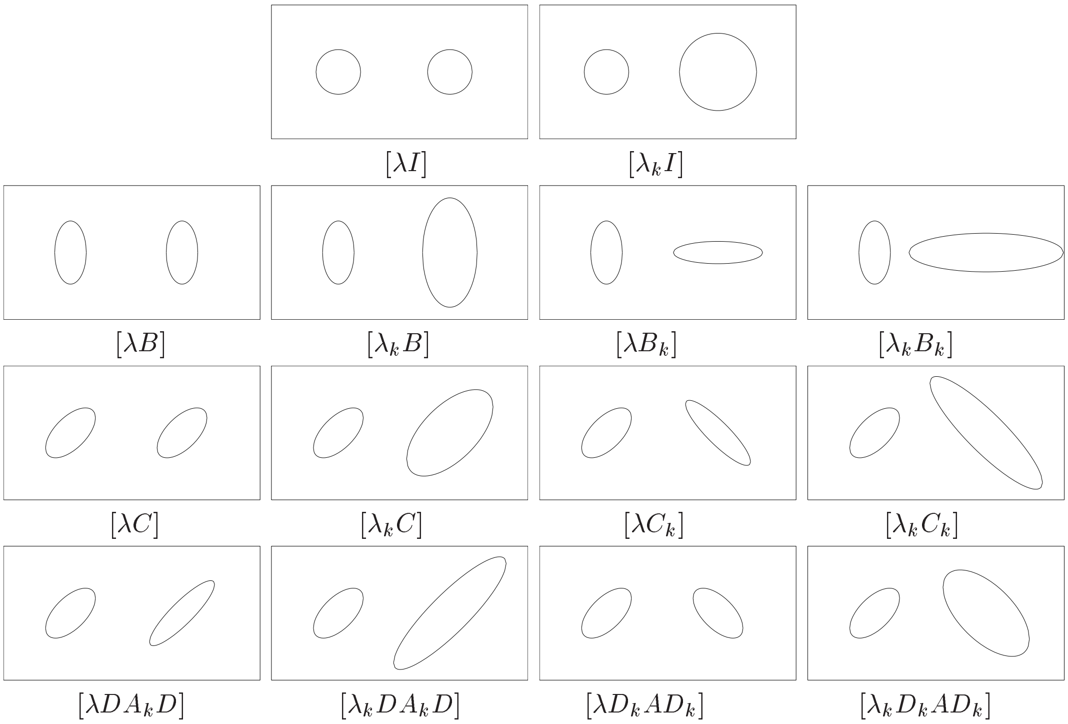


FIG. 3.2 – *Courbes isodensités des 14 modèles gaussiens, représentés ici dans le cas de deux états cachés par des couples de matrices de variance. Le paramètre λ détermine le volume des ellipses de concentration, A leur forme et D leur orientation.*

Remarque 3.1 – *Dans le cas de modèles de mélanges indépendants, la densité de Y s'écrit*

$$f(y) = \sum_{k=1}^{K} \pi_k f_{(\mu_k,\Sigma_k)}(y) \tag{3.1}$$

où $\pi_k = P(S = k)$ représente la proportion de la classe k. Il est alors possible de contraindre les π_k à être égaux (en l'occurrence à $\frac{1}{k}$), ce qui double le nombre de modèles possibles. La notation $[\pi_k]$ désigne les modèles à proportions non contraintes et $[\pi]$ désigne les modèles à proportions égales. Par exemple, dans le contexte de mélanges gaussiens, la notation $[\pi\lambda D_k A D_k]$ désigne un modèle à volumes égaux et formes égales mais à orientations et proportions différentes.
– *Dans le cas de modèles de chaînes ou d'arbres de Markov cachés stationnaires, l'équation (3.1) s'applique également à la loi marginale de Y. On distingue donc les modèles non stationnaires $[\pi_k]$, où $\pi_k = P(S_1 = k)$ est non contraint, des modèles stationnaires $[\pi]$, où $\pi = (\pi_k)_k$ désigne la loi stationnaire qui se déduit de la matrice de transition P. Par exemple, dans le contexte de chaînes de Markov cachées à lois d'émission gaussiennes, la notation $[\pi\lambda D_k A D_k]$ désigne un modèle stationnaire à volumes égaux et formes égales mais à orientations différentes.*

3.2.3 Enjeu de la sélection de modèles

Les motivations de la sélection de modèles sont étudiées en détail par Burnham et Anderson, 1998 [17], dans le contexte de la modélisation statistique en biologie. Les auteurs étudient les conséquences de l'usage d'un modèle sur- ou sous- paramétré, ou tout simplement inadéquat. Leur postulat est qu'une inférence valide se base sur un modèle valide, toute la question étant alors de pouvoir formaliser la notion *de validité* ou *d'adéquation*. Nous partons du principe, à notre avis répandu, qu'un modèle est conçu avec un objectif défini a priori. Ce modèle est censé aider à répondre à un problème ou à une question et cette intention du modélisateur est alors incluse dans le modèle d'une manière ou d'une autre. Dans le cas de modèles de Markov cachés, il est fréquent que le but soit la prise en compte des différentes zones dans lequel le processus observé a un comportement (ou régime) homogène. La question pourra alors être *"combien y-a-t-il de types de régime ?"* (résolution par choix du nombre d'états cachés) puis *"quelles sont les zones homogènes ?"* (résolution par restauration des états cachés). Un autre objectif d'un modèle peut être d'effectuer des prévisions. La notion de prévision est d'ailleurs présente de manière implicite ou explicite dans des critères classiques de sélection de modèle. Un modèle qui est adapté à un jeu de données $\boldsymbol{y}$ particulier, mais à aucun autre échantillon simulé suivant la loi de $\boldsymbol{Y}$, ou aucun jeu de données réel de même nature que $\boldsymbol{y}$, risque d'avoir peu d'intérêt. On désire en général qu'un modèle réponde au problème posé pour toute réalisation $\boldsymbol{y}$ de $\boldsymbol{Y}$ suivant la (vraie) loi de ce processus. Nous verrons dans la section 3.4 le rôle de la prédiction dans la sélection de modèles basée sur la théorie de l'information. En conclusion, le concept de *modèle valide* est lié à l'intention du modélisateur et de même que cette intention est intégrée au modèle (par exemple, par l'introduction d'états cachés indiquant le type de régime), la phase de sélection de modèles tient compte de cette intention.

Puisqu'il s'agit de trouver un modèle adéquat, l'idée d'avoir recours à des tests d'adéquation à une famille de modèles $\mathcal{M}(\mathcal{G}, \mathcal{S}, \mathcal{P})$ semble naturelle. On dispose des données $\boldsymbol{y}$, réalisation d'un processus aléatoire $\boldsymbol{Y}$ de loi P_0. Il s'agit alors de tester l'hypothèse

$$H_0 : P_0 \in \mathcal{M}(\mathcal{G}, \mathcal{S}, \mathcal{P})$$

contre une hypothèse alternative, par exemple,

$$H_1 : P_0 \notin \mathcal{M}(\mathcal{G}, \mathcal{S}, \mathcal{P}),$$

ou encore contre l'hypothèse

$$H_1' : P_0 \in \mathcal{M}(\mathcal{G}', \mathcal{S}', \mathcal{P}'),$$

où en général les modèles de $\mathcal{M}(\mathcal{G}, \mathcal{S}, \mathcal{P})$ appartiennent à $\mathcal{M}(\mathcal{G}', \mathcal{S}', \mathcal{P}')$. L'approche des tests n'est cependant pas celle de Burnham et Anderson qui, rappelons-le, s'intéressent à la modélisation de phénomènes biologiques où les données sont peu susceptibles d'être vraiment la réalisation de variables aléatoires de loi connue, et somme toute assez simple, si l'on considère tous les paramètres ayant pu, dans la réalité, avoir une influence sur les mesures ayant produit les données $\boldsymbol{y}$. Ceci se traduit par le fait que, dans ce contexte, le test de H_0 contre H_1 va avoir tendance à rejeter H_0 quand le nombre n de données observées est suffisamment élevé. Dans le cas du test de H_0 contre H_1', le fait que l'hypothèse H_0 soit rejetée ne signifie pas forcément que H_1' soit un *modèle valide*. Le test de

$$P_0 \in \mathcal{M}(\mathcal{G}', \mathcal{S}', \mathcal{P}')$$

contre

$$P_0 \notin \mathcal{M}(\mathcal{G}', \mathcal{S}', \mathcal{P}')$$

aidera à savoir si le modèle alternatif est adéquat.

Les auteurs, refusant de croire en l'existence d'un vrai modèle connu qui aurait généré les données, proposent de recourir à une notion de *meilleur modèle approximant* la loi de $\boldsymbol{Y}$. Vu que la vraie loi P_0 de $\boldsymbol{Y}$ ne peut être déterminée, il s'agit de sélectionner, parmi les lois de probabilités $\tilde{P}$ dans l'ensemble de modèles $\{\mathcal{M}(\mathcal{G}_i, \mathcal{S}_i, \mathcal{P}_i)\}_i$, celle qui minimise une certaine mesure de dissimilarité entre $\tilde{P}$ et P_0, en l'occurrence la divergence de Kullback-Leibler

$$\mathrm{KL}(P_0, \tilde{P}) = \int P_0(\boldsymbol{y}) \log \frac{P_0(\boldsymbol{y})}{\tilde{P}(\boldsymbol{y})}\, d\boldsymbol{y}. \tag{3.2}$$

Cette méthode s'inspire de la théorie de l'information, qui s'est développée dans les années 1940 et recouvre des théories fondamentales de plusieurs disciplines scientifiques (information de Fisher en statistique et entropie de Boltzmann en thermodynamique, entre autres). D'autres fonctions que la divergence de Kullback-Leibler peuvent être utilisées pour mesurer la dissimilarité entre lois de probabilités – ce que Linhart et Zucchini, 1986 [84] appellent *le manque d'adéquation.* Ces auteurs développent une théorie générale de la sélection de modèles basée sur la notion de meilleur modèle approximant, au sens d'une mesure de dissimilarité donnée (par exemple la divergence de Kullback-Leibler et les distances de Kolmogorov, de Cramer-von Mises, de Pearson, de Neyman et de Gauss, entre autres).

Ainsi, il existe différentes approches possibles pour la sélection de modèles. Les sections suivantes offrent un panorama de certaines de ces approches et décrivent les méthodes qui en découlent. La section 3.3 aborde les méthodes basées sur le formalisme des tests. La section 3.4 traite du critère d'information AIC et de la validation croisée, tous deux liés à la théorie de l'information. La section 3.5 présente la sélection de modèles basée sur le critère d'information bayésienne BIC. Nous donnons en section 3.6 des arguments, basés sur les travaux de Gassiat, 2002 [52] pour la sélection de modèles basée sur des critères de vraisemblance marginale pénalisée. En section 3.7, nous présentons une approche issue de la sélection de modèles en classification automatique, le critère ICL. Enfin, une application de la sélection de modèles est réalisée sur des données simulées en section 3.8.1 et sur des données réelles en section 3.8.2.

3.3 Tests d'hypothèses

La théorie des tests d'hypothèses peut être utilisée pour la sélection de modèles. L'ouvrage de McLachlan et Peel, 2000 [91] et la thèse de C. Biernacki, 1997 [12] offrent, à notre avis, un très bon résumé de l'application de cette méthodologie à la sélection de modèles dans le cadre de mélanges indépendants. Le principe de cette méthode est de tester une hypothèse H_0 contre une hypothèse H_1 avec

$$\begin{array}{rcl} H_0 & : & P_0 \in \mathcal{M}_0 \\ H_1 & : & P_0 \in \mathcal{M}_1, \end{array}$$

où $\mathcal{M}_i = \mathcal{M}(\mathcal{G}_i, \mathcal{S}_i, \mathcal{P}_i)$ pour $i \in \{0;1\}$, avec de plus $\mathcal{M}_0 \subset \mathcal{M}_1$ (les modèles sont dits *emboîtés*). L'hypothèse H_0 correspond au fait que le vrai modèle est $\mathcal{M}(\mathcal{G}_0, \mathcal{S}_0, \mathcal{P}_0)$ et l'hypothèse H_1 au fait que le vrai modèle est $\mathcal{M}(\mathcal{G}_1, \mathcal{S}_1, \mathcal{P}_1)$. Le rapport des vraisemblances maximales est défini par

$$r(\boldsymbol{y}) = \frac{\max\limits_{\lambda \in \mathcal{M}_0} \mathcal{L}_{\boldsymbol{y}}(\lambda)}{\max\limits_{\lambda \in \mathcal{M}_1} \mathcal{L}_{\boldsymbol{y}}(\lambda)}.$$

L'élaboration du test repose sur la détermination de la loi de $r(\boldsymbol{Y})$. Cette loi est souvent impossible à obtenir, c'est pourquoi on essaye d'obtenir sa loi asymptotique. Sous certaines conditions de régularité (dont la normalité asymptotique de l'estimateur de maximum de vraisemblance, ou EMV), la loi asymptotique de $-2\ln(r(\boldsymbol{Y}))$ est une loi du χ^2.

Choix du nombre d'états cachés

Dans le cas du choix du nombre d'états cachés K, on est amené à poser

$$\begin{array}{rcl} H_0 & : & K = k \\ H_1 & : & K = k+1 \end{array}$$

pour une certaine valeur de k. La difficulté de recourir à des tests découle de ce que sous H_0, l'EMV ne suit pas, asymptotiquement, une loi normale de plein rang. Ceci provient du fait que H_0 correspond à l'appartenance de λ à un sous-ensemble de l'espace des paramètres où le mélange est non identifiable, ou à l'appartenance de λ à la frontière de cet espace. En effet, l'hypothèse H_0 peut être spécifiée par l'annulation d'une proportion d'un mélange à $k+1$ composants. Le paramètre π appartient alors à la frontière de l'espace des paramètres. De plus, si $\pi_i = 0$ pour l'état caché i, alors la vraisemblance ne dépend pas de θ_i. D'autre part, H_0 est également vraie lorsque $\theta_i = \theta_j$ pour deux états cachés distincts i et j. La vraisemblance est alors la même, sous H_0, pour toutes les valeurs de π_i et de π_j situées sur une certaine droite d'équation $\pi_i + \pi_j = \text{Constante}$. Ce manque d'identifiabilité conduit à une dégénérescence de la matrice d'information lorsque l'on considère la loi de $-2\ln(r(\boldsymbol{Y}))$ sous H_0. On s'attend alors à ce que la loi asymptotique de $-2\ln(r(\boldsymbol{Y}))$ ne soit pas une loi du χ^2 – du moins cette loi est-elle inconnue.

Différentes approches ont été envisagées pour déterminer la loi, asymptotique ou non, de la statistique de test $-2\ln(r(\boldsymbol{Y}))$. Ainsi, l'approche de McLachlan et Peel, 1996 [90] consiste à obtenir la loi de $-2\ln(r(\boldsymbol{Y}))$ par rééchantillonnage. L'estimateur $\hat{\lambda}_0$ de λ est calculé par maximum de vraisemblance sous H_0 (c'est-à-dire avec un modèle à k états cachés). Puis on tire R échantillons $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_R$ de taille n sous H_0, autrement dit sous la loi $P_{\hat{\lambda}_0}$. Pour chacune de ces réplications, on obtient la statistique de test en estimant le paramètre sous H_0 et sous H_1. On dispose ainsi d'une estimation de la vraie loi de $-2\ln(r(\boldsymbol{Y}))$ sous H_0. Le test au niveau de signification α se fait en rejetant H_0 si la valeur de la statistique pour l'échantillon d'origine $-2\ln(r(\boldsymbol{y}_r))$ est plus grande que la $j^{\text{ème}}$ plus petite valeur de l'échantillon $-2\ln(r(\boldsymbol{y}_1)), \ldots, -2\ln(r(\boldsymbol{y}_R))$, où j est défini par

$$j = (R+1)(1-\alpha).$$

McLachlan et Peel remarquent, sur la base de simulations, que les estimateurs des quantiles de la loi de $-2\ln(r(\boldsymbol{Y}))$ obtenus par rééchantillonnage sont biaisés et proposent une méthode alternative basée sur des réplications de $-2\ln(r(\boldsymbol{y}))$ simulées suivant la loi $P_{\hat{\lambda}_0^*}$, où $\hat{\lambda}_0^*$ est obtenu en tirant plusieurs réplications de $\boldsymbol{Y}$ sous H_0 (en utilisant le paramètre $\hat{\lambda}_0$) puis en calculant $\hat{\lambda}_0^*$ à partir de l'échantillon ainsi obtenu.

Dans Lo, Mendell et Rubin, 2001 [86] les auteurs établissent que dans le cas de mélanges indépendants gaussiens, la statistique $-2\ln(r(\boldsymbol{Y}))$ a même loi qu'une somme pondérée de variables aléatoires indépendantes de loi du χ^2 à un degré de liberté.

Le test du nombre de composants d'un mélange gaussien est traité dans le cas d'indépendance (avec une extension au choix de modèles ARMA) dans Dacunha-Castelle et Gassiat, 1999 [34] par une méthode de paramétrisation localement conique. Il est établi, moyennant des hypothèses de régularité assez fortes (qui interdisent les mélanges gaussiens à variance arbitrairement petite) que le rapport de maximum de log-vraisemblance a même distribution sous H_0 que le supremum, sur un ensemble de scores directionnels, d'une fonction d'un processus gaussien continu.

Parmi les procédures ci-dessus, certaines sont basées sur l'hypothèse d'indépendance des variables aléatoires de $\boldsymbol{Y}$. Le résultat de Lo *et al.*, visiblement, s'appuye fortement sur cette hypothèse et l'adaptation au cas de modèles de Markov cachés généraux semble ne pas être immédiate. En revanche, l'approche de McLachlan et Peel repose essentiellement sur le principe du *bootstrap* dont l'application aux modèles de la famille $\mathcal{D}$ est possible. L'existence de dépendances entre les variables aléatoires complique la maximisation de la vraisemblance évaluée sur un sous-échantillon mais nous présentons en section 3.4.3 une procédure prenant en compte ces dépendances. Enfin, le résultat de Dacunha-Castelle et Gassiat a été généralisé aux chaînes de Markov cachées par Gassiat et Kéribin, 2000 [53] par des calculs beaucoup plus compliqués. L'adaptation de cette méthode à des modèles de Markov cachés plus généraux que les chaînes est sans aucun doute plus complexe encore.

Pourquoi sélectionner un modèle par des tests d'hypothèse ?

Tout d'abord, la sélection de modèles par les tests n'est possible que si les modèles sont emboîtés, ce qui peut être restrictif. Par exemple, les modèles gaussiens présentés figure 3.2 ne sont généralement pas emboîtés : leur sélection est donc exclue par les tests, sauf dans certains cas particuliers. Dans ce qui suit, nous considérons le choix de l'ordre du modèle, ce qui peut effectivement se formuler sous la forme d'un test.

Ainsi, certains problèmes se formulent naturellement sous la forme d'un test de $K = k$ contre $K = k + 1$, comme celui présenté dans Garel, 2002 [51]. L'application considérée consiste à déterminer si un patient est atteint d'une maladie. On sait, certainement pour des raisons biologiques, que la distribution du taux d'une certaine substance présente dans le sang est normale, centrée, de variance inconnue. Chez un patient malade, cette distribution est un mélange de gaussiennes centrées, de variance inconnue. La détection de la maladie chez un patient peut donc se formuler comme le test d'une loi gaussienne contre un mélange de gaussiennes, dans la mesure où l'hypothèse que la vraie loi appartient à l'une des deux familles est réaliste.

Dans d'autres situations, l'hypothèse que les données $\boldsymbol{y}$ sont réellement issues d'un modèle de Markov caché n'est visiblement pas réaliste. Nous verrons au chapitre 4 un exemple de situation où le modèle de chaîne de Markov cachée est un modèle peu réaliste et met néanmoins en évidence certaines caractéristiques du processus observé très utiles pour son interprétation. Le choix du nombre d'états cachés par les tests d'hypothèse pose alors les problèmes suivants :

1. la vraie loi de $\boldsymbol{Y}$ n'appartient pas à l'ensemble des modèles en compétition et les hypothèses H_0 et H_1 des tests sont "fausses" (c'est-à-dire peu réalistes), sauf à considérer des modèles très complexes ;
2. lorsque le nombre n de données observées tend vers $+\infty$, on s'attend à ce que si les hypothèses H_0 et H_1 sont peu réalistes, l'hypothèse H_0 sera systématiquement rejetée ;
3. lorsque toutes les valeurs de l'ensemble $\{1, \ldots, K\}$ sont a priori possibles pour le nombre d'états cachés, on est amené à réaliser des tests successifs. Dans ce cas, il n'y a pas de fondement théorique pour comparer un grand nombre de *p-values*, issues de tests multiples (voir à ce sujet (Kass et Raftery, 1995 [69]) et (Burnham et Anderson, 1998 [17])).

Concernant le troisième point, il est fréquent de tester l'hypothèse $H_0 : K = 1$ contre

$H_1 : K = 2$ en choisissant le modèle correspondant à $K = 1$ si H_0 n'est pas rejetée. Dans le cas contraire, on teste $H_0 : K = k$ contre $H_1 : K = k + 1$ avec des valeurs croissantes de k jusqu'à ce que H_0 ne soit pas rejetée, ce qui permet de choisir le modèle le plus parcimonieux. Le problème de cette approche est qu'alors, les valeurs supérieures K ne sont pas envisagées alors qu'elles pourraient donner des résultats "meilleurs" (y compris au sens des tests, pour ne citer qu'eux). L'exemple choisi dans McLachlan et Peel, 2000 [91] illustre ce problème à travers le jeu de données philatéliques Hidalgo de Mexico 1872. La méthodologie ci-dessus conduit à choisir un mélange gaussien à trois composants parce que le test de $H_0 : K = 3$ contre $H_1 : K = 4$ est le premier à ne pas rejeter H_0, alors que le test de $H_0 : K = 4$ contre $H_1 : K = 5$ rejetterait H_0 et la continuation des tests conduirait au choix d'un modèle à sept composants.

Dans Burnham et Anderson, 1998 [17] les auteurs comparent les procédures de tests successifs avec une procédure basée sur le critère d'information AIC dans le cadre de modèles linéaires pour la modélisation de phénomènes réels en biologie. Leur conclusion est que AIC a tendance à choisir des modèles plus parcimonieux que les tests successifs, alors même qu'en général ce critère tend à choisir des modèles trop complexes (voir par exemple Koehler et Murphee, 1988 [73]). Cependant, le fait que AIC sélectionne des modèles plus parcimonieux que les tests se conçoit, dans le cas par exemple d'un nombre de données très élevé. Les expérimentations réalisées dans cette thèse ne comprennent pas de cas de sélection de modèles par les tests mais il aurait été instructif de mettre en œuvre ces techniques pour voir leur comportement sur les modèles de Markov cachés, dans le contexte de données simulées et réelles.

3.4 Approches issues de la théorie de l'information

Burnham et Anderson, 1998 [17] appellent *approches issues de la théorie de l'information* des approches basées sur les hypothèses et le principe suivants :

- les données $\boldsymbol{y}$ sont issues d'une vraie loi P_0 inconnue ;
- l'ensemble des différents modèles candidats pour l'inférence ne contient pas P_0 ;
- le principe est de déterminer le modèle $\mathcal{M}(\mathcal{G}, \mathcal{S}, \mathcal{P})$ et la loi $\tilde{P}$ dans ce modèle minimisant la divergence de Kullback-Leibler de $\tilde{P}$ à P_0.

La divergence de Kullback-Leibler de $\tilde{P}$ à P_0 est définie par l'équation (3.2). Cette divergence s'interprète comme la *perte d'information* due à l'approximation de la loi P_0 par la loi $\tilde{P}$. Sélectionner un modèle par cette méthode revient donc, pour chaque modèle $\mathcal{M}(\mathcal{G}_i, \mathcal{S}_i, \mathcal{P}_i)$, à minimiser $\mathrm{KL}(P_0, P_{\lambda_i})$ sur tous les $\lambda_i \in \mathcal{M}(\mathcal{G}_i, \mathcal{S}_i, \mathcal{P}_i)$, puis à sélectionner le couple $(\mathcal{M}(\mathcal{G}_i, \mathcal{S}_i, \mathcal{P}_i), \lambda_i)$ qui réalise ce minimum. La principale difficulté provient du fait que P_0 et λ sont inconnus. Dans cette section, nous présentons deux méthodes remplaçant λ par son estimateur de maximum de vraisemblance $\hat{\lambda}$. La première méthode est basée sur les développements limités de la vraisemblance au voisinage de λ et conduit aux critères de vraisemblance pénalisée de type AIC (voir Burnham et Anderson, 1998 [17]). La deuxième méthode est basée, à l'origine, sur l'évaluation de la capacité prédictive des modèles mais peut néanmoins être rattachée à la théorie de l'information : il s'agit de la validation croisée (voir Stone, 1974 [113] et Stone, 1977 [114]). Après avoir explicité l'origine de ces deux approches, nous présentons un algorithme pour la mise en œuvre de la validation croisée dans le cadre des modèles de Markov cachés de la famille $\mathcal{D}$. Cet algorithme, en substance, répond à la question du calcul de probabilités et de l'estimation des paramètres dans le contexte de la suppression d'observations à valeurs continues, laissée en suspend dans le chapitre 2.

Remarque 3.2 *L'approche dite "Minimum Discrimination Information" ou MDI, de Shore et Johnson, 1980 [109], peut être rattachée aux méthodes issues de la théorie de l'information. Il s'agit en effet d'une méthode d'estimation des paramètres dans les modèles de Markov cachés (à l'origine, les chaînes de Markov cachées), qui consiste à minimiser la divergence de Kullback-Leibler, parmi toutes les distributions vérifiant des contraintes données sur les moments et tous les paramètres possibles. L'approche MDI est une généralisation de l'approche par maximum d'entropie de Cover et Thomas, 1991 [30]. En comparant la valeur des minima pour différents modèles, cette approche peut également servir de base à des méthodes de sélection. Cependant, sa mise en œuvre semble difficile (en particulier pour la détermination de l'ensemble des lois satisfaisant les contraintes de moments, voir Ephraim et Mehrav, 2002 [47]), c'est pourquoi nous ne présentons pas cette approche plus en détail.*

3.4.1 Critères d'information

Dans cette section, nous présentons les critères d'information AIC et ses variantes. Ces critères de sélection de modèles sont obtenus par une approximation de la quantité $\mathrm{KL}(P_0, P_\lambda)$ à minimiser, par des développements limités de la log-vraisemblance (cf. Burnham et Anderson, 1998 [17]). Nous sommes alors amenés à faire l'hypothèse qu'il existe une unique valeur du paramètre minimisant $\mathrm{KL}(P_0, P_\lambda)$ quand λ décrit $\mathcal{M}(\mathcal{G}, \mathcal{S}, \mathcal{P})$, cette valeur étant notée λ_0[1]. Nous appellerons alors la quantité $\mathrm{KL}(P_0, P_{\lambda_0})$ la *divergence de Kullback-Leibler du modèle* $\mathcal{M}(\mathcal{G}, \mathcal{S}, \mathcal{P})$ *à la loi* P_0. La divergence de Kullback-Leibler est correctement définie sous des conditions données dans White, 1982 [119], à savoir que les quantités $|\ln(P_\lambda)|$ sont majorées par une fonction mesurable (par rapport à P_0) indépendante de λ. L'existence de valeurs minimisant $\mathrm{KL}(P_0, P_\lambda)$ est assurée sous des conditions données dans Bunke et Milhaud, 1998 [16]. L'une d'entre elles est que Λ soit un convexe fermé de $\mathbb{R}^d$ d'intérieur non vide et que le support de P_λ ne dépende pas de λ. Les autres conditions portent sur le caractère fini du supremum de la divergence de Kullback-Leibler sur une sphère de centre λ, pour tout $\lambda \in \Lambda$, et sur la régularité de la dérivée de P_λ, ces hypothèses assurant en particulier la continuité de KL, le fait que KL atteigne ses minima et que l'ensemble des minima soit compact. L'unicité de λ_0 est supposée dans les références ci-dessus ; elle est liée, entre autres, à la notion d'identifiabilité.

Le paramètre λ_0 dépend de la vraie loi P_0 inconnue, donc il est lui-même inconnu. Puisque l'on dispose d'une réalisation $\boldsymbol{y}$ de $\boldsymbol{Y}$ suivant la loi P_0, on peut utiliser l'estimateur empirique suivant associé à la divergence de Kullback-Leibler :

$$-\frac{1}{n} \sum_{u=1}^{n} \ln(P_\lambda(Y_u = y_u)), \tag{3.3}$$

[1] La valeur λ_0 est parfois appelée "pseudo-vraie" valeur du paramètre.

égal à la log-vraisemblance $\ln(\mathcal{L}_{\boldsymbol{y}}(\lambda))$, au facteur $-n^{-1}$ près, dans le cas où les variables aléatoires observées sont indépendantes (dans le cas contraire, la quantité (3.3) est appelée *log-vraisemblance marginale* ; nous la noterons $\ln(\tilde{\mathcal{L}}_{\boldsymbol{y}}(\lambda))$). Par la loi forte des grands nombres, on a

$$-\frac{1}{n}\sum_{u=1}^{n}\ln(P_{\lambda}(Y_u=y_u)) \xrightarrow{P_0\,p.s.} -\int P_0(\tilde{\boldsymbol{y}})\ln(P_{\lambda}(\tilde{\boldsymbol{y}}))\,d\tilde{\boldsymbol{y}}.$$

C'est pourquoi il est naturel d'estimer λ_0 par l'EMV $\hat{\lambda}(\boldsymbol{y})$, cette notation explicitant la dépendance de cet estimateur vis-à-vis des données. De plus, sous des conditions de régularité données dans White, 1982 [119], l'EMV $\hat{\lambda}$ tend P_0-presque sûrement vers λ_0. La quantité $\mathrm{KL}(P_0, P_{\lambda_0})$ inconnue est alors estimée par l'approximation suivante de la divergence de Kullback-Leibler

$$\mathrm{KL}(P_0, P_{\hat{\lambda}(\boldsymbol{y})}) = \int P_0(\tilde{\boldsymbol{y}})\ln\frac{P_0(\tilde{\boldsymbol{y}})}{P_{\hat{\lambda}(\boldsymbol{y})}(\tilde{\boldsymbol{y}})}\,d\tilde{\boldsymbol{y}}. \tag{3.4}$$

Dans le contexte fréquent où l'on désire qu'un modèle ait en moyenne une bonne capacité prédictive, on choisira comme critère de sélection la valeur moyenne de la quantité (3.4) pour des réplications indépendantes de $\boldsymbol{y}$, autrement dit l'espérance de la quantité par rapport à $\boldsymbol{Y}$, soit

$$\mathbb{E}_{P_0}[\mathrm{KL}(P_0, P_{\hat{\lambda}(\boldsymbol{Y})})]$$

également noté

$$\mathbb{E}_{\boldsymbol{Y}}[\mathrm{KL}(P_0, P_{\hat{\lambda}(\boldsymbol{Y})})],$$

étant entendu que $\boldsymbol{Y}$ suit la loi P_0. Nous nous basons principalement sur Burnham et Anderson, qui justifient les critères d'information comme suit. Les auteurs notent que la quantité (3.4) se réécrit

$$\mathrm{KL}(P_0, P_{\hat{\lambda}(\boldsymbol{y})}) = \mathrm{Constante} - \mathbb{E}_{\tilde{\boldsymbol{Y}}}[\ln(P_{\hat{\lambda}(\boldsymbol{y})}(\tilde{\boldsymbol{Y}}))],$$

où $\tilde{\boldsymbol{Y}}$ représente un échantillon test de même taille que $\boldsymbol{Y}$. Le critère à minimiser se réécrit alors

$$\mathbb{E}_{\boldsymbol{Y}}[\mathrm{KL}(P_0, P_{\hat{\lambda}(\boldsymbol{Y})})] = \mathrm{Constante} - \mathbb{E}_{\boldsymbol{Y}}[\mathbb{E}_{\tilde{\boldsymbol{Y}}}[\ln(P_{\hat{\lambda}(\boldsymbol{Y})}(\tilde{\boldsymbol{Y}}))]].$$

Par conséquent, cette méthode revient à déterminer le modèle $\mathcal{M}(\mathcal{G}_i, \mathcal{S}_i, \mathcal{P}_i)$ qui minimise en moyenne, parmi tous les modèles $\{\mathcal{M}(\mathcal{G}_j, \mathcal{S}_j, \mathcal{P}_j)\}_j$, un estimateur de la divergence de Kullback-Leibler du modèle à la vraie loi. Le fait de considérer la double espérance par rapport à $\boldsymbol{Y}$ et à $\tilde{\boldsymbol{Y}}$ revient à utiliser, pour valider un modèle, un échantillon indépendant de celui qui a servi à l'estimer.

Il est tentant d'estimer la quantité $\ln(P_{\hat{\lambda}(\boldsymbol{y})}(\tilde{\boldsymbol{y}}))$ par le maximum de vraisemblance $\ln(P_{\hat{\lambda}(\boldsymbol{y})}(\boldsymbol{y}))$, vu la ressemblance entre ces deux quantités ; cependant Akaike, qui est à l'origine du critère (3.5), a montré que la vraisemblance est un estimateur biaisé positivement de ce critère (voir Akaike, 1973 [3]). Ce biais provient du fait que dans le cas où la log-vraisemblance est utilisée comme approximation de $\ln(P_{\hat{\lambda}(\boldsymbol{y})}(\tilde{\boldsymbol{y}}))$, les données $\boldsymbol{y}$ sont utilisées à la fois pour identifier le modèle et pour le valider ; ainsi, $\mathbb{E}_{\boldsymbol{Y}}[\ln(P_{\hat{\lambda}(\boldsymbol{Y})}(\boldsymbol{Y}))]$ surestime systématiquement $\mathbb{E}_{\boldsymbol{Y}}[\mathbb{E}_{\tilde{\boldsymbol{Y}}}[\ln(P_{\hat{\lambda}(\boldsymbol{Y})}(\tilde{\boldsymbol{Y}}))]]$. Par ailleurs, il est connu que dans le cas de modèles emboîtés, le modèle le plus complexe maximise toujours la vraisemblance. Les critères d'informations essayent donc d'estimer ce biais. Vu que

$$\mathbb{E}_{\boldsymbol{Y}}[\ln(P_{\hat{\lambda}(\boldsymbol{Y})}(\boldsymbol{Y}))] = \mathbb{E}_{\boldsymbol{Y}}[\mathbb{E}_{\tilde{\boldsymbol{Y}}}[\ln(P_{\hat{\lambda}(\boldsymbol{Y})}(\tilde{\boldsymbol{Y}}))]] + \text{biais},$$

on est amené à maximiser des critères de la forme

$$\ln(P_{\hat{\lambda}(\boldsymbol{y})}(\boldsymbol{y})) - \text{estimation du biais}.$$

Ces critères pénalisent la vraisemblance par un terme positif qui *pénalise* les modèles les plus complexes (c'est-à-dire ayant le nombre le plus élevé de paramètres) et s'écrivent donc

$$\text{IC} = \ln(\mathcal{L}_{\boldsymbol{y}}(\hat{\lambda})) - \text{pénalité} \ ;$$

ce sont des critères de vraisemblance pénalisée. Ainsi, le critère AIC (*An Information Criterion*) d'Akaike essaye d'estimer ce biais en effectuant un développement limité en λ de la quantité $\ln(P_\lambda(\boldsymbol{y}))$ dans un voisinage de λ_0 contenant $\hat{\lambda}$. Les détails de ce développement limité sont donnés dans Burnham et Anderson, 1998 [17] et conduisent au résultat suivant

$$\mathbb{E}_{\boldsymbol{Y}}[\mathbb{E}_{\tilde{\boldsymbol{Y}}}[\ln(P_{\hat{\lambda}(\boldsymbol{Y})}(\tilde{\boldsymbol{Y}}))]] \approx \mathbb{E}_{\boldsymbol{Y}}[\ln(P_{\hat{\lambda}(\boldsymbol{Y})}(\boldsymbol{Y}))] - \text{tr}(\mathcal{J}_{\lambda_0}\mathcal{I}_{\lambda_0}^{-1}) \qquad (3.5)$$

où

$$\mathcal{J}_\lambda = \mathbb{E}_{P_0}\left[(\nabla_\lambda \ln(P_\lambda(\boldsymbol{Y})))^t(\nabla_\lambda \ln(P_\lambda(\boldsymbol{Y})))\right]$$

et $\mathcal{I}_\lambda$ est la matrice

$$\mathcal{I}_\lambda = \mathbb{E}_{P_0}[-\nabla_\lambda^2 \ln(P_\lambda(\boldsymbol{Y}))],$$

où la notation ∇_λ^2 désigne l'opérateur de dérivée seconde par rapport à λ. Ainsi, $\mathcal{I}_{\lambda_0}$ est l'espérance de la dérivée seconde, prise en $\lambda = \lambda_0$, de la fonction $\lambda \longrightarrow \ln(P_\lambda(\boldsymbol{Y}))$.

Notons que Ripley, 1996 [104], sous l'hypothèse d'indépendance et avec des conditions de régularité standard, utilise un point de vue un peu différent en cherchant directement un développement limité de l'espérance de la *déviance*

$$D(\tilde{Y}) = 2[\ln(P_0(\tilde{Y})) - \ln(P_{\hat{\lambda}(\boldsymbol{Y})}(\tilde{Y}))] = 2\text{KL}(P_0, P_{\hat{\lambda}(\boldsymbol{Y})}),$$

où $\tilde{Y}$ représente un "individu test" indépendant de $\boldsymbol{Y}$. Le processus $\tilde{\boldsymbol{Y}}$ désigne encore une fois un "échantillon test", c'est-à-dire un échantillon de même taille que $\boldsymbol{Y}$ et indépendant de $\boldsymbol{Y}$. L'espérance de la déviance ci-dessus est prise par rapport à $\tilde{Y}$: il ne s'agit plus, comme dans Burnham et Anderson, de la double espérance prise par rapport à $\tilde{Y}$ puis $\boldsymbol{Y}$. L'utilisation d'un échantillon test $\tilde{\boldsymbol{Y}}$ est due au fait que $\hat{\lambda}(\boldsymbol{y})$ est déterminé de manière à minimiser la déviance sur les données d'apprentissage $\boldsymbol{y}$ et sous-estime donc la déviance sur un ensemble test de taille comparable. L'auteur montre l'approximation suivante

$$n\mathbb{E}_{\tilde{Y}}[D(\tilde{Y})] = \mathbb{E}_{\tilde{\boldsymbol{Y}}}[\ln(P_0(\tilde{\boldsymbol{Y}})) - \ln(P_{\hat{\lambda}(\boldsymbol{y})}(\tilde{\boldsymbol{Y}}))] + 2\text{tr}(\mathcal{J}_{\lambda_0}\mathcal{I}_{\lambda_0}^{-1}) + \mathcal{O}_{P_0}(\frac{1}{\sqrt{n}}). \qquad (3.6)$$

Le critère AIC peut être obtenu à partir des équations (3.5) ou (3.6). Dans le cas de l'équation (3.5), la quantité inconnue $\mathbb{E}_{\boldsymbol{Y}}[\ln(P_{\hat{\lambda}(\boldsymbol{Y})}(\boldsymbol{Y}))]$ est estimée par le maximum de log-vraisemblance $\ln(P_{\hat{\lambda}}(\boldsymbol{Y}))$. Le biais de cette approximation est alors estimé par $\mathrm{tr}(\mathcal{J}_{\lambda_0}\mathcal{I}_{\lambda_0}^{-1})$, grâce à un développement limité dans un voisinage de λ_0 contenant $\hat{\lambda}$. Dans le cas de l'expression (3.6), la quantité $\mathbb{E}_{\tilde{\boldsymbol{Y}}}[\ln(P_0(\tilde{\boldsymbol{Y}}))]$ constante par rapport à $\boldsymbol{y}$ est éliminée du critère à minimiser. La quantité $\mathbb{E}_{\tilde{\boldsymbol{Y}}}[\ln(P_{\hat{\lambda}(\boldsymbol{y})}(\tilde{\boldsymbol{Y}}))]$ est estimée de manière empirique par la vraisemblance

$$\sum_u \ln(P_{\hat{\lambda}(\boldsymbol{y})}(Y_u = y_u)),$$

ce qui crée des fluctuations d'échantillonnage. Par le théorème central limite, l'erreur d'approximation est d'ordre $\mathcal{O}_{P_0}(\sqrt{n})$. Dans les deux cas, la quantité à minimiser est le critère

$$-\ln(\mathcal{L}_{\boldsymbol{y}}(\hat{\lambda})) + \mathrm{tr}(\mathcal{J}_{\lambda_0}\mathcal{I}_{\lambda_0}^{-1}),$$

appelé NIC, où la trace fait intervenir des quantités qui dépendent du paramètre inconnu λ_0, et doit donc être approximée. Dans le critère AIC, cette trace est estimée par le nombre d de paramètres réels indépendants du modèle, d'où la définition

$$\mathrm{AIC}(\mathcal{M}(\mathcal{G},\mathcal{S},\mathcal{P})) = -\ln(\mathcal{L}_{\boldsymbol{y}}(\hat{\lambda})) + d.$$

Cette approximation est motivée par le fait que si $P_0 \in \mathcal{M}(\mathcal{G},\mathcal{S},\mathcal{P})$, alors $\mathcal{J}_{\lambda_0} = \mathcal{I}_{\lambda_0}$. Ce point de vue peut sembler contradictoire avec l'approche à l'origine de AIC, à savoir qu'aucun modèle ne contient P_0 et que le but est justement de déterminer le modèle le plus proche de P_0 au sens de la divergence de Kullback-Leibler. D'après Burnham et Anderson, cette approximation est utilisable si l'ensemble des modèles en compétition contient une loi proche de P_0, auquel cas l'approximation $\mathrm{tr}(\mathcal{J}_{\lambda_0}\mathcal{I}_{\lambda_0}^{-1}) = d$ est raisonnable. Cependant, alors que les calculs ci-dessus permettaient de contrôler l'erreur d'approximation, le fait d'estimer $\mathrm{tr}(\mathcal{J}_{\lambda_0}\mathcal{I}_{\lambda_0}^{-1})$ par d dans le cas où P_0 n'appartient pas à $\{P_\lambda\}_\lambda$ crée également un biais qui n'est pas pris en compte par AIC. En réalité, lorsque les modèles considérés ne contiennent pas P_0, AIC a tendance à choisir des modèles de plus en plus complexes lorsque n augmente (voir Ripley, 1996 [104]).

D'autres estimateurs de cette trace ont été proposés et ont conduit à autant de critères d'information. Citons par exemple le critère TIC de Takeuchi, 1976 [115] qui est basé sur une estimation empirique des matrices $\mathcal{J}_{\lambda_0}$ et $\mathcal{I}_{\lambda_0}$, utilisant typiquement des méthodes de bootstrap. Burnham et Anderson notent la forte variabilité des estimateurs obtenus et s'interrogent sur la pertinence de cette méthode, à moins peut-être de disposer d'échantillons de très grande taille.

Utilisation du critère AIC pour le choix du nombre d'états cachés

Parmi les hypothèses qui justifient les équations (3.5) et (3.6) figure celle de normalité asymptotique de l'estimateur $\hat{\lambda}$ sous P_0, à savoir

$$\sqrt{n}(\hat{\lambda} - \lambda_0) \longrightarrow \mathcal{N}_d(\mathcal{I}_{\lambda_0}^{-1}\mathcal{J}_{\lambda_0}\mathcal{I}_{\lambda_0}^{-1}).$$

Des conditions nécessaires pour que cette hypothèse soit satisfaite sont données dans White, 1982 [119]. Or nous avons vu dans la section 3.3 que la normalité asymptotique de $\hat{\lambda}$ peut ne pas être vérifiée dans le contexte du choix du nombre d'états cachés dans les modèles de Markov cachés, suivant la loi P_0. Ceci est dû à la possibilité que le modèle soit "non identifiable", en un sens légèrement différent de la notion d'identifiabilité vue en section 2.2. Par exemple, dans le cas de mélanges indépendants, si la famille de lois d'émission est identifiable et si l'on contraint les proportions à être non nulles et les paramètres d'émission à être distincts, les modèles sont identifiables. Cependant, dans le cas de la sélection de modèles, nous sommes conduits à considérer des modèles à proportions nulles ou à paramètres d'émission égaux lorsque le nombre de composants est supérieur au "vrai" (ou au "pseudo-vrai") nombre de composants.

Pour cette raison, la justification théorique de AIC n'est pas valide dans ce cadre (voir à ce sujet, par exemple, McLachlan et Peel, 2000 [91]). En outre, il est connu que ce critère a tendance à surestimer le nombre d'états cachés, dans le contexte des mélanges indépendants (voir par exemple les expérimentations de Cutler et Windham, 1993 [33] ou encore Celeux et Soromenho, 1996 [25]). Des précisions sur la surestimation du nombre d'états cachés par AIC sont données dans la section 3.6. Enfin, notons que la justification du critère AIC donnée par Burnham et Anderson, 1998 [17] ne s'appuie pas sur l'hypothèse d'indépendance des variables aléatoires observées. Cependant, les hypothèses effectuées sont plus difficiles à vérifier pour les modèles de Markov cachés (en particulier la normalité asymptotique de l'EMV). En tout état de cause, ces hypothèses peuvent être fausses dans le contexte du choix du nombre d'états cachés.

3.4.2 Validation croisée multiple

La validation croisée est une technique ancienne, citée dans des articles de modélisation psychométrique dans les années 1930. Les données sont divisées en deux groupes : le premier est utilisé pour l'ajustement du modèle et le second pour sa validation. Dans les articles fondateurs de la validation croisée, une seule observation à la fois est utilisée pour la validation (on parle alors de *validation croisée "à une donnée exclue[2]"*). La procédure d'identification des paramètres et d'évaluation du modèle est alors répétée n fois, c'est-à-dire autant de fois qu'il y a de données.

Jusqu'à l'article de Stone, 1974 [113] cette technique était vue avant tout comme un moyen d'évaluer un modèle, mais l'auteur suggère son utilisation pour choisir entre plusieurs modèles. Dans certains contextes, dont celui des modèles de Markov cachés, il paraît déraisonnable d'évaluer le modèle avec une seule observation, notamment à cause des dépendances entre variables aléatoires observées. De plus, répéter la procédure n fois conduit à un nombre très important de calculs, vu la complexité de l'algorithme d'estimation des paramètres. C'est pourquoi la validation croisée a été généralisée à la validation avec M données ; nous la nommerons dans ce cas *validation croisée multiple* ou *"à M données exclues"* [3].

[2] *"leave-one-out cross validation* en anglais.

[3] *multifold cross validation* en anglais.

Principe

Dans le contexte de la sélection de modèles linéaires, Zhang 1993 [126] propose les variantes suivantes de la validation croisée multiple : supposons que $n = Rm$.

- Par définition, le critère MCV* (pour *Multifold Cross Validation*) est obtenu en considérant une partition fixée des observations en R classes, en général déterminée de manière aléatoire. Chaque classe $\boldsymbol{y}^{(r)}$ composant la partition est alors utilisée à son tour pour évaluer le modèle, l'ensemble des données observées restantes, noté $\boldsymbol{y}^{(-r)}$, étant utilisé pour calculer l'EMV $\hat{\lambda}^{(-r)}$ du paramètre. Le critère MCV*, qui dépend de la partition utilisée, est alors défini par
$$\sum_{r=1}^{R} \ln(P_{\hat{\lambda}^{(-r)}}(\boldsymbol{Y}^{(r)} = \boldsymbol{y}^{(r)})).$$
- Le critère MCV est défini comme la moyenne, sur toutes les partitions possibles des observations en R classes, des critères MCV* associés à chaque partition.
- Par définition, le critère RLT (pour *Repeated Learning-Testing*) est obtenu en considérant un nombre fixé V de partitions des observations en R classes, chaque partition étant tirée aléatoirement. Pour chaque valeur de v, on choisit au hasard l'une des classes, notée $\boldsymbol{y}^{(r_v)}$, pour évaluer le modèle, l'ensemble des données observées restantes $\boldsymbol{y}^{(-r_v)}$ étant utilisé pour calculer l'EMV $\hat{\lambda}^{(-r_v)}$ du paramètre. Le critère RLT est défini par
$$\sum_{v=1}^{V} \ln(P_{\hat{\lambda}^{(-r_v)}}(\boldsymbol{Y}^{(r_v)} = \boldsymbol{y}^{(r_v)})).$$

Le calcul du critère MCV nécessite de considérer toutes les partitions possibles des données en R classes, ce qui est en général infaisable vu le temps de calcul en jeu. Les critères MCV* et RLT ont donc été élaborés, afin d'approcher le critère MCV. Notons que le critère MCV* est sujet à une variabilité due au tirage aléatoire de la partition. Cette variabilité peut être atténuée par répétition des tirages, de manière à en obtenir une version *Monte-Carlo*, notée MCMCV*.

Les critères de validation croisée vus comme critères d'information

De manière évidente, les critères de validation croisée sélectionnent les modèles sur la base de leur capacité prédictive. Nous rappelons ci-dessous les liens entre la validation croisée et les critères d'information. Nous avons évoqué dans la section 3.4.1 que les critères d'information reposent en partie sur l'utilisation d'un échantillon pour valider le modèle, indépendant de celui qui a servi à l'identifier.

Un premier point de vue permettant d'expliciter ce lien avec les critères d'information est d'introduire la validation croisée dans le cadre d'une estimation empirique de la déviance
$$D(\tilde{Y}) = 2[\ln(P_0(\tilde{Y})) - \ln(P_{\hat{\lambda}(\boldsymbol{Y})}(\tilde{Y}))] = 2\mathrm{KL}(P_0, P_{\hat{\lambda}(\boldsymbol{Y})})$$
qui est une mesure de l'adéquation du modèle. Lorsque l'on utilise la déviance en sélection de modèles, $\hat{\lambda}(\boldsymbol{y})$ est déterminé de manière à minimiser cette déviance sur les données

d'apprentissage (autrement dit à minimiser la quantité $2[\ln(P_0(\boldsymbol{Y} = \boldsymbol{y})) - \ln(P_\lambda(\boldsymbol{Y} = \boldsymbol{y}))]$), ce qui fait que la déviance évaluée sur un ensemble test de taille comparable est en général plus grande. Un estimateur empirique de la déviance basé uniquement sur les données d'apprentissage est alors biaisé, au sens où il sous-estime systématiquement la déviance réelle. Notons que les critères présentés dans la section 3.4.1 tentent également d'estimer le biais résultant d'une estimation de la déviance minimale en utilisant le maximum de vraisemblance, essentiellement grâce à des développements limités et des arguments asymptotiques.

Un estimateur empirique de ce biais est obtenu par validation croisée, en estimant de manière naturelle la différence entre la déviance estimée sur un ensemble d'apprentissage et sur un ensemble de test. On considère alors une partition $\{\boldsymbol{y}_0, \boldsymbol{y}_1\}$ de $\boldsymbol{y}$, où typiquement les deux ensembles sont de même cardinal. Le paramètre λ est estimé par $\hat{\lambda}_i$ à partir de $\boldsymbol{y}_i$ pour $i \in \{0; 1\}$. L'estimateur empirique de la déviance est

$$2\ln(P_0(\boldsymbol{y}_0)) - 2\ln(P_{\hat{\lambda}_0}(\boldsymbol{y}_0)) + 2\ln(P_0(\boldsymbol{y}_1)) - 2\ln(P_{\hat{\lambda}_1}(\boldsymbol{y}_1)). \tag{3.7}$$

D'après ce qui précède, le biais de cet estimateur est estimé par

$$\begin{aligned} \mathrm{B}(\boldsymbol{y}) = \quad & 2\ln(P_0(\boldsymbol{y}_0)) - 2\ln(P_{\hat{\lambda}_0}(\boldsymbol{y}_0)) - [2\ln(P_0(\boldsymbol{y}_1)) - 2\ln(P_{\hat{\lambda}_0}(\boldsymbol{y}_1))] & (3.8) \\ & + 2\ln(P_0(\boldsymbol{y}_1)) - 2\ln(P_{\hat{\lambda}_1}(\boldsymbol{y}_1)) - [2\ln(P_0(\boldsymbol{y}_0)) - 2\ln(P_{\hat{\lambda}_1}(\boldsymbol{y}_0))] & (3.9) \end{aligned}$$

où $2\ln(P_0(\boldsymbol{y}_0)) - 2\ln(P_{\hat{\lambda}_0}(\boldsymbol{y}_0))$ est un estimateur empirique de la déviance sur l'ensemble d'apprentissage et où $2\ln(P_0(\boldsymbol{y}_1)) - 2\ln(P_{\hat{\lambda}_0}(\boldsymbol{y}_1))$ estime la déviance sur un ensemble test, et de même en inversant les rôles de $\boldsymbol{y}_0$ et de $\boldsymbol{y}_1$. Ainsi, la formule (3.8) représente la différence entre la déviance sur un ensemble d'apprentissage et un ensemble test et $\frac{1}{2}\mathrm{B}(\boldsymbol{y})$ correspond à une différence moyenne. Le biais estimé $\mathrm{B}(\boldsymbol{y})$ est alors soustrait à l'estimateur empirique biaisé (3.7), ce qui donne l'estimateur

$$2\ln(P_0(\boldsymbol{y}_0)) - 2\ln(P_{\hat{\lambda}_0}(\boldsymbol{y}_0)) + 2\ln(P_0(\boldsymbol{y}_1)) - 2\ln(P_{\hat{\lambda}_1}(\boldsymbol{y}_1)) - \mathrm{B}(\boldsymbol{y}). \tag{3.10}$$

Cette expression est ensuite simplifiée en utilisant la définition de $\mathrm{B}(\boldsymbol{y})$ ci-dessus et en éliminant les termes indépendants des $\hat{\lambda}_i$. On est finalement amené à minimiser l'expression

$$-2\ln(P_{\hat{\lambda}_0}(\boldsymbol{y}_1)) - 2\ln(P_{\hat{\lambda}_1}(\boldsymbol{y}_0)),$$

donc à maximiser un critère de validation croisée. Dans le cas où les $\boldsymbol{y}_i$ sont effectivement des sous-échantillons de même taille, on parle de demi-échantillonnage[4], ce qui est bien un cas particulier de validation croisée multiple avec $R = 2$ et $m = \dfrac{n}{2}$.

D'autre part, Stone 1977 [114] explicite le lien entre la validation croisée à une donnée exclue et les critères d'information, dans le cas de variables aléatoires observées indépendantes. Nous notons $\hat{\lambda}$ l'estimateur de maximum de vraisemblance – calculé à partir de $\boldsymbol{y}$ – et $\hat{\lambda}_{-u}$ l'EMV calculé à partir de $\boldsymbol{y}$ *privé de l'observation* y_u. Nous notons également $l_y(\lambda) = \ln(P_\lambda(Y = y))$. Le critère de validation croisée à une donnée exclue est défini par

$$\mathrm{CV} = \sum_u l_{\hat{\lambda}_{-u}}(y_u).$$

[4] *half-sampling* en anglais.

La preuve de Stone se base en partie sur un développement de Taylor de la vraisemblance et de sa dérivée dans un voisinage de $\hat{\lambda}$, d'où

$$\mathrm{CV} = \ln(\mathcal{L}_{\boldsymbol{y}}(\hat{\lambda})) + \sum_u {}^t(\hat{\lambda}_{-u} - \hat{\lambda}) l'_{y_u}(\hat{\lambda} + a_u(\hat{\lambda}_{-u} - \hat{\lambda}))$$

et pour tout u dans $\{1, \ldots, n\}$,

$$(\ln \mathcal{L}_{\boldsymbol{y}})'(\hat{\lambda}_{-u}) = \left[(\ln \mathcal{L}_{\boldsymbol{y}})''(\hat{\lambda} + b_u(\hat{\lambda}_{-u} - \hat{\lambda}))\right](\hat{\lambda}_{-u} - \hat{\lambda}),$$

où les $(a_u)_{u\in\{1,\ldots,n\}}$ et les $(b_u)_{u\in\{1,\ldots,n\}}$ sont certains scalaires de norme inférieure à un. Alors sous les hypothèses suivantes :

(i) unicité de l'estimateur de maximum de vraisemblance,

(ii) unicité du paramètre λ_0 minimisant la distance de Kullback-Leibler du modèle $\mathcal{M}(\mathcal{G}, \mathcal{S}, \mathcal{P})$ à la loi P_0,

(iii) convergence P_0 presque-sûre de $\hat{\lambda}$ vers λ_0 quand $n \to +\infty$,

(iv) convergence P_0 presque-sûre de $\hat{\lambda}_{-u}$ vers λ_0 quand $n \to +\infty$, pour la suppression d'une quelconque des observations y_u,

(v) inversibilité de la matrice $(\ln \mathcal{L}_{\boldsymbol{y}})''(\hat{\lambda} + b_u(\hat{\lambda}_{-u} - \hat{\lambda}))$,

(vi) convergence P_0 presque-sûre vers $-\mathcal{I}_{\lambda_0}$ de la quantité

$$\frac{1}{n}(\ln \mathcal{L}_{\boldsymbol{y}})''(\hat{\lambda} + b_u(\hat{\lambda}_{-u} - \hat{\lambda})),$$

(vii) convergence P_0 presque-sûre vers $\mathcal{J}_{\lambda_0}$ de la quantité

$$\frac{1}{n}\sum_u l'_{y_u}(\hat{\lambda} + a_u(\hat{\lambda}_{-u} - \hat{\lambda}))^t l'_{y_u}(\hat{\lambda}_{-u}) \ ;$$

le critère CV est équivalent quand $n \to +\infty$ au critère d'information NIC

$$\ln(\mathcal{L}_{\boldsymbol{y}}(\hat{\lambda})) - \mathrm{tr}(\mathcal{J}_{\lambda_0}\mathcal{I}_{\lambda_0}^{-1}),$$

quantité dont nous savons qu'elle estime, à une constante près et avec un biais qui tend vers zéro, l'espérance de la divergence de Kullback-Leibler de $P_{\hat{\lambda}}$ à P_0. Ce résultat de Stone établit que la validation croisée, sous les hypothèses ci-dessus, conduit à choisir asymptotiquement le modèle le plus proche de P_0 au sens de la divergence de Kullback-Leibler. L'intérêt de la validation croisée par rapport au critère AIC est que l'hypothèse que $\mathcal{J}_{\lambda_0} = \mathcal{I}_{\lambda_0}$ n'est pas utile.

Si l'on souhaite prouver que pour les modèles de Markov cachés, le critère de validation croisée conduit également à choisir asymptotiquement le modèle le plus proche de P_0 au sens de la divergence de Kullback-Leibler (ce qui permet aussi de montrer la consistance de ce critère), on peut envisager d'adaptater la preuve de Stone. Dans le cadre des modèles de Markov cachés, cette adaptation comporte a priori deux difficultés :

1. la prise en compte des dépendances entre variables aléatoires observées, qui fait que l'équation
$$\ln(\mathcal{L}_{\boldsymbol{y}}(\lambda)) = \sum_u l_{y_u}(\lambda)$$
n'est plus vérifiée ;
2. la vérification d'hypothèses de régularité analogues à celles de Stone pour les modèles de mélange.

Le problème de la prise en compte des dépendances pourrait éventuellement être éliminée par une validation basée sur la log-vraisemblance marginale au lieu de la log-vraisemblance (voir section 3.6). D'autre part, les hypothèses de régularité (i) et (ii) ne sont pas vérifiées dans le cas de mélanges, ne serait-ce que par invariance de la vraisemblance et de la divergence de Kullback-Leibler par permutation des états cachés. On peut contourner ce problème en considérant la relation d'équivalence d'égalité des paramètres à une permutation près des états cachés, puis en se plaçant dans l'espace quotient des paramètres. Mais le problème, soulevé également dans le cas du critère AIC, lié à ce que pour certaines lois P_0 l'EMV risque de n'être pas asymptotiquement de loi normale, paraît plus profond. Il est lié à la possibilité d'une non identifiabilité du modèle. Peut-être est-il possible de s'appuyer sur une paramétrisation alternativee, analogue à celle proposée par Dacunha-Castelle et Gassiat, 1999 [34], permettant de résoudre cette difficulté. La vérification des autres conditions de régularité peut vraisemblablement s'appuyer sur les travaux de Mevel, 1997 [92] qui a montré, sous certaines conditions, la consistance et la normalité asymptotique de l'estimateur de maximum de vraisemblance dans le contexte des chaînes de Markov cachées. Ses résultats s'appuient sur de nombreuses propositions intermédiaires concernant les propriétés asymptotiques de la vraisemblance et de ses dérivées et devraient pouvoir être adaptées à la vérification des hypothèses (iii) à (vii).

Remarquons enfin que, comme nous l'avons vu au début de cette section, la validation croisée est avant tout justifiée par une estimation empirique du biais résultant d'une estimation de la déviance par la quantité (3.7). Ses propriétés asymptotiques peuvent être étudiées en se basant uniquement sur la définition (3.10), sans faire l'hypothèse, par exemple, que l'EMV est asymptotiquement normal. Dans cette perspective, on peut calculer le biais de l'estimateur (3.10), égal à

$$\begin{aligned}&\mathbb{E}_{\tilde{\boldsymbol{Y}}_0,\tilde{\boldsymbol{Y}}_1}[\ln(P_0(\tilde{\boldsymbol{Y}}_0,\tilde{\boldsymbol{Y}}_1))] - \sum_{i=0}^{1} \mathbb{E}_{\tilde{\boldsymbol{Y}}_i}[\ln(P_0(\tilde{\boldsymbol{Y}}_i))]\\ &-\mathbb{E}_{\tilde{\boldsymbol{Y}}_0,\tilde{\boldsymbol{Y}}_1}[\ln(P_{\hat{\lambda}(\tilde{\boldsymbol{Y}}_0,\tilde{\boldsymbol{Y}}_1)}(\tilde{\boldsymbol{Y}}_0,\tilde{\boldsymbol{Y}}_1))] + \sum_{i=0}^{1} \mathbb{E}_{\tilde{\boldsymbol{Y}}_{1-i}}[\mathbb{E}_{\tilde{\boldsymbol{Y}}_i}[\ln(P_{\hat{\lambda}(\tilde{\boldsymbol{Y}}_i)}(\tilde{\boldsymbol{Y}}_{i-i}))]].\end{aligned}$$

Ce biais est essentiellement dû à l'utilisation de la moitié seulement des données pour estimer le paramètre, et tend vers zéro dans le cas où $\tilde{\boldsymbol{Y}}_0$ et $\tilde{\boldsymbol{Y}}_1$ sont indépendants. Même dans le cas non indépendant, on s'attend à ce que ce biais soit faible en général.

3.4.3 Mise en œuvre de la validation croisée

Nous avons vu dans la section précédente que la validation croisée repose sur le principe suivant :

– choisir au hasard un sous-ensemble A des indices du processus observé et utiliser la partie $\boldsymbol{y}_A$ des données observées $\boldsymbol{y}$ pour identifier le modèle – en principe par calcul de l'estimateur de maximum de vraisemblance $\hat{\lambda}_A$;
– valider le modèle à l'aide du reste des données $\boldsymbol{y}_B$ (où B est le complémentaire de A dans l'ensemble des indices) par le calcul de $P_{\hat{\lambda}_A}(\boldsymbol{Y}_B = \boldsymbol{y}_B)$.

Dans le contexte des modèles de Markov cachés, deux difficultés apparaissent. La première consiste à calculer, pour un paramètre λ connu, la probabilité $P_\lambda(\boldsymbol{Y}_B = \boldsymbol{y}_B)$ pour une sous-partie $\boldsymbol{y}_B$ quelconque des données observées. La deuxième difficulté concerne l'estimation des paramètres dans un cadre où des données $\boldsymbol{y}_B$, observées en principe, ont été supprimées en réalité.

Suppression régulière de motifs dans des modèles dynamiques

L'un des principes de la validation croisée est que le sous-échantillon d'apprentissage est déterminé *au hasard*, c'est-à-dire indépendamment des valeurs prises par les données observées. Cependant, les indices définissant ce sous-échantillon peuvent être choisis de manière déterministe. Nous considérons le cas d'un modèle de Markov caché de la famille $\mathcal{D}$ dynamique (*i.e.* dont la structure est basée sur la répétition d'un motif) et homogène. Par définition, sa structure est basée sur des répétitions de motifs graphiques identiques. On choisira alors par exemple de définir l'ensemble d'apprentissage en supprimant un motif sur deux. Le modèle obtenu a une structure de même nature que le modèle d'origine avec de nouveaux paramètres qui se déduisent fonctionnellement des paramètres d'origine. On utilise alors les résultats du chapitre 2 pour estimer les paramètres, puis pour évaluer la vraisemblance sur les données de test.

Cette méthode est illustrée ci-dessous par l'exemple des chaînes de Markov cachées. On montre aisément (voir Celeux et Durand, 2002 [22] ou Durand, 2001 [44]) que si $(Y_t)_{t\in\{1,\dots,2n\}}$ suit une loi de chaîne de Markov cachée à K états de paramètres $(\pi, A, \theta_1, \dots, \theta_K)$, alors $(Y_{2t+1})_{1\leq t\leq n-1}$ est une chaîne de Markov cachée à K états de paramètres $(\pi, A^2, \theta_1, \dots, \theta_K)$ et $(Y_{2t})_{1\leq t\leq n}$ est une chaîne de Markov cachée à K états de paramètres $(\pi A, A^2, \theta_1, \dots, \theta_K)$. La preuve vient de ce qu'une propriété semblable est vérifiée pour la chaîne de Markov sous-jacente $(S_t)_{t\in\{1,\dots,2n\}}$. Cette propriété est illustrée figure 3.3. Dans le cas où la chaîne cachée est stationnaire, il vient $\pi A = \pi$ et les deux processus (à savoir celui des indices pairs et celui des indices impairs) ont même loi.

Par conséquent, le modèle s'identifie aisément par l'algorithme EM (algorithme dit *de Baum-Welch* dans le contexte des chaînes de Markov cachées), en utilisant uniquement les données d'indice impair. La vraisemblance du paramètre évaluée en les données d'indice pair se fait par l'algorithme *avant-arrière*. On permute ensuite le rôle des données d'indice pair et impair pour implémenter une version particulière, car déterministe, du demi-échantillonnage. Le temps de calcul d'une itération de l'algorithme EM est linéaire en n, de même pour le calcul de la vraisemblance. Par conséquent, l'estimation des paramètres et le calcul du critère de validation croisée est de complexité équivalente au calcul de l'EMV avec la chaîne complète.

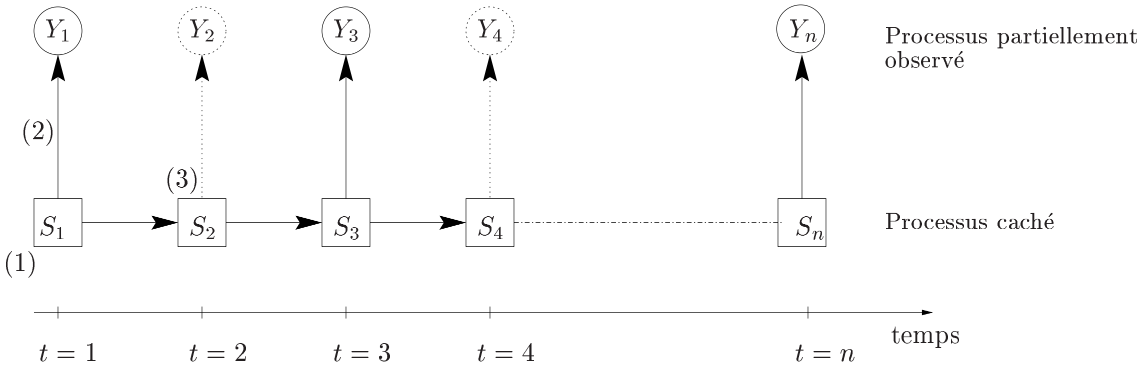


FIG. 3.3 – *Suppression des observations d'indice pair dans une chaîne de Markov cachée. Le processus correspondant aux indices impairs est encore une chaîne de Markov cachée : (1) de même loi initiale π ; (2) de même paramètres d'émission $(\theta_i)_{i\in\{1,\ldots,K\}}$; (3) de matrice de transition A^2. Ceci provient du fait que le processus caché $(S_t)_{t\in\{1,\ldots,n-1\}}$ est une chaîne de Markov de matrice de transition A^2.*

Inférence dans les modèles de Markov cachés à observations supprimées

Dans le cas où l'on souhaite implémenter la validation croisée générale (par exemple parce que le modèle considéré est non homogène ou non dynamique), on est amené à calculer la vraisemblance des paramètres d'un modèle et à identifier ce modèle dans un contexte où certaines données en principe observées sont en réalité manquantes. Notons que ce problème apparaît également dans d'autres contextes, en particulier celui où la collecte des données a abouti à la suppression de certaines d'entre elles (voir à ce sujet l'application présentée au chapitre 4).

Nous avons vu dans le chapitre 2 que pour tout modèle appartenant à la famille $\mathcal{D}$, il existe des algorithmes efficaces pour le calcul de probabilités et pour l'estimation des paramètres. Dans le cas où seules les valeurs de variables aléatoires observées Y_u à valeurs discrètes sont manquantes, le processus effectivement observé a même loi qu'un modèle de Markov caché où un Y_u manquant est remplacé par la variable cachée S_u. Ce modèle équivalent appartient toujours à la famille $\mathcal{D}$ et admet les mêmes paramètres que la loi d'origine. Ceci est une propriété remarquable de la famille $\mathcal{D}$ et justifie sa définition. On utilisera alors, pour estimer les paramètres et calculer des probabilités, les méthodes du chapitre 2 au nouveau modèle où les variables manquantes à valeurs discrètes Y_u sont remplacées par les variables cachées S_u.

Un problème se pose en réalité dans le cas où certaines variables aléatoires Y_u à valeurs continues sont, de fait, inobservées. Rappelons que $\mathcal{U}_\theta$ représente l'ensemble des indices des variables aléatoires observées à valeurs continues. Nous allons donc considérer le cas où Obs est une partie de $\mathcal{U}_\theta$, notre objectif étant alors de maximiser par rapport à λ la vraisemblance définie par

$$\mathcal{L}_{\boldsymbol{y}_{Obs}}(\lambda) = P_\lambda(\boldsymbol{Y}_{Obs} = \boldsymbol{y}_{Obs}).$$

Nous désignons par Man le complémentaire de Obs dans $\mathcal{U}_\theta$, c'est-à-dire l'ensemble des indices des variables aléatoires manquantes à valeurs continues.

Compte tenu de l'existence de données supprimées ou manquantes, il est naturel de recourir une fois de plus à la méthodologie EM pour estimer les paramètres. Comme nous l'avons vu dans la section 2.3, le cas de modèles non homogènes et celui où plusieurs réalisations indépendantes du processus observé sont utilisées pour estimer les paramètres, se déduisent facilement du cas homogène avec une seule réalisation du processus observé disponible, c'est pourquoi nous nous plaçons uniquement dans ce cadre.

Estimation des paramètres par EM :
Simplification de l'étape E

Rappelons l'expression (2.10) de la log-vraisemblance complétée pour ces modèles :

$$\begin{aligned}\ln(\mathcal{L}_{\boldsymbol{x}}(\lambda)) &= \sum_{\bar{u}\in\bar{\mathcal{U}}_\theta}\sum_{u\in\bar{u}}\sum_{\boldsymbol{a}\in\mathcal{X}_{\mathrm{pa}(u)}} \ln(P_{\theta_{\boldsymbol{a}}^{(\bar{u})}}(y_u))\mathbb{1}_{\{\boldsymbol{x}_{\mathrm{pa}(u)}=\boldsymbol{a}\}}\\ &+ \sum_{\bar{u}\in\bar{\mathcal{U}}_p}\sum_{u\in\bar{u}}\sum_{\boldsymbol{a}\in\mathcal{X}_{\mathrm{pa}(u)}}\sum_{i\in\mathcal{X}_u} \ln(p_{\boldsymbol{a},i}^{(\bar{u})})\mathbb{1}_{\{x_u=i,\boldsymbol{x}_{\mathrm{pa}(u)}=\boldsymbol{a}\}}\\ &+ \sum_{\bar{u}\in\bar{\mathcal{U}}_\pi}\sum_{u\in\bar{u}}\sum_{i\in\mathcal{X}_u}\ln(\pi_i^{(\bar{u})})\mathbb{1}_{\{x_u=i\}}\end{aligned}$$

Nous sommes alors conduit à calculer la fonction Q définie par

$$Q(\lambda,\lambda^{(\eta-1)}) = \mathbb{E}_{\lambda^{(\eta-1)}}[\ln(P_\lambda(\boldsymbol{S},\boldsymbol{Y}))|\boldsymbol{Y}_{Obs}=\boldsymbol{y}_{Obs}].$$

On en déduit l'expression suivante de $Q(\lambda,\lambda^{(\eta-1)})$

$$\begin{aligned}&Q(\lambda,\lambda^{(\eta-1)}) = \qquad (3.11)\\ &\sum_{\bar{u}\in\bar{\mathcal{U}}_\theta}\sum_{u\in\bar{u}}\sum_{\boldsymbol{a}\in\mathcal{X}_{\mathrm{pa}(u)}} \mathbb{E}_{\lambda^{(\eta-1)}}[\ln(P_{\theta_{\boldsymbol{a}}^{(\bar{u})}}(Y_u))\mathbb{1}_{\{\boldsymbol{S}_{\mathrm{pa}(u)}=\boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}},\boldsymbol{Y}_{\mathrm{pa}(u)}=\boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}}\}}|\boldsymbol{Y}_{Obs}=\boldsymbol{y}_{Obs}]\\ &+\sum_{\bar{u}\in\bar{\mathcal{U}}_p}\sum_{u\in\bar{u}}\sum_{\boldsymbol{a}\in\mathcal{X}_{\mathrm{pa}(u)}}\sum_{i\in\mathcal{X}_u}\ln(p_{\boldsymbol{a},i}^{(\bar{u})})P_{\lambda^{(\eta-1)}}(X_u=i,\boldsymbol{S}_{\mathrm{pa}(u)}=\boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}},\boldsymbol{Y}_{\mathrm{pa}(u)}=\boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}}|\boldsymbol{Y}_{Obs}=\boldsymbol{y}_{Obs})\\ &+\sum_{\bar{u}\in\bar{\mathcal{U}}_\pi}\sum_{u\in\bar{u}}\sum_{i\in\mathcal{X}_u}\ln(\pi_i^{(\bar{u})})P_{\lambda^{(\eta-1)}}(X_u=i|\boldsymbol{Y}_{Obs}=\boldsymbol{y}_{Obs}).\end{aligned}$$

L'espérance intervenant dans cette expression peut être simplifiée dans la mesure où

$$\begin{aligned}&\mathbb{E}_{\lambda^{(\eta-1)}}[\ln(P_{\theta_{\boldsymbol{a}}^{(\bar{u})}}(Y_u))\mathbb{1}_{\{\boldsymbol{S}_{\mathrm{pa}(u)}=\boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}},\boldsymbol{Y}_{\mathrm{pa}(u)}=\boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}}\}}|\boldsymbol{Y}_{Obs}=\boldsymbol{y}_{Obs}]\\ &= P_{\lambda^{(\eta-1)}}(\boldsymbol{S}_{\mathrm{pa}(u)}=\boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}},\boldsymbol{Y}_{\mathrm{pa}(u)}=\boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}}|\boldsymbol{Y}_{Obs}=\boldsymbol{y}_{Obs})\\ &\times\ \mathbb{E}_{\lambda^{(\eta-1)}}[\ln(P_{\theta_{\boldsymbol{a}}^{(\bar{u})}}(Y_u))|\boldsymbol{S}_{\mathrm{pa}(u)}=\boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}},\boldsymbol{Y}_{\mathrm{pa}(u)}=\boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}},\boldsymbol{Y}_{Obs}=\boldsymbol{y}_{Obs}]\end{aligned}$$

Ceci provient de la propriété $\mathbb{E}[f(X)Z]=\mathbb{E}[f(X)\mathbb{E}[Z|f(X)]]$ appliquée à l'espérance conditionnelle $\mathbb{E}_{\lambda^{(\eta-1)}}[\,.\,|\boldsymbol{Y}_{Obs}=\boldsymbol{y}_{Obs}]$ et aux variables aléatoires

$$Z=\ln(P_{\theta_{\boldsymbol{a}}^{(\bar{u})}}(Y_u)) \text{ et } f(X)=\mathbb{1}_{\{\boldsymbol{S}_{\mathrm{pa}(u)}=\boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}},\boldsymbol{Y}_{\mathrm{pa}(u)}=\boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}}\}}.$$

Nous utilisons ensuite la propriété

$$\mathbb{E}[\mathbb{1}_{\{X\in A\}}\mathbb{E}[Z|\mathbb{1}_{\{X\in A\}}]] = P(X\in A)\mathbb{E}[Z|X\in A].$$

D'autre part, Y_u n'a pas de descendant d'après les hypothèses caractérisant la famille $\mathcal{D}$, car cette variable aléatoire est à valeurs continues. Donc Y_u est conditionnellement indépendante de toutes les autres variables aléatoires, sachant $\{\boldsymbol{S}_{\text{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}}\} \cap \{\boldsymbol{Y}_{\text{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}}\}$. Il s'ensuit que

$$\begin{aligned}&\mathbb{E}_{\lambda^{(\eta-1)}}[\ln(P_{\boldsymbol{\theta}_{\boldsymbol{a}}^{(\bar{u})}}(Y_u))|\boldsymbol{S}_{\text{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}}, \boldsymbol{Y}_{\text{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}}, \boldsymbol{Y}_{Obs} = \boldsymbol{y}_{Obs}] \\ &\quad = \mathbb{E}_{\lambda^{(\eta-1)}}[\ln(P_{\boldsymbol{\theta}_{\boldsymbol{a}}^{(\bar{u})}}(Y_u))|\boldsymbol{S}_{\text{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}}, \boldsymbol{Y}_{\text{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}}].\end{aligned}$$

À ce point de l'exposé, la simplification ci-dessus de l'expression (3.11) conduit à

$$\begin{aligned}&Q(\lambda, \lambda^{(\eta-1)}) = \qquad\qquad (3.12)\\ &\sum_{\bar{u}\in\bar{\mathcal{U}}_\theta} \sum_{u\in\bar{u}\cap Obs} \sum_{\boldsymbol{a}\in\mathcal{X}_{\text{pa}(u)}} \ln(P_{\boldsymbol{\theta}_{\boldsymbol{a}}^{(\bar{u})}}(y_u)) P_{\lambda^{(\eta-1)}}(\boldsymbol{S}_{\text{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}}, \boldsymbol{Y}_{\text{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}} | \boldsymbol{Y}_{Obs} = \boldsymbol{y}_{Obs})\\ &+ \sum_{\bar{u}\in\bar{\mathcal{U}}_\theta} \sum_{u\in\bar{u}\cap Man} \sum_{\boldsymbol{a}\in\mathcal{X}_{\text{pa}(u)}} \Big[\mathbb{E}_{\lambda^{(\eta-1)}}[\ln(P_{\boldsymbol{\theta}_{\boldsymbol{a}}^{(\bar{u})}}(Y_u))|\boldsymbol{S}_{\text{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}}, \boldsymbol{Y}_{\text{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}}]\\ &\qquad \times P_{\lambda^{(\eta-1)}}(\boldsymbol{S}_{\text{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}}, \boldsymbol{Y}_{\text{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}} | \boldsymbol{Y}_{Obs} = \boldsymbol{y}_{Obs})\Big]\\ &+ \sum_{\bar{u}\in\bar{\mathcal{U}}_p} \sum_{u\in\bar{u}} \sum_{\boldsymbol{a}\in\mathcal{X}_{\text{pa}(u)}} \sum_{i\in\mathcal{X}_u} \ln(p_{\boldsymbol{a},i}^{(\bar{u})}) P_{\lambda^{(\eta-1)}}(X_u = i, \boldsymbol{S}_{\text{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}}, \boldsymbol{Y}_{\text{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}} | \boldsymbol{Y}_{Obs} = \boldsymbol{y}_{Obs})\\ &+ \sum_{\bar{u}\in\bar{\mathcal{U}}_\pi} \sum_{u\in\bar{u}} \sum_{i\in\mathcal{X}_u} \ln(\pi_i^{(\bar{u})}) P_{\lambda^{(\eta-1)}}(X_u = i | \boldsymbol{Y}_{Obs} = \boldsymbol{y}_{Obs}).\end{aligned}$$

Or sachant $\{\boldsymbol{S}_{\text{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}}\} \cap \{\boldsymbol{Y}_{\text{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}}\}$ et sous $P_{\lambda^{(\eta-1)}}$, Y_u suit la loi $P_{\boldsymbol{\theta}_{\boldsymbol{a}}^{(\eta-1)}}$ (nous omettons d'expliciter la dépendance des paramètres vis-à-vis de $\bar{u}$ pour simplifier les notations). Nous sommes donc amenés à calculer l'espérance de $\ln(P_{\boldsymbol{\theta}_{\boldsymbol{a}}}(Y))$ lorsque Y suit la loi $P_{\boldsymbol{\theta}_{\boldsymbol{a}}^{(\eta-1)}}$. Les détails du calcul, voire la possibilité de l'effectuer, dépendent de la loi considérée, mais dans le cas de lois d'émission de la famille exponentielle, nous obtenons le résultat général ci-dessous. Rappelons que dans ce contexte,

$$\ln(P_{\Phi_{\boldsymbol{a}}}(y)) = -\ln(\alpha(\Phi_{\boldsymbol{a}})) + \ln(b(y)) + {}^t\theta(\Phi_{\boldsymbol{a}})T(y),$$

où $\Phi_{\boldsymbol{a}}$ est le paramètre inconnu que nous cherchons à réestimer à l'étape M de l'algorithme EM et où Y suit la loi $P_{\Phi_{\boldsymbol{a}}^{(\eta-1)}}$, $\Phi_{\boldsymbol{a}}^{(\eta-1)}$ étant connu à l'itération η de l'algorithme. Par conséquent,

$$\begin{aligned}&\mathbb{E}_{\lambda^{(\eta-1)}}[\ln(P_{\boldsymbol{\theta}_{\boldsymbol{a}}^{(\bar{u})}}(Y))|\boldsymbol{X}_{\text{pa}(u)} = \boldsymbol{a}]\\ &\quad = -\ln(\alpha(\Phi_{\boldsymbol{a}})) + \mathbb{E}_{\lambda^{(\eta-1)}}[\ln(b(Y))|\boldsymbol{X}_{\text{pa}(u)} = \boldsymbol{a}] + {}^t\theta(\Phi_{\boldsymbol{a}})\mathbb{E}_{\lambda^{(\eta-1)}}[T(Y)|\boldsymbol{X}_{\text{pa}(u)} = \boldsymbol{a}].\end{aligned}$$

La quantité $\mathbb{E}_{\lambda^{(\eta-1)}}[\ln(b(Y))|\boldsymbol{X}_{\text{pa}(u)} = \boldsymbol{a}]$ est indépendante de $\Phi_{\boldsymbol{a}}$ et apparaît donc comme une constante dans l'expression $Q(\lambda, \lambda^{(\eta-1)})$ prise comme une fonction de λ. D'autre part, nous avons vu dans la section 2.3.3 que

$$\mathbb{E}_{\lambda^{(\eta-1)}}[T(Y)|\boldsymbol{X}_{\text{pa}(u)} = \boldsymbol{a}] = \Phi_{\boldsymbol{a}}^{(\eta-1)}.$$

On obtient donc l'expression suivante de $Q(\lambda, \lambda^{(\eta-1)})$, dans le cadre de lois de la famille exponentielle :

$$Q(\lambda, \lambda^{(\eta-1)}) = \tag{3.13}$$
$$\sum_{\bar{u}\in\bar{\mathcal{U}}_\theta} \sum_{u\in\bar{u}\cap Obs} \sum_{\boldsymbol{a}\in\mathcal{X}_{\mathrm{pa}(u)}} \ln(P_{\theta_{\boldsymbol{a}}^{(\bar{u})}}(y_u)) P_{\lambda^{(\eta-1)}}(\boldsymbol{S}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}}, \boldsymbol{Y}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}} | \boldsymbol{Y}_{Obs} = \boldsymbol{y}_{Obs})$$
$$+ \sum_{\bar{u}\in\bar{\mathcal{U}}_\theta} \sum_{u\in\bar{u}\cap Man} \sum_{\boldsymbol{a}\in\mathcal{X}_{\mathrm{pa}(u)}} \Big[(-\ln(\alpha(\Phi_{\boldsymbol{a}})) + \mathbb{E}_{\lambda^{(\eta-1)}}[\ln(b(Y))|\boldsymbol{X}_{\mathrm{pa}(u)} = \boldsymbol{a}] + {}^t\theta(\Phi_{\boldsymbol{a}})\Phi_{\boldsymbol{a}}^{(\eta-1)})$$
$$\times P_{\lambda^{(\eta-1)}}(\boldsymbol{S}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}}, \boldsymbol{Y}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}} | \boldsymbol{Y}_{Obs} = \boldsymbol{y}_{Obs})\Big]$$
$$+ \sum_{\bar{u}\in\bar{\mathcal{U}}_p} \sum_{u\in\bar{u}} \sum_{\boldsymbol{a}\in\mathcal{X}_{\mathrm{pa}(u)}} \sum_{i\in\mathcal{X}_u} \ln(p_{\boldsymbol{a},i}^{(\bar{u})}) P_{\lambda^{(\eta-1)}}(X_u = i, \boldsymbol{S}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}}, \boldsymbol{Y}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}} | \boldsymbol{Y}_{Obs} = \boldsymbol{y}_{Obs})$$
$$+ \sum_{\bar{u}\in\bar{\mathcal{U}}_\pi} \sum_{u\in\bar{u}} \sum_{i\in\mathcal{X}_u} \ln(\pi_i^{(\bar{u})}) P_{\lambda^{(\eta-1)}}(X_u = i | \boldsymbol{Y}_{Obs} = \boldsymbol{y}_{Obs}).$$

Étape M

De même que dans la section 2.3.3, on obtient les formules de réestimation suivantes : pour les lois des sources,

$$\hat{\pi}_i^{(\bar{u})} = \frac{\sum_{u\in\bar{u}} P_{\lambda^{(\eta-1)}}(X_u = i | \boldsymbol{Y}_{Obs} = \boldsymbol{y}_{Obs})}{\mathrm{card}(\bar{u})}.$$

Pour les probabilités de transition, les formules de réestimation sont

$$\hat{p}_{\boldsymbol{a},i}^{(\bar{u})} = \frac{\sum_{u\in\bar{u}} P_{\lambda^{(\eta-1)}}(X_u = i, \boldsymbol{S}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}}, \boldsymbol{Y}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}} | \boldsymbol{Y}_{Obs} = \boldsymbol{y}_{Obs})}{\sum_{u\in\bar{u}} P_{\lambda^{(\eta-1)}}(\boldsymbol{S}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}}, \boldsymbol{Y}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}} | \boldsymbol{Y}_{Obs} = \boldsymbol{y}_{Obs})}.$$

Les formules de réestimation des paramètres d'émission sont données par la résolution de

$$\nabla_{\Phi_{\boldsymbol{a}}} \left[\sum_{u\in\bar{u}\cap Obs} \ln(P_{\theta_{\boldsymbol{a}}^{(\bar{u})}}(y_u)) P_{\lambda^{(\eta-1)}}(\boldsymbol{S}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}}, \boldsymbol{Y}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}} | \boldsymbol{Y}_{Obs} = \boldsymbol{y}_{Obs}) \right]$$
$$+\nabla_{\Phi_{\boldsymbol{a}}} \Bigg[\sum_{u\in\bar{u}\cap Man} \big((-\ln(\alpha(\Phi_{\boldsymbol{a}})) + {}^t\theta(\Phi_{\boldsymbol{a}})\Phi_{\boldsymbol{a}}^{(\eta-1)})$$
$$\times P_{\lambda^{(\eta-1)}}(\boldsymbol{S}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}}, \boldsymbol{Y}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}} | \boldsymbol{Y}_{Obs} = \boldsymbol{y}_{Obs})\big) \Bigg] = 0,$$

compte tenu de

$$\nabla_{\Phi_{\boldsymbol{a}}} \Bigg[\sum_{u\in\bar{u}\cap Man} \big(\mathbb{E}_{\lambda^{(\eta-1)}}[\ln(b(Y))|\boldsymbol{X}_{\mathrm{pa}(u)} = \boldsymbol{a}]$$
$$\times P_{\lambda^{(\eta-1)}}(\boldsymbol{S}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}}, \boldsymbol{Y}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}} | \boldsymbol{Y}_{Obs} = \boldsymbol{y}_{Obs})\big) \Bigg] = 0.$$

Or d'après la section 2.3.3, on a

$$\nabla_{\Phi_{\boldsymbol{a}}}[-\ln(\alpha(\theta(\Phi_{\boldsymbol{a}})))] = -\nabla_{\Phi_{\boldsymbol{a}}}[\theta(\Phi_{\boldsymbol{a}})]\Phi_{\boldsymbol{a}}.$$

De plus,

$$\nabla_{\Phi_{\boldsymbol{a}}}[{}^t\theta(\Phi_{\boldsymbol{a}})\Phi_{\boldsymbol{a}}^{(\eta-1)}] = \nabla_{\Phi_{\boldsymbol{a}}}[\theta(\Phi_{\boldsymbol{a}})]\Phi_{\boldsymbol{a}}^{(\eta-1)}.$$

On en déduit, si $\nabla_{\Phi_{\boldsymbol{a}}}[\theta(\Phi_{\boldsymbol{a}})]$ est inversible, que $\Phi_{\boldsymbol{a}}$ vérifie l'équation

$$\begin{aligned}&\sum_{u\in\bar{u}\cap Obs} P_{\lambda^{(\eta-1)}}(\boldsymbol{S}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}}, \boldsymbol{Y}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}} | \boldsymbol{Y}_{Obs} = \boldsymbol{y}_{Obs})(T(y_u) - \Phi_{\boldsymbol{a}})\\ &+ \sum_{u\in\bar{u}\cap Man} P_{\lambda^{(\eta-1)}}(\boldsymbol{S}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}}, \boldsymbol{Y}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}} | \boldsymbol{Y}_{Obs} = \boldsymbol{y}_{Obs})(\Phi_{\boldsymbol{a}}^{(\eta-1)} - \Phi_{\boldsymbol{a}}) = 0,\end{aligned}$$

d'où la formule de réestimation pour les paramètres d'émission

$$\begin{aligned}\hat{\Phi}_{\boldsymbol{a}} = &\frac{\sum\limits_{u\in\bar{u}\cap Obs} P_{\lambda^{(\eta-1)}}(\boldsymbol{S}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}}, \boldsymbol{Y}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}} | \boldsymbol{Y}_{Obs} = \boldsymbol{y}_{Obs})T(y_u)}{\sum\limits_{u\in\bar{u}} P_{\lambda^{(\eta-1)}}(\boldsymbol{S}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}}, \boldsymbol{Y}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}} | \boldsymbol{Y}_{Obs} = \boldsymbol{y}_{Obs})}\\ &+\frac{[\sum\limits_{u\in\bar{u}\cap Man} P_{\lambda^{(\eta-1)}}(\boldsymbol{S}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}}, \boldsymbol{Y}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}} | \boldsymbol{Y}_{Obs} = \boldsymbol{y}_{Obs})]\Phi_{\boldsymbol{a}}^{(\eta-1)}}{\sum\limits_{u\in\bar{u}} P_{\lambda^{(\eta-1)}}(\boldsymbol{S}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}}, \boldsymbol{Y}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}} | \boldsymbol{Y}_{Obs} = \boldsymbol{y}_{Obs})}.\end{aligned}$$

Nous notons

$$\begin{aligned}\xi_u^{(\eta-1)}(\boldsymbol{a}) &= P_{\lambda^{(\eta-1)}}(\boldsymbol{X}_{\mathrm{pa}(u)} = \boldsymbol{a} | \boldsymbol{Y}_{Obs} = \boldsymbol{y}_{Obs})\\ &= P_{\lambda^{(\eta-1)}}(\boldsymbol{S}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{S}}}, \boldsymbol{Y}_{\mathrm{pa}(u)} = \boldsymbol{a}_{\mathcal{U}_{\boldsymbol{Y}}} | \boldsymbol{Y}_{Obs} = \boldsymbol{y}_{Obs})\end{aligned}$$

puis donnons, à titre d'illustration, les formules de réestimation dans le cas gaussien :

$$\hat{\mu}_{\boldsymbol{a}} = \frac{\sum\limits_{u\in\bar{u}\cap Obs} \xi_u^{(\eta-1)}(\boldsymbol{a})y_u + [\sum\limits_{u\in\bar{u}\cap Man} \xi_u^{(\eta-1)}(\boldsymbol{a})]\mu_{\boldsymbol{a}}^{(\eta-1)}}{\sum\limits_{u\in\bar{u}} \xi_u^{(\eta-1)}(\boldsymbol{a})}$$

et

$$\begin{aligned}\hat{\Sigma}_{\boldsymbol{a}} = &\frac{\sum\limits_{u\in\bar{u}\cap Obs} \xi_u^{(\eta-1)}(\boldsymbol{a})^t(y_u - \hat{\mu}_{\boldsymbol{a}})(y_u - \hat{\mu}_{\boldsymbol{a}})}{\sum\limits_{u\in\bar{u}} \xi_u^{(\eta-1)}(\boldsymbol{a})}\\ &+\frac{[\sum\limits_{u\in\bar{u}\cap Man} \xi_u^{(\eta-1)}(\boldsymbol{a})][\Sigma_{\boldsymbol{a}}^{(\eta-1)} + {}^t(\mu_{\boldsymbol{a}}^{(\eta-1)} - \hat{\mu}_{\boldsymbol{a}})(\mu_{\boldsymbol{a}}^{(\eta-1)} - \hat{\mu}_{\boldsymbol{a}})]}{\sum\limits_{u\in\bar{u}} \xi_u^{(\eta-1)}(\boldsymbol{a})}.\end{aligned}$$

On remarque, d'après les formules de réestimation des lois des sources et des probabilités de transition, que l'interprétation a posteriori de l'algorithme EM donnée dans la section 2.3.4 est encore valide dans le contexte de la suppression de données observées à valeurs continues. On constate en effet que dans les formules de réestimation ci-dessus, les variables aléatoires manquantes sont remplacées par leur espérance conditionnelle sachant les données observées. Cependant, l'algorithme EM ne consiste pas, en général, à remplacer les données manquantes par leur espérance conditionnelle. Ainsi, dans le cas particulier gaussien, même si les y_u manquants sont remplacés par $\mu_{\boldsymbol{a}}^{(\eta-1)}$ pour estimer $\mu_{\boldsymbol{a}}$, une interprétation analogue n'est pas possible pour la formule de réestimation de $\Sigma_{\boldsymbol{a}}$. En effet, la substitution d'un y_u par son espérance, dans le cas où cette donnée est supprimée, a pour effet de réduire la variance estimée. Ceci est compensé par l'ajout du terme $\Sigma_{\boldsymbol{a}}^{(\eta-1)}$. Néanmoins, dans le cas particulier de lois d'émission de la famille exponentielle, rappelons que $\Phi_{\boldsymbol{a}}^{(\eta-1)}$ est l'espérance de $T(Y_u)$ quand Y_u suit la loi $P_{\theta_{\boldsymbol{a}}^{(\eta-1)}}$. Si les $(\Phi_{\boldsymbol{a}})_{\boldsymbol{a}}$ sont utilisés pour paramétrer le modèle, alors la formule de réestimation de $\Phi_{\boldsymbol{a}}$ correspond à l'estimateur du maximum de vraisemblance où les variables aléatoires manquantes sont remplacées par leur espérance conditionnelle sachant les variables observées et on retrouve l'interprétation a posteriori des formules de réestimation donnée au chapitre 2.3.4.

Étape E et calcul de probabilités

De même qu'en section 2.3.3, les probabilités conditionnelles qui interviennent dans l'étape E sont des produits d'indicatrices concernant les variables aléatoires observées $\boldsymbol{Y}_{Obs}$ et des probabilités conditionnelles, sachant $\boldsymbol{Y}_{Obs} = \boldsymbol{y}_{Obs}$, de variables manquantes appartenant à $\{u\}\cup\mathrm{pa}(u)$ (qu'il s'agisse d'états cachés S_v ou de variables observées mais supprimées Y_v). Or d'après la proposition 1, pour tout sommet $u \in \mathcal{U}$, il existe une clique $\mathcal{C}$ de l'arbre de jonction contenant $\{u\}\cup\mathrm{pa}(u)$. Il suffit donc de savoir calculer

$$P(\boldsymbol{S}_{\mathcal{C}} = \boldsymbol{s}_{\mathcal{C}}, \boldsymbol{Y}_{\mathcal{C}\cap Man} = \boldsymbol{y}_{\mathcal{C}\cap Man} | \boldsymbol{Y}_{Obs} = \boldsymbol{y}_{Obs})$$

pour toutes les cliques $\mathcal{C}$ de l'arbre de jonction pour pouvoir implémenter l'étape E.

Dans ce qui suit, nous supposons λ fixé et connu et nous notons $P = P_\lambda$. Comme dans la section 2.4, nous supposons également fixée la valeur $\boldsymbol{y}_{Obs}$ du processus $\boldsymbol{Y}_{Obs}$. Nous allons alors adapter l'algorithme arrière-avant des sections 2.4.2 et 2.4.3 pour permettre le calcul des probabilités requises pour l'étape E. L'adaptation repose essentiellement sur le fait que dans les modèles de la famille $\mathcal{D}$, les sommets u de $\mathcal{U}_\theta$ (c'est-à-dire les variables aléatoires observées Y_u à valeurs continues) sont des puits et appartiennent à exactement une clique. Nous verrons que l'algorithme arrière-avant consiste à remplacer $P_{\theta_{\boldsymbol{x}_{\mathrm{pa}(u)}}}(y_u)$ par la valeur un lorsque Y_u est supprimée. Le modèle $P_{\boldsymbol{Y}_{Obs}}$ est alors identique au modèle $P_{\boldsymbol{Y}}$ où les sommets de $\boldsymbol{Y}_{Man}$ sont supprimés du graphe d'indépendance conditionnelle. Le fait que les sommets de $\mathcal{U}_\theta$ soient des puits découle tout simplement du fait qu'ils n'admettent que des arcs entrants. L'autre propriété, prouvée ci-dessous, établit qu'un sommet u de $\mathcal{U}_\theta$ appartient à une unique clique $\mathcal{C}$ de l'arbre de jonction.

Proposition 10 *Soit $u \in \mathcal{U}_\theta$ un sommet du graphe $\mathcal{G}$ d'indépendance conditionnelle. Alors il existe une unique clique $\mathcal{C}$ de l'arbre de jonction contenant u.*

Démonstration

L'existence d'une clique contenant u est immédiate, elle tient au fait que u admet au moins un parent v, l'ensemble $\{u; v\}$ engendrant un graphe complet.
L'unicité de cette clique se montre en considérant deux cliques $\mathcal{C}$ et $\mathcal{C}'$ contenant u. Soit v un sommet de $\mathcal{C}$ et v' un sommet de $\mathcal{C}'$. Alors v et u sont dans $\mathcal{C}$ donc ces sommets sont adjacents. Comme $u \in \mathcal{U}_\theta$, u n'admet aucun arc sortant : on en déduit $v \to u$. De même, $v' \to u$. Comme le graphe $\mathcal{G}$ est moral, il vient $v \leftrightarrow v'$. On en déduit que le graphe engendré par $\mathcal{C} \cup \mathcal{C}'$ est complet. Par maximalité de $\mathcal{C}$ et de $\mathcal{C}'$, ceci n'est possible que si $\mathcal{C} = \mathcal{C}'$.
En conclusion, u appartient à une unique clique $\mathcal{C}$.

Nous avons annoncé ci-dessus qu'il suffit de savoir calculer les probabilités

$$P(\boldsymbol{S}_\mathcal{C} = \boldsymbol{s}_\mathcal{C}, \boldsymbol{Y}_{\mathcal{C}\cap Man} = \boldsymbol{y}_{\mathcal{C}\cap Man} | \boldsymbol{Y}_{Obs} = \boldsymbol{y}_{Obs})$$

pour implémenter l'étape E. D'après la proposition 10, l'ensemble $\mathcal{C} \cap Man$ contient en fait un seul élément Y_u au plus. De plus, vu qu'un tel Y_u appartient à une seule clique, Y_u n'appartient à aucun séparateur de cliques.

Notre algorithme arrière-avant pour les variables manquantes à valeurs continues repose sur le travail effectué dans les sections 2.4.2 et 2.4.3, mais en considérant que les Y_u pour $u \in \mathcal{U}_\theta$ sont des variables aléatoires cachées. En pratique, on utiliserait un algorithme basé sur la décomposition des probabilités de lissage $P(X_u = x_u | \boldsymbol{Y}_{Obs} = \boldsymbol{y}_{Obs})$, similaire à celui de la section 2.4.5. Cependant, pour éviter des complications dans les notations et pour pouvoir nous focaliser sur le problème de suppression des variables aléatoires à valeurs continues, et lui seul, nous commençons par présenter une méthode de décomposition des probabilités jointes $P(X_u = x_u, \boldsymbol{Y}_{Obs} = \boldsymbol{y}_{Obs})$.

L'algorithme arrière-avant du chapitre 2 repose, pour chaque arc a de l'arbre de jonction, sur des quantités notées $\tilde{\beta}_a$ et $\tilde{\alpha}_a$, calculées de manière inductive : nous redéfinissons donc ces quantités de manière à prendre en compte les Y_u supprimés. Les formules d'initialisation, de propagation et de terminaison faisaient intervenir, dans le chapitre 2, des sommes sur toutes les valeurs possibles des états cachés. Les formules ci-dessous feront donc intervenir, en plus de ces sommes, des intégrales sur toutes les valeurs possibles des variables observées supprimées. Nous montrerons que ces intégrales se calculent en fait très simplement. Nous conservons les notations du chapitre 2 ; en particulier $\mathcal{C}_0$ désigne la racine de l'arbre de jonction. Rappelons qu'il s'agit d'une clique égale à son graphe ancestral.

Phase arrière

La phase arrière est un parcours de graphe qui part des feuilles de l'arbre des cliques et remonte jusqu'à la racine $\mathcal{C}_0$, parcourant ainsi les arcs de l'arbre en sens contraire de leur orientation. Elle permet le calcul de la vraisemblance $P(\boldsymbol{Y}_{Obs} = \boldsymbol{y}_{Obs})$ et également le calcul de certaines probabilités notées $\tilde{\beta}_a(\boldsymbol{y}_{\mathcal{K}_a \cap Obs}, \boldsymbol{x}_{\mathcal{S}_a})$ et définies par

$$P(\boldsymbol{Y}_{\mathcal{K}_a\cap Obs} = \boldsymbol{y}_{\mathcal{K}_a\cap Obs} | \boldsymbol{Y}_{\mathcal{S}_a\cap Man} = \boldsymbol{y}_{\mathcal{S}_a\cap Man}, \boldsymbol{Y}_{\mathcal{S}_a\cap Obs} = \boldsymbol{y}_{\mathcal{S}_a\cap Obs}, \boldsymbol{S}_{\mathcal{S}_a} = \boldsymbol{s}_{\mathcal{S}_a})$$

où a est un arc de l'arbre de jonction $\mathcal{T}$, avec $a = (\mathcal{C}_i, \mathcal{C}_j)$. Rappelons que dans notre contexte, les données observées sont fixées et égales à $\boldsymbol{y}_{Obs}$. Les seules variables observées supprimées appartiennent à $\mathcal{U}_\theta$, et donc n'appartiennent à aucun séparateur de cliques d'après la propriété 10. Donc $\mathbb{S}_a \cap Man = \emptyset$ et $\mathbb{S}_a \cap Obs = \mathbb{S}_a$, la définition de $\tilde{\beta}_a$ se résumant à

$$\tilde{\beta}_a(\boldsymbol{s}_{\mathbb{S}_a}) = P(\boldsymbol{Y}_{\mathcal{K}_a \cap Obs} = \boldsymbol{y}_{\mathcal{K}_a \cap Obs} | \boldsymbol{Y}_{\mathbb{S}_a} = \boldsymbol{y}_{\mathbb{S}_a}, \boldsymbol{S}_{\mathbb{S}_a} = \boldsymbol{s}_{\mathbb{S}_a}).$$

La dépendance de $\tilde{\beta}_a(\boldsymbol{s}_{\mathbb{S}_a})$ vis-à-vis de $\boldsymbol{y}_{Obs}$ n'étant pas importante dans le cadre de notre étude, nous n'en tenons pas compte dans les notations.

Initialisation de la phase arrière

La phase arrière est initialisée en les arcs $a = (\mathcal{C}_i, \mathcal{C}_j)$ de l'arbre de jonction, $\mathcal{C}_j$ étant une clique feuille, par le calcul des quantités

$$\tilde{\beta}_a(\boldsymbol{s}_{\mathbb{S}_a}) = P(\boldsymbol{Y}_{(\mathcal{C}_j \cap Obs)\setminus \mathbb{S}_a} = \boldsymbol{y}_{(\mathcal{C}_j \cap Obs)\setminus \mathbb{S}_a} | \boldsymbol{X}_{\mathbb{S}_a} = \boldsymbol{x}_{\mathbb{S}_a}).$$

Dans le cas où $\mathcal{C}_j$ ne contient aucune variable aléatoire supprimée Y_u à valeurs continues, la formule d'initialisation demeure l'expression (2.24). Dans le cas contraire, $\mathcal{C}_j$ contient exactement une variable aléatoire supprimée à valeurs continues, notée Y_u. La quantité $\tilde{\beta}_a(\boldsymbol{s}_{\mathbb{S}_a})$ s'obtient alors à partir de la loi

$$P(\boldsymbol{Y}_{\mathcal{C}_j\setminus \mathbb{S}_a} = \boldsymbol{y}_{\mathcal{C}_j\setminus \mathbb{S}_a}, \boldsymbol{S}_{\mathcal{C}_j\setminus \mathbb{S}_a} = \boldsymbol{s}_{\mathcal{C}_j\setminus \mathbb{S}_a} | \boldsymbol{X}_{\mathbb{S}_a} = \boldsymbol{x}_{\mathbb{S}_a}),$$

donnée par les équations (2.22) ou (2.23), par sommation sur toutes les valeurs de $\boldsymbol{s}_{\mathcal{C}_j\setminus \mathbb{S}_a}$ dans $\mathcal{S}_{\mathcal{C}_j\setminus \mathbb{S}_a}$ et par intégration sur y_u. L'intégration se fait suivant le même principe que dans la remarque 2.13, c'est-à-dire en remplaçant $P_{\theta_{\boldsymbol{x}_{\mathrm{pa}(u)}}}(y_u)$ par la valeur un. Ceci provient du fait que y_u intervient uniquement à travers $P_{\theta_{\boldsymbol{x}_{\mathrm{pa}(u)}}}(y_u)$ dans les expressions (2.22) et (2.23). En définitive, $\tilde{\beta}_a(\boldsymbol{s}_{\mathbb{S}_a})$ est donné par l'expression (2.24) où $P_{\theta_{\boldsymbol{x}_{\mathrm{pa}(u)}}}(y_u)$ est remplacé par la valeur un si Y_u est une variable aléatoire supprimée à valeurs continues.

Propagation dans la phase arrière

Soit un arc $a = (\mathcal{C}_i, \mathcal{C}_j)$ de l'arbre de jonction. Par définition de la phase arrière, les quantités $\tilde{\beta}_{a_{j_l}}(\boldsymbol{s}_{\mathbb{S}_{a_{j_l}}})$ sont supposées connues pour tous les arcs a_{j_l} incidents à $\mathcal{C}_j$ autres que l'arc a (qui est le seul menant vers $\mathcal{C}_0$) et pour toutes les valeurs possibles de $\boldsymbol{s}_{\mathbb{S}_{a_{j_l}}} \in \mathcal{S}_{\mathbb{S}_{a_{j_l}}}$. Le but de la propagation est de calculer $\tilde{\beta}_a(\boldsymbol{s}_{\mathbb{S}_a})$ à partir des $\tilde{\beta}_{a_{j_l}}(\boldsymbol{s}_{\mathbb{S}_{a_{j_l}}})$.

La formule (2.19)

$$\tilde{\beta}_a(\boldsymbol{s}_{\mathbb{S}_a}) = \sum_{\boldsymbol{s}_{\mathcal{H}_a}} P(\boldsymbol{Y}_{\mathcal{C}_j\setminus \mathbb{S}_a} = \boldsymbol{y}_{\mathcal{C}_j\setminus \mathbb{S}_a}, \boldsymbol{S}_{\mathcal{H}_a} = \boldsymbol{s}_{\mathcal{H}_a} | \boldsymbol{X}_{\mathbb{S}_a} = \boldsymbol{x}_{\mathbb{S}_a}) \prod_l \tilde{\beta}_{a_{j_l}}(\boldsymbol{s}_{\mathbb{S}_{a_{j_l}}}),$$

reste valable puisque $\mathcal{H}_a = \bigcup_l \mathbb{S}_{a_{j_l}} \setminus \mathbb{S}_a$ est l'ensemble des variables aléatoires de $\mathcal{C}_j$ étant présentes dans au moins l'une des cliques $\mathcal{C}_{j_l}$ mais pas dans $\mathcal{C}_i$. L'ensemble $\mathcal{H}_a$ ne peut donc contenir de variable aléatoire Y_u à valeurs continues, car ces dernières ne peuvent appartenir à aucun séparateur de cliques (voir proposition 10). Cependant, en général, la loi conditionnelle

$$P(\boldsymbol{Y}_{\mathcal{C}_j \setminus \mathcal{S}_a} = \boldsymbol{y}_{\mathcal{C}_j \setminus \mathcal{S}_a}, \boldsymbol{S}_{\mathcal{H}_a} = \boldsymbol{s}_{\mathcal{H}_a} | \boldsymbol{X}_{\mathcal{S}_a} = \boldsymbol{x}_{\mathcal{S}_a})$$

ne peut être obtenue que par sommation (ou intégration, dans le cas de variables continues), de la loi conditionnelle

$$P(\boldsymbol{Y}_{\mathcal{C}_j \setminus \mathcal{S}_a} = \boldsymbol{y}_{\mathcal{C}_j \setminus \mathcal{S}_a}, \boldsymbol{S}_{\mathcal{C}_j \setminus \mathcal{S}_a} = \boldsymbol{s}_{\mathcal{C}_j \setminus \mathcal{S}_a} | \boldsymbol{X}_{\mathcal{S}_a} = \boldsymbol{x}_{\mathcal{S}_a}) \tag{3.14}$$

qui est donnée en fonction des paramètres par la formule (2.22) ou la formule (2.23).

Dans le cas où $\mathcal{C}_j$ contient une variable aléatoire supprimée Y_u à valeurs continues, nous sommes donc amenés à intégrer la quantité

$$\sum_{\boldsymbol{s}_{\mathcal{C}_j \setminus \mathcal{S}_a}} P(\boldsymbol{X}_{\mathcal{C}_j \setminus \mathcal{S}_a} = \boldsymbol{x}_{\mathcal{C}_j \setminus \mathcal{S}_a} | \boldsymbol{X}_{\mathcal{S}_a} = \boldsymbol{x}_{\mathcal{S}_a}) \prod_l \tilde{\beta}_{a_{j_l}}(\boldsymbol{s}_{\mathcal{S}_{a_{j_l}}})$$

par rapport à y_u. L'intégration se fait en remarquant que les quantités $(\tilde{\beta}_{a_{j_l}}(\boldsymbol{s}_{\mathcal{S}_{a_{j_l}}}))_l$ sont algébriquement indépendantes de y_u. Seule la quantité (3.14) dépend de y_u, à travers $P_{\theta_{\boldsymbol{x}_{\mathrm{pa}(u)}}}(y_u)$ uniquement. D'après la remarque 2.13, l'intégration se fait en remplaçant $P_{\theta_{\boldsymbol{x}_{\mathrm{pa}(u)}}}(y_u)$ par la valeur un. Compte tenu de ce qui précède, l'équation de propagation est donc, dans le cas où $\mathcal{C}_j$ contient une variable aléatoire supprimée Y_u à valeurs continues :

$$\tilde{\beta}_a(\boldsymbol{s}_{\mathcal{S}_a}) = \sum_{\boldsymbol{s}_{\mathcal{C}_j \setminus \mathcal{S}_a}} \prod_l \tilde{\beta}_{a_{j_l}}(\boldsymbol{s}_{\mathcal{S}_{a_{j_l}}}) \int_{\mathcal{Y}_u} P(\boldsymbol{X}_{\mathcal{C}_j \setminus \mathcal{S}_a} = \boldsymbol{x}_{\mathcal{C}_j \setminus \mathcal{S}_a} | \boldsymbol{X}_{\mathcal{S}_a} = \boldsymbol{x}_{\mathcal{S}_a}) \, dy_u \, ; \tag{3.15}$$

dans le cas contraire, il s'agit de l'équation usuelle (2.20).

Terminaison

La vraisemblance de λ est obtenue lorsque tous les arcs de l'arbre de jonction ont été parcourus. Nous notons $a_1, \ldots, a_l$ les arcs incidents à $\mathcal{C}_0$ dans l'arbre de jonction. Par un raisonnement analogue à celui utilisé pour la propagation et d'après l'équation (2.25), la vraisemblance est donnée, dans le cas où $\mathcal{C}_0$ contient une variable aléatoire supprimée Y_u à valeurs continues, par

$$P(\boldsymbol{Y}_{Obs} = \boldsymbol{y}_{Obs}) = \sum_{\boldsymbol{s}_{\mathcal{C}_0} \in \mathcal{S}_{\mathcal{C}_0}} \prod_l \tilde{\beta}_{a_l}(\boldsymbol{s}_{\mathcal{S}_{a_l}}) \int_{\mathcal{Y}_u} P(\boldsymbol{X}_{\mathcal{C}_0} = \boldsymbol{x}_{\mathcal{C}_0}) \, dy_u \, ; \tag{3.16}$$

et dans le cas contraire, par l'équation (2.25). Rappelons que dans les deux cas, la loi jointe de $\boldsymbol{X}_{\mathcal{C}_0}$ est

$$P(\boldsymbol{X}_{\mathcal{C}_0} = \boldsymbol{x}_{\mathcal{C}_0}) = \prod_i P(X_0^{(i)} | X_0^{(1)}, \ldots, X_0^{(i-1)}),$$

où pour tout i, les probabilités $P(X_0^{(i)} = x_0^{(i)} | X_0^{(1)} = x_0^{(1)}, \ldots, X_0^{(i-1)} = x_0^{(i-1)})$ sont données par les paramètres du modèle, vu que $\mathrm{pa}(X_0^{(i)}) = \{X_0^{(1)}, \ldots, X_0^{(i-1)}\}$. De plus, dans l'équation (3.16), l'intégration se fait en remplaçant $P_{\theta_{\boldsymbol{x}_{\mathrm{pa}(u)}}}(y_u)$ par la valeur un.

Phase avant

La phase avant est un parcours de graphe qui part de $\mathcal{C}_0$ et descend jusqu'aux feuilles de l'arbre des cliques, parcourant ainsi les arcs de cet arbre dans le même sens que leur orientation. Son but est le calcul, pour chaque clique $\mathcal{C}$, des probabilités

$$P(\boldsymbol{S}_{\mathcal{C}} = \boldsymbol{s}_{\mathcal{C}}, \boldsymbol{Y}_{\mathcal{C}\cap Man} = \boldsymbol{y}_{\mathcal{C}\cap Man} | \boldsymbol{Y}_{Obs} = \boldsymbol{y}_{Obs})$$

intervenant dans l'algorithme EM de la section courante 3.4.3. Elle utilise les probabilités $\tilde{\alpha}_a$ définies comme suit : soit a un arc de l'arbre de jonction $\mathcal{T}$, avec $a = (\mathcal{C}_i, \mathcal{C}_j)$, la clique $\mathcal{C}_i$ étant donc située sur le chemin entre $\mathcal{C}_0$ et $\mathcal{C}_j$ dans l'arbre de jonction. La fonction $\tilde{\alpha}_a$ est définie par

$$\tilde{\alpha}_a(\boldsymbol{s}_{\mathbb{S}_a}) = P(\boldsymbol{Y}_{\mathcal{K}_a^c\cap Obs} = \boldsymbol{y}_{\mathcal{K}_a^c\cap Obs}, \boldsymbol{Y}_{\mathbb{S}_a} = \boldsymbol{y}_{\mathbb{S}_a}, \boldsymbol{S}_{\mathbb{S}_a} = \boldsymbol{s}_{\mathbb{S}_a})$$

pour $\boldsymbol{s}_{\mathbb{S}_a} \in \mathcal{S}_{\mathbb{S}_a}$. De même que pour la définition des quantités $\tilde{\beta}_a(\boldsymbol{s}_{\mathbb{S}_a})$, les variables aléatoires du processus $\boldsymbol{Y}_{\mathbb{S}_a}$ sont supposées être toutes observées. En outre, la phase avant utilise des quantités calculées dans la phase arrière et doit donc être effectuée après celle-ci (voir l'équation (3.18) et aussi la remarque 2.17 concernant le cas particulier des chaînes de Markov cachées).

La phase avant est initialisée comme suit : soit $a = (\mathcal{C}_0, \mathcal{C}_j)$ un arc incident à $\mathcal{C}_0$. On note $(a_{j_l})_l$ les autres arcs incidents à $\mathcal{C}_0$. Par définition, la quantité $\tilde{\alpha}_a(\boldsymbol{s}_{\mathbb{S}_a})$ est initialisée en calculant la probabilité $P(\boldsymbol{Y}_{\mathcal{K}_a^c\cap Obs} = \boldsymbol{y}_{\mathcal{K}_a^c\cap Obs}, \boldsymbol{X}_{\mathbb{S}_a} = \boldsymbol{x}_{\mathbb{S}_a})$. Celle-ci se déduit des probabilités conditionnelles $(P(\boldsymbol{Y}_{\mathcal{K}_{a_{j_l}}} = \boldsymbol{y}_{\mathcal{K}_{a_{j_l}}} | \boldsymbol{X}_{\mathbb{S}_{a_{j_l}}} = \boldsymbol{x}_{\mathbb{S}_{a_{j_l}}}))_l$, suivant un raisonnement analogue à celui aboutissant à la formule d'initialisation (2.26) de la section 2.4.3 et la formule (3.16) ci-dessus pour le calcul de la vraisemblance. Ainsi, dans le cas où $\mathcal{C}_0$ contient une variable aléatoire supprimée Y_u à valeurs continues,

$$\tilde{\alpha}_a(\boldsymbol{s}_{\mathbb{S}_a}) = \sum_{\boldsymbol{s}_{\mathcal{C}_0 \setminus \mathbb{S}_a}} \prod_l \tilde{\beta}_{a_{j_l}}(\boldsymbol{s}_{\mathbb{S}_{a_{j_l}}}) \int_{\mathcal{Y}_u} P(\boldsymbol{X}_{\mathcal{C}_0} = \boldsymbol{x}_{\mathcal{C}_0}) \, dy_u \,, \tag{3.17}$$

où l'intégrale

$$\int_{\mathcal{Y}_u} P(\boldsymbol{X}_{\mathcal{C}_0} = \boldsymbol{x}_{\mathcal{C}_0}) \, dy_u$$

a déjà été calculée dans l'étape terminale de la phase arrière (rappelons qu'il suffit de remplacer $P_{\theta_{\boldsymbol{x}_{\mathrm{pa}(u)}}}(y_u)$ par la valeur un). Dans le cas où $\mathcal{C}_0$ ne contient pas de variable aléatoire supprimée à valeurs continues, l'équation (2.26) reste valide pour l'initialisation de $\tilde{\alpha}_a(\boldsymbol{s}_{\mathbb{S}_a})$.

La propagation dans la phase avant se fait en considérant chaque arc $a = (\mathcal{C}_i, \mathcal{C}_j)$ de l'arbre des cliques. On note a_{i_0} l'arc incident à $\mathcal{C}_i$ qui intervient dans le chemin entre $\mathcal{C}_0$ et $\mathcal{C}_i$ dans l'arbre de jonction et $(a_{i_l})_l$ les autres arcs, différents de a et incidents à $\mathcal{C}_i$. Le raisonnement précédent est aisément adapté pour obtenir une équation de propagation analogue à la formule (2.27). Ainsi, l'équation de propagation dans la phase avant, dans le cas où $\mathcal{C}_j$ contient une variable aléatoire supprimée Y_u à valeurs continues, est la suivante :

$$\tilde{\alpha}_a(\boldsymbol{s}_{\mathcal{S}_a}) = \sum_{\boldsymbol{s}_{\mathcal{C}_i \backslash \mathcal{S}_a}} \tilde{\alpha}_{i_0}(\boldsymbol{s}_{\mathcal{S}_{a_{i_0}}}) \prod_l \tilde{\beta}_{a_{i_l}}(\boldsymbol{s}_{\mathcal{S}_{a_{i_l}}}) \int_{\mathcal{Y}_u} P(\boldsymbol{X}_{\mathcal{C}_i \backslash \mathcal{S}_{a_{i_0}}} = \boldsymbol{x}_{\mathcal{C}_i \backslash \mathcal{S}_{a_{i_0}}} | \boldsymbol{X}_{\mathcal{S}_{a_{i_0}}} = \boldsymbol{x}_{\mathcal{S}_{a_{i_0}}}) \, dy_u \; ; \tag{3.18}$$

dans le cas contraire, il s'agit de l'équation usuelle (2.27). L'équation (3.18) provient du fait que les quantités $(\tilde{\beta}_{a_{i_l}}(\boldsymbol{s}_{\mathcal{S}_{a_{i_l}}}))_l$ et $\tilde{\alpha}_{i_0}(\boldsymbol{s}_{\mathcal{S}_{a_{i_0}}})$ sont algébriquement indépendantes de y_u. Seule la quantité

$$P(\boldsymbol{X}_{\mathcal{C}_i \backslash \mathcal{S}_{a_{i_0}}} = \boldsymbol{x}_{\mathcal{C}_i \backslash \mathcal{S}_{a_{i_0}}} | \boldsymbol{X}_{\mathcal{S}_{a_{i_0}}} = \boldsymbol{x}_{\mathcal{S}_{a_{i_0}}})$$

dépend de y_u, à travers $P_{\boldsymbol{\theta}_{\boldsymbol{x}_{\mathrm{pa}(u)}}}(y_u)$ uniquement, c'est pourquoi d'après la remarque 2.13, il suffit de remplacer $P_{\boldsymbol{\theta}_{\boldsymbol{x}_{\mathrm{pa}(u)}}}(y_u)$ par la valeur un dans l'expression de la loi conditionnelle de $\boldsymbol{X}_{\mathcal{C}_i \backslash \mathcal{S}_{a_{i_0}}}$ pour calculer l'intégrale ci-dessus.

Calcul des probabilités intervenant dans l'étape E

Rappelons qu'il suffit de connaître les probabilités

$$P(\boldsymbol{S}_{\mathcal{C}} = \boldsymbol{s}_{\mathcal{C}}, \boldsymbol{Y}_{\mathcal{C} \cap Man} = \boldsymbol{y}_{\mathcal{C} \cap Man} | \boldsymbol{Y}_{Obs} = \boldsymbol{y}_{Obs})$$

pour pouvoir implémenter l'étape E de l'algorithme EM. L'algorithme ci-dessus permet en fait de calculer une probabilité jointe au lieu d'une probabilité conditionnelle. Le calcul peut se baser sur une décomposition directe de cette probabilité jointe, inspirée de l'équation (2.28), de manière à obtenir (en utilisant les mêmes notations que pour la propagation dans l'algorithme avant)

$$P(\boldsymbol{S}_{\mathcal{C}_i} = \boldsymbol{s}_{\mathcal{C}_i}, \boldsymbol{Y}_{\mathcal{C}_i \cap Man} = \boldsymbol{y}_{\mathcal{C}_i \cap Man}, \boldsymbol{Y}_{Obs} = \boldsymbol{y}_{Obs}) = P(\boldsymbol{X}_{\mathcal{C}_i \backslash \mathcal{S}_{a_{i_0}}} = \boldsymbol{x}_{\mathcal{C}_i \backslash \mathcal{S}_{a_{i_0}}} | \boldsymbol{X}_{\mathcal{S}_{a_{i_0}}} = \boldsymbol{x}_{\mathcal{S}_{a_{i_0}}}) \tilde{\alpha}_{a_{i_0}}(\boldsymbol{s}_{\mathcal{S}_{a_{i_0}}}) \tilde{\beta}_a(\boldsymbol{s}_{\mathcal{S}_a}) \prod_l \tilde{\beta}_{a_{i_l}}(\boldsymbol{s}_{\mathcal{S}_{a_{i_l}}}). \tag{3.19}$$

La vraisemblance $P(\boldsymbol{Y}_{Obs} = \boldsymbol{y}_{Obs})$ étant calculée par l'équation (3.16), on en déduit les probabilités conditionnelles voulues.

Notons qu'en pratique, comme le montrent les formules de réestimation, on a besoin de connaître également, pour certains sommets u, les probabilités $P(\boldsymbol{X}_{\mathrm{pa}(u)} = \boldsymbol{a} | \boldsymbol{Y}_{Obs} = \boldsymbol{y}_{Obs})$, où les variables aléatoires $\boldsymbol{X}_{\mathrm{pa}(u)}$ appartiennent à une même clique $\mathcal{C}_i$ et où les variables aléatoires $\boldsymbol{Y}_{\mathrm{pa}(u)}$ sont à valeurs discrètes. Cette probabilité se déduit de

$$P(\boldsymbol{S}_{\mathcal{C}_i} = \boldsymbol{s}_{\mathcal{C}_i}, \boldsymbol{Y}_{\mathcal{C}_i \cap Man} = \boldsymbol{y}_{\mathcal{C}_i \cap Man} | \boldsymbol{Y}_{Obs} = \boldsymbol{y}_{Obs})$$

par sommation sur les variables discrètes de $\boldsymbol{X}_{\mathcal{C}_i \backslash \mathrm{pa}(u)}$ et par intégration sur les variables continues de ce même processus. Par la proposition 10, $\boldsymbol{X}_{\mathcal{C}_i \backslash \mathrm{pa}(u)}$ contient au plus une variable aléatoire à valeurs continues ; dans ce cas nous la notons Y_v. Nous sommes alors amenés à calculer

$$\int_{\mathcal{Y}_v} P(\boldsymbol{S}_{\mathcal{C}_i} = \boldsymbol{s}_{\mathcal{C}_i}, \boldsymbol{Y}_{\mathcal{C}_i \cap Man} = \boldsymbol{y}_{\mathcal{C}_i \cap Man}, \boldsymbol{Y}_{Obs} = \boldsymbol{y}_{Obs}) \, dy_v = \tilde{\alpha}_{a_{i_0}}(\boldsymbol{s}_{\mathcal{S}_{a_{i_0}}}) \tilde{\beta}_a(\boldsymbol{s}_{\mathcal{S}_a}) \prod_l \tilde{\beta}_{a_{i_l}}(\boldsymbol{s}_{\mathcal{S}_{a_{i_l}}}) \int_{\mathcal{Y}_v} P(\boldsymbol{X}_{\mathcal{C}_i \backslash \mathcal{S}_{a_{i_0}}} = \boldsymbol{x}_{\mathcal{C}_i \backslash \mathcal{S}_{a_{i_0}}} | \boldsymbol{X}_{\mathcal{S}_{a_{i_0}}} = \boldsymbol{x}_{\mathcal{S}_{a_{i_0}}}) \, dy_v \, ,$$

car les quantités $\tilde{\alpha}_{a_{i_0}}(\boldsymbol{s}_{\mathcal{S}_{a_{i_0}}})$, $\tilde{\beta}_a(\boldsymbol{s}_{\mathcal{S}_a})$ et $(\tilde{\beta}_{a_{i_l}}(\boldsymbol{s}_{\mathcal{S}_{a_{i_l}}}))_l$, de même que précédemment, ne dépendent pas de y_v. L'intégrale

$$\int_{\mathcal{Y}_v} P(\boldsymbol{X}_{\mathcal{C}_i \backslash \mathcal{S}_{a_{i_0}}} = \boldsymbol{x}_{\mathcal{C}_i \backslash \mathcal{S}_{a_{i_0}}} | \boldsymbol{X}_{\mathcal{S}_{a_{i_0}}} = \boldsymbol{x}_{\mathcal{S}_{a_{i_0}}})\, dy_v\,,$$

d'après la remarque 2.13, se calcule en remplaçant $P_{\theta_{\boldsymbol{x}_{\mathrm{pa}(u)}}}(y_v)$ par la valeur un dans l'expression de la loi conditionnelle de $\boldsymbol{X}_{\mathcal{C}_i \backslash \mathcal{S}_{a_{i_0}}}$.

Remarque 3.3 *Il est possible de justifier l'algorithme arrière-avant ci-dessus de manière légèrement différente, en se basant sur l'équation (2.28). À partir de la probabilité* $P(\boldsymbol{S}_{\mathcal{C}_i} = \boldsymbol{s}_{\mathcal{C}_i}, \boldsymbol{Y} = \boldsymbol{y})$, *on calcule alors*

$$P(\boldsymbol{S}_{\mathcal{C}_i} = \boldsymbol{s}_{\mathcal{C}_i}, \boldsymbol{Y}_{\mathcal{C}_i \cap Man} = \boldsymbol{y}_{\mathcal{C}_i \cap Man}, \boldsymbol{Y}_{Obs} = \boldsymbol{y}_{Obs}) = \int_{\mathcal{Y}_{Man \backslash \mathcal{C}_i}} P(\boldsymbol{S}_{\mathcal{C}_i} = \boldsymbol{s}_{\mathcal{C}_i}, \boldsymbol{Y} = \boldsymbol{y})\, d\boldsymbol{y}_{Man \backslash \mathcal{C}_i}.$$

En observant que les ensembles $\bar{\mathcal{K}}^c_{a_{i_0}}$, $\mathcal{C}_i$ *et* $(\bar{\mathcal{K}}_{a_{i_l}})_l$ *définissent une partition des variables aléatoires à valeurs continues, et en utilisant la factorisation (2.28) de* $P(\boldsymbol{S}_{\mathcal{C}_i} = \boldsymbol{s}_{\mathcal{C}_i}, \boldsymbol{Y} = \boldsymbol{y})$, *on est amené, par le théorème de Fubini, à intégrer séparément les quantités*

$$P(\boldsymbol{Y}_{\mathcal{K}^c_{a_{i_0}}} = \boldsymbol{y}_{\mathcal{K}^c_{a_{i_0}}}, \boldsymbol{Y}_{\mathcal{S}_{a_{i_0}}} = \boldsymbol{y}_{\mathcal{S}_{a_{i_0}}}, \boldsymbol{S}_{\mathcal{S}_{a_{i_0}}} = \boldsymbol{s}_{\mathcal{S}_{a_{i_0}}})$$

et

$$P(\boldsymbol{Y}_{\mathcal{K}_b} = \boldsymbol{y}_{\mathcal{K}_b} | \boldsymbol{Y}_{\mathcal{S}_b} = \boldsymbol{y}_{\mathcal{S}_b}, \boldsymbol{S}_{\mathcal{S}_b} = \boldsymbol{s}_{\mathcal{S}_b}),$$

pour chaque arc b dans $\{a\} \cup \{a_{i_l}\}_l$. *On en déduit un algorithme arrière-avant où les nouvelles quantités arrière* $\tilde{\beta}_a$ *sont les intégrales des quantités arrière définies dans la section 2.4.2 et où les nouvelles quantités avant* $\tilde{\alpha}_a$ *sont les intégrales des quantités avant définies dans la section 2.4.3. Cet algorithme est exactement l'algorithme présenté ci-dessus, mais présenté en adoptant un point de vue différent.*

Application aux chaînes de Markov cachées

L'application des résultats précédents aux chaînes de Markov cachées conduit à l'algorithme avant-arrière ci-dessous. On suppose que la chaîne est de longueur n mais que seules les variables aléatoires $(Y_t)_{t \in Obs}$ sont observées, avec $Obs \subset \{1, \ldots, n\}$. La phase avant qui, rappelons-le, peut dans ce cas particulier être exécutée avant la phase arrière, est basée sur les quantités

$$\tilde{\alpha}_t(j) = P(\bigcap_{\substack{1 \leq u \leq t \\ u \in Obs}} \{Y_u = y_u\} \cap \{S_t = j\}).$$

Ces quantités sont initialisées par

$$\tilde{\alpha}_1(j) = \begin{cases} \pi_j f_{\theta_j}(y_t) \text{ si } t \in Obs\ ; \\ \pi_j \text{ sinon.} \end{cases}$$

La propagation dans la phase avant est donnée par

$$\tilde{\alpha}_{t+1}(k) = \begin{cases} \sum_{j=1}^{K} a_{jk}\tilde{\alpha}_t(j) f_{\theta_k}(y_{t+1}) \text{ si } t \in Obs \ ; \\ \sum_{j=1}^{K} a_{jk}\tilde{\alpha}_t(j) \text{ sinon.} \end{cases}$$

La phase arrière est basée sur les quantités $\tilde{\beta}_t(j)$, égales à

$$\tilde{\beta}_t(j) = P(\bigcap_{\substack{t<u\leq n \\ u\in Obs}} \{Y_u = y_u\} | S_t = j).$$

Elle est initialisée par $\tilde{\beta}_n(i) = 1$ et la propagation se fait suivant la récursion

$$\tilde{\beta}_t(j) = \begin{cases} \sum_{k=1}^{K} a_{jk}\tilde{\beta}_{t+1}(k) f_{\theta_k}(y_{t+1}) \text{ si } t \in Obs \ ; \\ \sum_{k=1}^{K} a_{jk}\tilde{\beta}_{t+1}(k) \text{ sinon.} \end{cases}$$

On constate donc que les formules d'initialisation et de propagation dans les phases avant et arrière correspondent aux formules usuelles (2.37) et (2.36) où les quantités $f_{\theta_k}(y_t)$, lorsqu'elles sont inconnues, sont remplacées par la valeur un. Ceci coïncide avec le résultat annoncé au début de la partie concernant l'étape E et le calcul de probabilités dans les modèles à observations supprimées. De plus, ces formules amènent des commentaires qui se généralisent à tous les modèles de la famille $\mathcal{D}$.

Tout d'abord, on constate que dans le cas où Y_t est supprimée, Y_{t-1} et Y_{t+1} étant observées, les quantités $(\tilde{\alpha}_{t+1}(k))_k$ sont obtenues en fonction des $(\tilde{\alpha}_{t-1}(l))_l$ par

$$\tilde{\alpha}_{t+1}(k) = \sum_j a_{jk} \sum_l a_{lj}\tilde{\alpha}_{t-1}(l) f_{\theta_k}(y_{t+1}) = \sum_l \left[\sum_j a_{lj}a_{jk}\right] \tilde{\alpha}_{t-1}(l) f_{\theta_k}(y_{t+1}).$$

On voit donc apparaître les coefficients de la matrice A^2. Ceci est lié au fait que pour une chaîne de Markov $\boldsymbol{S}$, la transition entre S_{t-1} et S_{t+1} est régie par la matrice A^2. On montre aisément par récurrence que si un nombre n' de données consécutives est supprimé entre deux observations, la quantité avant concernant la deuxième observation fait intervenir la matrice $A^{n'+1}$. Ceci est illustré sur la figure 3.4 ci-dessous. La même interprétation est valable pour la phase arrière de l'algorithme.

Rappelons que d'autre part, $\mathcal{G}$ désigne la structure d'une chaîne de Markov cachée et λ ses paramètres. On constate alors que Y_{Obs} a même loi que le modèle graphique de paramètres λ défini par le graphe $\mathcal{G}$ privé des sommets $\boldsymbol{Y}_{Man}$ et des arcs incidents à ces sommets.

Conclusion sur le calcul de probabilités

Notre algorithme arrière-avant permet donc le calcul de la vraisemblance et des quantités intervenant dans l'étape E de l'algorithme EM, dans le cas de données observées

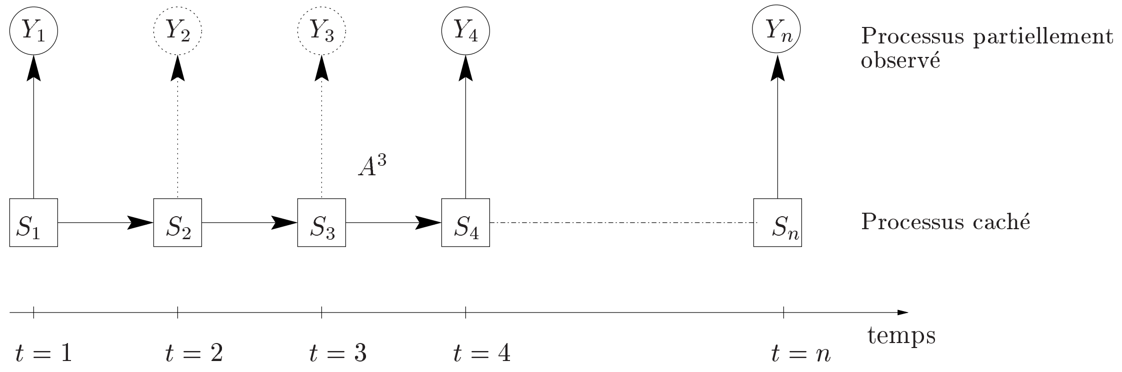


FIG. 3.4 – *Suppression d'observations quelconques dans une chaîne de Markov cachée. Dans le cas où les variables aléatoires observées Y_{t+1} à $Y_{t+n'}$ sont supprimées, l'algorithme avant passe de $\tilde{\alpha}_t$ à $\tilde{\alpha}_{t+n'+1}$ en calculant $A^{n'+1}$. Ceci est illustré ci-dessus pour $n' = 2$.*

supprimées. Ceci permet essentiellement l'implémentation de la validation croisée et l'estimation de paramètres quand la collecte des données n'a pas permis d'obtenir $\boldsymbol{y}$ dans sa totalité. De plus, le problème général du calcul de probabilités énoncé dans la section 1.4 est à présent totalement résolu pour les modèles de la famille $\mathcal{D}$. Autrement dit, le calcul des probabilités conditionnelles $P(\boldsymbol{X}_A = \boldsymbol{x}_A | \boldsymbol{X}_B = \boldsymbol{x}_B)$ est possible pour tous les sous-ensembles A et B des sommets du graphe $\mathcal{G}$ et pour toutes les valeurs possibles de $\boldsymbol{x}_A$ dans $\mathcal{X}_A$ et de $\boldsymbol{x}_B$ dans $\mathcal{X}_B$.

L'algorithme arrière-avant dans le cas d'observations supprimées est rendu possible par le fait que dans la famille $\mathcal{D}$, il n'existe pas d'arc entre variables aléatoires à valeurs continues. Les conséquences en sont multiples et donnent lieu aux interprétations suivantes de ce nouvel algorithme :

1. la prise en compte des Y_u non observés à valeurs continues se fait simplement en remplaçant $P_{\theta_{\boldsymbol{a}}}(y_u)$ par la valeur un dans l'algorithme arrière-avant des sections 2.4.2 et 2.4.3. Les Y_u supprimés à valeurs discrètes n'ayant pas d'arc sortant sont également traités de cette manière, c'est-à-dire en posant $p_{\boldsymbol{a},y_u} = 1$ (voir le point 2). En revanche, les autres Y_u à valeurs discrètes ayant été supprimés sont traités comme des états cachés S_u ;

2. la non observation de sommets $Man \subset \mathcal{U}$ correspondant à des variables aléatoires puits, *i.e.* sans arc sortant, aboutit à une loi $P_{\lambda,\boldsymbol{Y}_{Obs}}$ égale à la loi $P_{\lambda,\boldsymbol{Y}'}$ associée au modèle graphique $\mathcal{G}$ privé des sommets $\boldsymbol{X}_{Man}$ et des arcs incidents à ces sommets. Ceci n'est pas vrai pour les sommets autres que les puits : dans une chaîne de Markov cachée, le processus $\boldsymbol{S}$ est non observé. Pour autant, le modèle graphique obtenu en supprimant, dans la figure 3.4, les variables cachées, ne conduit pas à la même loi pour $\boldsymbol{Y}$. Une conséquence de cette propriété est que les états cachés puits sont inutiles dans les modèles de la famille $\mathcal{D}$ et peuvent être ignorés ;

3. le fait que certaines variables aléatoires puits Y_u soient supprimées conduit, pour les variables Y_{Obs}, à calculer des produits entre tenseurs de transitions p associés aux variables supprimées.

D'autre part, pour les raisons détaillées dans la section 2.4.5, les quantités $\tilde{\beta}_a$ tendent vers zéro lorsque l'arc a de l'arbre de jonction se rapproche de la clique racine $\mathcal{C}_0$. Dans le cas où le nombre de données observées Y_{Obs} tend vers l'infini, des limitations numériques surviennent lors de l'exécution de l'algorithme. C'est pourquoi on utilisera un algorithme arrière-avant basé sur des probabilités de type "lissage" en remplaçant $P_{\theta_{\boldsymbol{a}}}(y_u)$ par la valeur un pour les Y_u manquants à valeurs continues, dans l'algorithme de la section 2.4.5. L'algorithme du MAP pour la restauration des données cachées de la section 2.5 s'adapte de la même manière que ci-dessus, c'est-à-dire en remplaçant $P_{\theta_{\boldsymbol{a}}}(y_u)$ par la valeur un pour les Y_u manquants à valeurs continues.

3.5 Critère d'information bayésienne BIC

L'approche bayésienne consiste à sélectionner le modèle le plus probable sachant les observations. On suppose qu'il y a M modèles en compétition, formant l'ensemble $\{\mathcal{M}_m\}_{m\in\{1,\ldots,M\}}$. Le modèle inconnu $\boldsymbol{\mathcal{M}}$ est une variable aléatoire dont la loi est caractérisée par $(P(\boldsymbol{\mathcal{M}} = \mathcal{M}_m))_{m\in\{1,\ldots,M\}}$. Le modèle sélectionné est celui maximisant $P(\boldsymbol{\mathcal{M}} = \mathcal{M}_m|\boldsymbol{Y} = \boldsymbol{y})$, cette probabilité étant déduite, par le théorème de Bayes, de $P(\boldsymbol{\mathcal{M}} = \mathcal{M}_m)$ et de la vraisemblance intégrée

$$\begin{aligned} P(\boldsymbol{Y} = \boldsymbol{y}|\boldsymbol{\mathcal{M}} = \mathcal{M}_m) &= \int\limits_{\Lambda_m} P(\boldsymbol{Y} = \boldsymbol{y}, \boldsymbol{\lambda}_m = \lambda_m|\boldsymbol{\mathcal{M}} = \mathcal{M}_m)\, d\lambda_m \\ &= \int\limits_{\Lambda_m} P(\boldsymbol{Y} = \boldsymbol{y}|\boldsymbol{\lambda}_m = \lambda_m, \boldsymbol{\mathcal{M}} = \mathcal{M}_m) P(\boldsymbol{\lambda}_m = \lambda_m|\boldsymbol{\mathcal{M}} = \mathcal{M}_m)\, d\lambda_m \qquad (3.20) \end{aligned}$$

où $P(\boldsymbol{Y} = \boldsymbol{y}|\boldsymbol{\lambda}_m = \lambda_m, \boldsymbol{\mathcal{M}} = \mathcal{M}_m)$ est la vraisemblance $\mathcal{L}_{\boldsymbol{y},\mathcal{M}_m}(\lambda_m)$ sous le modèle $\mathcal{M}_m$, où $P(\boldsymbol{\lambda}_m = \lambda_m|\boldsymbol{\mathcal{M}} = \mathcal{M}_m)$ est la loi a priori de $\boldsymbol{\lambda}_m$ sous $\mathcal{M}_m$ et où Λ_m est l'espace des paramètres sous $\mathcal{M}_m$. Par la suite, nous supposerons que $P(\boldsymbol{\mathcal{M}} = \mathcal{M}_m) = \frac{1}{M}$: comparer les probabilités $P(\boldsymbol{\mathcal{M}} = \mathcal{M}_m|\boldsymbol{Y} = \boldsymbol{y})$ revient alors à comparer les vraisemblances intégrées $P(\boldsymbol{Y} = \boldsymbol{y}|\boldsymbol{\mathcal{M}} = \mathcal{M}_m)$.

Facteurs de Bayes et critère BIC

Pour les modèles de Markov cachés – comme d'ailleurs pour la plupart des autres modèles – la vraisemblance intégrée ne peut être calculée directement. On peut alors l'approximer de manière asymptotique par la méthode de Laplace. La présentation ci-dessous de différentes variantes pour l'approximation de la vraisemblance intégrée s'inspire de Kass et Raftery, 1995 [69].

L'emploi de la méthode de Laplace est basé sur l'hypothèse que la loi conditionnelle $P(\boldsymbol{\lambda} = \lambda|\boldsymbol{Y} = \boldsymbol{y}, \boldsymbol{\mathcal{M}} = \mathcal{M})$ (cette probabilité étant proportionnelle à $\mathcal{L}_{\boldsymbol{y},\mathcal{M}}(\lambda)P(\boldsymbol{\lambda} = \lambda|\boldsymbol{\mathcal{M}} = \mathcal{M})$), est concentrée autour de son maximum $\tilde{\lambda}$ qui est le MAP. Cette hypothèse est en général vérifiée si $\mathcal{L}_{\boldsymbol{y}}(\lambda)$ est concentrée autour de son maximum $\hat{\lambda}$, ce qui est le cas quand n tend vers l'infini. De plus, en faisant l'hypothèse que la loi a priori $P(\boldsymbol{\lambda} = \lambda)$ est centrée asymptotiquement sur une valeur proche de $\tilde{\lambda}$, on assure une approximation plus précise. On pose alors $\tilde{l}_{\boldsymbol{y},\mathcal{M}}(\lambda) = \log(\mathcal{L}_{\boldsymbol{y},\mathcal{M}}(\lambda)P(\boldsymbol{\lambda} = \lambda|\boldsymbol{\mathcal{M}} = \mathcal{M}))$. Par un développement limité à l'ordre deux de $\tilde{l}_{\boldsymbol{y},\mathcal{M}}$ autour de $\tilde{\lambda}$ et en considérant l'exponentielle de ce développement limité, on obtient une approximation de $\tilde{l}_{\boldsymbol{y},\mathcal{M}}(\lambda)$ ayant la forme d'une densité normale de moyenne $\tilde{\lambda}$ et de matrice de covariance $\tilde{\Sigma}$ définie par l'inverse de la hessienne de $\tilde{l}_{\boldsymbol{y},\mathcal{M}}(\lambda)$ en $\lambda = \tilde{\lambda}$. Intégrer cette approximation – ou plus rigoureusement, appliquer la méthode de Laplace à l'intégrale

$$I = \int_{\Lambda} P(\boldsymbol{\lambda} = \lambda | \boldsymbol{\mathcal{M}} = \mathcal{M}) \exp\left\{-n\left[-n^{-1}\log(\mathcal{L}_{\boldsymbol{y},\mathcal{M}}(\lambda))\right]\right\} d\lambda$$

– conduit à l'approximation suivante

$$I \approx (2\pi)^{\frac{d}{2}} |\tilde{\Sigma}|^{\frac{1}{2}} \mathcal{L}_{\boldsymbol{y},\mathcal{M}}(\tilde{\lambda}) P(\boldsymbol{\lambda} = \tilde{\lambda} | \boldsymbol{\mathcal{M}} = \mathcal{M})$$

où d est la dimension de Λ. Sous des conditions de régularité données par Kass, Tierney et Kadane, 1990 [70], l'erreur relative commise est $\mathcal{O}(n^{-1})$.

Il est également possible d'effectuer un développement limité à l'ordre deux de $\tilde{l}_{\boldsymbol{y},\mathcal{M}}$ autour de $\hat{\lambda}$ pour obtenir

$$\hat{I} = (2\pi)^{\frac{d}{2}} |\hat{\Sigma}|^{\frac{1}{2}} \mathcal{L}_{\boldsymbol{y},\mathcal{M}}(\hat{\lambda}) P(\boldsymbol{\lambda} = \hat{\lambda} | \boldsymbol{\mathcal{M}} = \mathcal{M}) \tag{3.21}$$

où $\hat{\Sigma}$ est encore obtenue à partir de la hessienne de $\tilde{l}_{\boldsymbol{y},\mathcal{M}}(\lambda)$, évaluée cette fois en $\lambda = \hat{\lambda}$. L'erreur relative est une fois de plus d'ordre $\mathcal{O}(n^{-1})$, *i.e.* $I = \hat{I}(1 + \mathcal{O}(n^{-1}))$. La matrice $\hat{\Sigma}$, en général inconnue, est approchée par l'inverse de la matrice d'information de Fisher $\mathcal{I}_{\lambda_0}$ où λ_0 est la limite de la suite $(\hat{\lambda}_n)_n$, supposée unique. Si l'on suppose qu'il existe une "vraie" valeur du paramètre, celle-ci vaut λ_0. L'approximation de $\hat{\Sigma}$ par $\mathcal{I}_{\lambda_0}^{-1}$ est justifiée par le théorème de la limite centrale. L'erreur relative d'approximation qui en résulte est plus importante (à savoir $\mathcal{O}(n^{-\frac{1}{2}})$) mais reste en général suffisamment faible pour la sélection de modèles.

La sélection de modèles peut se faire par l'utilisation des *facteurs de Bayes*, définis par le rapport des vraisemblances intégrées d'un modèle $\mathcal{M}_m$ et d'un modèle de référence $\mathcal{M}_{m_0}$, soit

$$\mathrm{B}_{m,m_0} = \frac{P(\boldsymbol{Y} = \boldsymbol{y} | \boldsymbol{\mathcal{M}} = \mathcal{M}_m)}{P(\boldsymbol{Y} = \boldsymbol{y} | \boldsymbol{\mathcal{M}} = \mathcal{M}_{m_0})}.$$

Si le facteur de Bayes est plus grand que un, le modèle $\mathcal{M}_m$ est le plus probable sachant les données observées.

En pratique, on approxime le logarithme du facteur de Bayes en utilisant l'expression (3.21). Si les termes indépendants de n sont ignorés, on obtient une approximation égale au critère ci-dessous, dû à Schwarz, 1978 [107] :

$$\mathrm{S}_m = \ln(\mathcal{L}_{\boldsymbol{y},\mathcal{M}_m}(\hat{\lambda}_m)) - \ln(\mathcal{L}_{\boldsymbol{y},\mathcal{M}_0}(\hat{\lambda}_0)) - \frac{1}{2}(d_m - d_{m_0})\ln(n), \tag{3.22}$$

où d_m est la dimension de Λ_m. L'expression (3.22) est la différence entre deux quantités de la forme $\ln(\mathcal{L}_{\boldsymbol{y},\mathcal{M}}(\hat{\lambda})) - \frac{d}{2}\ln(n)$. Comparer l'approximation du logarithme du facteur de Bayes revient donc à comparer le critère BIC (*Bayesian Information Criterion*) associé à chaque modèle $\mathcal{M}_m$ et défini par

$$\mathrm{BIC}(\mathcal{M}_m) = \ln(\mathcal{L}_{\boldsymbol{y},\mathcal{M}_m}(\hat{\lambda}_m)) - \frac{d_m}{2}\ln(n),$$

puis à sélectionner le modèle maximisant ce critère. Le critère BIC peut être vu comme une approximation d'ordre $\mathcal{O}(1)$ de la log-vraisemblance intégrée. L'approximation de cette dernière basée sur l'équation (3.21) fait intervenir la probabilité a priori $P(\boldsymbol{\lambda}_m = \lambda|\boldsymbol{\mathcal{M}} = \mathcal{M}_m)$, indépendante de n. Par conséquent, ce terme est absorbé dans le résidu $\mathcal{O}(1)$ et le choix explicite de cette loi n'est pas nécessaire. Ceci explique l'utilisation du critère BIC, d'origine bayésienne, dans un cadre non bayésien.

D'autre part, contrairement à l'approximation (3.21), l'erreur relative commise dans (3.22), due à l'approximation de $\ln(P(\boldsymbol{Y} = \boldsymbol{y}|\boldsymbol{\mathcal{M}} = \mathcal{M}_m))$ par $\mathrm{BIC}(\mathcal{M}_m)$, est d'ordre $\mathcal{O}(1)$, puisque le terme constant

$$\frac{d}{2}\ln(2\pi) - \frac{1}{2}\ln(|\mathcal{I}_{\lambda_0}|) + \ln(P(\boldsymbol{\lambda}_m = \hat{\lambda}|\boldsymbol{\mathcal{M}} = \mathcal{M}_m)) \tag{3.23}$$

est ignoré. Par conséquent, le critère de Schwarz ne tend pas vers le logarithme du facteur de Bayes en général. Cependant, l'erreur reste bornée, ce qui suffit à montrer la consistance du critère de Schwarz sous des conditions assez générales. Étant donné que

$$\frac{\mathrm{S}_m - \ln(\mathrm{B}_{m,m_0})}{\ln(\mathrm{B}_{m,m_0})} \xrightarrow[n\to+\infty]{} 0,$$

le critère de Schwarz S_m peut être vu comme une approximation grossière du logarithme du facteur de Bayes.

Cependant, dans l'approximation de la log-vraisemblance intégrée par le BIC, la précision peut être améliorée par un choix particulier de la loi a priori : le choix de la loi $\mathcal{N}(\lambda_0, \mathcal{I}_{\lambda_0}^{-1})$ assure la convergence vers zéro du terme (3.23), à une vitesse $\mathcal{O}(n^{-\frac{1}{2}})$. Comme l'approximation de $\ln(\mathcal{L}_{\boldsymbol{y},\mathcal{M}}(\tilde{\lambda}))$ par $\ln(\mathcal{L}_{\boldsymbol{y},\mathcal{M}}(\hat{\lambda}))$ est également d'ordre $\mathcal{O}(n^{-\frac{1}{2}})$, le BIC est une approximation $\mathcal{O}(n^{-\frac{1}{2}})$ de la log-vraisemblance intégrée. Le choix de la loi a priori $\mathcal{N}(\lambda_0, \mathcal{I}_{\lambda_0}^{-1})$ revient à faire l'hypothèse que la quantité d'information apportée par cette loi est égale à la quantité d'information pour une observation (voir Kass et Wasserman, 1995 [71], pour le détail sur une telle utilisation de BIC, dans le cas de tests d'hypothèses basés sur les facteurs de Bayes pour des modèles emboîtés).

Notons que lorsqu'une approximation de la log-vraisemblance intégrée plus précise que $\mathcal{O}(1)$ est souhaitée, se pose le problème du choix de la pénalité. Considérons en effet un échantillon de n vecteurs gaussiens de $\mathbb{R}^p$ dont les coordonnées sont indépendantes. Les coordonnées ont même loi qu'un échantillon gaussien de taille pn à valeurs dans $\mathbb{R}$. La pénalité intervenant dans BIC doit-elle être en $\ln(pn)$ ou en $\ln(n)$? En réalité, d'après Kass et Wasserman, 1995 [71], n représente, dans la définition de BIC, le taux de croissance de la hessienne de la log-vraisemblance. Dans le cas d'un échantillon, n devient le nombre de valeurs contribuant à la somme apparaissant dans la formule de la hessienne. Ainsi, dans le cas gaussien multivarié, la pénalité est $\ln(pn)$. De la même manière, dans le cas de R réplications indépendantes de chaînes de Markov cachées, on est amené à étudier précisément la contribution de chaque donnée à la croissance de la hessienne. D'après la proposition 5.3 de Mevel, 1997 [92], qui étudie le comportement asymptotique de la hessienne de la log-vraisemblance dans le cas d'une seule réalisation de chaîne de Markov cachée, la croissance est linéaire en n sous certaines hypothèses. Dans le cas de R réplications indépendantes, la pénalité à considérer est donc $\ln(Rn)$. Il serait intéressant d'étudier l'extension de ces résultats aux modèles de la famille $\mathcal{D}$.

Utilisation du critère BIC pour le choix du nombre d'états cachés

La méthode de Laplace permettant l'approximation par BIC de la vraisemblance intégrée est valide sous des conditions données dans Kass, Tierney et Kadane, 1990 [70] et appelées par les auteurs *Laplace-régularité* du modèle. Ces conditions portent sur l'identifiabilité du modèle, la consistance de l'EMV et aussi sur des conditions de régularité de la vraisemblance qui concernent sa dérivée sixième, ce qui est particulièrement pénible à vérifier pour des modèles de chaînes de Markov cachées (la thèse de Mevel porte sur l'étude de la dérivée seconde et les calculs sont déjà ardus). La présence de dépendances et d'états cachés complique donc la vérification de la Laplace-régularité des modèles de Markov cachés.

D'autre part, le critère BIC est couramment utilisé pour la sélection de modèles de Markov cachés (voir Geiger *et al.*, 2001 [58]). Cependant, dans le contexte du choix du nombre d'états cachés, la justification du BIC donnée ci-dessus se heurte une fois de plus au problème de non normalité asymptotique de l'EMV sous la vraie loi de $\boldsymbol{Y}$ (voir sections 3.3 et 3.4.1). Toutefois, il est parfois possible de montrer que le critère BIC est consistant sans avoir à vérifier la validité de l'approximation de Laplace, mais en adoptant un autre point de vue (voir section 3.6).

3.6 Critères de vraisemblance marginale pénalisée

Nous avons vu dans la section 3.4.1 qu'un estimateur empirique naturel de la divergence de Kullback-Leibler de P_λ à P_0 est la log-vraisemblance marginale (3.3), notée $\ln(\tilde{\mathcal{L}}_{\boldsymbol{y}}(\lambda))$. De plus, la valeur de λ maximisant $\ln(\tilde{\mathcal{L}}_{\boldsymbol{y}}(\lambda))$ est un estimateur naturel du paramètre λ_0 réalisant le minimum de $\mathrm{KL}(P_0, P_\lambda)$ sur $\mathcal{M}(\mathcal{G}, \mathcal{S}, \mathcal{P})$ (voir également Akaike, 1973 [3]). Ainsi, certains critères se basent sur le maximum de vraisemblance marginale au lieu du maximum de vraisemblance pour la sélection de modèles.

C'est le cas des critères suivants, utilisés par Leroux et Puterman, 1992 [82] puis par Gassiat, 2002 [52], dans le contexte de l'estimation de l'ordre de chaînes de Markov cachées :

$$\max_{\lambda \in \Lambda(K)} \ln(\tilde{\mathcal{L}}_{\boldsymbol{y}}(\lambda)) - a_n(K) \tag{3.24}$$

où K désigne le nombre d'états cachés du modèle, $\Lambda(K)$ désigne l'espace des paramètres pour un modèle $\mathcal{M}(K)$ à K états et $a_n(K)$ est une pénalité telle que

(i) la fonction $K \to a_n(K)$ est croissante ;

(ii) si $K_1 > K_2$, alors $a_n(K_1) - a_n(K_2) \xrightarrow[n\to+\infty]{} +\infty$;

(iii) pour toute valeur de (K_1, K_2), $\dfrac{a_n(K_1) - a_n(K_2)}{n} \xrightarrow[n\to+\infty]{} 0$.

Notons que la pénalité utilisée par le critère BIC : $a_n(K) = \dfrac{d_K}{2}\ln(n)$, où d_K représente la dimension de l'espace des paramètres d'un modèle à K états cachés, vérifie les hypothèses ci-dessus. Notons également que pour cette pénalité, le critère BIC coïncide avec le critère (3.24) pour les modèles de mélanges indépendants. Cependant, dans les cas non indépendants et en particulier pour les chaînes de Markov cachées, il n'y a pas coïncidence entre BIC et le critère (3.24) dans la mesure où le maximum de log-vraisemblance marginale diffère du maximum de log-vraisemblance. En revanche, la pénalité utilisée par le critère AIC ($\tilde{a}_n(K) = d_K$) ne vérifie pas la condition (ii). Par la suite, nous utiliserons la pénalité $a_n(K) = \frac{d_K}{2}\ln(n)$, en particulier dans les expérimentations de la section 3.8.

La consistance des critères (3.24) est prouvée dans Gassiat, 2002 [52]. Autrement dit, il est montré que si la vraie loi de $\boldsymbol{Y}$ est celle d'une chaîne de Markov cachée admettant comme nombre d'états cachés la valeur K_0 (K_0 étant supposée minimale ; en effet il existe également, dans ce cas, des lois identiques avec K états cachés, $K > K_0$), la probabilité que le critère (3.24) choisisse la valeur K_0 tend vers un quand n tend vers l'infini. A contrario, la consistance des critères basés sur les tests successifs, le critère AIC, la validation croisée et BIC n'est pas établie. Leroux et Puterman, 1992 [81] ont montré que l'utilisation d'un critère de log-vraisemblance marginale pénalisée conduit au choix d'un modèle qui, asymptotiquement, a au moins K_0 états cachés. La preuve, donnée par Gassiat, que ce critère ne surestime pas non plus asymptotiquement K_0, repose sur les hypothèses de stationnarité et de β-mélangeance de la chaîne cachée $\boldsymbol{S}$, et sur les hypothèses suivantes :

1. $\int_0^1 \sqrt{H_{[],\beta}(u)}\,du$ est finie, où β représente la fonction de taux de mélangeance de la chaîne cachée et $H_{[]\beta}(u)$ l'entropie à crochets de la classe
$$\mathcal{F} = \left\{ \frac{\frac{P_\lambda}{P_{\lambda_0}} - 1}{\|\frac{P_\lambda}{P_{\lambda_0}} - 1\|} | P_\lambda \in \bigcup_K \mathcal{M}_K \right\},$$
pour des crochets de largeur u ;
2. la paramétrisation de P_θ est continue, pour tout y, par rapport au paramètre θ, et il existe une fonction intégrable (indépendante du paramètre λ) dominant la log-densité marginale de Y pour tout P_λ dans $\bigcup_K \mathcal{M}_K$.

La condition de stationnarité portant sur $\boldsymbol{S}$ fait que la log-vraisemblance marginale s'écrit en fonction des proportions π et des paramètres $(\theta_j)_j$ seulement, et non pas en fonction de la matrice de transition A. Les conditions de stationnarité et de β-mélangeance de $\boldsymbol{S}$ sont suffisantes pour que les mêmes propriétés soient satisfaites par le processus observé $\boldsymbol{Y}$. La condition 1 est liée à l'utilisation d'un résultat de Doukhan, Massart et Rio, 1995 [42] énonçant un théorème de la limite centrale uniforme pour le processus empirique et assure que le maximum des rapports de log-vraisemblance marginale est borné en probabilité. La condition 1, dans le théorème de Doukhan, Massart et Rio, assure que la classe $\mathcal{F}$ est suffisamment restreinte pour pouvoir être de Donsker et vérifier uniformément un théorème de la limite centrale. Les conditions 1 et 2 entraînent le fait que l'ensemble des rapports de vraisemblance vérifie uniformément une loi des grands nombres. Notons que la condition 2 exclut par exemple les mélanges gaussiens à variances inconnues, non minorées.

L'hypothèse de β-mélangeance correspond à une faible dépendance entre les variables aléatoires cachées. Le cas d'indépendance vérifie cette condition : étant donné que la vraisemblance et la vraisemblance marginale coïncident alors, on en déduit que le critère BIC est consistant pour les modèles de mélanges indépendants. L'extension de la preuve de Gassiat à des modèles de Markov cachés où le processus caché est stationnaire et β-mélangeant (comme des arbres de Markov cachés par exemple) semble directe, sous réserve d'adapter la notion de (β-) mélangeance à des processus non temporels ; en réalité l'hypothèse de Markov n'intervient pas explicitement dans la preuve. Enfin, l'approche ci-dessus, spécialisée dans la détermination du nombre d'états cachés, repose fortement sur l'hypothèse que $\boldsymbol{Y}$ suit effectivement une loi de chaîne de Markov cachée ; sa modification pour déterminer la valeur de K minimisant la divergence de Kullback-Leibler des modèles de Markov cachés à la vraie loi de $\boldsymbol{Y}$, supposée quelconque, est difficile.

3.7 Une approche issue de la classification : ICL

Le critère ICL (*Integrated Classification Likelihood*) a été introduit par Biernacki, Celeux et Govaert, 2001 [11], dans le cadre de la sélection de modèles pour la classification automatique dans le cas d'indépendance. Il s'agit de déterminer le modèle qui sépare au mieux les classes définies par les états cachés. Contrairement aux critères précédents, il ne s'agit pas de mesurer l'adéquation entre le modèle et les données dans une perspective d'estimation de densité. Le critère retenu pour mesurer l'aptitude d'un modèle à partitionner les données observées est la vraisemblance complétée (appelée aussi vraisemblance classifiante) $\mathcal{L}_{\boldsymbol{y},\boldsymbol{s}}(\lambda) = P_\lambda(\boldsymbol{Y} = \boldsymbol{y}, \boldsymbol{S} = \boldsymbol{s})$. La log-vraisemblance classifiante est liée à la log-vraisemblance $\ln(\mathcal{L}_{\boldsymbol{y}}(\lambda))$ par la relation

$$\ln(P_\lambda(\boldsymbol{Y} = \boldsymbol{y}, \boldsymbol{S} = \boldsymbol{s})) = \ln(P_\lambda(\boldsymbol{S} = \boldsymbol{s}|\boldsymbol{Y} = \boldsymbol{y})) + \ln(P_\lambda(\boldsymbol{Y} = \boldsymbol{y})),$$

soit

$$\ln(\mathcal{L}_{\boldsymbol{y},\boldsymbol{s}}(\lambda)) = \ln(\mathcal{L}_{\boldsymbol{y}}(\lambda)) - \mathrm{EC}_{\boldsymbol{y},\boldsymbol{s}}(\lambda), \tag{3.25}$$

où $\mathrm{EC}_{\boldsymbol{y},\boldsymbol{s}}(\lambda) = -\ln(P_\lambda(\boldsymbol{S} = \boldsymbol{s}|\boldsymbol{Y} = \boldsymbol{y}))$ est la réalisation d'une variable aléatoire $\mathrm{EC}_{\boldsymbol{y},\boldsymbol{S}}(\lambda)$ d'espérance conditionnelle égale à l'entropie

$$\mathrm{E}_{\boldsymbol{y},\boldsymbol{s}}(\lambda) = \mathbb{E}_\lambda[-\ln(P_\lambda(\boldsymbol{S}|\boldsymbol{Y} = \boldsymbol{y}))|\boldsymbol{Y} = \boldsymbol{y}]$$

du tenseur de classification floue $(P_\lambda(\boldsymbol{S} = \boldsymbol{s}|\boldsymbol{Y} = \boldsymbol{y}))_{\boldsymbol{s}\in\mathcal{S}}$. L'entropie mesure la capacité du modèle P_λ à établir une partition pertinente des données observées $\boldsymbol{y}$. Si les données se partitionnent bien (ce qui correspond, dans le cas de mélanges, à des composants bien séparés), les probabilités $P_\lambda(\boldsymbol{S} = \boldsymbol{s}|\boldsymbol{Y} = \boldsymbol{y})$ sont proches de zéro ou proches de un et $\mathrm{E}_{\boldsymbol{y},\boldsymbol{s}}(\lambda) \approx 0$. Dans le cas contraire, la valeur de $\mathrm{E}_{\boldsymbol{y},\boldsymbol{s}}(\lambda)$ est élevée. Par conséquent, d'après l'équation (3.25), la log-vraisemblance classifiante peut être vue comme un critère de pénalisation de la log-vraisemblance par l'entropie, favorisant ainsi les modèles de Markov cachés induisant une partition des données observées la plus pertinente.

Le critère ICL a été développé dans un cadre bayésien, mais nous verrons qu'il peut être utilisé également dans un cadre non bayésien, de façon analogue au critère BIC. La sélection de modèles repose alors sur le pendant classifiant de la vraisemblance intégrée (3.20), nommé *vraisemblance classifiante intégrée* et défini pour un modèle $\mathcal{M}$ par

$$P(\boldsymbol{Y}=\boldsymbol{y},\boldsymbol{S}=\boldsymbol{s}|\boldsymbol{\mathcal{M}}=\mathcal{M}) = \int\limits_{\Lambda} P(\boldsymbol{Y}=\boldsymbol{y},\boldsymbol{S}=\boldsymbol{s},\boldsymbol{\lambda}=\lambda|\boldsymbol{\mathcal{M}}=\mathcal{M})\,d\lambda$$
$$= \int\limits_{\Lambda(\mathcal{M})} P(\boldsymbol{Y}=\boldsymbol{y},\boldsymbol{S}=\boldsymbol{s}|\boldsymbol{\lambda}=\lambda,\boldsymbol{\mathcal{M}}=\mathcal{M})P(\boldsymbol{\lambda}=\lambda|\boldsymbol{\mathcal{M}}=\mathcal{M})\,d\lambda. \tag{3.26}$$

Il est encore une fois tentant d'utiliser la méthode de Laplace pour approximer l'intégrale (3.26) de manière asymptotique, mais on se heurte une fois de plus à la non normalité asymptotique de l'estimateur de certains paramètres, qui peuvent être à la frontière de $\Lambda(\mathcal{M})$. Dans le cas de la vraisemblance classifiante intégrée, cependant, on peut tirer parti du fait que les états cachés sont fixés pour conditionner par rapport à $\boldsymbol{s}$ et pour séparer les contributions dues respectivement à $\boldsymbol{s}$ et à $\boldsymbol{y}$. Cela est possible si les variables aléatoires observées sont conditionnellement indépendantes sachant les variables aléatoires cachées et si l'on choisit une loi a priori telle que les paramètres $\boldsymbol{\theta}=(\boldsymbol{\theta}_j)_j$ – concernant la loi de $\boldsymbol{Y}$ sachant $\boldsymbol{S}=\boldsymbol{s}$ – soient indépendants des paramètres $\boldsymbol{\pi}$ et $\boldsymbol{p}$ – concernant la loi de $\boldsymbol{S}$.

D'une part, on a

$$P(\boldsymbol{Y}=\boldsymbol{y},\boldsymbol{S}=\boldsymbol{s}|\boldsymbol{\mathcal{M}}=\mathcal{M}) = P(\boldsymbol{Y}=\boldsymbol{y}|\boldsymbol{S}=\boldsymbol{s},\boldsymbol{\mathcal{M}}=\mathcal{M})P(\boldsymbol{S}=\boldsymbol{s}|\boldsymbol{\mathcal{M}}=\mathcal{M})$$

et d'autre part, du fait que

$$P(\boldsymbol{\theta}=\theta,\boldsymbol{p}=p,\boldsymbol{\pi}=\pi|\boldsymbol{\mathcal{M}}=\mathcal{M}) = P(\boldsymbol{\theta}=\theta|\boldsymbol{\mathcal{M}}=\mathcal{M})P(\boldsymbol{p}=p,\boldsymbol{\pi}=\pi|\boldsymbol{\mathcal{M}}=\mathcal{M}),$$

on obtient

$$P(\boldsymbol{Y}=\boldsymbol{y}|\boldsymbol{S}=\boldsymbol{s},\boldsymbol{\mathcal{M}}=\mathcal{M})$$
$$= \int\limits_{\Theta(\mathcal{M})} P(\boldsymbol{Y}=\boldsymbol{y}|\boldsymbol{S}=\boldsymbol{s},\boldsymbol{\theta}=\theta,\boldsymbol{\mathcal{M}}=\mathcal{M})P(\boldsymbol{\theta}=\theta|\boldsymbol{\mathcal{M}}=\mathcal{M})\,d\theta.$$

La méthode de Laplace est alors applicable à l'intégrale ci-dessus, ce qui donne

$$\ln(P(\boldsymbol{Y}=\boldsymbol{y}|\boldsymbol{S}=\boldsymbol{s},\boldsymbol{\mathcal{M}}=\mathcal{M})) \approx \max_{\theta} \ln(P(\boldsymbol{Y}=\boldsymbol{y}|\boldsymbol{S}=\boldsymbol{s},\boldsymbol{\theta}=\theta,\boldsymbol{\mathcal{M}}=\mathcal{M})) - \frac{t_{\mathcal{M}}}{2}\ln(n)$$

où $t_{\mathcal{M}}$ est le nombre de coordonnées indépendantes du paramètre θ. La probabilité $P(\boldsymbol{S}=\boldsymbol{s}|\boldsymbol{\mathcal{M}}=\mathcal{M})$ est directement obtenue en calculant l'intégrale

$$\int\limits_{P(\mathcal{M})\times\Pi(\mathcal{M})} P(\boldsymbol{S}=\boldsymbol{s}|\boldsymbol{p}=p,\boldsymbol{\pi}=\pi,\boldsymbol{\mathcal{M}}=\mathcal{M})P(\boldsymbol{p}=p,\boldsymbol{\pi}=\pi|\boldsymbol{\mathcal{M}}=\mathcal{M})\,dp\,d\pi\,,$$

sous l'hypothèse que les lignes de $\boldsymbol{p}$ et de $\boldsymbol{\pi}$ sont indépendantes a priori et de loi de Dirichlet $\mathcal{D}(\frac{1}{2},\ldots,\frac{1}{2})$ (distribution non informative). On obtient alors une expression de $\ln(P(\boldsymbol{S}=\boldsymbol{s}|\boldsymbol{\mathcal{M}}=\mathcal{M}))$ faisant intervenir la fonction Γ et les comptages

$$n_j = \mathrm{card}\{u \in \mathcal{U}|s_u = j\}.$$

Étant donné que le processus $\boldsymbol{S}$ est inobservé, les états cachés $\boldsymbol{s}$ doivent être estimés. En pratique, on utilise l'estimateur du MAP $\hat{\boldsymbol{s}}$. On en déduit également un estimateur empirique $\hat{n}_j$ des n_j. Dans le cas où ceux-ci sont suffisamment grands, on peut utiliser la formule de Stirling pour approximer la fonction Γ et ignorer les termes constants de manière à obtenir une approximation $\mathcal{O}(1)$, de même que pour BIC. On obtient alors le critère

$$\mathrm{ICL}(\mathcal{M}) = \ln(\mathcal{L}_{\boldsymbol{y},\hat{\boldsymbol{s}},\mathcal{M}}(\hat{\lambda})) - \frac{d_{\mathcal{M}}}{2}\ln(n)$$

approximant la log-vraisemblance classifiante intégrée ($d_{\mathcal{M}}$ représente la dimension de $\Lambda(\mathcal{M})$). Notons que ce critère n'est en général pas consistant, mais il n'est pas construit dans l'optique de déterminer un "vrai" nombre de composants. Son utilité est de déterminer le modèle qui sépare au mieux les classes correspondant aux états cachés. De même que le critère BIC, il est également utilisé dans un cadre non bayésien.

3.8 Expérimentations : sélection de chaînes de Markov cachées

Nous avons présenté ci-dessus les fondements théoriques de plusieurs critères de sélection de modèles applicables aux modèles de Markov cachés. Certains d'entre eux sont couramment utilisés pour les problèmes de sélection de tels modèles, comme AIC, BIC ou les tests d'hypothèses. D'autres critères, assez souvent utilisés dans d'autres contextes, n'ont pas encore été appliqués aux modèles de Markov cachés. C'est le cas de la validation croisée, dont l'implémentation est délicate dans le cas de variables observées dépendantes. Les méthodes développées dans la section 3.4.3 permettent à présent sa mise en œuvre. Enfin, les critères de log-vraisemblance marginale pénalisée et le critère ICL sont des critères relativement originaux, spécifiques aux modèles de mélanges, qui ne semblent pas avoir fait l'objet d'études expérimentales dans le cadre des modèles de Markov cachés. Dans cette section, nous étudions le comportement de ces différents critères (mis à part les tests) pour la sélection de chaînes de Markov cachées. Nos conclusions sont que le critère BIC et le demi-échantillonnage sélectionnent des modèles pertinents et parcimonieux aussi bien pour des données réelles que simulées.

Dans la section 3.8.1, nous considérons des données simulées suivant une chaîne de Markov cachée. Le but est d'estimer l'ordre (*i.e.* le nombre d'états cachés) à partir d'une réalisation de ce modèle. Nous étudions le comportement des critères en fonction de la méthode d'estimation des paramètres, de la séparation des classes, de la mélangeance de la chaîne et du nombre de données observées. Il en ressort que ces critères sont peu sensibles au degré de mélange et à la mélangeance de la chaîne cachée, mais fortement dépendants de la condition d'arrêt de l'algorithme EM et de sa méthode d'initialisation. La plupart des critères ont des performances d'autant meilleures que le nombre de données disponibles est élevé.

Nous présentons en section 3.8.2 une application complète des chaînes de Markov cachées à la modélisation du processus des corrections et défaillances de logiciels. Nous insistons tout particulièrement sur l'application des méthodes de sélection abordées dans ce chapitre au choix de l'ordre de la chaîne, mais également au choix du type de matrice de transition (matrice bidiagonale, tridiagonale ou non contrainte), qui modélise la manière dont évolue le taux de défaillance au cours du temps. Cette étude met en évidence la sélection, par la quasi-totalité des critères, de modèles remarquablement parcimonieux.

3.8.1 Données simulées : choix de l'ordre d'une chaîne de Markov cachée

Nous proposons une étude comparative des critères de validation croisée, AIC, BIC, le critère de log-vraisemblance marginale pénalisée et ICL pour le choix du nombre d'états cachés (également appelé l'*ordre*) d'une chaîne de Markov cachée, à partir de ses réalisations. Nous étudions l'influence, sur ces méthodes de choix de modèles, de la séparation des classes, de la mélangeance de la chaîne cachée (dans le cas de la validation croisée) et du nombre de données disponibles.

Jeux de données simulés

Les données simulées sont dans $I\!R^2$. Elles sont tirées suivant la loi d'un modèle de chaîne de Markov cachée à $K = 3$ états, stationnaire, de matrice de transition

$$A = \begin{bmatrix} 0,7 & 0,15 & 0,15 \\ 0,1 & 0,7 & 0,2 \\ 0,1 & 0,1 & 0,8 \end{bmatrix}.$$

La chaîne de Markov correspondante admet pour unique loi stationnaire la loi

$$\pi = \begin{bmatrix} 0,2500 & 0,2812 & 0,4688 \end{bmatrix}.$$

Nous choisissons donc cette loi stationnaire comme loi de l'état initial π de la chaîne cachée. Les lois d'émission sont des lois normales, de matrice de variance-covariance Σ_j égales à la matrice identité, pour chacun des 3 états cachés. Nous envisageons trois situations pour les moyennes (μ_1, μ_2, μ_3) de ces lois d'émission, qui correspondent à des séparations décroissantes :

– degré de mélange faible (tableaux 3.1 et 3.4)

$$\mu_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \mu_2 = \begin{bmatrix} 2,7 \\ 2,7 \end{bmatrix} \quad \mu_3 = \begin{bmatrix} 2,7 \\ -2,7 \end{bmatrix} \quad ;$$

– degré de mélange moyen (tableaux 3.2 et 3.5)

$$\mu_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \mu_2 = \begin{bmatrix} 2,2 \\ 2,2 \end{bmatrix} \quad \mu_3 = \begin{bmatrix} 2,2 \\ -2,2 \end{bmatrix} \quad ;$$

– degré de mélange fort (tableaux 3.3 et 3.6)

$$\mu_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \mu_2 = \begin{bmatrix} 2,0 \\ 1,9 \end{bmatrix} \quad \mu_3 = \begin{bmatrix} 2,0 \\ -1,9 \end{bmatrix}.$$

Biernacki, 1997 [12] propose une fonction pour mesurer le degré de mélange ; nous conservons les valeurs de μ_j et de Σ_j proposées dans la partie de sa thèse dédiée à la sélection du nombre de classes pour des mélanges gaussiens indépendants à partir de données simulées. Nous conservons également, dans un premier temps, son choix de tirer 30 réalisations indépendantes de taille 200, ces 200 individus étant eux-mêmes indépendants dans la thèse de Biernacki mais dépendants dans notre contexte. La figure 3.5 illustre les différents degrés de mélange par le tracé dans le plan de données obtenues par simulation.

Les modèles considérés ont de 1 à 4 états cachés. Pour chaque valeur possible du nombre d'états cachés K, les paramètres du modèle sont estimés comme suit. Nous comparons, pour commencer, les critères de demi-échantillonnage au critère BIC, considéré comme le critère de référence. D'autre part, nous réglons les paramètres intervenant dans l'identification de ces modèles (initialisation et nombre d'itérations de l'algorithme EM) et nous étudions l'influence de la séparation et de la mélangeance sur ces critères. Puis nous fixons un jeu de 30 réalisations de chaînes de Markov cachées pour lequel nous comparons tous les critères de sélection de modèle.

Le paramètre est estimé à partir de la chaîne complète par maximum de vraisemblance, en utilisant l'algorithme EM *à la Gibbs*. Le paramètre de plus grande vraisemblance est utilisé comme paramètre initial pour une nouvelle exécution de l'algorithme EM classique, en utilisant un nombre maximal d'itérations fixé à $50K$, où K est le nombre d'états cachés du modèle considéré. En effet, l'identification de modèles complexes a tendance à nécessiter plus d'itérations que les modèles plus simples. Si la croissance relative de la log-vraisemblance passe en dessous d'un certain seuil, l'algorithme s'arrête également. Cette procédure est répétée trois fois, en partant de trois paramètres initiaux différents. Nous considérons que la valeur de sortie de plus grande vraisemblance est l'EMV.

Cette solution $\hat{\lambda}^{(i)} = (\hat{A}^{(i)}, \hat{\theta}_1^{(i)}, \ldots, \hat{\theta}_K^{(i)})$, qualifiée *d'initiale*, sert à initialiser l'algorithme EM pour l'estimation des paramètres dans le cas de données supprimées, ce qui est utilisé dans la mise en œuvre de la validation croisée. Cette méthode accélère la convergence de l'algorithme mais va également à l'encontre du principe de la validation croisée, qui stipule que les données de test ne sont absolument pas utilisées pour l'apprentissage (dans notre cas, elles le sont indirectement, à travers l'initialisation d'EM). Nous verrons qu'une initialisation aléatoire améliore les résultats.

La partition des observations {données d'apprentissage, données de test} est tirée aléatoirement et une seule fois, pour implémenter le demi-échantillonnage simple (tableaux 3.1 à 3.6). Pour la méthode des chaînes paire et impaire, l'algorithme EM est initialisé par $((\hat{A}^{(i)})^2, \hat{\theta}_1^{(i)}, \ldots, \hat{\theta}_K^{(i)})$, ce qui offre a priori le même avantage et le même inconvénient que ceux évoqués au paragraphe ci-dessus. Enfin, dans tout ce qui suit, le modèle gaussien des modèles candidats est celui qui a servi à générer les données, en l'occurrence le modèle $[\pi\lambda I]$ (modèle stationnaire, matrice de variance-covariance égale à l'identité à un coefficient multiplicatif inconnu près, commun à tous les états cachés).

Les résultats concernant la sélection de l'ordre du modèle par le critère de validation croisée utilisant les sous-chaînes paires et impaires (critère “pair impair” PI), par demi-échantillonnage simple (critère DES) et par le critère BIC, sont reportés dans les tableaux 3.1 à 3.3 pour les chaînes de taille 200. Chaque tableau donne le pourcentage de séquences affectées par les trois critères aux quatre valeurs possibles pour l'ordre du modèle, et correspond à un degré de séparation différent. Le symbole '-' correspond à un pourcentage nul.

Méthode	Ordre			
	1	2	3	4
PI	-	-	**87** %	13 %
DES	-	-	**77** %	23 %
BIC	-	-	**93** %	7 %

TAB. 3.1 – *200 données, mélange bien séparé.*

Méthode	Ordre			
	1	2	3	4
PI	-	-	**90** %	10 %
DES	-	3 %	**53** %	43 %
BIC	-	-	**87** %	13 %

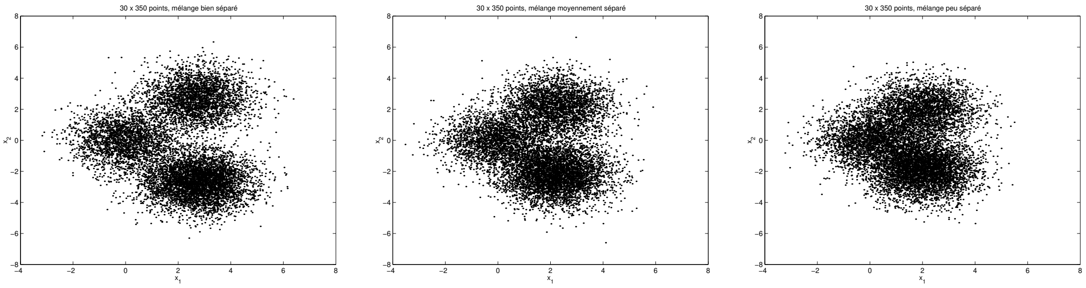
TAB. 3.2 – *200 données, mélange moyennement séparé.*

Méthode	Ordre			
	1	2	3	4
PI	-	3 %	**90** %	7 %
DES	-	10 %	**60** %	30 %
BIC	-	3 %	**93** %	3 %

TAB. 3.3 – *200 données, mélange peu séparé.*

De manière générale, le critère "pair impair" comme le critère BIC donnent tous deux de bons résultats, malgré une légère tendance à surestimer le nombre d'états cachés. Ceci est surprenant, vu les propriétés théoriques de ces critères, mais nous verrons ultérieurement que la cause en est une initialisation de EM non appropriée. Dans les deux cas, le pourcentage de bonnes réponses est peu sensible à la séparation du mélange En revanche, le demi-échantillonnage simple donne des résultats médiocres : cette méthode a une tendance nette à surestimer le nombre d'états cachés. Dans le cas du degré de mélange le plus fort (mélange peu séparé), les trois critères ont une très légère tendance à sous-estimer le nombre de composants. Dans ce cas, des modèles à deux composants peuvent être acceptables, vu que la longueur des séquences reste modérée.

Dans les expérimentations ci-dessus, nous nous sommes basés sur la thèse de Biernacki, 1997 [12] pour choisir le modèle selon lequel les séquences sont simulées (appelé *modèle générateur*) et le nombre de données. Étant donné que, du fait des dépendances entre états cachés, nos modèles sont plus complexes que ceux étudiés par Biernacki, nous augmentons le nombre de données disponibles pour la sélection de modèles afin d'obtenir des conditions comparables à celles de sa thèse. Ainsi, pour conserver le même rapport nombre de données/nombre de paramètres que dans le cas d'indépendance, nous tirons 30 réalisations indépendantes de chaînes de Markov cachées de longueur 350, avec les trois degrés de séparation précédents (tableaux 3.4, 3.5 et 3.6). La figure 3.5 illustre les différents degrés de mélange en représentant les 30 séquences bidimensionnelles de longueur 350. L'aspect temporel n'est pas représenté dans la figure : l'axe des abscisses représente la première dimension et celui des ordonnées la seconde dimension.



Composants bien séparés — *Composants moyennement séparés* — *Composants peu séparés*

FIG. 3.5 – *Les différents degrés de mélange dans les données simulées avec trois états cachés. La première coordonnée de chaque point est notée x_1 (en abscisse) et la seconde x_2 (en ordonnée).*

D'autre part, nous avons constaté que le critère de demi-échantillonnage simple (DES) a tendance à surestimer le nombre d'états cachés du modèle, c'est pourquoi nous considérons une borne maximale plus élevée pour le nombre d'états cachés des modèles en compétition, à savoir sept états cachés. Enfin, nous avons évoqué en section 3.4.2 la variabilité des critères de validation croisée multiple. Pour évaluer l'influence de cette variabilité sur la sélection de modèles afin de la réduire, nous considérons également le critère de demi-échantillonnage répété (ou demi-échantillonnage Monte-Carlo). Il s'agit donc d'un critère de type MCMCV* où la séquence de départ est partitionnée en deux classes de même effectif : l'ensemble d'apprentissage et l'ensemble de test ont même cardinal. On tire 15 fois cette partition.

Les résultats concernant la sélection de l'ordre du modèle par le critère "pair impair" PI, par le demi-échantillonnage simple DES, par le demi-échantillonnage Monte-Carlo (MCDE) et par le critère BIC, sont reportés dans les tableaux 3.1 à 3.3, selon le degré de mélange.

Les quatre critères considérés restent peu sensibles au degré de mélange. Le critère "pair impair' PI et le critère BIC donnent de bons résultats, malgré une légère tendance, une fois de plus, à surestimer le nombre d'états cachés. En revanche, le demi-échantillonnage répété donne des résultats moins bons : il possède une tendance plus forte que les deux critères précédents à sélectionner des modèles trop complexes (quatre états cachés). Le critère de demi-échantillonnage simple surestime largement le nombre d'états cachés, encore plus qu'avec des séquences de longueur 200 et d'autant plus également que le degré de mélange est élevé. La forte variabilité de ce critère est mise en évidence par les meilleures performances de sa version Monte-Carlo. Enfin, l'augmentation de la longueur des séquences améliore peu le comportement du critère BIC et améliore légèrement celui du critère PI. Dans le cas du degré de mélange le plus fort, le phénomène de sous-estimation du nombre d'états cachés, présent pour des séquences de longueur 200, disparaît totalement avec des séquences de longueur 350.

Méthode	Ordre						
	1	2	3	4	5	6	7
PI	-	-	**97** %	-	3 %	-	-
DES	-	-	**47** %	27 %	13 %	13 %	-
MCDE	-	-	**83** %	13 %	4 %	-	-
BIC	-	-	**90** %	10 %	-	-	-

TAB. 3.4 – *350 données, mélange bien séparé.*

Méthode	Ordre						
	1	2	3	4	5	6	7
PI	-	-	**97** %	3 %	-	-	-
DES	-	-	32 %	**37** %	17 %	7 %	7 %
MCDE	-	-	**70** %	30 %	-	-	-
BIC	-	-	**83** %	17 %	-	-	-

TAB. 3.5 – *350 données, mélange moyennement séparé.*

Méthode	Ordre						
	1	2	3	4	5	6	7
PI	-	-	**90** %	10 %	-	-	-
DES	-	-	27 %	27 %	**30** %	10 %	6 %
MCDE	-	-	**73** %	27 %	-	-	-
BIC	-	-	**90** %	10 %	-	-	-

TAB. 3.6 – *350 données, mélange peu séparé.*

À titre de complément, nous représentons sur la figure 3.6 la moyenne, sur les 30 séquences simulées, des quatre critères considérés ci-dessus, en fonction du nombre d'états cachés. Le modèle ayant servi à générer ces séquences correspond à un degré de mélange fort (classes peu séparées, voir tableau 3.6). Les courbes associées à chaque séquence sont peu différentes des courbes moyennes. En général, la courbe du critère de demi-échantillonnage répété (MCDE) décroît faiblement à partir de trois états cachés. La courbe associée au demi-échantillonnage simple (DES) varie autour de la courbe MCDE et ces fluctuations, dues au tirage aléatoire de la partition, entraînent une surestimation du nombre d'états cachés. Quand on effectue la moyenne de ce critère sur un nombre croissant de retirages, la courbe tend (uniformément) vers la courbe limite, dont une approximation est représentée figure 3.6. Ceci explique que le comportement du critère MCDE soit meilleur mais pas idéal : il s'agit d'une moyenne du critère DES sur 15 retirages indépendants de la partition. Les critères BIC et "pair-impair" (PI), au contraire, présentent en général un maximum marqué et correctement localisé : ils tendent à sélectionner un nombre d'états correct.

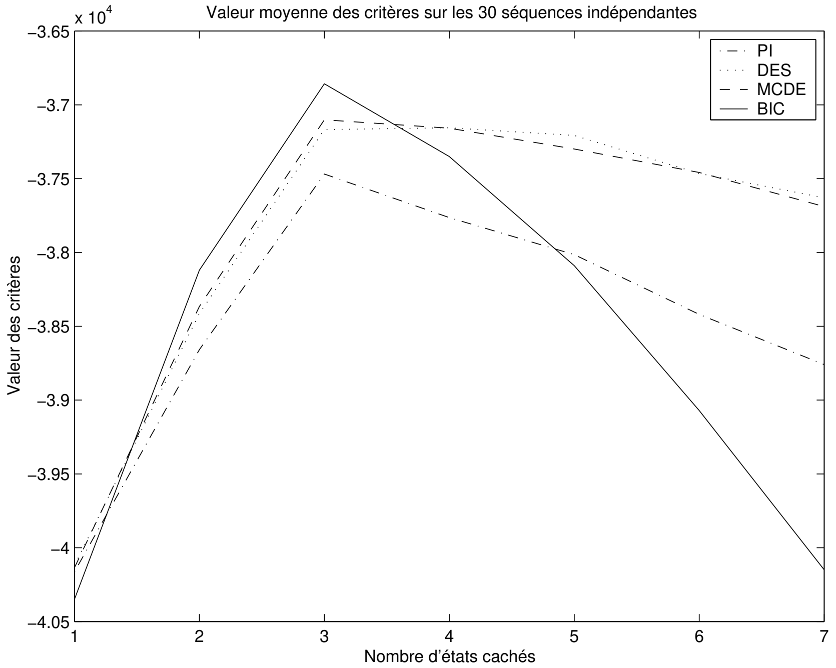


FIG. 3.6 – *Moyenne, sur les 30 séquences simulées, des quatre critères considérés ci-dessus, en fonction du nombre d'états cachés (en abscisse). Les séquences sont générées suivant le modèle de séparation la plus faible des composants.*

Influence de la mélangeance de la chaîne cachée

Dans le cas limite d'une chaîne cachée à deux états, périodique et de période 2, la chaîne des indices impairs (respectivement pairs) ne fait intervenir qu'un seul état. On s'attend donc à ce que les propriétés de mélangeance de la chaîne cachée influencent le comportement des techniques de demi-échantillonnage, tout particulièrement celle basée sur les chaînes paire et impaire. Les expérimentations ci-dessous ont pour objet d'étudier cette influence. Pour ce faire, nous simulons 30 réalisations indépendantes de longueur 350 de chaînes de Markov cachées en considérant successivement les matrices de transition ci-dessous, de plus en plus mélangeantes

$$A = A_1 = \begin{bmatrix} 0,7 & 0,15 & 0,15 \\ 0,1 & 0,7 & 0,2 \\ 0,1 & 0,1 & 0,8 \end{bmatrix} \quad A_2 = \begin{bmatrix} 0,5 & 0,4 & 0,1 \\ 0,2 & 0,4 & 0,4 \\ 0,2 & 0,3 & 0,5 \end{bmatrix} \quad A_3 = \begin{bmatrix} 0,2 & 0,4 & 0,4 \\ 0,6 & 0,1 & 0,3 \\ 0,3 & 0,4 & 0,3 \end{bmatrix}.$$

Le taux de mélangeance d'une chaîne de Markov est lié à la plus grande valeur propre ν_i de sa matrice de transition A_i, différente de un (rappelons que la valeur un est toujours valeur propre d'une matrice de transition et que c'est sa plus grande valeur propre). Plus cette valeur propre est proche de un, moins la chaîne mélange rapidement. Les valeurs $(\nu_i)_{i=1,2,3}$ sont $\nu_1 = 0,6$; $\nu_2 = 0,3$ et $\nu_3 = -0,1$. Pour chacune des matrices ci-dessus et pour chacun des trois triplets (μ_1, μ_2, μ_3) définissant la séparation du mélange, nous déterminons le pourcentage de sous-estimation, d'estimation correcte et de surestimation du nombre d'états cachés, en considérant le critère "pair impair" PI, le demi-échantillonnage simple DES, le demi-échantillonnage Monte-Carlo MCDE et le critère BIC. Le tableau 3.7 résume ces résultats. Le degré de mélange est désigné par la notation s_i où s_1 désigne un mélange bien séparé, s_2 un mélange moyennement séparé et s_3 un mélange peu séparé.

Matrice	A_1			A_2			A_3		
Séparation	s_1	s_2	s_3	s_1	s_2	s_3	s_1	s_2	s_3
K	= - / +	= - / +	= - / +	= - / +	= - / +	= - / +	= - / +	= - / +	= - / +
PI	**97** 0 / 3	**97** 0 / 3	**90** 0 / 10	**87** 0 / 13	**97** 0 / 3	**100** 0 / 0	**100** 0 / 0	**97** 0 / 3	**100** 0 / 0
DES	47 0 / **53**	33 0 / **66**	27 0 / **73**	**53** 0 / 47	**53** 0 / 47	40 0 / **60**	**53** 0 / 47	**60** 0 / 40	**50** 0 / **50**
MCDE	**83** 0 / 17	**70** 0 / 30	**73** 0 / 27	**87** 0 / 13	**83** 0 / 17	**87** 0 / 13	**77** 0 / 23	**87** 0 / 13	**87** 0 / 13
BIC	**90** 0 / 10	**83** 0 / 17	**90** 0 / 10	**100** 0 / 0	**100** 0 / 0	**94** 3 / 3	**93** 0 / 7	**97** 0 / 3	**97** 0 / 3

TAB. 3.7 – *Pourcentage d'estimation correcte (=), de sous-estimation (-) et de surestimation du nombre K d'états cachés suivant le degré de mélange s_i et la mélangeance de la chaîne cachée, pour le critère "pair impair" PI, le demi-échantillonnage simple DES, le demi-échantillonnage Monte-Carlo MCDE et le critère BIC. Une séparation s_1 désigne un mélange bien séparé, s_2 un mélange moyennement séparé et s_3 un mélange peu séparé.* La matrice de transition A_1 correspond à une chaîne peu mélangeante, A_2 à une chaîne moyennement mélangeante et A_3 à une chaîne très mélangeante. Les séquences simulées sont de longueur 350.

D'après le tableau 3.7, l'influence de la mélangeance de la chaîne cachée, suivant la séparation du mélange, sur le comportement des différents critères, n'est pas sensible. Les différences observées pourraient être dues au hasard résultant du tirage des 30 séquences. Cela expliquerait les variations des performances de BIC, qui n'a pas de raison d'être influencé par la mélangeance. D'autre part, le critère "pair-impair" sélectionne un nombre d'états correct d'autant plus fréquemment que la séparation du mélange est faible, ce qui va à l'encontre des résultats attendus et aussi des résultats obtenus avec d'autres critères par Biernacki, 1997 [12], pour les modèles de mélanges indépendants. Vu que la mélangeance de la chaîne cachée n'a pas d'influence sur les performances de la sélection par la méthode des chaînes paire et impaire, nous pressentons que les cas réellement pathologiques sont seulement les cas périodiques ou très proches de la périodicité (ce qui se traduit par exemple par des matrices de transitions proches de matrices de permutations) et où la période est paire.

Nos conclusions sur les différents critères de sélection restent les mêmes que celles de la partie précédente : les critères BIC et “pair-impair” ne sous-estiment pas le nombre K d'états cachés et le surestiment rarement, le critère de demi-échantillonnage répété surestime parfois le nombre d'états cachés (d'une unité le plus souvent, dans le cas présent), et le demi-échantillonnage simple surestime K fréquemment.

Influence de l'initialisation d'EM et de la condition d'arrêt

Dans les expérimentations ci-dessus, nous utilisons chaque séquence complète pour estimer les paramètres par maximum de vraisemblance (par l'algorithme EM). L'estimateur obtenu est alors réutilisé comme valeur initiale de l'algorithme EM pour estimer les paramètres associés à la séquence incomplète, lors du demi-échantillonnage. Cette méthode permet en pratique d'obtenir une convergence plus rapide qu'avec une initialisation aléatoire. Cependant, ce principe est en contradiction avec celui de la validation croisée, qui exige que les données ne soient pas utilisées à la fois pour l'identification du modèle et pour sa validation.

C'est pourquoi nous étudions une méthode d'initialisation alternative, où le paramètre initial est déterminé aléatoirement. On peut envisager de recommencer cette procédure en partant de plusieurs valeurs initiales aléatoires et de conserver le paramètre de plus grande vraisemblance ; cependant, dans l'expérimentation ci-dessous, nous considérons une unique valeur initiale du paramètre. Les données sont simulées à partir d'un modèle de chaîne de Markov cachée de matrice de transition A_1 (faible mélangeance), le mélange étant fortement séparé.

La partie supérieure du tableau 3.8 est un rappel des modèles sélectionnés par les critères quand l'algorithme EM est initialisé, pour le demi-échantillonnage, avec le paramètre estimé à partir de la séquence entière. Pour l'estimation de ce dernier, trois valeurs initiales aléatoires du paramètre sont utilisées et les 50 premières itérations sont de type EM *à la Gibbs*.

La partie inférieure du tableau 3.8 est obtenue par une initialisation aléatoire de l'algorithme EM pour le demi-échantillonnage et pour le critère BIC, avec une seule valeur aléatoire du paramètre et également 50 itérations d'EM *à la Gibbs*. Dans les deux cas, la condition d'arrêt est identique à celle des expérimentations précédentes.

La comparaison des parties supérieure et inférieure du tableau 3.8 montre que les résultats de ces méthodes se dégradent, globalement, quand l'algorithme EM est initialisé aléatoirement. Nous verrons ci-après, cependant, que ceci ne remet pas en cause l'initialisation aléatoire. La dégradation des performances des critères est due à ce que l'algorithme EM ne converge pas vers un maximum local de la vraisemblance, à cause d'un nombre d'itérations insuffisant. L'insuffisance du nombre d'itérations est compensée, dans les expérimentations associées à la partie supérieure du tableau, par une initialisation astucieuse qui permet une convergence plus rapide ; cependant, nous ne pouvons considérer cette méthode comme la meilleure, comme nous le verrons ci-dessous.

Notons également la forte dégradation des performances du BIC. Celle-ci est due

Initialisation	Critère	Ordre						
		1	2	3	4	5	6	7
séquence complète	PI	-	-	**97** %	-	3 %	-	-
	DES	-	-	**47** %	27 %	13 %	13 %	-
	MCDE	-	-	**83** %	13 %	4 %	-	-
	BIC	-	-	**90** %	10 %	-	-	-
aléatoire	PI	-	-	**83** %	17 %	-	-	-
	DES	-	-	**51** %	33 %	13 %	3 %	-
	MCDE	-	-	**50** %	43 %	7 %	-	-
	BIC	-	-	**63** %	30 %	7 %	-	-

TAB. 3.8 – *350 données par séquence, nombre d'itérations maximal proportionnel au nombre d'états, initialisation d'EM en utilisant la séquence complète ou aléatoire.*

uniquement au fait qu'une seule valeur initiale du paramètre est utilisée dans l'algorithme EM, au lieu de trois. On s'attendrait à ce que les itérations d'EM *à la Gibbs* qui précèdent l'algorithme EM rendent le paramètre final peu dépendant de la valeur initiale, mais les résultats ci-dessus montrent que l'initialisation de l'algorithme EM par trois valeurs initiales au lieu d'une est importante, pour l'estimation des paramètres avec la séquence complète.

D'autre part, les critères de demi-échantillonnage considérés se basent sur l'estimation par maximum de vraisemblance, réalisée par l'algorithme EM. Il est donc important que la condition d'arrêt de EM soit telle que le paramètre de sortie de l'algorithme soit de vraisemblance aussi grande que possible. Cela n'est pas le cas si l'algorithme s'arrête prématurément. Pour éviter ce problème, on peut envisager de poursuivre les itérations d'EM jusqu'à ce que la croissance relative de la log-vraisemblance passe en dessous d'un certain seuil (en l'occurrence 10^{-6} dans les expérimentations ci-dessus). L'inconvénient de baser la condition d'arrêt d'EM uniquement sur ce critère est que cela conduit souvent à réaliser un nombre d'itérations largement supérieur à mille, avec en sortie des paramètres peu différents, et une log-vraisemblance pratiquement égale, à ce qu'on peut obtenir avec uniquement mille itérations. De plus, ce critère d'arrêt basé uniquement sur la croissance de la log-vraisemblance conduit à un nombre d'itérations qui croît avec le nombre d'états cachés.

Le critère d'arrêt utilisé dans les expérimentations précédentes est celui basé sur la croissance relative de la log-vraisemblance, mais où un nombre maximal d'itérations est fixé, égal à 50 fois le nombre K d'états cachés. On s'aperçoit que cette limite n'est pas toujours suffisante pour que la log-vraisemblance se stabilise. Dans l'étude ci-dessus, nous augmentons la limite du nombre d'itérations à mille (indépendamment, donc, du nombre d'états cachés du modèle), ce qui est suffisant pour atteindre la stabilisation de la log-vraisemblance dans la plupart des cas. L'expérimentation est menée sur des données simulées à partir d'un modèle de chaîne de Markov cachée de matrice de transition A_1, à classes fortement séparées.

La fréquence de sélection de chaque nombre K d'états cachés par les critères est rappelée dans la partie supérieure du tableau 3.8, lorsque le nombre d'itérations de l'algorithme EM est limité à $50K$. Nous comparons cette méthode avec l'alternative consistant à fixer à mille le nombre maximal d'itérations. Dans les deux cas, la valeur du paramètre obtenue sert à calculer BIC et à initialiser EM lors du demi-échantillonnage.

Nombre d'itérations	Critère	Ordre K						
		1	2	3	4	5	6	7
limité à $50K$	PI	-	-	**97** %	-	3 %	-	-
	DES	-	-	**47** %	27 %	13 %	13 %	-
	MCDE	-	-	**83** %	13 %	4 %	-	-
	BIC	-	-	**90** %	10 %	-	-	-
limité à 1000	PI	-	-	**100** %	-	-	-	-
	DES	3 %	-	**48** %	13 %	13 %	13 %	10 %
	MCDE	-	-	**83** %	17 %	-	-	-
	BIC	-	-	**100** %	-	-	-	-

TAB. 3.9 – *350 données par séquence, initialisation d'EM en utilisant la séquence complète, nombre d'itérations maximal proportionnel au nombre K d'états cachés ou limité à 1000.*

La comparaison des parties supérieure et inférieure du tableau 3.9 montre que le comportement des critères est globalement amélioré par l'augmentation du nombre d'itérations d'EM, notamment pour le critère BIC. Par conséquent, dans les expérimentations précédentes, la convergence de l'algorithme EM n'était vraisemblablement pas atteinte lors de l'arrêt, ce qui conduisait à des valeurs du paramètre non optimales localement.

Enfin, nous concluons les expérimentations sur les paramètres de l'algorithme EM (initialisation et condition d'arrêt) et leur influence sur les critères de sélection en examinant l'effet combiné de ces deux facteurs. Ainsi, nous fixons le nombre maximal d'itérations de l'algorithme EM à mille et nous initialisation cet algorithme de manière aléatoire pour l'estimation dans les techniques de demi-échantillonnage. Cette fois, l'algorithme EM *à la Gibbs* n'est pas utilisé mais nous partons de trois valeurs aléatoires pour l'initialisation. Le pourcentage de sélection de chaque nombre de composants par le critère "pair impair" PI, par le demi-échantillonnage simple DES, par le demi-échantillonnage Monte-Carlo MCDE et par le critère BIC est alors donné par le tableau 3.10.

Méthode	Ordre						
	1	2	3	4	5	6	7
PI	-	-	**93** %	7 %	-	-	-
DES	-	-	**84** %	13 %	3 %	-	-
MCDE	-	-	**100** %	-	-	-	-
BIC	-	-	**100** %	-	-	-	-

TAB. 3.10 – *350 données par séquence, nombre d'itérations maximal fixé à 1000, initialisation aléatoire de l'algorithme EM.*

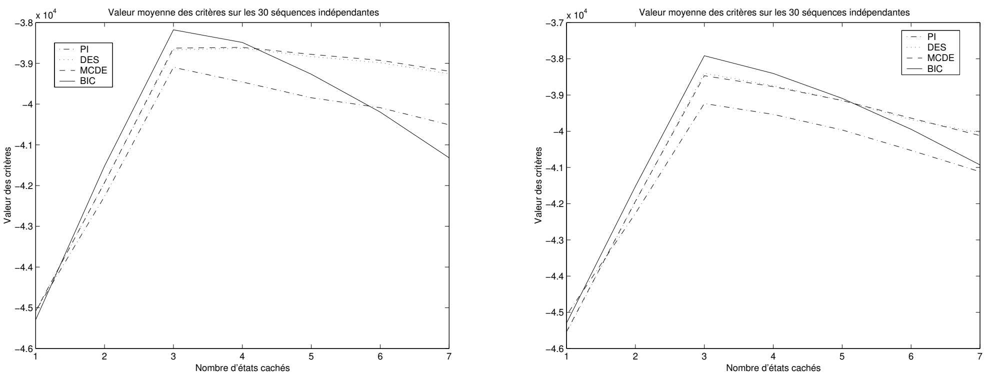
La comparaison du tableau 3.10 et des tableaux 3.8 et 3.9 montre que les meilleurs résultats, globalement, sont obtenus en limitant le nombre d'itérations d'EM à mille plutôt qu'à 50 fois le nombre d'états cachés, en utilisant trois valeurs de départ pour l'initialisation, ces valeurs de départ devant être déterminées sans utiliser la séquence complète dans le cas du demi-échantillonnage. Ces résultats sont conformes à ce qui était attendu : d'une part, le demi-échantillonnage repose sur le fait que la séquence de test ne doit pas être utilisée pour l'apprentissage, même indirectement. D'autre part, l'algorithme EM reste dépendant de la valeur initiale du paramètre, même si des itérations d'EM *à la Gibbs* sont utilisées ; enfin, le nombre d'itérations requis pour la convergence d'EM reste élevé en général. Les tentatives effectuées pour diminuer le nombre d'itérations d'EM se soldent par une dégradation du comportement des critères, aussi bien pour le demi-échantillonnage que pour le critère BIC.

Il est intéressant de comparer les courbes moyennes, sur les 30 séquences simulées, des quatre critères considérés ci-dessus, en fonction du nombre d'états cachés. En effet, pour un critère donné, les courbes associées à chaque séquence sont en général peu différentes de la courbe moyenne. On peut ainsi étudier le comportement moyen des critères suivant l'initialisation d'EM et suivant le nombre d'itérations maximal. Les courbes moyennes des critères, pour le jeu de données étudié ci-dessus, sont représentées figure 3.7. Dans le cas du demi-échantillonnage simple et Monte-Carlo (critères DES et MCDE), un nombre d'itérations insuffisant et une initialisation de l'algorithme EM utilisant la séquence complète provoque, en moyenne, un plateau de la courbe entre $K = 3$ et $K = 4$, ce qui explique les confusions plus nombreuses entre les deux modèles et une tendance à surestimer le nombre d'états cachés (courbes de gauche). Dans le cas d'un nombre d'itérations plus élevé et d'une initialisation aléatoire, le pic des courbes pour $K = 3$ est plus marqué en moyenne et le nombre d'erreurs dans le choix de K est moins élevé. C'est la raison pour laquelle, dans les expérimentations suivantes, nous utiliserons en principe un nombre d'itérations maximal égal à mille et une initialisation aléatoire de l'algorithme EM en partant de trois valeurs initiales, en particulier pour la validation croisée.

En définitive, lorsque l'algorithme EM est utilisé correctement, les critères "pair-impair", le demi-échantillonnage répété et BIC choisissent un nombre correct d'états cachés. Le demi-échantillonnage simple a une certaine tendance à surestimer le nombre d'états cachés.

Comparaison de tous les critères

Les expérimentations précédentes nous ont permis de comparer les critères de demi-échantillonnage (simple, répété et "pair-impair") et le critère BIC. Elles nous ont également permis de déterminer une méthode d'initialisation de l'algorithme EM et une condition d'arrêt donnant des résultats satisfaisants. Enfin, l'étude de l'influence de la séparation des classes du mélange et de la mélangeance de la chaîne cachée ont mis en évidence le fait que ces facteurs sont peu déterminants (du moins dans le contexte précédent, qui reste raisonnable en termes de séparation des classes et de mélangeance de la chaîne). Empiriquement, le jeu de données simulées engendrant le plus fort taux de surestimation du nombre d'états cachés pour la plupart des critères de sélection correspond à une séparation moyenne des classes et à une chaîne faiblement mélangeante



Initialisation avec la séquence complète. Nombre d'itérations maximal : 50 fois le nombre d'états cachés.

Initialisation aléatoire. Nombre d'itérations maximal : mille itérations

FIG. 3.7 – *Moyenne, sur les 30 séquences simulées, des quatre critères considérés ci-dessus, en fonction du nombre d'états cachés (en abscisse). Les séquences sont générées suivant le modèle de séparation la plus forte des composants.*

(voir tableau 3.7).

Nous utilisons ce jeu de données pour comparer les critères BIC, ICL, AIC, la log-vraisemblance marginale pénalisée VMP et différentes variantes de la validation croisée multiple : demi-échantillonnage simple, répété et "pair-impair", validation croisée multiple MCV* et apprentissage-test répété RLT. Rappelons que le critère MCV* est obtenu en tirant une partition aléatoire des séquences (en l'occurrence en dix classes). Chaque classe est utilisée successivement pour la validation, les neuf autres servant à l'apprentissage. Pour le critère RLT, dix partitions aléatoires sont utilisées et pour chaque d'elles, une seule classe est utilisée pour la validation et les neuf autres pour l'apprentissage. Pour le demi-échantillonnage Monte-Carlo, dix partitions en deux classes de même effectif sont utilisées. Les résultats sont présentés dans le tableau 3.11.

Méthode	Ordre						
	1	2	3	4	5	6	7
PI	-	-	**76** %	17 %	7 %	-	-
DES	-	-	**80** %	20 %	-	-	-
MCDE	-	-	**100** %	-	-	-	-
MCV*	-	-	27 %	**36** %	20 %	10 %	7 %
RLT	-	-	13 %	**38** %	23 %	13 %	13 %
BIC	-	-	**90** %	10 %	-	-	-
ICL	-	-	**90** %	10 %	-	-	-
AIC	-	-	**83** %	17 %	-	-	-
VMP	-	-	**90** %	10 %	-	-	-

TAB. 3.11 – *350 données, mélange moyennement séparé, chaîne cachée peu mélangeante.*

Le nombre d'états cachés sélectionné par les critères MCDE, BIC, VMP, ICL et AIC

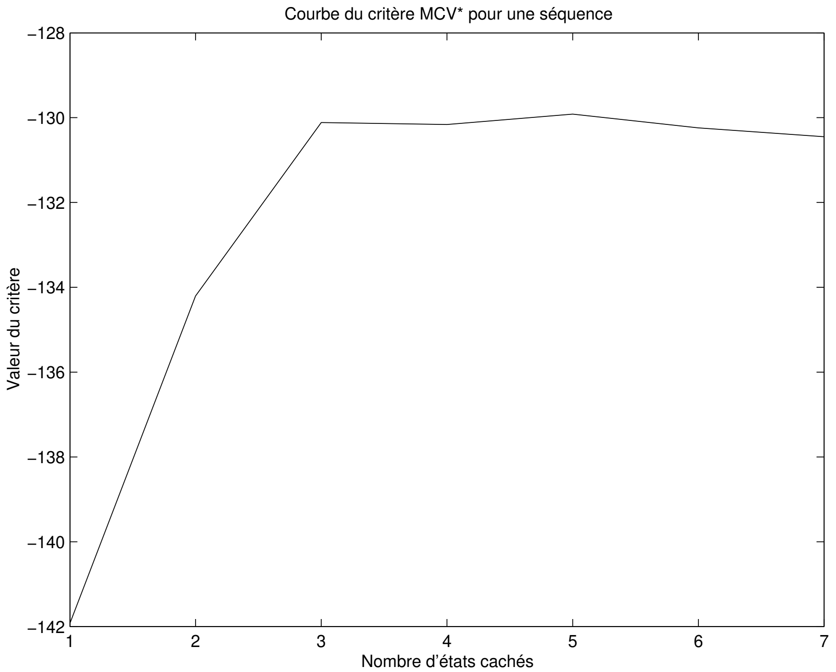


FIG. 3.8 – *Pour certains jeux de données, la courbe du critère MCV* possède un plateau. À un modèle à cinq états cachés (valeur où le critère est maximal), un modèle à trois états cachés (valeur la plus "à gauche" du plateau) est préférable.*

est très fréquemment correct et plus rarement surestimé d'une unité. La tendance un peu plus forte des critères PI et DES à surestimer le nombre K d'états cachés, qui devient très prononcée pour les critères MCV* et RLT, s'explique avant tout par un aspect en plateau de leurs courbes (voir figure 3.8). La sélection automatique considère uniquement le maximum strict du critère, alors que dans le cas de plateaux, des états plus parcimonieux doivent être choisis. Dans la partie concernant les données réelles, nous choisirons le modèle le plus parcimonieux tel que le critère est supérieur à sa valeur maximale moins son écart-type empirique. Dans l'expérimentation ci-dessus, une telle méthode ramène à un même niveau la fréquence de bonne estimation de K par les critères de demi-échantillonnage. De plus, elle réduit nettement la fréquence de surestimation par les critères MCV* et RLT.

Cas d'un modèle générateur trop complexe

Souvent, l'enjeu de la sélection de modèles est de trouver un modèle parcimonieux, compte tenu du nombre de données disponibles, plutôt que de rechercher la "vraie loi" (voir section 3.2.3). Son but ne se réduit pas à retrouver le modèle générateur de données simulées. Dans ce cas, en effet, il se peut que ce dernier soit trop complexe par rapport au nombre de données disponibles.

Ainsi, dans le cas de réalisations d'un modèle de chaîne de Markov cachée ayant cinq états cachés très peu séparés, un modèle à trois ou quatre états cachés pourra peut-être convenir si le nombre de données observées est modéré. C'est pourquoi nous proposons d'étudier le comportement des critères étudiés jusqu'ici, en considérant des réalisations de longueur 350 d'une chaîne de Markov cachées stationnaire bi-dimensionnelle à cinq états, ayant la matrice de transition suivante :

$$A = \begin{bmatrix} 0,7 & 0,1 & 0,1 & 0,05 & 0,05 \\ 0,05 & 0,8 & 0,05 & 0,07 & 0,03 \\ 0,03 & 0,15 & 0,75 & 0,03 & 0,04 \\ 0,02 & 0,01 & 0,05 & 0,85 & 0,07 \\ 0,04 & 0,02 & 0,02 & 0,02 & 0,9 \end{bmatrix}.$$

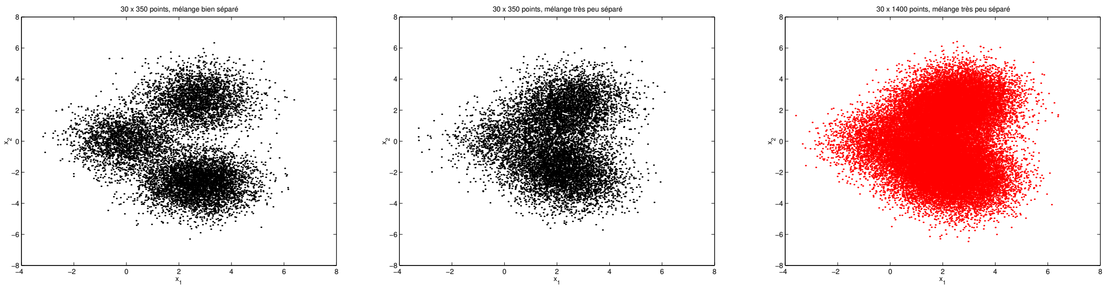
La chaîne de Markov correspondante admet pour unique loi stationnaire la loi

$$\pi = \begin{bmatrix} 0,1072 & 0,2104 & 0,1522 & 0,2074 & 0,3228 \end{bmatrix}.$$

On suppose que la loi de l'état initial de la chaîne cachée est cette loi stationnaire. Les matrices de variance-covariance des lois normales conditionnelles sont égales à l'identité pour chacun des 5 états. Les moyennes μ_k de l'état k valent

$$\mu_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \mu_2 = \begin{bmatrix} 2,7 \\ 2,7 \end{bmatrix} \quad \mu_3 = \begin{bmatrix} 2,7 \\ -2,7 \end{bmatrix} \quad \mu_4 = \begin{bmatrix} 2,0 \\ 1,5 \end{bmatrix} \quad \mu_5 = \begin{bmatrix} 2,0 \\ -1,5 \end{bmatrix}.$$

Remarquons que les moyennes μ_1, μ_2 et μ_3 sont identiques aux paramètres du modèle utilisé dans les expérimentations ci-dessus et correspondent à la séparation la plus faible. Les paramètres μ_4 et μ_5 appartiennent à l'enveloppe convexe de $\{\mu_1; \mu_2\ ; \mu_3\}$, ce qui fait que les composants du mélange sont très peu séparées. La figure 3.9 représente les 30 séquences bidimensionnelles de longueur 350 simulées suivant un modèle à trois états cachés (moyennes $(\mu_i)_{i=1,\ldots,3}$) puis suivant un modèle à cinq états cachés (moyennes $(\mu_i)_{i=1,\ldots,5}$), puis 30 séquences de longueur 1 400 simulées suivant le même modèle à cinq états cachés, l'aspect temporel n'étant pas représenté dans cette figure.



Chaînes de longueur 350, trois états cachés — *Chaînes de longueur 350, cinq états cachés* — *Chaînes de longueur 1400, cinq états cachés*

FIG. 3.9 – *Jeux de données simulés à partir d'un modèle à trois états bien séparés et d'un modèle à cinq états très peu séparés, avec 30 séquences de longueur 350 ou 1 400.*

Nous considérons 30 échantillons de longueur 350. Les modèles en compétition ont entre un et sept états cachés et le modèle gaussien est celui qui a servi à générer les données, en l'occurrence le modèle $[\pi\lambda I]$ (modèle stationnaire, matrice de variance-covariance égale à l'identité à un coefficient multiplicatif inconnu près, commun à tous les états cachés). L'estimation des paramètres par l'algorithme EM se fait de la manière ayant donné les meilleurs résultats dans l'étude précédente (trois valeurs aléatoires initiales du paramètre, mille itérations au maximum). Les résultats sont résumés dans le tableau 3.12, qui donne le pourcentage de sélection de chaque modèle pour les critères BIC, AIC, ICL, la log-vraisemblance marginale pénalisée VMP et différentes variantes de la validation croisée multiple : demi-échantillonnage répété MCDE et "pair-impair" PI, validation croisée multiple MCV* et apprentissage-test répété RLT. Comme dans l'expérimentation précédente, le nombre de répétitions est de dix pour la validation croisée et les partitions utilisées ont également dix classes.

Méthode	Ordre						
	1	2	3	4	5	6	7
PI	-	-	**63** %	**21** %	13 %	3 %	-
MCDE	-	-	**33** %	**33** %	31 %	3 %	-
MCV*	-	-	10 %	17 %	**39** %	17 %	17 %
LRT	-	-	-	21 %	**33** %	23 %	23 %
BIC	-	-	**90** %	10 %	-	-	-
ICL	-	13 %	**84** %	3 %	-	-	-
AIC	-	-	**37** %	33 %	20 %	7 %	3 %
VMP	-	30 %	**70** %	-	-	-	-

TAB. 3.12 – *Séquences de longueur 350 simulées suivant un modèle complexe.*

D'après le tableau 3.12, les critères de demi-échantillonnage "pair-impair" PI et le critère BIC sélectionnent fréquemment des modèles à trois ou quatre états cachés, ce qui nous semble être des valeurs raisonnables vu la faible séparation des composants du mélange et vu le nombre modéré d'observations. Les critères de demi-échantillonnage répété MCDE et AIC sélectionnent le plus fréquemment des modèles ayant de trois à cinq états cachés. Bien que modèle à cinq états soit le modèle générateur, on peut considérer que ces deux critères ont une certaine tendance à surestimer la complexité du modèle. Les critères de validation croisée multiple RLT et MCV* ont une tendance plus forte à choisir des modèles trop complexes, qui s'explique en grande partie par un aspect en plateau (voir figure 3.8) qui n'est pas pris en compte par notre protocole. Le critère ICL, au contraire, sélectionne des modèles séparant au mieux les classes. Vu que celles-ci sont très peu séparées, ce critère choisit des modèles plus simples que les critères précédents. Enfin, le critère de vraisemblance marginale pénalisée a tendance à choisir des modèles légèrement trop parcimonieux.

Comportement asymptotique

Nous avons évoqué ci-dessus que le but de la sélection de modèles ne se résume pas à retrouver le modèle ayant généré des données simulées. Cependant, une propriété souhaitable d'un critère de sélection de modèle est de sélectionner un modèle susceptible d'avoir généré les données simulées – le plus parcimonieux possible lorsque plusieurs lois sont identiques, par exemple dans le cas de modèles emboîtés – avec une probabilité qui tend vers un lorsque le nombre de données simulées tend vers $+\infty$ (consistance du critère). Rappelons qu'il n'y a que le critère de vraisemblance marginale pénalisée dont la consistance soit établie dans le contexte des chaînes de Markov cachées (voir section 3.6). Nous vérifions empiriquement si c'est le cas pour les critères de sélection étudiés, en reprenant l'expérimentation précédente (dont les résultats sont résumés tableau 3.12) mais en considérant cette fois des séquences de longueur 1 400. Les autres conditions sont identiques à celles de l'expérimentation précédente, à ceci près que le nombre d'itérations maximal pour l'algorithme EM est porté à 2000. Les résultats sont présentés dans le tableau 3.13.

Méthode	Ordre						
	1	2	3	4	5	6	7
PI	-	-	-	3 %	**73** %	21 %	3 %
MCDE	-	-	-	-	43 %	**54** %	3 %
MCV*	-	-	27 %	10 %	17 %	**30** %	17 %
RLT	-	-	3 %	20 %	13 %	**37** %	27 %
BIC	-	-	-	40 %	**60** %	-	-
ICL	-	-	30 %	**53** %	17 %	-	-
AIC	-	-	-	-	**47** %	40 %	13 %
VMP	-	-	**100** %	-	-	-	-

TAB. 3.13 – *Séquences de longueur 1 400 simulées suivant un modèle complexe.*

La comparaison des tableaux 3.12 et 3.13 montre que quand le nombre de données observées augmente, tous les critères ont tendance à sélectionner un nombre d'états cachés plus proche de la valeur cinq (celle du modèle ayant généré les données). Les critères de demi-échantillonnage "pair-impair" PI et BIC choisissent le plus fréquemment des modèles à cinq états cachés, mais alors que PI a une légère tendance à surestimer la complexité du modèle, BIC la sous-estime fréquemment, ce qui paraît moins gênant vu la très faible séparation du mélange. La prise en compte de la forme "en plateau" des deux critères conduirait à choisir des modèles plus parcimonieux. Les critères de vraisemblance marginale pénalisée VMP et ICL ont une forte tendance à sous-estimer le nombre d'états cachés. En théorie, VMP est consistant mais apparemment, ce critère nécessite plus de données observées pour sélectionner le modèle générateur, dans le cadre d'un mélange de degré si élevé. De plus, il est difficile de vérifier toutes les hypothèses assurant la validité de ce critère ($\beta-$mélangeance et intégrabilité de l'entropie à crochets). Enfin, les autres critères de validation croisée ont une forte tendance à surestimer la complexité du modèle, de même que le critère AIC – ce résultat ayant déjà été constaté dans le cadre des mélanges indépendants (voir McLachlan et Peel, 2000 [91] ou Biernacki, 1997 [12]).

Pour confirmer cette tendance des critères, nous considérons quelques séquences de longueur 15 000, pour lesquelles nous comparons les critères PI, BIC, AIC, ICL et VMP. Les courbes des critères sont alors très peu dépendantes de la séquence simulée considérée. Pour l'une d'elles, nous représentons sur la figure 3.10 les différents critères en fonction du nombre d'états cachés.

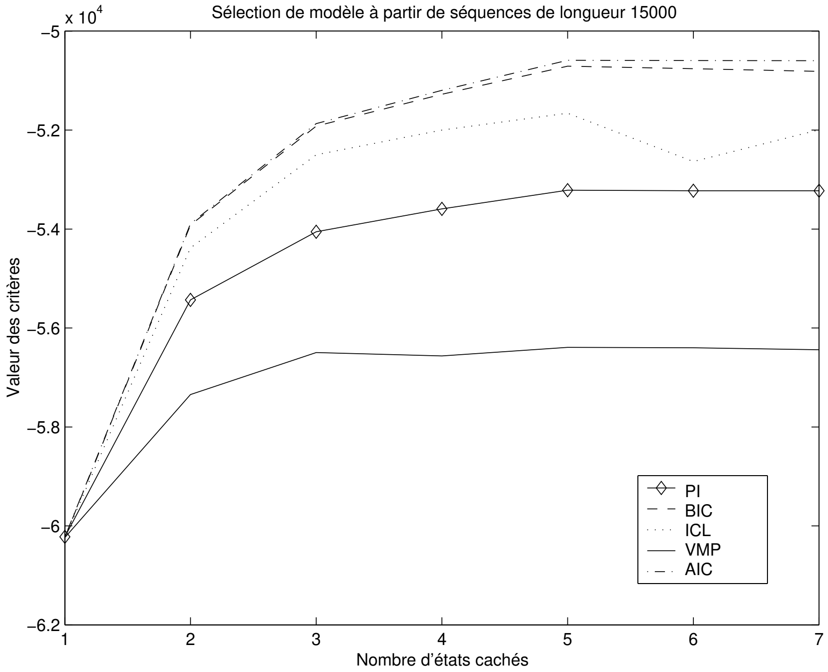


FIG. 3.10 – *Valeurs de cinq critères pour l'une des séquences de longueur 15 000, en fonction du nombre d'états cachés (en abscisse).*

De manière générale, tous les critères sélectionnent très fréquemment le modèle générateur. Notons que la valeur du critère de vraisemblance marginale pénalisée différencie très peu les modèles ayant plus de trois états cachés ; cependant le maximum de sa courbe est correctement localisé. Les courbes des critères BIC et ICL possèdent un pic plus marqué localisé en la valeur $K = 5$. Les critères PI et AIC ont plutôt un aspect en plateau débutant à la valeur $K = 5$.

Conclusion

Les expérimentations réalisées sur ces données simulées mettent tout d'abord en évidence la sensibilité des critères à la méthode d'estimation des paramètres. Dans le cas général, on doit s'affranchir de l'influence de la valeur initiale de l'algorithme EM en partant par exemple de plusieurs valeurs aléatoires. Il est également essentiel de privilégier une condition d'arrêt basée sur la croissance relative de la log-vraisemblance ; cependant le nombre maximal d'itérations peut être limité si on constate que les paramètres et la log-vraisemblance varient peu au-delà de ce seuil. Si l'on utilise la validation croisée, il paraît important de ne pas initialiser l'algorithme EM en estimant les paramètres avec la séquence complète.

Il apparaît que BIC et les critères de demi-échantillonnage, particulièrement la méthode des séquences paire et impaire, choisissent un nombre d'états cachés pertinent. Notons que les temps de calcul mis en jeu pour chaque critère sont équivalents entre eux, pour un nombre fixé d'itérations de l'algorithme EM, étant donné que la complexité de l'algorithme avant-arrière pour le calcul de la vraisemblance ou l'estimation des paramètres est linéaire par rapport au nombre de données observées (voir chapitre 2). Cependant, lorsque les ensembles d'apprentissage et de test sont déterminés au hasard, le critère de demi-échantillonnage possède une assez forte variabilité, compensée par la répétition du tirage des partitions. Ceci multiplie la complexité de la procédure par le nombre de tirages, et rend le critère "pair-impair" préférable. Cependant, on peut remarquer que la procédure de validation croisée multiple se parallélise très facilement, ce qui permet, si l'on possède assez d'ordinateurs, de calculer les critères en un temps identique à celui nécessité par l'identification d'un seul modèle. D'autre part, ces critères sont peu sensibles à la séparation des composants du mélange, si l'on compare aux résultats obtenus avec BIC et d'autres critères par Biernacki, 1997 [12], dans le cas indépendant. Enfin, les critères de demi-échantillonnage sont également peu sensibles à la mélangeance de la chaîne cachée.

Les critères de validation croisée multiple ont tendance à surestimer le nombre d'états cachés, de même que le critère AIC. Même lorsque le nombre d'observations augmente, la validation croisée multiple conserve cette tendance. A contrario, le critère ICL tend à choisir des modèles plus parcimonieux que le modèle générateur, surtout dans les cas où les composants du mélange sont peu séparés. Cela s'explique par le fait que ce critère sélectionne le modèle séparant au mieux les composants. Cependant, il semble que ce critère permette de retrouver le modèle générateur lorsque le nombre de données observées augmente.

Enfin, le critère de vraisemblance marginale pénalisée estime correctement le nombre d'états cachés lorsque le degré de mélange est faible. En revanche, il semble sous-estimer le nombre d'états cachés lorsque le nombre de données observées est modéré compte tenu du degré de mélange. Le nombre de données nécessaire à la sélection du modèle générateur apparaît, dans les cas considérés, comme plus important que celui requis par exemple par les critères BIC et "pair-impair".

3.8.2 Données réelles : sélection de modèles en fiabilité de logiciels

Dans cette section, nous présentons une application des chaînes de Markov cachées à la modélisation du processus des défaillances et des corrections d'un logiciel. L'accent est mis sur la sélection de modèles par les critères BIC, ICL, AIC, la vraisemblance marginale pénalisée et la validation croisée, mais nous présentons également l'étape de modélisation et d'interprétation du modèle, dans le contexte de modèles classiques et généraux en fiabilité de logiciels (processus de Poisson non homogènes et modèles à intensité de défaillance étagée). Cette approche originale de la fiabilité de logiciel a été présentée dans Durand et Gaudoin, 2002 [45].

Contexte : la modélisation en fiabilité de logiciels

Les études concernant la modélisation en fiabilité de logiciels ont débuté il y a une trentaine d'années. À ce jour, plus de cinquante modèles stochastiques ont été proposés pour le processus des défaillances et des corrections de logiciels. Leur rôle est essentiellement d'estimer la fiabilité actuelle et future du logiciel, sur la base d'observations de défaillances passées et de corrections, où une défaillance est définie comme une sortie du logiciel ne correspondant pas à sa spécification. Un état de l'art récent sur le sujet est disponible dans Lyu, 1996 [88] et Pham, 2000 [100].

Un cadre général pour les modèles stochastiques en fiabilité de logiciels est celui des processus ponctuels auto-excités, proposé par Gaudoin, 1990 [54], ou encore Chen et Singpurwalla, 1997 [28]. Soient $(T_i)_{i\geq 1}$ les instants successifs de défaillance du logiciel, avec pour origine $T_0 = 0$. Après chaque défaillance, le logiciel est corrigé ou non puis redémarré. Habituellement, on considère que la durée des corrections est négligeable ou non prise en compte. Les variables aléatoires $X_i = T_i - T_{i-1}$ sont alors les temps inter-défaillances successifs. On nomme N_t le nombre de défaillances survenues entre les instants 0 et t. Le processus des défaillances est, de manière équivalente, l'un des processus aléatoires $(T_i)_{i\geq 1}$, $(X_i)_{i\geq 1}$ ou $(N_t)_{t\geq 0}$. Sa loi est entièrement définie par la donnée de l'intensité de défaillance

$$\lambda_t = \lim_{dt\to 0} \frac{1}{dt} P\left(N_{t+dt} - N_t = 1|\mathcal{F}_t\right)$$

où $\mathcal{F}_t = \sigma\left(\{N_s\}_{0\leq s\leq t}\right)$ désigne la filtration interne du processus des défaillances.

La fiabilité $R_t(\tau)$ du logiciel à l'instant t est la probabilité qu'aucune défaillance ne survienne pendant une durée quelconque τ après t, conditionnellement au passé du processus des défaillances :

$$R_t(\tau) = P\left(N_{t+\tau} - N_t = 0|\mathcal{F}_t\right) = P\left(T_{N_t+1} - t > \tau|\mathcal{F}_t\right) = \exp\left(-\int_t^{t+\tau} \lambda_s ds\right).$$

La plupart des modèles de fiabilité de logiciel supposent que $(N_t)_{t\geq 0}$ est un processus de Poisson non homogène (PPNH), pour lequel l'intensité de défaillance est une fonction déterministe continue de t : $\lambda_t = \lambda(t)$. Les plus courants d'entre eux sont :

- le modèle de Duane, 1964 [43] ou processus en loi de puissance (PLP) où $\lambda(t) = \alpha\beta t^{\beta-1}$, $\alpha \in I\!R^+, \beta \in I\!R^+$;
- le modèle de Goel et Okumoto, 1979 [60] (GO) où $\lambda(t) = \lambda e^{-\phi t}$, $\lambda \in I\!R^+, \phi \in I\!R$;
- le modèle en S de Yamada *et al.*, 1983 [125] (S) où $\lambda(t) = \alpha\beta^2 te^{-\beta t}$, $\alpha \in I\!R^+, \beta \in I\!R^+$.

La raison principale de l'utilisation répandue de PPNH en fiabilité de logiciels est leur simplicité d'utilisation, mais leur inconvénient principal est l'hypothèse que l'intensité de défaillance est une fonction continue du temps t : il est plus réaliste de considérer que les corrections induisent des discontinuités dans l'intensité de défaillance.

En outre, une caractéristique primordiale d'un logiciel, par rapport au matériel, est qu'il est sans usure : si un logiciel n'est pas modifié, les temps inter-défaillances X_i ont même loi et l'intensité de défaillance entre deux exécutions du logiciel doit être constante. C'est pourquoi une autre classe de modèles est basée sur l'hypothèse que les X_i sont indépendants, de loi exponentielle de paramètre λ_i. L'intensité de défaillance est alors une fonction étagée. On obtient des modèles plus sophistiqués en considérant que λ_i est la réalisation d'une variable aléatoire Λ_i et en modélisant la loi du processus $(\Lambda_i)_{i\geq 1}$.

Il paraît réaliste de faire l'hypothèse que la correction d'un logiciel après une défaillance ne dépend que des sources du logiciel à cet instant et non pas de tout l'historique des sources. Cette hypothèse se traduit par la vérification de la propriété de Markov par le processus $(\Lambda_i)_{i\geq 1}$. Ce résultat est montré de manière plus rigoureuse dans Soler, 1988 [111], en faisant notamment des hypothèses sur le processus des entrées du logiciel, qui est également source de hasard. Sous ces hypothèses, $(\Lambda_i)_{i\geq 1}$ est une chaîne de Markov et sachant $\{\Lambda_i = \lambda_i\}_{i\geq 1}$, les temps inter-défaillances X_i sont indépendants, de loi exponentielle de paramètres respectifs les $(\lambda_i)_{i\geq 1}$. La quantité λ_i peut être vue comme le taux de défaillance après la $i^{\text{ème}}$correction. Dans ce contexte, définir un modèle de fiabilité de logiciel revient à modéliser la chaîne de Markov $(\Lambda_i)_{i\geq 1}$. C'est le cas des modèles suivants :

- lorsque les Λ_i sont déterministes, les temps inter-défaillances sont indépendants de loi exponentielle. Les deux modèles les plus connus vérifiant ces hypothèses sont :
  - ⋆ le modèle de Jelinski et Moranda, 1972 [66] (JM) : $\Lambda_i = \phi(N - i + 1)$, $\phi \in \mathbb{R}^+, N \in \mathcal{N}$;
  - ⋆ le modèle géométrique de Moranda, 1979 [93] (GM) : $\Lambda_i = \lambda c^{i-1}$, $\lambda \in \mathbb{R}^+, c \in]0,1]$;
- le modèle de Littlewood et Verral, 1973 [85] (LV), peut être vu comme un modèle de cette catégorie où les Λ_i sont indépendants et de loi Gamma de paramètres respectifs $(\alpha, \beta_1 + i\beta_2)$, $(\alpha, \beta_1, \beta_2) \in \mathbb{R}^{+3}$;
- Gaudoin, Lavergne et Soler, 1994 [56], définissent une classe de modèles, appelés modèles proportionnels, tels que

$$\forall i \geq 1,\ \Lambda_{i+1} = \Lambda_i e^{-\Theta_i}$$

  où Λ_i et Θ_i sont indépendants. Les Θ_i représentent les effets successifs du processus des corrections :
  - $\Theta_i = 0 \Longleftrightarrow \Lambda_{i+1} = \Lambda_i$ signifie que la correction n'a pas d'effet ;
  - $\Theta_i > 0 \Longleftrightarrow \Lambda_{i+1} < \Lambda_i$ signifie que la correction réduit le taux de défaillance du logiciel qui se voit ainsi amélioré ;
  - $\Theta_i < 0 \Longleftrightarrow \Lambda_{i+1} > \Lambda_i$ signifie que la correction détériore le logiciel ;
- le modèle de correction imparfaite proposé par Gaudoin, 1999 [55], fait l'hypothèse que

$$\forall i \geq 1,\ \Lambda_i = (1 - \alpha_i - \beta_i)\,\Lambda_{i-1} + \mu\beta_i$$

  où les α_i sont des taux de bonnes corrections et les β_i des taux de mauvaises corrections. On obtient un modèle simple en posant $\alpha_i = \alpha$ et $\beta_i = \beta$ pour tout i.

Un point commun à tous ces modèles est qu'ils supposent qu'une correction est effectuée après chaque défaillance. En pratique, après des défaillances, les ordinateurs sont souvent redémarrés sans qu'aucune correction ne soit faite. Les corrections commencent à être effectuées quand un nombre de défaillances suffisamment élevé a été observé. Lorsque un logiciel est dans sa phase opérationnelle, il connaît des mises à jour ou des changements de version plutôt que des corrections d'erreurs par paquets successifs, mais ces deux concepts peuvent être traités de la même manière. Dans le cas où une correction est effectuée après chaque défaillance, il arrive souvent que la plupart d'entre elles soient mineures, quelques corrections majeures ayant parfois lieu, ce qui peut être considéré comme équivalent à un changement de version du logiciel.

Les données en fiabilité de logiciel consistent généralement en une liste de temps inter-défaillances : le fait que des corrections ont été ou non effectuées et que ces corrections sont mineures ou majeures est inobservé. Il est donc intéressant de prendre en compte ces aspects dans les modèles de fiabilité de logiciels : c'est ce que font les chaînes de Markov cachées.

Les chaînes de Markov cachées et leur interprétation en fiabilité de logiciels

Utiliser des chaînes de Markov cachées – au sens de la définition de la section 1.3 – pour modéliser le processus $\boldsymbol{X} = (X_i)_{i\geq 1}$ revient à faire l'hypothèse que la chaîne de Markov $\boldsymbol{\Lambda} = (\Lambda_i)_{i\geq 1}$ est à valeurs dans un ensemble fini de cardinal K, noté $\{\lambda^{(1)}, \ldots, \lambda^{(K)}\}$. Rappelons alors les hypothèses sur le processus complet $(\boldsymbol{X}, \boldsymbol{\Lambda})$:

– $\boldsymbol{\Lambda}$ est une chaîne de Markov à valeurs discrètes inobservée ;
– sachant $\{\Lambda_i = \lambda_i\}_{i\geq 1}$, les temps inter-défaillances $(X_i)_{i\geq 1}$ sont indépendants ;
– sachant $\{\Lambda_i = \lambda^{(j)}\}$, X_i est de loi exponentielle $\varepsilon(\lambda^{(j)})$.

Ainsi, $\boldsymbol{X}$ est une chaîne de Markov cachée à lois d'émission exponentielles, dont les états cachés sont les taux de défaillance Λ_i.

Une trajectoire de $\boldsymbol{X}$ peut être séparée en zones homogènes au sein desquelles la valeur du taux de défaillance Λ_i est constante, égale à $\lambda^{(j)}$. Des sauts apparaissent dans le processus des défaillances lorsque la chaîne $\boldsymbol{\Lambda}$ effectue une transition, ce qui crée un changement de zone homogène : à la $i^{\text{ème}}$défaillance, le taux de défaillance passe de $\Lambda_i = \lambda^{(j)}$ à $\Lambda_i = \lambda^{(l)}$. Dans la période de test du logiciel, les zones homogènes peuvent être interprétées comme des périodes où aucune correction n'a été réalisée après les défaillances, ou seulement des corrections mineures qui n'ont pas eu d'effet significatif sur le taux de défaillance. Les transitions correspondent alors à des corrections dans le premier cas et à des corrections majeures dans le second. Pour un logiciel en phase opérationnelle, les transitions peuvent être interprétées comme une mise à jour ou la sortie d'une nouvelle version.

L'avantage du modèle de chaînes de Markov cachées sur les PPNH est qu'il prend en compte les discontinuités induites par les corrections, aussi bien que l'absence d'usure du logiciel. Son avantage par rapport aux modèles (JM), (GM), (MV) et aux autres modèles pour la chaîne $\boldsymbol{\Lambda}$ est la prise en compte des périodes homogènes correspondant à la possibilité de ne pas effectuer de correction. Son inconvénient est que le processus des taux de défaillance est minoré, ce qui supprime la possibilité d'améliorations arbitrairement bonnes du logiciel, asymptotiquement. De plus, quand des corrections sont apportées au logiciel après chaque défaillance, le taux de défaillance croît de manière continue, ce que sont incapables de prendre en compte les chaînes de Markov cachées.

D'autre part, différentes hypothèses peuvent être effectuées concernant le processus des corrections, qui conduisent à des contraintes sur la matrice de transition $P = (p_{jl})_{j,l}$. Considérer une matrice de transition générale revient à admettre la possibilité d'améliorations et de détériorations quelconques du taux de défaillance. Ceci conduit à des modèles ne reflétant pas forcément le processus des corrections de manière fine et ayant un nombre de paramètres indépendants équivalent à K^2. On peut envisager des modèles alternatifs où chaque zone homogène est parcourue une seule fois. C'est le cas des modèles de croissance de fiabilité au sens strict, où chaque correction fait décroître strictement le taux de défaillance. Dès lors, toute transition d'un état vers un autre état rencontré précédemment doit être interdite. Cette hypothèse induit une matrice de transition surdiagonale, de la forme suivante (à une permutation des états près)

$$P = \begin{bmatrix} p_{1,1} & p_{1,2} & 0 & \dots & 0 \\ 0 & p_{2,2} & p_{2,3} & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & p_{K-1,K-1} & p_{K-1,K} \\ 0 & \dots & \dots & 0 & 1 \end{bmatrix}.$$

Les états cachés sont donc ordonnés : dans le cas où le taux de défaillance avant la première correction est $\lambda^{(j)}$, les états cachés visités par la suite par la chaîne de Markov $\mathbf{\Lambda}$ seront $\lambda^{(j+1)}, \dots, \lambda^{(K)}$, pourvu que la séquence soit suffisamment longue et que les $p_{l,l+1}$ soient strictement positifs. De plus, le nombre de paramètres du modèle est une fonction linéaire de K.

En pratique, il est important de pouvoir prendre en compte la possibilité de corrections imparfaites. Cela se traduit par le fait que le retour à des états précédemment visités doit être rendu possible par la matrice de transition P. La manière de le faire la plus simple et la moins coûteuse (en termes de complexité du modèle) est d'autoriser deux transitions uniquement pour chaque état caché (sauf le "premier" et le "dernier") : l'une vers le dernier état visité et l'autre vers le prochain. Cette hypothèse est vérifiée si P est tridiagonale, de la forme suivante (à une permutation des états près)

$$P = \begin{bmatrix} p_{1,1} & p_{1,2} & 0 & \dots & 0 \\ p_{2,1} & p_{2,2} & p_{2,3} & \ddots & 0 \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & p_{K-1,K} \\ 0 & \dots & 0 & p_{K,K-1} & p_{K,K} \end{bmatrix}.$$

Une fois de plus, une telle matrice de transition induit un certain ordre sur les états cachés. Le nombre de paramètres du modèle reste une fonction linéaire de K.

Les données et leur interprétation compte tenu du modèle

Nous utilisons le modèle de chaîne de Markov cachée sur deux groupes de données réelles en fiabilité de logiciel. Le premier est constitué des temps inter-défaillances de neuf logiciels américains de contrôle-commande, dans leur phase de test et leur phase opérationnelle. Les neufs séquences, appelées *données de Musa, 1979 [95]*, sont désignées par M1, M2, M3, M4, M6, M14C, M17, M27 et M40. Leur longueur varie de 38 à 136. Le second groupe est constitué des temps inter-défaillances de quatre logiciels français en période de test (voir Gaudoin, 1990 [54]). Les quatre séquences, de longueur allant de 40 à 395, sont désignées par C1, C2, C3 et C4.

Les paramètres du modèle de Markov caché sont estimés par l'algorithme EM, où le nombre d'itérations maximal est fixé à mille. L'initialisation est aléatoire et se fait à partir de trois valeurs initiales pour lesquelles 50 itérations sont effectuées, le paramètre de sortie maximisant la vraisemblance servant de valeur initiale à une nouvelle exécution d'EM (la justification de ces choix est donnée en partie 3.8.1). Les données étant toutes issues de logiciels différents, on suppose qu'elles sont les réalisations d'autant de modèles de chaîne de Markov cachée, dont les paramètres sont estimés successivement.

Les considérations ci-dessus sur le choix de contraintes concernant la matrice de transition conduisent à considérer des chaînes cachées non stationnaires, fortement susceptibles au contraire d'admettre des états transitoires et des états absorbants. On choisit donc un modèle tel que la loi $(\pi_j)_j = (P(\Lambda_1 = \lambda^{(j)}))_j$ de l'état initial est quelconque. D'après la remarque 2.2 de la section 2.3.3, vu qu'une seule séquence est utilisée pour estimer chaque paramètre, on sait que l'estimateur de maximum de vraisemblance de π correspond à un état initial déterministe. D'autre part, la vraisemblance est invariante par permutation des états cachés. Nous faisons donc l'hypothèse que le taux de défaillance avant la première correction est $\lambda^{(1)}$. Dans le cas de matrices de transition tridiagonales, la permutation des états cachés perturbe leur forme particulière, mais cela n'a pas d'influence sur la loi du modèle, qui est invariante par ces permutations. Dans le cas de matrices de transition surdiagonales, le fait de commencer presque sûrement dans l'état $\lambda^{(1)}$ évite la possibilité que certains états ne soient jamais visités.

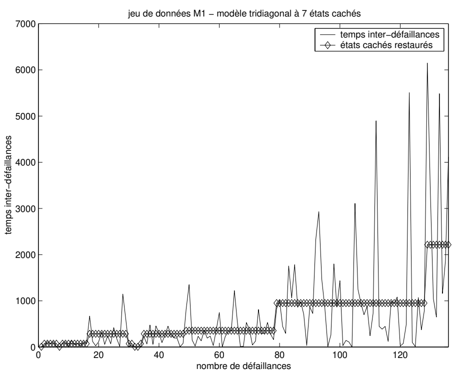


FIG. 3.11a). *Séquence cachée la plus probable globalement.*

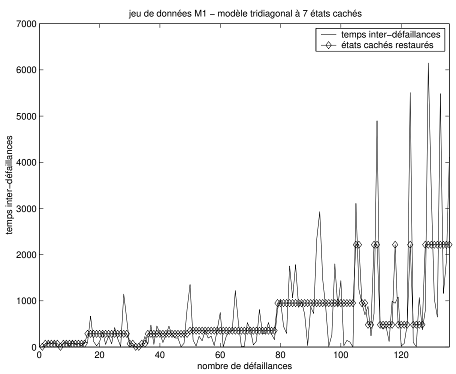


FIG. 3.11b). *États cachés les plus probables individuellement.*

FIG. 3.11 – *Comparaison des méthodes globale (par le MAP) et locale (par les probabilités de lissage) pour la restauration des états cachés. L'algorithme de Viterbi respecte les contraintes de la matrice de transition et favorise les zones homogènes.*

L'interprétation des temps inter-défaillances en termes de corrections majeures ou de versions du logiciel repose avant tout sur la restauration des états cachés. Celle-ci est effectuée par l'algorithme du MAP, dit *de Viterbi* pour les chaînes de Markov cachées (voir section 2.5). Rappelons qu'il s'agit de trouver la séquence $\hat{\boldsymbol{\lambda}} = (\hat{\lambda}_i)_{i \geq 1}$ maximisant la probabilité $P(\boldsymbol{\Lambda} = \boldsymbol{\lambda} | \boldsymbol{X} = \boldsymbol{x})$. Une méthode alternative consiste à restaurer les états cachés par une méthode locale à partir des probabilités de lissage :

$$\hat{\lambda}_i = \arg\max_j P(\Lambda_i = \lambda^{(j)} | \boldsymbol{X} = \boldsymbol{x}).$$

Les inconvénients de cette méthode est que la séquence restaurée par la méthode locale peut être de probabilité nulle dans le cas où des probabilités de transition sont nulles également. D'autre part, l'algorithme de Viterbi tend à donner des périodes homogènes plus stables, par prise en compte plus forte de la dynamique markovienne des états cachés. Ces deux aspects sont illustrés figure 3.11. On considère le jeu de données M1 et un modèle tridiagonal à sept états cachés. Les paramètres estimés sont tels que l'état 7 correspond au paramètre $\lambda^{(7)} \approx 480^{-1}$ et l'état 5 à $\lambda^{(5)} \approx 950^{-1}$. Notons que la comparaison entre les données observées et les états cachés est aisée dans la mesure où $\mathbb{E}[X_i|\Lambda_i = \lambda^{(j)}] = \frac{1}{\lambda^{(j)}}$. La transition de l'état 5 vers l'état 7, interdite par la forme tridiagonale de la matrice, est pourtant présente dans la séquence restaurée par la méthode locale (figure 3.11b), entre la 108$^{\text{ème}}$et la 109$^{\text{ème}}$défaillance). La restauration par le MAP évite ce problème (figure 3.11a)). De plus, dans ce dernier cas, il est plus facile de repérer les zones homogènes qui s'interprètent comme des périodes sans correction majeure du logiciel ; en revanche, les "oscillations" de la séquence restaurée par la méthode locale rendent cette interprétation plus difficile.

D'autre part, l'introduction de contraintes sur la matrice de transition conduit à des modèles essentiellement différents pour le processus $\boldsymbol{X}$. Ces différences se traduisent avant tout dans les paramètres mais sont également reflétées par la restauration des états cachés. La figure 3.12 représente les données du jeu M1 superposées à la séquence cachée la plus probable au sens du MAP, où le modèle considéré a cinq états cachés.

Sur la figure 3.12a), un modèle à matrice de transition non contrainte a été utilisé, ce qui autorise a priori toutes les transitions. Sur la figure 3.12b), il s'agit d'un modèle à matrice de transition surdiagonale. On constate alors que les retours à des états précédemment visités sont interdits, ce qui facilite l'interprétation des zones homogènes. En effet, les transitions entre états cachés sont acceptées moins facilement par l'algorithme du MAP qu'avec les autres types de matrice de transition ; de plus les transitions sont irréversibles, ce qui permet d'interpréter directement les zones homogènes comme des périodes sans correction majeure du logiciel. La figure 3.12b) représente les états cachés restaurés pour un modèle à matrice de transition tridiagonale. Le retour à des états déjà visités est autorisé mais doit se faire en respectant l'ordre des états induits par la matrice de transition. Par exemple, le passage de l'état 2 (pour lequel $\lambda^{(2)} \approx 210^{-1}$) à l'état 4 ($\lambda^{(4)} \approx 1360^{-1}$) doit se faire en passant par l'état 3 ($\lambda^{(3)} \approx 10^{-1}$), ce qui se produit entre la 26$^{\text{ème}}$et la 28$^{\text{ème}}$défaillance . Notons enfin qu'il n'y a pas coïncidence entre les paramètres $\lambda^{(j)}$ quand on change de type de matrice de transition : l'état 3 de la figure 3.12a) ($\lambda^{(3)} \approx 1330^{-1}$) n'a d'homologue ni sur la figure 3.12b) ni sur la figure 3.12c).

Validation et comparaison de modèles par la prédiction

Comme nous l'avons évoqué en introduction, le rôle des modèles de fiabilité de logiciel est essentiellement d'estimer la fiabilité actuelle et future du logiciel, en se basant sur l'observation des défaillances passées – bien que les modèles de Markov cachés puissent être utilisés, en plus de cela, pour identifier les périodes où le logiciel n'a pas été significativement modifié. Ainsi, la qualité des prédictions de la fiabilité fournies par les différents modèles de chaîne de Markov cachée doit être évaluée et comparée à celle des modèles classiques de croissance de fiabilité, présentés en début de section. La méthode

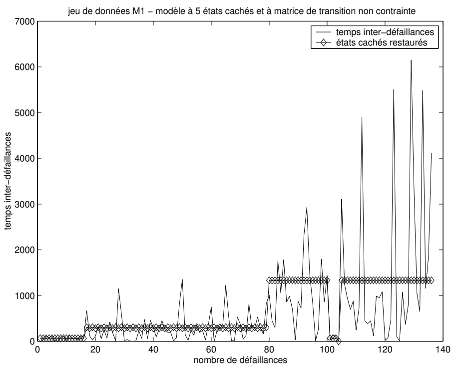


FIG. 3.12a). *Modèle à matrice non contrainte*

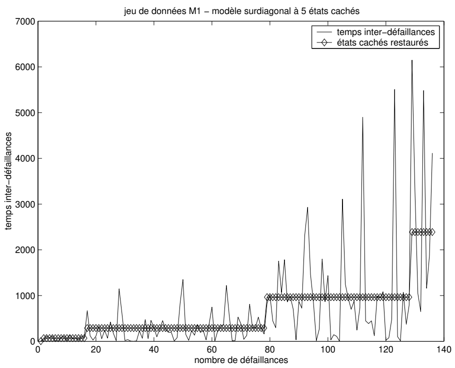


FIG. 3.12b). *Modèle à matrice surdiagonale*

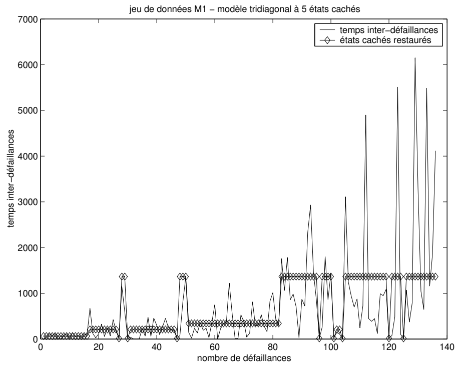


FIG. 3.12c). *Modèle à matrice tridiagonale*

FIG. 3.12 – *Comparaison des trois types de matrice de transition*

usuelle pour comparer les prédictions en fiabilité de logiciels est le critère nommé *U-plot* (*U* pour *Uniform*), présenté par exemple dans Keiller *et al.*, 1983 [72].

Rappelons que la notation $\boldsymbol{X}_1^i$ désigne les i premières valeurs du processus $\boldsymbol{X}$. Pour tout modèle de paramètre θ, soit $P_{\hat{\theta}_i}(X_i \leq x | \boldsymbol{X}_1^{i-1} = \boldsymbol{x}_1^{i-1})$ la fonction de répartition prédictive de l'instant de la prochaine défaillance X_i sachant les $i-1$ premiers temps inter-défaillances, où $\hat{\theta}_i$ est un estimateur de θ obtenu à partir de $\boldsymbol{x}_1^{i-1}$. L'idée générale de la méthode *U-plot* est de calculer les $u_i = P_{\hat{\theta}_i}(X_i \leq x_i | \boldsymbol{X}_1^{i-1} = \boldsymbol{x}_1^{i-1})$. Si l'estimation est de bonne qualité et que le modèle est pertinent au sens de la prédiction de la fiabilité, alors $\boldsymbol{u}_1^n$ doit être la réalisation d'un processus $\boldsymbol{U}_1^n$ approximativement de loi uniforme sur $[0;1]$, du fait que si Z est une variable aléatoire de fonction de répartition F, alors $F(Z)$ suit la loi uniforme. La proximité de la loi de $\boldsymbol{U}_1^n$ avec la loi uniforme, qui mesure la validité prédictive du modèle, est quantifiée par la distance de Kolmogorov-Smirnov KS entre la fonction de répartition empirique de $(\boldsymbol{u}_1^n)$ et la fonction de répartition théorique de la loi uniforme, qui est $F(x) = x$ pour $x \in [0;1]$. Le graphe *U-plot* est le tracé de la fonction de répartition empirique de $\boldsymbol{u}_1^n$. Lorsque plusieurs modèles sont en compétition, le "meilleur modèle" est celui de distance KS minimale. De plus, pour pouvoir comparer entre eux les jeux de données, on ramène cette distance à une même unité en la divisant par $\sqrt{n}$.

Dans le cas de chaînes de Markov cachées, on prouve aisément par intégration des probabilités avant $\alpha_i(j) = P(\Lambda_i = \lambda^{(j)} | \boldsymbol{X}_1^i = \boldsymbol{x}_1^i)$ (ou probabilités *de filtrage*, au sens des modèles à espace d'états), que

$$P(X_i \leq x | \boldsymbol{X}_1^{i-1} = \boldsymbol{x}_1^{i-1}) = \sum_l F_{\lambda^{(l)}}(x) \sum_k \alpha_{i-1}(k) p_{kl}$$

où F_λ désigne la fonction de répartition de la loi exponentielle de paramètre λ.

Remarque 3.4 *Dans d'autres contextes que la fiabilité de logiciels, on peut de même être amené à calculer des fonctions de répartition conditionnelles, en utilisant des modèles de la famille $\mathcal{D}$ plus généraux que les chaînes de Markov cachées. Si l'on ne dispose que de l'algorithme d'arbre de jonction de Jensen, Lauritzen et Olesen, 1990 [67] pour l'inférence, elles ne pourront être calculées, sauf à faire de l'intégration numérique de densités conditionnelles. En revanche, si l'on utilise les formules arrière-avant de la section 2.4, on obtient aisément une expression analytique de ces fonctions de répartitions faisant intervenir celles des lois d'émission, les quantités arrière et éventuellement les quantités avant.*

La figure 3.13 représente le graphe *U-plot* pour le jeu de données M1 de Musa, 1979 [95], en utilisant une chaîne de Markov cachée à trois états à matrice de transition surdiagonale. Ce graphe comporte la fonction de répartition de la loi uniforme (en pointillés) et la fonction de répartition empirique des $(u_i)_{i \geq 1}$ (en trait plein). Sur cet exemple, $n = 136$ et $\frac{KS}{\sqrt{n}} \approx 1,5$.

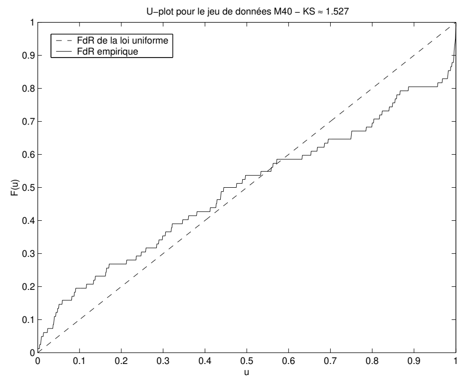


FIG. 3.13 – *Graphe U-plot pour le jeu de données M40, modèle surdiagonal à 3 états cachés. La courbe en pointillés représente la fonction de répartition de la loi uniforme et la courbe en trait plein la fonction de répartition empirique. La distance de KS normalisée est de 1,5.*

Remarquons que dans son principe, le critère *U-plot* est assez similaire à la validation croisée à une donnée exclue, à ceci près que seules les $i-1$ premières données sont utilisées pour l'apprentissage (au lieu de toutes les données sauf la $i^{\text{ème}}$) et que la validation se fait par comparaison de fonctions de répartitions théorique et empirique au lieu de la vraisemblance.

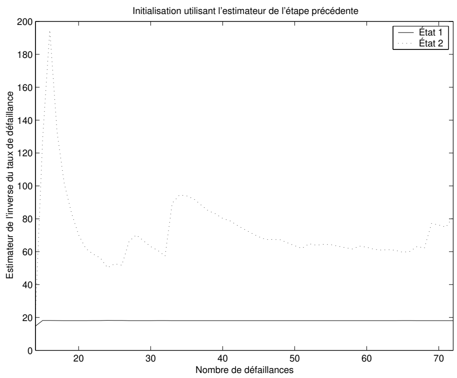


FIG. 3.14a). *Initialisation non aléatoire*

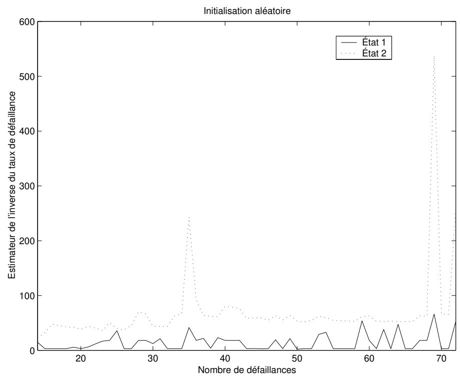


FIG. 3.14b). *Initialisation aléatoire*

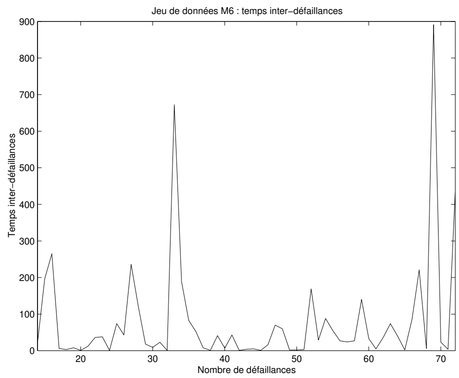


FIG. 3.14c). *Jeu de données M6*

FIG. 3.14 – *Comparaison des deux méthodes d'initialisation de U-Plot :* FIG. *3.14a) Initialisation de EM avec $\hat{\theta}_{i-1}$.* FIG. *3.14b) Initialisation aléatoire de EM.* FIG. *3.14c) Jeu de données M6.*

Enfin, l'utilisation du critère *U-plot* requiert le calcul d'environ n estimateurs de maximum de vraisemblance par l'algorithme EM. Or on s'attend à ce que $\hat{\theta}_i$ soit peu différent de $\hat{\theta}_{i-1}$. Pour accélérer la convergence d'EM, on peut être tenté d'utiliser $\hat{\theta}_{i-1}$ comme paramètre initial de l'algorithme pour calculer $\hat{\theta}_i$. En réalité, cette méthode aboutit à une sous-estimation de la séparation des composants du mélange et conduit à des estimateurs ne maximisant pas la vraisemblance ; il est donc essentiel d'initialiser l'algorithme EM de manière aléatoire. Ceci est illustré par la figure 3.14, où un modèle à deux états cachés et à matrice de transition surdiagonale est utilisé sur le jeu de données M6. Sur la figure 3.14a), l'initialisation de l'algorithme EM se fait par $\hat{\theta}_{i-1}$ et sur la figure 3.14b), l'initialisation de l'algorithme est aléatoire. Dans le cas d'une initialisation avec $\hat{\theta}_{i-1}$, les estimateurs varient de manière plus régulière en fonction de i et restent coincés autour d'un maximum local de la vraisemblance. Le calcul du critère *U-plot* nécessite 280 itérations. L'initialisation aléatoire, au contraire, permet d'atteindre des valeurs plus élevées de la vraisemblance. Cependant, 880 itérations sont nécessaires. La figure 3.14c) représente les données du jeu M6.

Sélection d'un modèle

Nous appliquons les méthodes de sélection de modèles étudiées dans la section 3.8.1 sur des données simulées au choix de l'ordre et du type de matrice de transition des chaînes de Markov cachées pour la fiabilité de logiciels. Tous ces critères utilisent l'estimation des paramètres par maximum de vraisemblance, cette méthode devant être consistante. Notons que cela n'est a priori pas le cas quand les chaînes cachées ont des états transitoires et des états absorbants (cas des matrices de transition surdiagonales). Le critère de vraisemblance marginale pénalisée est un cas particulier puisque les paramètres sont estimés sous un modèle indépendant ; cependant cette méthode s'applique en théorie à des chaînes cachées stationnaires, hypothèse peu réaliste dans notre contexte. Ce critère ne permet donc pas, en particulier, de choisir un type de matrice de transition. Vu que le nombre d'observations par séquence est en général plutôt faible (neuf des treize séquences sont de longueur inférieure à 80), nous étudions également le critère de validation croisée à une donnée exclue, ce qui permet également la comparaison avec le critère *U-plot*, également calculé.

Considérer un nombre maximal d'états cachés indépendant du nombre de données, par exemple sept états comme dans la section 3.8.1, conduit à des problèmes d'estimation des paramètres par l'algorithme EM quand un petit nombre de données est utilisé. De plus, l'utilisation de critères basés sur des approximations asymptotiques n'est pas raisonnable dans ce cas – rappelons que le nombre de paramètres pour un modèle général à sept états cachés est de 49, alors qu'on dispose la plupart du temps de moins de 80 données. C'est pourquoi, suivant Bozdogan, 1993 [13], nous limitons le nombre d'états cachés à $\min(n^{0,3}; 7)$, la valeur sept étant choisie par analogie avec la partie concernant les données simulées.

D'autre part, les critères de validation croisée présentent en général un aspect en forme de plateau. Dans ce cas, on ne choisira pas le modèle maximisant strictement le critère de sélection, mais le modèle le plus parcimonieux ayant une valeur supérieure ou égale à la valeur maximale moins l'écart-type du critère. Ainsi, la figure 3.15 représente le critère MCMCV* pour le jeu de données C2 : il s'agit d'une courbe comportant un plateau à partir du nombre d'états cachés $K = 2$. Bien que le critère soit en fait maximal pour $K = 5$, on choisit deux états cachés.

Le tableau 3.14 donne le couple (type de matrice de transition, nombre d'états cachés) pour chaque jeu de données et chaque critère. Le symbole 'p' désigne une matrice de transition pleine (*i.e.* non contrainte), 's' désigne une matrice de transition surdiagonale et 't' une matrice de transition tridiagonale. Les critères considérés sont BIC, AIC, ICL et les différentes variantes de la validation croisée : demi-échantillonnage simple DES, répété MCDE, "pair-impair" PI, validation croisée multiple MCV* et MCMCV*, apprentissage-test répété RLT et validation croisée à une donnée exclue VCS. La répétition dans les variantes Monte-Carlo de la validation croisée est effectuée six fois. Les partitions utilisées dans MCV*, MCMCV* et dans RLT possèdent cinq classes (quatre d'entre elles étant utilisées pour l'estimation des paramètres et la cinquième pour la validation). On considère également le critère RLTP avec trente tirages des partitions, pour obtenir un critère RLT avec un nombre d'apprentissages et de tests égal à celui de MCMCV*. Nous présentons également les modèles choisis par le critère *U-plot*, en maximisant le critère

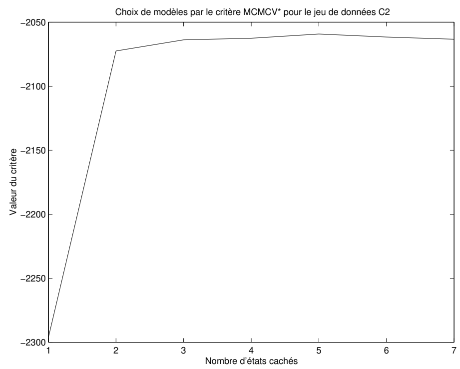


FIG. 3.15 – *Traitement des courbes de critères en plateau : ici MCMCV*. Le critère est maximal pour 5 états cachés mais la courbe est en plateau à partir de deux états cachés : nous choisissons donc la valeur de début de plateau.*

Données	C1	C2	C3	C4	M1	M2	M3	M4	M6	M17	M27	M40	M14C
n	169	395	40	46	136	54	38	53	73	38	41	101	36
PI	s,3	p,2	s,2	t,3	s,3	p,3	p,2	s,2	p,2	s,2	s,2	s,2	p,2
DES	s,2	p,2	s,2	t,3	s,2	s,2	s,2	p,2	p,2	s,2	s,2	p,2	p,2
MCDE	p,2	p,2	s,2	s,3	s,4	s,2	s,2	p,2	p,2	s,2	s,2	p,2	p,2
MCV*	t,3	p,2	s,2	s,3	s,3	s,2	s,2	s,2	p,2	p,1	s,3	s,3	p,2
MCMCV*	s,3	p,2	s,2	s,3	t,3	s,2	s,2	s,2	p,2	t,3	s,2	t,3	p,2
RLT	s,2	p,2	p,2	p,2	s,4	s,3	p,2	s,3	p,2	p,2	s,2	s,2	t,3
RLTP	s,2	p,2	s,3	p,3	p,3	s,3	p,2	s,3	p,2	p,2	p,2	s,2	p,2
VC	s,2	s,2	s,2	s,4	s,3	s,2	s,2	p,2	p,3	s,3	s,2	s,2	p,2
KS	t,2	s,2	s,3	s,3	s,5	t,4	s,3	s,3	t,3	s,3	t,2	s,2	p,2
BIC	p,2	p,2	p,1	s,2	s,2	s,2	s,2	s,2	p,2	s,2	s,2	s,2	p,2
ICL	p,2	p,2	p,1	s,2	s,2	s,2	s,2	s,2	p,2	s,2	s,2	s,2	p,2
AIC	p,2	p,2	s,2	s,2	s,2	p,2	s,2	s,2	p,2	p,2	s,2	s,2	p,2
VMP	p,2	p,2	p,1	p,2	p,2	p,2	p,2	p,2	p,2	p,2	p,2	p,2	p,2

TAB. 3.14 – *Couple (type de matrice de transition, nombre d'états cachés) sélectionnés par les différents critères en fonction du jeu de données. Le symbole 'p' désigne une matrice de transition pleine, 's' désigne une matrice de transition surdiagonale et 't' une matrice de transition tridiagonale.*

numérique KS.

Tout d'abord, ces résultats mettent en évidence le peu d'intérêt du modèle 't' à matrice de transition tridiagonale, qui n'est presque jamais sélectionné. Notons que pour $K \leq 2$, ce modèle est identique au modèle non contraint 'p'. Mis à part le critère *U-plot*, qui accepte visiblement des modèles trop complexes, et peut-être le critère de validation croisée VCS à une donnée exclue, les modèles sélectionnés par les critères sont plausibles. On vérifie en considérant les paramètres estimés et par restauration des états cachés que pour des modèles plus complexes (typiquement avec cinq états cachés), des paramètres d'émission égaux ou des états jamais visités apparaissent. Ainsi, sur la figure 3.12 où on considère un modèle à cinq états cachés, certains d'entre eux sont pratiquement inutiles et sont associés à une unique donnée observée. Dans d'autre cas, l'égalité de deux paramètres d'émission se traduit par des oscillations artificielles dans la séquence restaurée.

Notons le comportement très similaire des critères BIC, ICL et AIC sur pratiquement tous les jeux de données. De manière général, ces trois critères choisissent des modèles extrêmement parcimonieux (jamais plus de deux états cachés, soit quatre paramètres indépendants). D'avis d'expert, des modèles si parcimonieux sont inattendus et nous pensions trouver plus de périodes homogènes dans les temps inter-défaillances. Cependant, les modèles sélectionnés semblent adéquats pour l'interprétation des données, sauf peut-être pour les jeux de données C3, C4 et M1.

En effet, un modèle à deux états cachés à matrice de transition surdiagonale – choisi par les critères de demi-échantillonnage – paraît plus adapté au jeu de données C3 que le modèle exponentiel d'indépendance (correspondant à un état caché) – choisi par les critères BIC, ICL et VMP. La figure 3.16a) compare les états cachés restaurés pour des modèles à un et à deux états cachés.

Pour le jeu de données C4, un modèle à trois états cachés – de nouveau choisi par les critères de demi-échantillonnage – permet une interprétation plus évidente que le modèle à trois états cachés – choisi par les critères AIC, BIC, ICL et VMP (voir figure 3.16b). En revanche, on peut vérifier par un examen des paramètres estimés qu'un modèle plus complexe est inutile (par exemple le modèle (s,4), choisi par validation croisée à une donnée exclue VCS).

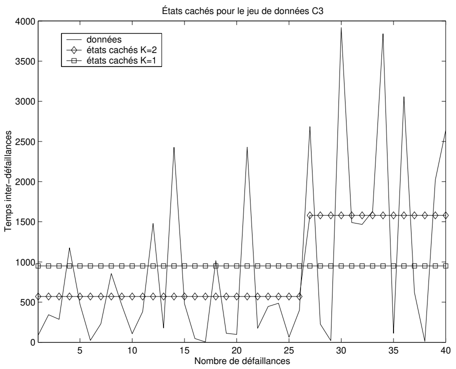


FIG. 3.16a). *Jeu de données C3*

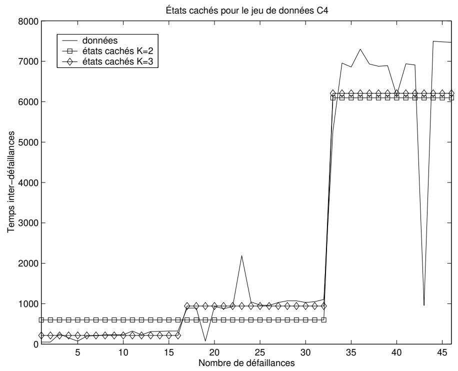


FIG. 3.16b). *Jeu de données C4*

FIG. 3.16 – *Comparaison des séquences cachées restaurées :* FIG. *3.16a) Modèles à un ou deux états cachés pour le jeu de données C3 ;* FIG. *3.16a) Modèles à un ou deux états cachés pour le jeu de données C4.*

Enfin, pour le jeu de données M1, le choix par les critères AIC, BIC, ICL et VMP d'un modèle à deux états cachés est étonnant. Un modèle à trois voire à quatre états cachés, choisis respectivement par la validation croisée "paire-impaire" et le demi-échantillonnage Monte-Carlo, paraît plus judicieux d'un point de vue visuel et pour l'interprétation des données. La restauration des états cachés pour le modèle (s,4) est analogue à la figure

3.12 b).

Notons que pour les jeux de données M4 à M40 et M14C, les critères de validation croisée multiple (c'est-à-dire tous sauf VCS) sont assez bien en accord avec BIC, ICL, AIC et VMP. Les critères de demi-échantillonnage "pair-impair" et BIC coïncident exactement. Tous jeux de données confondus, les critères de demi-échantillonnage tendent à sélectionner des modèles moins complexes que les autres critères de validation croisée. Les critères de validation croisée à une donnée exclue et le critère *U-plot*, qui sont plutôt similaires du point de vue du calcul, ne coïncident pas mais sélectionnent tous deux des modèles visiblement plus complexes que nécessaire, le critère *U-plot* choisissant parfois un nombre d'états cachés peu raisonnable.

Pour conclure, les critères de demi-échantillonnage PI et MCDE fournissent des modèles pas trop complexes mais permettant une interprétation aisée et visuelle, en terme de périodes homogènes, des temps inter-défaillances. Les critères BIC, ICL, AIC et VMP choisissent la plupart du temps les mêmes modèles que PI et MCDE mais parfois des modèles moins complexes, d'interprétation moins évidente. Les autres critères de validation croisée ont une légère tendance à préférer des modèles plus complexes (cas de RLT, RLTP, MCV* et MCMCV*), voire des modèles trop complexes (VCS) ou dans certains cas déraisonnables (*U-plot*, autrement dit KS).

Comparaison des chaînes de Markov cachées avec les modèles classiques en fiabilité de logiciels

Le tableau 3.15 rappelle, pour chaque jeu de données, le modèle sélectionné par le critère BIC. Puis la capacité prédictive de ce modèle, mesurée par le critère *U-plot*, est comparée à celle des modèles classiques en fiabilité de logiciels. Le tableau indique le rang de chaque modèle.

Jeu de données	C1	C2	C3	C4	M1	M2	M3	M4	M6	M17	M27	M40	M14C
matrice	p	p	p	p	s	s	s	s	p	s	s	s	p
K	2	2	1	2	2	2	2	2	2	2	2	2	2
CMC	1	1	5	5	4	4	4	2	1	2	2	2	1
PLP	4	2	4	3	2	3	3	1	2	5	4	3	5
GO	2	4	1	2	3	2	2	4	4	3	3	4	6
S	5	6	6	4	5	5	5	5	5	6	5	5	3
JM	6	5	2	6	6	6	6	6	6	1	6	6	2
GM	3	3	3	1	1	1	1	3	3	4	1	1	4

TAB. 3.15 – *Comparaison des chaînes de Markov cachées avec les modèles classiques de fiabilité de logiciels. Les colonnes indiquent, pour chaque jeu de données, le rang des modèles ordonnés par leur valeur du critère U-Plot.*

Le rang moyen de chaque modèle, calculé en utilisant le rang obtenu par ce modèle pour chaque jeu de données, est présenté dans le tableau 3.16. Ce rang est fonction du critère utilisé pour la sélection des modèles de chaîne de Markov cachée.

	BIC	KS
CMC	2,6	2,2
PLP	3,2	3,4
GO	3,1	3,2
S	5,0	5,0
JM	4,9	4,9
GM	2,2	2,3

TAB. 3.16 – *Rang moyen, sur les différents jeux de données, des modèles de fiabilité de logiciels, classés en fonction du critère U-plot.*

Il apparaît donc que, relativement aux modèles classiques de fiabilité de logiciels, les chaînes de Markov cachées ont une bonne capacité prédictive en moyenne. Leur performance prédictive peut être améliorée en basant la sélection de ces modèles (nombre d'états cachés et type de matrice de transition) sur le critère *U-plot* au lieu d'autres critères (par exemple BIC, dans le tableau 3.15 et la première colonne du tableau 3.16). Cependant, les modèles choisis par *U-plot* ont tendance à être plus complexes que nécessaire, du point de vue de l'estimation de densité.

Le modèle de chaîne de Markov cachée se révèle en général performant, au sens du critère *U-plot*, quand la fiabilité du logiciel reste constante par morceaux. En revanche, quand celle-ci évolue de manière continue au cours des défaillances successives, ce modèle se révèle incapable d'anticiper la croissance de la fiabilité, contrairement aux autres modèles qui, justement, essaient de modéliser cette croissance. Cependant, sur certains jeux de données comme par exemple C4 (représenté figure 3.16b), où les données sont nettement étagées, on s'attendrait à ce qu'une chaîne de Markov cachée ait une bonne capacité prédictive. Or il s'agit d'un des modèles les moins performants : ceci s'explique sans doute par le fait que dans le calcul du critère *U-plot*, l'estimation des paramètres se fait en utilisant les $i-1$ premières valeurs de la séquence, pour chaque valeur de i. Pour le modèle à trois états cachés utilisé sur la figure 3.16b), la signification des états change donc pour chaque $\hat{\theta}_i$ et il est difficile, dans ce cas, d'essayer de relier les résultats du critère *U-plot* à l'interprétation visuelle de la chaîne de Markov cachée.

Conclusion et perspectives

En définitive, l'approche par chaînes de Markov cachées offre de nouvelles possibilités en modélisation de la fiabilité de logiciels. Ces modèles prennent en compte des facteurs qui ne sont pas représentés par d'autres modèles usuels, comme l'existence de périodes homogènes. Leur capacité prédictive est plutôt bonne en général. Dans les autres cas, cette approche reste intéressante pour repérer les instants des corrections majeures.

Une limite des chaînes de Markov cachées, par rapport aux modèles concurrents, est leur incapacité à anticiper la croissance de la fiabilité quand celle-ci évolue de manière progressive et non par paliers. Il est donc intéressant de considérer l'extension suivante, qui prend en compte l'évolution de la fiabilité suivant un modèle de croissance, à l'intérieur des périodes homogènes. Ceci conduit à des mélanges de modèles usuels avec un régime markovien pour les transitions. Par exemple, un "modèle géométrique de Moranda de Markov caché" correspond à l'hypothèse que sachant $\Lambda_i = \lambda^{(j)}$, X_i est de loi $\varepsilon(\lambda^{(j)} c_j^{i-1})$, le processus $(\Lambda_i)_i$ demeurant une chaîne de Markov.

3.9 Conclusion sur la sélection de modèles

Les expérimentations menées sur les modèles de chaîne de Markov cachée avec des données simulées et des données réelles mettent en évidence, en premier lieu, l'importance des précautions suivantes lors de l'estimation des paramètres par l'algorithme EM. D'une part, cet algorithme est véritablement sensible à la valeur du paramètre initial. L'algorithme EM *à la Gibbs* ne parvient pas à atténuer suffisamment cette sensibilité et il est préférable de considérer plusieurs valeurs initiales aléatoires du paramètre. De plus, le critère d'arrêt est lui aussi déterminant : il doit être de préférence basé sur la croissance relative de la log-vraisemblance. Les performances des méthodes de sélection de modèles envisagées sont déterminées par celles de l'estimation des paramètres par l'algorithme EM.

D'autre part, les critères BIC et de demi-échantillonnage, en particulier PI, basé sur les sous-séquences d'indices pairs et d'indices impairs, sélectionnent généralement des modèles raisonnablement parcimonieux, compte tenu du nombre de données disponibles. Nous recommandons donc ces deux méthodes de sélection de modèles. Le critère ICL, en accord avec son objectif, rejette les modèles à composants peu séparés, ce qui conduit en général au choix de modèles plus parcimonieux que ceux sélectionnés par BIC et PI. Le critère de vraisemblance marginale pénalisée sélectionne un nombre d'états cachés correct lorsque les composants sont séparés. Dans le cas de mélanges très imbriqués, il différencie assez peu les modèles les plus plausibles, ce qui aboutit à la sélection de modèles parcimonieux (par rapport à BIC et PI) à moins que le nombre de données observées ne soit très élevé. Enfin, le critère AIC et la validation croisée multiple ou à une donnée exclue ont tendance à favoriser des modèles trop complexes, en particulier dans le cas de données réelles. Quant aux méthodes basées sur les tests, leur mise en œuvre serait intéressante, pour déterminer leur comportement par rapport aux autres critères.

Notons, dans le cadre du choix du nombre d'états cachés, le manque de résultats théoriques sur la loi de la valeur sélectionnée par les différents critères, mis à part pour la vraisemblance marginale pénalisée. De plus, toutes les méthodes considérées sont a priori inadaptées à la sélection de modèles dans le cas de chaînes cachées possédant des états transitoires ou des états absorbants, pourtant d'un grand intérêt pour de nombreuses applications. Il n'en reste pas moins qu'elles sont utiles et performantes dans les situations rencontrées au cours de ce chapitre, c'est pourquoi leurs propriétés théoriques gagneraient à être étudiées. Nous rencontrerons au chapitre 4 un problème de sélection de modèles où l'application des méthodes ci-dessus ne conduit pas à des modèles utiles vis-à-vis de l'usage que nous souhaitons en faire, ce qui est l'occasion de rappeler l'importance, dans la sélection d'un modèle, de la prise en compte de son utilisation finale.

Chapitre 4

Application : étude de courbes de consommation électrique

4.1 Introduction

Dans cette section, nous présentons une étude réalisée en collaboration avec le département "clientèle" de EDF-DRD CLAMART et ayant pour objet l'analyse statistique de courbes de consommation électrique. Dans un premier temps, nous justifions l'emploi de chaînes de Markov cachées pour la modélisation de ces courbes de consommation. Puis nous étudions l'effet sur la log-consommation de facteurs contrôlés (mois, jour, heure, type de contrat et puissance maximale souscrite), par une analyse de variance. Les résidus sont alors modélisés par une chaîne de Markov cachée. Ensuite, les états cachés sont restaurés puis interprétés grâce à un tableau de contingence mettant en relation les états cachés restaurés et la consommation de différents appareils électriques (ou *usages*) lorsque celle-ci est disponible. Un objectif de cette étude est de permettre l'estimation de la consommation de chaque usage dans le cas où elle est inconnue ; nous donnons donc une méthode pour estimer ces consommations. Cette technique se base sur le tableau de contingence et les états cachés restaurés. Nous montrons comment la prise en compte d'informations a priori sur la consommation permet de rendre cette estimation plus réaliste. Enfin, nous abordons le problème de la sélection de modèles. Les critères classiques conduisent à des modèles extrêmement complexes qui rendent très difficile l'interprétation des états cachés par leur mise en correspondance avec les usages. C'est pourquoi nous présentons des méthodes alternatives pour choisir un modèle. Les résultats de cette étude sont présentés, avec en particulier l'interprétation des états cachés et l'estimation des usages sur des courbes de test.

4.2 Problématique

Les données

Il s'agit d'exploiter les résultats d'une campagne de mesure des principaux usages domestiques de la clientèle résidentielle d'EDF : le chauffage électrique (désigné par l'abréviation CHA), l'eau chaude sanitaire (ECS), l'éclairage halogène (H), l'éclairage non halogène (NH), les autres types d'éclairage (ECL), le lave-vaisselle (LV), le lave-linge (LL), le sèche-linge (SL), la cuisson classique (CUI), les équipements de télévision et chaîne hi-fi (TVH), les appareils de réfrigération (FRD), les autres appareils électriques (AUT). La consommation totale du logement, tous usages et appareils confondus, est également mesurée. Cette campagne concerne 100 clients dont les usages sont mesurés sur une année entière, toutes les dix minutes.

En réalité, nous ne disposons pas, pour notre étude, de ces données mesurées, mais de données simulées fournies par EDF : 750 courbes de charge donnant la consommation totale journalière du logement (une mesure toutes les dix minutes). Ces courbes sont simulées de manière à respecter les variations de la consommation totale dues au jour de l'année, à l'heure de la journée et au type de tarif, entre autres. Nous disposons également de 250 courbes comportant la décomposition par usage (même fréquence des mesures). Il s'agit de courbes obtenues par ajout de bruit à des courbes réelles.

Enjeux de cette campagne

Les apports attendus d'un modèle statistique pour l'analyse de ces courbes de consommation sont :

1. l'étude des variations de la consommation en fonction de différents facteurs contrôlés : le mois, le jour de la semaine ou du mois, l'heure, le type de tarif (normal, nuit ou EJP) et la puissance souscrite ;
2. l'estimation des usages quand seule la consommation totale est connue ;
3. la prévision de la consommation future de logements ;
4. la simulation de scénarii de consommation électrique ;
5. la classification de courbes de charges, afin de créer une typologie des clients ;
6. l'étude, plus globalement, des habitudes de consommation des clients en fonction des caractéristiques de leur logement et du type de contrat souscrit.

Dans ce qui suit, nous nous intéressons principalement aux deux premiers points, en nous basant sur le modèle des chaînes de Markov cachées. Les points 3 et 4 peuvent être traités facilement en utilisant les algorithmes des chapitres 2 et 3. Enfin, comme perspective de ce travail, nous donnons quelques pistes permettant de traiter les points 5 et 6.

4.3 Modélisation

L'idée à l'origine de la modélisation de la consommation électrique par des chaînes de Markov cachées est la suivante. Nous faisons l'hypothèse que la consommation électrique d'un logement s'explique essentiellement par l'existence de périodes homogènes du point de vue de la consommation électrique et qu'à l'intérieur de chacune de ces périodes, les fluctuations de la consommation sont dues au hasard. Par exemple, pour un ménage, les périodes de comportement homogènes peuvent être interprétées en termes de type d'activité domestique, qui induit différents niveaux de consommation électrique : repos, lessive, préparation de repas, veillée avec utilisation de la télévision ou de la chaîne hi-fi, etc. De telles activités, relativement homogènes par tranche de temps, sont modélisées par des chaînes de Markov. Or nous ne disposons pas du type d'activité des ménages, mais seulement de leur consommation totale, c'est pourquoi les valeurs de la chaîne de Markov sont cachées. Du fait que, dans une période homogène donnée, les fluctuations de la consommation sont dues au hasard uniquement, les consommations instantanées sont conditionnellement indépendantes sachant l'état latent et suivent une loi de type connu (par exemple une loi gaussienne).

En définitive, le modèle est donc le suivant. La consommation électrique est décrite par deux processus : le processus observé $(Y_1, \ldots, Y_n) = \boldsymbol{Y}_1^n$ de la consommation électrique proprement dite, à valeurs dans $\mathbb{R}$, et un processus caché $(S_1, \ldots, S_n) = \boldsymbol{S}_1^n$, à nombre fini d'états et à valeurs dans $\{1, ..., K\}$, qui sont tels que :

- $\boldsymbol{S}_1^n$ est une chaîne de Markov homogène, irréductible et stationnaire de matrice de transition P et de distribution stationnaire π ;
- les Y_t sont indépendants conditionnellement aux S_t ;
- sachant $S_t = j$, Y_t suit une loi de densité f_{θ_j} appartenant à une famille paramétrée $(P_\theta)_{\theta\in\Theta}$.

Il s'ensuit que la loi de $\boldsymbol{Y}_1^n$ est celle d'une chaîne de Markov cachée – au sens de la définition de la section 1.3.

Remarque 4.1

- *Le choix de la famille $(P_\theta)_{\theta\in\Theta}$ des lois d'émission, qui modélise la nature du bruit, est bien sûr important. Nous choisirons une famille gaussienne, de sorte que $\theta = (\mu, \sigma^2)$ où μ représente la moyenne et σ^2 la variance. Ce choix est justifié dans les sections 4.4 et 4.5.2.*
- *Sous les hypothèses ci-dessus, pour chaque t compris entre 1 et n, la loi marginale de Y_t est une loi de mélange, car*

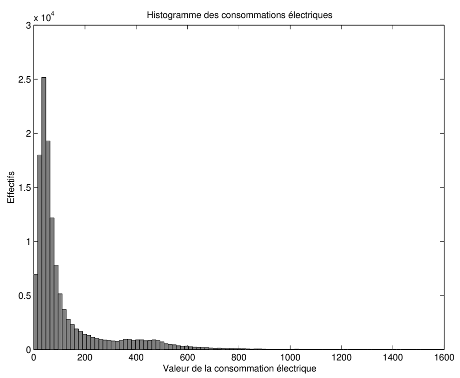
$$f_{Y_t}(y) = \sum_{j=1}^{K} \pi_j f_{\theta_j}(y).$$

4.4 Prétraitement des données

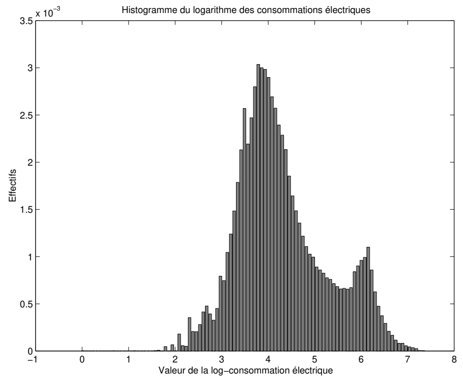
Nous considérons, dans un premier temps, les courbes de charge journalière de la consommation électrique totale, les usages n'étant pas disponibles. Dans le cas de mesures réelles de la consommation dans des logements, on peut faire l'hypothèse que les consommations journalières de différents logements sont des réalisations de plusieurs modèles distincts. Cependant, nous disposons en l'occurrence de courbes simulées, que nous supposons être des réalisations mutuellement indépendantes d'un même modèle. Préalablement, nous effectuons une analyse de variance en fonction de différents facteurs contrôlés. En réalité, un modèle à structure cachée vise à mettre en évidence des effets non mesurables directement à partir des covariables disponibles. Il est donc important de travailler orthogonalement aux effets contrôlés. Ainsi, la modélisation par chaînes de Markov cachées doit se faire sur les résidus d'un modèle validé d'analyse de variance.

Notons que de la sorte, on consolide l'hypothèse de stationnarité du processus observé (sa non stationnarité implique celle du processus caché).

Dans l'analyse de variance, les variables aléatoires sont supposées gaussiennes. L'histogramme de gauche sur la figure 4.1 représente la répartition de tous les relevés de consommation électrique globale $(G_t)_t$, tandis que celui de droite représente celle de leur logarithme. Nous constatons (visuellement) que l'hypothèse d'une loi gaussienne ou de mélange gaussien est nettement plus plausible pour les $(\log(G_t))_t$ que pour les $(G_t)_t$.



Consommations électriques



Logarithme des consommations électriques

FIG. 4.1 – *Histogramme des consommations électriques et leur logarithme.*

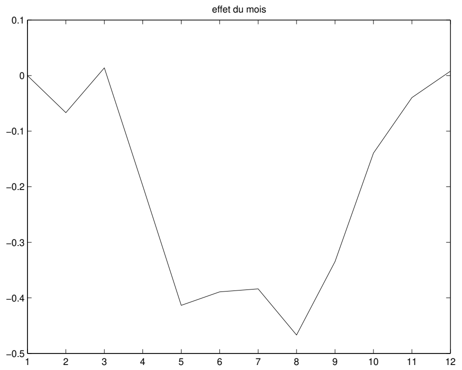
Les facteurs contrôlés disponibles sont les suivants : *type de contrat* (ou *tarif*) c, *puissance souscrite* p, *mois* m, *heure du jour* h, *jour du mois* et *jour de la semaine*. La partie concernant la sélection des facteurs et de leurs interactions dans l'analyse de variance est reportée dans la section 4.7, dédiée à la sélection de modèles. Nos conclusions sont que la prise en compte des effets *jour du mois* et *jour de la semaine* contribue à n'expliquer que peu de déviance par rapport aux autres facteurs contrôlés ; ces effets sont donc ignorés. En outre, la prise en compte d'interactions entre facteurs autres que le mois et le tarif n'est pas pertinente non plus. De plus, les tarifs "nuit" et "EJP" sont regroupés dans une même modalité de l'effet c – l'autre modalité étant le tarif "normal".

En définitive, nous considérons le modèle suivant d'analyse de variance pour le logarithme de la consommation électrique globale $G_{m,h,p,c,t}$:

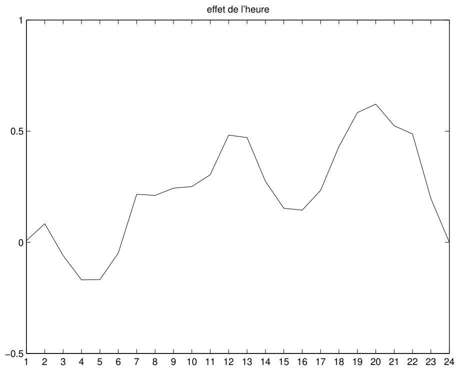
$$\log(G_{m,h,p,c,t}) = \alpha + \beta_m + \gamma_h + \delta_p + \eta_c + \zeta_{c,m} + \varepsilon. \tag{4.1}$$

Les estimateurs des coefficients associés aux facteurs *mois* et *heure* sont représentés figure 4.2. Par convention, pour chaque effet contrôlé, la valeur du paramètre pour la première modalité est nulle.

Nous proposons ensuite de modéliser le processus des $(\varepsilon_t)_t$ par une chaîne de Markov cachée, de sorte que par cohérence avec les notations utilisées dans la section 4.3, la séquence $(\varepsilon_1, \ldots, \varepsilon_n)$ est notée $\boldsymbol{Y}_1^n$. Dans le modèle d'analyse de variance, les résidus $(\varepsilon_t)_t$ sont supposés indépendants et de loi normale de moyenne nulle et de variance σ^2. L'idée présidant à notre tentative de modélisation est précisément que ces hypothèses d'indépendance et de normalité sont douteuses dans le contexte de courbes de consommation électrique.



Effet du mois



Effet de l'heure

FIG. 4.2 – *Estimateurs de paramètres de l'analyse de variance pour les effets dus au mois et à l'heure.*

4.5 Interprétation des états cachés

Une compréhension fine du modèle passe par l'interprétation des états cachés. D'une part, celle-ci permet d'associer un niveau de consommation moyen à chaque état. En outre, la restauration des états cachés permet d'obtenir des histogrammes reflétant la distribution empirique conditionnelle de la consommation dans un état donné. Enfin, la mise en correspondance des usages et des états cachés permet leur interprétation en terme de type d'activité.

Dans ce qui suit, nous considérons un modèle de chaîne de Markov cachée à $K = 7$ états cachés. Ce choix est expliqué dans la section 4.7, dédiée à la sélection de modèles. Les paramètres du modèle sont estimés par l'algorithme EM à partir des résidus $(y_t)_t$ de l'analyse de variance. Pour ce faire, nous disposons de 856 courbes de longueur 144, supposées être des réalisations indépendantes d'un même modèle. Les 144 données par séquence correspondent à des mesures effectuées toutes les dix minutes pendant une journée. L'algorithme est initialisé de manière aléatoire, en considérant trois valeurs initiales du paramètre. Le nombre maximal d'itérations est fixé à 10 000. De nombreux relevés de la consommation sont manquants, c'est pourquoi nous utilisons l'algorithme EM de la section 3.4.3, adapté au cas de variables aléatoires Y_t supprimées.

4.5.1 Restauration des états cachés

Les états cachés $(\hat{s}_t)_t$ les plus probables sont calculés pour chaque courbe séparément par l'algorithme du MAP (dit *algorithme de Viterbi*, pour les chaînes de Markov cachées, voir section 2.5). Cet algorithme détermine les états cachés maximisant la probabilité conditionnelle de la chaîne cachée entière, sachant les observations. La figure 4.3 permet la comparaison, pour une courbe donnée (courbe numéro 1), de la consommation électrique globale $(c_t)_{1 \leq t \leq 144}$ (cadre supérieur), son logarithme $(\log(c_t))_{1 \leq t \leq 144}$ (cadre du milieu) et les résidus $(y_t)_{1 \leq t \leq 144}$, superposés aux états cachés restaurés par l'algorithme de Viterbi (cadre inférieur).

Les sept états cachés se distinguent par la valeur de la moyenne et de la variance

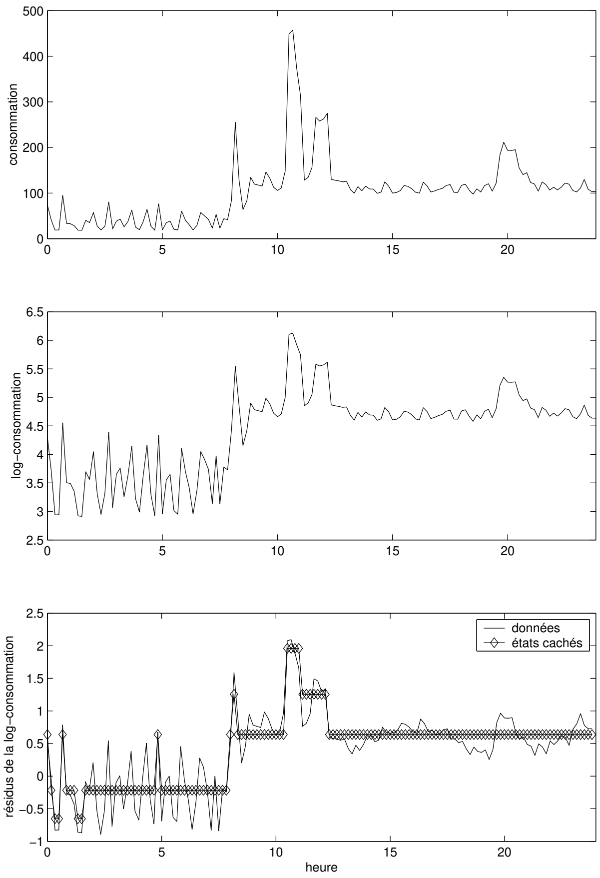


FIG. 4.3 – *Consommation, log consommation, résidus et états cachés pour la courbe numéro 1. En haut, la consommation électrique globale ; au milieu, la log-consommation ; en bas, les résidus de l'analyse de variance et les états cachés restaurés.*

des lois conditionnelles gaussiennes. Les moyennes estimées $(\hat{\mu}_j)_{1 \leq j \leq 7}$ sont données par l'équation (4.2). On peut renommer $(C1, \ldots, C7)$ les indices des états cachés en ordonnant ces derniers par valeur décroissante de $\hat{\mu}_j$.

$$\begin{bmatrix} \hat{\mu}_1 \\ \hat{\mu}_2 \\ \hat{\mu}_3 \\ \hat{\mu}_4 \\ \hat{\mu}_5 \\ \hat{\mu}_6 \\ \hat{\mu}_7 \end{bmatrix} = \begin{bmatrix} \hat{\mu}_{C3} \\ \hat{\mu}_{C2} \\ \hat{\mu}_{C7} \\ \hat{\mu}_{C4} \\ \hat{\mu}_{C6} \\ \hat{\mu}_{C1} \\ \hat{\mu}_{C5} \end{bmatrix} \approx \begin{bmatrix} 0,4 \\ 1,0 \\ -1,4 \\ 0,1 \\ -0,7 \\ 1,8 \\ -0,3 \end{bmatrix} \tag{4.2}$$

La variance estimée est du même ordre de grandeur pour les différents états cachés, à savoir d'environ 0,065 sauf pour l'état 3 (aussi appelé C7 et correspondant aux consommations les plus faibles), où elle est trois fois plus élevée. Ainsi, les états cachés sont essentiellement liés aux différents niveaux de la log-consommation, corrigée des variations saisonnières et de celles dues au type d'abonnement.

4.5.2 Estimation de la densité marginale et des densités conditionnelles

Le nombre d'états cachés peut être vu comme le nombre de composants de la loi marginale de Y_t, puisqu'il s'agit d'une loi de mélange (voir remarque 4.1). Pour apprécier visuellement la qualité de l'estimation de cette densité marginale, nous superposons l'histogramme des valeurs observées $(y_t)_t$ avec la densité théorique en utilisant les paramètres estimés. La figure 4.4 représente cet histogramme superposé à la densité marginale estimée, qui est une loi de mélange gaussien à sept composants.

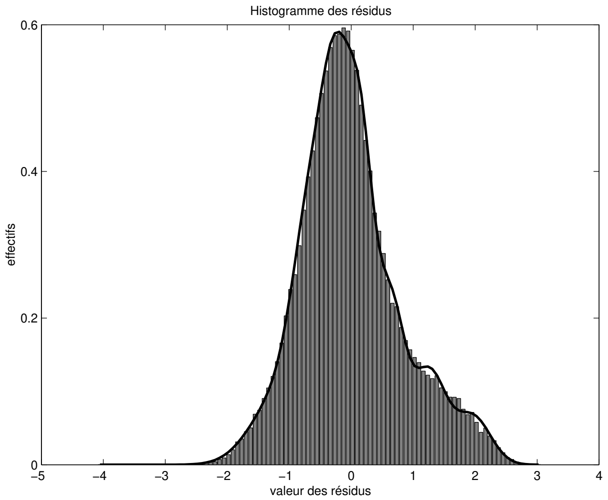


FIG. 4.4 – *Densité marginale théorique et empirique des résidus.*

De plus, la restauration des états cachés permet de tracer, pour chaque valeur k des états cachés, l'histogramme des résidus y_t dont l'état le plus probable est k (au sens du MAP). Nous superposons ensuite cet histogramme à la densité conditionnelle estimée de Y_t sachant $S_t = k$, en l'occurrence la loi $\mathcal{N}(\hat{\mu}_k, \hat{\sigma}_k^2)$ - voir figure 4.5. Nous constatons une bonne coïncidence entre les deux courbes pour chacun des états cachés. Ceci tend à justifier l'hypothèse que les lois d'émission sont en effet log-normales, si l'on considère la consommation électrique (nous étudions en réalité son logarithme). D'autre part, l'interprétation des états cachés en est facilitée. En revanche, une non coïncidence n'apporte pas d'information sur la qualité du modèle, car l'algorithme de Viterbi utilise une procédure d'affectation par un principe de maximum a posteriori (MAP) qui ne respecte pas nécessairement la répartition des composants, notamment lorsqu'ils sont peu séparés.

4.5.3 Mise en correspondance des usages et des états cachés

Dans cette section, nous exposons comment tirer parti des états cachés restaurés $(\hat{s}_t)_t$ et de la connaissance des usages pour certaines courbes (en nombre $R = 444$), pour interpréter les états cachés en termes d'usages privilégiés. Dans un premier temps,

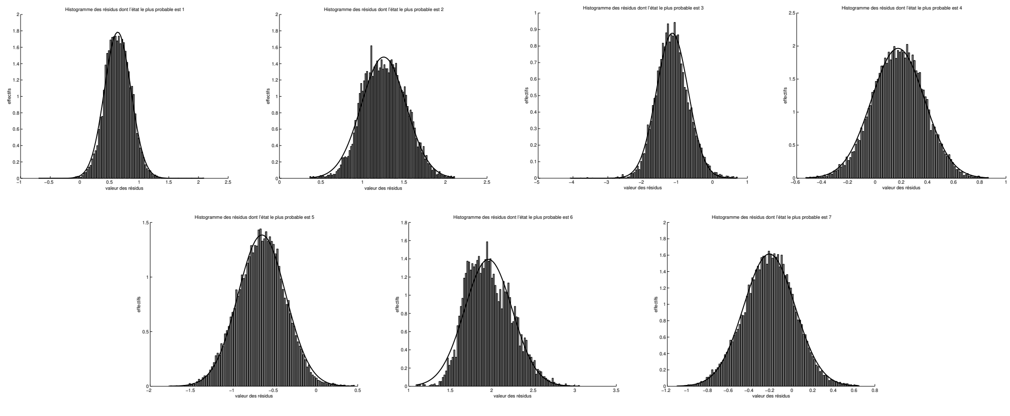


FIG. 4.5 – *Histogrammes des résidus conditionnellement à l'état caché le plus probable.*

nous utilisons les données concernant les usages pour calculer la quantité suivante, pour chaque usage u et chaque état caché j :

$$n_{ju} = \sum_{r=1}^{R} \sum_{t=1}^{144} \mathrm{conso}_u^r(t) \mathbb{1}_{\{\hat{s}_t = j\}}$$

où $\mathrm{conso}_u^r(t)$ représente la consommation à l'instant t due à l'usage u, pour la courbe de numéro r. La consommation globale, toutes courbes confondues, s'exprime donc par

$$\sum_j \sum_r \sum_t \mathrm{conso}_u^r(t) \mathbb{1}_{\{\hat{s}_t = j\}}.$$

On obtient un tableau de contingence (n_{ju}) dont les lignes j correspondent aux états et les colonnes u aux usages et pour lequel on peut calculer la statistique $\mathcal{D}_n^2$ du χ^2, mesurant l'écart à l'indépendance des douze usages et des K états. Nous normalisons cette statistique par le nombre de degrés de liberté, soit $11(K-1)$, afin de nous ramener à des variables aléatoires d'espérance unitaire. L'interprétation des états cachés par rapport aux usages passe par une analyse factorielle des correspondances.

L'inertie du nuage des $(n_{ju})_{j,u}$ a un pourcentage de représentation de 99% dans le premier plan principal de l'analyse des correspondances (figure 4.6). Tous les états et tous les usages sauf l'éclairage non halogène NH sont bien représentés dans ce plan. Les états sont situés sur une parabole sur laquelle ils sont ordonnés, suivant la valeur de la moyenne de leur loi d'émission. Il s'agit de l'*effet Guttman*, dû au fait que les états ont une signification quantitative et qu'il existe une relation non-linéaire entre les deux axes principaux (au sujet de l'effet Guttman, voir par exemple Lebart, Morineau et Piron, 1995 [80]). Les usages sont pratiquement situés sur cette parabole. Ainsi, le premier axe principal est très fortement lié à l'intensité de la consommation électrique et oppose les valeurs maximales (état C1) – associées principalement à l'eau chaude mais aussi au lave-vaisselle, lave-linge et sèche-linge – aux valeurs minimales (états C7, C6 et C5) – associées à la réfrigération. Le deuxième axe principal oppose les consommations intermédiaires – liées à l'halogène, qui a un profil atypique, et au chauffage –

aux consommations extrémales. Le quadrant "sud-ouest" correspond essentiellement aux consommations nocturnes où la réfrigération représente l'usage dominant, ainsi que la télévision quand elle est en veille.

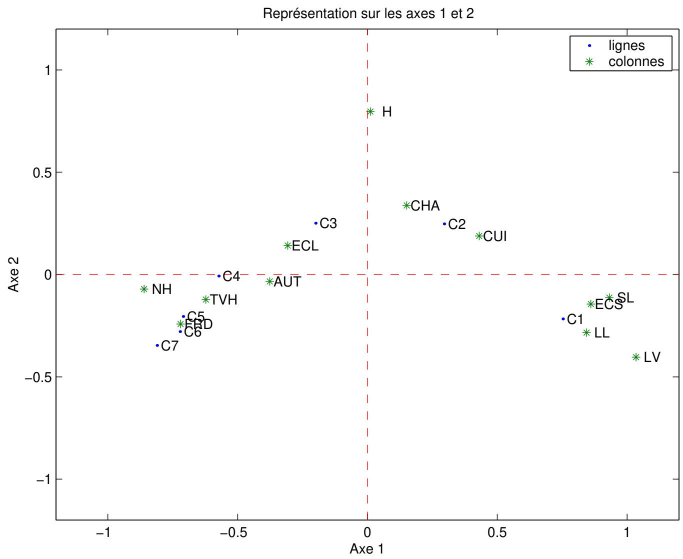


FIG. 4.6 – *Représentation des états et des usages dans le premier plan principal de l'AFC (modèle à 7 états cachés).*

4.6 Estimation de la consommation due aux usages

Nous présentons, dans cette section, une méthode permettant d'estimer la part de la consommation globale due à chaque usage, en utilisant la restauration des états cachés et le tableau de contingence mettant en correspondance les usages et les états cachés restaurés. Étant donné une courbe de la consommation électrique journalière totale $(g_t)_{1\leq t\leq 144}$ d'un ménage, éventuellement avec des relevés manquants, on souhaite estimer la consommation $(g_t^u)_{1\leq t\leq 144}$ due à chaque usage u. Pour ce faire, nous calculons les résidus $(y_t)_t$ des $(\log(g_t))_t$ par l'analyse de variance. Nous utilisons ensuite le modèle de chaîne de Markov cachée estimé pour restaurer les états cachés $\boldsymbol{s}_1^{144}$ par l'algorithme de Viterbi. Le tableau de contingence croisant usages et états cachés restaurés permet ensuite le calcul de la loi conditionnelle des usages sachant l'état caché le plus probable. Cette probabilité conditionnelle constitue une estimation de la part de la consommation globale due à l'usage u. On pose pour tout état j et tout usage u

$$f_{ju} = \frac{n_{ju}}{\sum\limits_{j'}\sum\limits_{u'} n_{j'u'}}.$$

La quantité f_{ju} est un estimateur de la probabilité jointe d'être dans l'état j pour l'usage u. La loi conditionnelle de l'usage u sachant l'état j est estimée par

$$f_{u|j} = \frac{f_{ju}}{\sum\limits_{u'} f_{u'j}}.$$

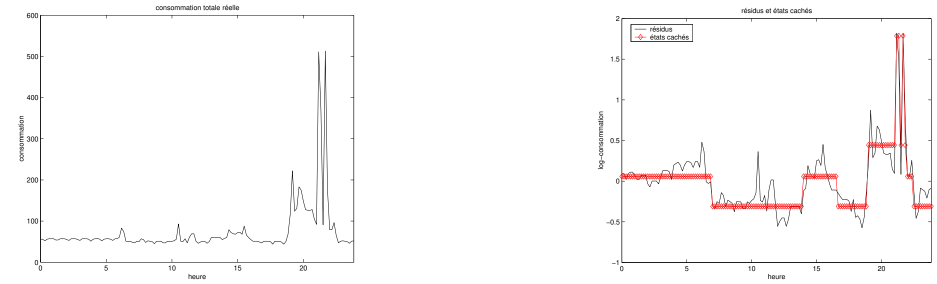
Enfin, si à l'instant t on a $\hat{s}_t = j$, alors l'estimateur de la consommation due à l'usage u est $\hat{g}_t^u = f_{u|j} g_t$. La figure 4.7 présente les courbes restaurées par cette méthode, avec, pour comparaison, la consommation réelle pour chaque usage (rappelons cependant qu'il s'agit de consommations bruitées). Les différences importantes entre ces deux dernières courbes mettent en évidence l'intérêt de prendre en compte des informations extérieures sur les habitudes de consommation des foyers.

L'information extérieure la plus facile à prendre en compte est l'absence d'un appareil électrique dans un logement. Par exemple, dans le cas de la figure 4.7, le logement ne comporte pas de lave-linge, de sèche-linge, de chauffage de l'eau et de chauffage électriques, ni d'éclairage non halogène. Ceci est pris en compte en imposant que les n_{ju} soient égaux à 0 pour toutes les valeurs de j et pour les valeurs de u correspondant aux usages absents. Les estimateurs des probabilités conditionnelles sont alors réactualisés en conséquence, en utilisant la même formule que dans la section précédente. Les trois courbes de la ligne inférieure de la figure 4.7 illustrent l'effet de la suppression des cinq usages cités ci-dessus. Il est également possible de supprimer certains usages à des heures précises de la journée et non plus forcément pour la journée entière. Ceci ne change pas le raisonnement précédent, valable pour des instants t quelconques.

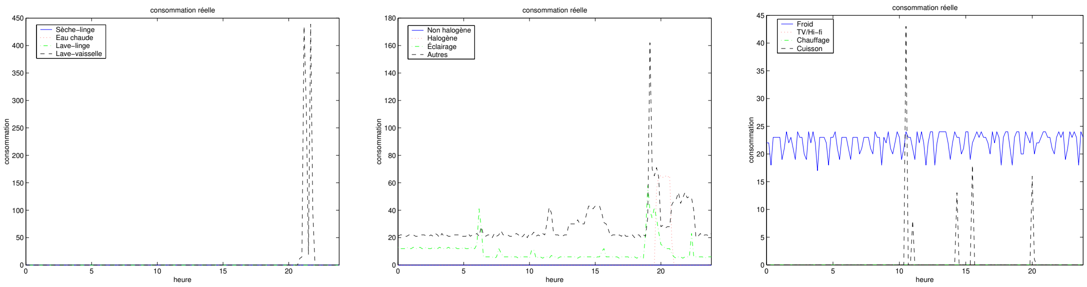
Sur la figure 4.7, la part de la consommation due à chaque usage, en particulier celle du lave-vaisselle, est mieux estimée quand on prend en compte les usages absents. Cependant, cette méthode d'estimation possède encore les inconvénients suivants : les grandeurs du tableau de contingence correspondent à des grandeurs de type consommation instantanée x temps, qui s'expriment en watt heure. Ces watt heures sont répartis globalement suivant les différents usages, mais c'est la consommation instantanée que nous cherchons à reconstituer. Par conséquent, l'estimation des usages se fait sur la base de la consommation moyenne et de manière proportionnelle à la consommation totale. Ceci ne prend pas compte le fait que la plupart des appareils qui consomment beaucoup fonctionnent brièvement (sèche-linge et lave-linge par exemple). L'estimation par notre modèle de la consommation de ces appareils conduit, au contraire, à des valeurs plus faibles (due à la répartition proportionnelle de la consommation entre les usages) mais pendant une durée plus longue que la durée réelle. D'autre part, l'usage *autres appareils* est le plus représenté. De plus, il n'est associé à aucun état spécifique. La part de la consommation affectée à cet usage est donc toujours élevée (voir par exemple le pic de la consommation estimée pour cet usage, vers 22 heures, figure 4.7).

Nous avons donc tenté de tenir compte des heures de pointe, de l'ordre de grandeur relatif de la consommation de chaque appareil et de la saisonnalité des usages, par l'utilisation d'une courbe type $(\tilde{g}_i^u)_i$ de la consommation pour chaque usage. Le choix d'un poids $p_u \in [0, 1]$ détermine l'importance donnée à la courbe type par rapport à la courbe estimée. La consommation à l'instant t due à l'usage u est estimée dans un premier temps par

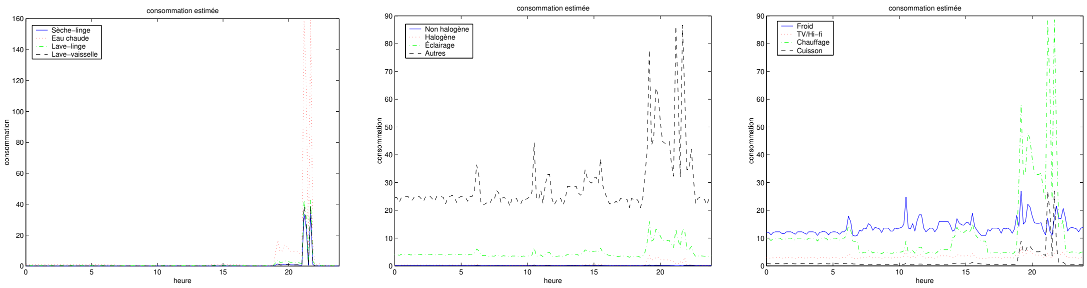
$$\hat{g}_t^u = p_u f_{u|j} g_t + (1 - p_u)\tilde{g}_t^u$$



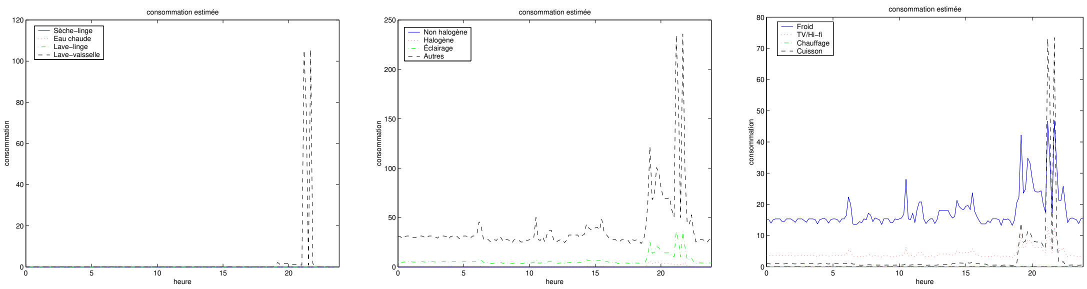
Consommation totale réelle *Résidus de l'ANOVA et états cachés restaurés.*



Consommation réelle pour chaque usage.



Consommation estimée pour chaque usage.



Estimation prenant en compte l'absence de certains appareils électriques.

FIG. 4.7 – *Estimation de la consommation due aux usages sur des données réelles bruitées.*

si $\hat{s}_t = j$. Comme on souhaite que la contrainte $\sum_u \hat{g}_t^u = g_t$ soit vérifiée, on réaffecte alors la consommation pour chaque usage de manière proportionnelle, par :

$$\hat{G}_t^u = G_t \frac{\hat{G}_t^u}{\sum_u \hat{G}_t^u}.$$

En réalité, l'application de ce principe ne nous a pas permis d'améliorer l'estimation. D'une part, ceci est dû à la difficulté d'établir des courbes types pertinentes. D'autre part, il reste le problème, évoqué plus haut, lié à l'affectation des watt heures aux différents usages alors qu'on veut estimer une consommation instantanée. Le problème de l'estimation de pics de consommation et de leur affectation à un appareil unique, plutôt qu'à un groupe d'appareils, demeure et ne peut être résolu par la modélisation proposée. Il paraît donc plus réaliste de tenter d'estimer la consommation due à un groupe d'usages similaires.

4.7 Sélection de modèles

Sélection du modèle d'analyse de variance

Le premier problème de sélection de modèles rencontré dans cette étude concerne la détermination du modèle d'analyse de variance, pour déterminer les effets contrôlés qui ont une influence sur la valeur moyenne de la log-consommation. Les modèles considérés a priori sont ceux faisant intervenir tout ou partie des facteurs suivants : *tarif*, *puissance souscrite*, *mois*, *jour* et *heure*, ainsi que toutes les interactions d'ordre un entre ces facteurs. Dans un premier temps, nous avons essayé de baser la sélection sur des tests. Cette méthode conduit à rejeter systématiquement le modèle le moins complexe. Cela est sans doute dû au nombre élevé de données (une centaine de milliers) qui rend le test très puissant, et au fait que l'hypothèse H_0 est peu réaliste vu les données (voir section 4.4). Nous avons considéré également le critère BIC, dont l'application aux modèles linéaires est réalisée dans Kass et Raftery, 1995 [69]. Le critère BIC conduit au même résultat, ce qui est lié au fait que la pénalité reste minime par rapport à la log-vraisemblance et par rapport à sa croissance suivant la complexité du modèle.

En réalité, il est suffisant, dans le cadre de cette étude, de disposer d'un modèle grossier mettant en évidence les effets les plus flagrants, vu qu'il s'agit uniquement de calculer les résidus de l'analyse de variance. Or ceux-ci sont apparus peu sensibles à la complexification du modèle, à partir d'un certain seuil. C'est pourquoi nous nous sommes contentés de sélectionner le modèle offrant un bon rapport entre la déviance expliquée et le nombre de degrés de liberté. Pour ce faire, nous avons considéré des modèles de plus en plus complexes jusqu'à ce que l'accroissement de ce rapport devienne inférieur à 0,1%. Le modèle finalement retenu met en jeu les facteurs *type de contrat* (ou tarif) c, *mois* m, *heure* h, *puissance* p et l'interaction entre *tarif* et *mois*. De plus, cette méthode nous a permis de regrouper les types de tarif "nuit" et "EJP", qui ont des effets similaires sur la moyenne de la log-consommation.

Choix du nombre d'états cachés

Dans cette étude, le principal paramètre à déterminer pour la sélection d'un modèle de chaîne de Markov cachée est son nombre d'états cachés K. L'ensemble des valeurs possibles pour K est choisi comme suit : la valeur maximale est fixée à douze, ce qui correspond au nombre d'usages disponibles. On espère pouvoir mettre en correspondance, de cette manière, les usages et les états cachés, par la méthode indiquée en section 4.5.3. La valeur minimale est fixée à sept par les experts d'EDF, au vu de l'histogramme des résidus $(y_t)_t$ (voir figure 4.4).

Nous envisageons tout d'abord un choix parmi les différents modèles, caractérisés par une valeur de K entre sept et douze, basé sur des critères de sélection du chapitre 3. Les critères considérés sont BIC, ICL et le demi-échantillonnage. Comme les différentes courbes sont supposées mutuellement indépendantes, la validation croisée utilise la moitié des séquences pour l'apprentissage et l'autre moitié pour l'évaluation du modèle. Contrairement aux expérimentations du chapitre 3, des séquences entières sont utilisées (et non pas des séquences à données supprimées au hasard).

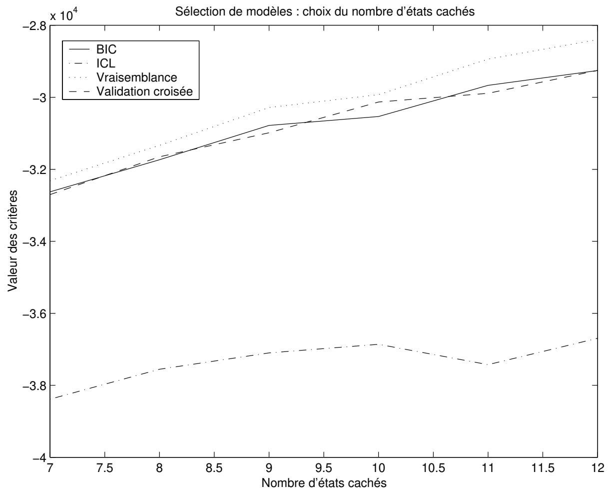


FIG. 4.8 – *Critères de choix d'un modèle pour la consommation électrique : BIC, ICL et validation croisée. La vraisemblance est tracée pour permettre la comparaison avec ces critères.*

La figure 4.8 représente les courbes pour ces critères (la vraisemblance est également tracée). Ces critères ont un comportement analogue à celui concernant la sélection d'un modèle pour l'analyse de variance, en ce qu'ils choisissent le modèle le plus complexe. Leur croissance est approximativement linéaire par rapport au nombre d'états cachés. Notons néanmoins le comportement original d'ICL par rapport aux autres critères, puisqu'il pénalise fortement un modèle à onze états pour lequel deux états se trouvent être pratiquement confondus (même moyenne, même variance). Ceci s'explique par le fait qu'ICL favorise les modèles qui aboutissent à une partition nette des données. D'autre part, les autres critères ont une croissance très similaire à celle de la log-vraisemblance. En réalité, la pénalité de BIC est négligeable par rapport à la valeur absolue de la log-vraisemblance et par rapport à sa croissance en fonction de K. De plus, le modèle de Markov caché apparaît en définitive comme un modèle peu réaliste pour les résidus de la log-consommation électrique, ce qui explique la sélection de modèles très complexes par tous les critères, d'autant que le nombre de données disponible est considérable (plus d'une centaine de milliers). Or nos objectifs d'interprétation des états, dont ne tiennent pas compte les critères ci-dessus, imposent des modèles plus parcimonieux. C'est pourquoi nous avons envisagé une méthode de sélection de modèles permettant de prendre en compte cet objectif d'interprétation des états cachés par rapport aux usages.

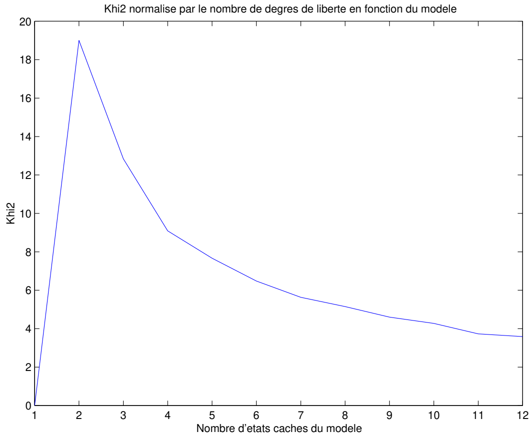


FIG. 4.9 – *La statistique du χ^2 normalisé en fonction du nombre d'états cachés.*

Rappelons que l'interprétation des états cachés se base avant tout sur le tableau de contingence mettant en relation les usages et les états cachés restaurés. Le modèle le plus facilement interprétable, du point de vue de la mise en relation des usages et des états cachés, est celui qui maximise l'écart à l'indépendance entre ces deux facteurs. Le tableau de contingence dépend de la valeur de K : nous calculons la statistique $\mathcal{D}_n^2(K)$ du χ^2 pour chacune d'entre elles. Cette statistique mesurant l'écart des données à l'indépendance vis-à-vis des douze usages et des K états, l'idée est de sélectionner le modèle (*i.e.* la valeur de K) qui maximise la statistique $\mathcal{D}_n^2(K)$ – après normalisation pour ramener son espérance à la valeur un. La figure 4.9 représente $\frac{\mathcal{D}_n^2(K)}{11(K-1)}$ en fonction de K, non seulement pour les valeurs de K entre 7 et 12 mais aussi, pour information, pour des valeurs comprises entre 1 et 6.

Un autre critère de comparaison est la *p-value* du test de H_0 contre H_1 avec

$$\begin{array}{lcl} H_0 & : & \text{indépendance entre les états et les usages} \\ H_1 & : & \text{dépendance entre les états et les usages.} \end{array}$$

En l'occurrence, cette *p-value* est $1 - F_{\chi^2_{11(K-1)}}(\mathcal{D}_n^2)$, où $F_{\chi^2_{11(K-1)}}$ est la fonction de répartition de la loi du $\chi^2_{11(K-1)}$. On obtient alors la courbe de la figure 4.10. Le modèle sélectionné est celui minimisant cette *p-value*.

Les courbes des figures 4.9 et 4.10 sont en accord et conduisent à rejeter les modèles les plus complexes. Autrement dit, l'augmentation du nombre d'états cachés rend plus difficile leur interprétation vis-à-vis des usages. Ceci est illustré par les résultats de l'analyse factorielle des correspondances pour un modèle à dix états cachés (figure 4.11), présentés à titre de comparaison avec la figure 4.6. Alors que les conclusions énoncées dans la section 4.5.3 ne sont pas modifiées, l'interprétation des états est rendue plus difficile par leur plus grand nombre.

Nous sommes conduits, en utilisant le critère $\mathcal{D}_n^2(K)$ normalisé ou la *p-value* du test d'indépendance, à accepter un modèle trivial à deux états cachés, opposant les consom-

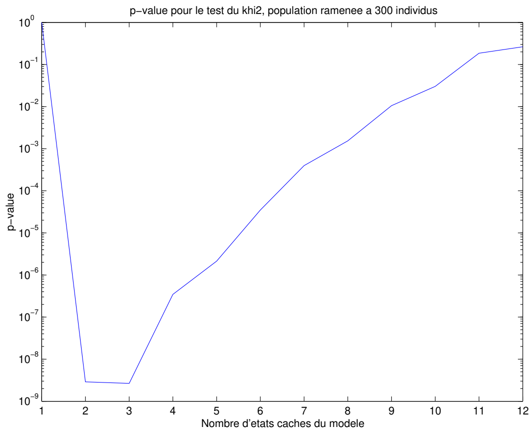


FIG. 4.10 – *p-value du test de H_0 : indépendance entre les états et les usages contre H_1 : dépendance entre ces variables, représenté sur une échelle logarithmique. Le nombre d'individus du tableau de contingence est ramené à 300.*

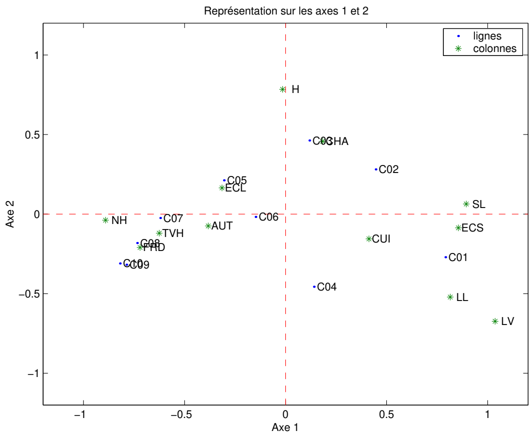


FIG. 4.11 – *Représentation des états et des usages dans le premier plan principal de l'AFC [modèle à 10 états cachés]*

mations faibles et les consommations élevées – qui est effectivement le plus facilement interprétable... Mais ces critères ne tiennent pas compte de l'adéquation entre le modèle et les données, cependant essentielle pour la validité de l'inférence et de l'analyse des courbes de consommation électrique à partir du modèle. La sélection de modèles ne peut donc se faire en utilisant ces critères seuls. Puisque les modèles considérés a priori ont de sept à douze états cachés, nous avons utilisé le modèle le plus simple parmi ceux-ci afin que l'interprétation des états cachés soit la plus facile possible, soit un modèle avec $K = 7$.

4.8 Discussion et perspectives

Voici un exemple d'application où les chaînes de Markov cachées sont un modèle peu réaliste vu les données disponibles mais qui offre un intérêt important, du point de vue de l'interprétation des états cachés en termes de niveaux de la consommation électrique ; de plus, ces niveaux sont mis en relation avec les usages disponibles par un tableau de contingence. On en déduit une méthode pour l'estimation des usages.

Sélection de modèles

Ce manque de réalisme des chaînes de Markov cachées, ainsi que le nombre considérable de données, rend impossible la sélection, par les critères classiques (comme BIC, ICL et la validation croisée), d'un modèle à la fois parcimonieux et interprétable. Clairement, la méthode retenue en définitive pour choisir un modèle n'est pas très satisfaisante : elle revient à déterminer, de manière ad hoc, un modèle réalisant un bon compromis entre l'adéquation aux données et l'interprétation. Il serait souhaitable de pouvoir quantifier ce compromis, ce qui pourrait être fait par le critère suivant. L'idée qui est en à l'origine est comparable au principe de l'utilisation de la log-vraisemblance classifiante pour la sélection de modèles (voir section 3.7).

La méthode s'applique au cas où l'on dispose d'une partition $\boldsymbol{Z}_1^n = (Z_t)_{1 \leq t \leq n}$ des variables aléatoires observées $\boldsymbol{Y}_1^n = (Y_t)_{1 \leq t \leq n}$, où Z_t est donc une covariable discrète à J modalités. L'équation

$$\ln(P_\lambda(\boldsymbol{Y} = \boldsymbol{y}, \boldsymbol{Z} = \boldsymbol{z})) = \ln(P_\lambda(\boldsymbol{Y} = \boldsymbol{y})) + \ln(P_\lambda(\boldsymbol{Z} = \boldsymbol{z} | \boldsymbol{Y} = \boldsymbol{y})),$$

permet alors de voir la probabilité $\ln(P_\lambda(\boldsymbol{Y} = \boldsymbol{y}, \boldsymbol{Z} = \boldsymbol{z}))$ comme une pénalisation de la vraisemblance par l'entropie du tenseur de classification floue a priori $(P_\lambda(\boldsymbol{Z} = \boldsymbol{z} | \boldsymbol{Y} = \boldsymbol{y}))_{\boldsymbol{z} \in \boldsymbol{Z}}$, suivant un principe analogue à celui de l'équation (3.25) mais où la partition considérée est $\boldsymbol{Z}$ plutôt que $\boldsymbol{S}$. La sélection de modèles dans un cadre bayésien se fait à partir du critère suivant, associé au modèle $\mathcal{M}$:

$$\begin{aligned} &P(\boldsymbol{Y} = \boldsymbol{y}, \boldsymbol{Z} = \boldsymbol{z} | \boldsymbol{\mathcal{M}} = \mathcal{M}) \\ &\quad = \int_{\Lambda(\mathcal{M})} P(\boldsymbol{Y} = \boldsymbol{y}, \boldsymbol{Z} = \boldsymbol{z} | \boldsymbol{\lambda} = \lambda, \boldsymbol{\mathcal{M}} = \mathcal{M}) P(\boldsymbol{\lambda} = \lambda | \boldsymbol{\mathcal{M}} = \mathcal{M}) \, d\lambda. \end{aligned}$$

Ce critère favorise donc les modèles prenant en compte les covariables $\boldsymbol{z}$ de manière pertinente, tout en pénalisant les modèles peu adéquats (qui ont une log-vraisemblance faible) où donnant lieu à une partition des données par les covariables de grande entropie.

Des approximations analogues à celles de BIC (voir section 3.5), basées sur la méthode de Laplace, conduisent à sélectionner le modèle maximisant le critère suivant :

$$\ln(P(\boldsymbol{Y} = \boldsymbol{y}, \boldsymbol{Z} = \boldsymbol{z} | \boldsymbol{\lambda} = \tilde{\lambda}, \boldsymbol{\mathcal{M}} = \mathcal{M})) - \frac{d}{2} \ln(n)$$

où d est la dimension de Λ et

$$\tilde{\lambda} = \arg\max_{\lambda} P(\boldsymbol{Y} = \boldsymbol{y}, \boldsymbol{Z} = \boldsymbol{z} | \boldsymbol{\lambda} = \lambda, \boldsymbol{\mathcal{M}} = \mathcal{M}).$$

Ce critère est assez similaire au critère BIC, dont nous rappelons l'expression :

$$\mathrm{BIC} = \ln(P(\boldsymbol{Y} = \boldsymbol{y}|\boldsymbol{\lambda} = \hat{\lambda}, \boldsymbol{\mathcal{M}} = \mathcal{M})) - \frac{d}{2}\ln(n).$$

En pratique, $\tilde{\lambda}$ est inconnu mais on dispose de $\hat{\lambda}$. Sous l'hypothèse que $\boldsymbol{Z}$ et $\boldsymbol{\lambda}$ sont indépendants et que $\tilde{\lambda} \approx \hat{\lambda}$, on a alors

$$\begin{aligned}&\ln(P(\boldsymbol{Y} = \boldsymbol{y}, \boldsymbol{Z} = \boldsymbol{z}|\boldsymbol{\lambda} = \tilde{\lambda}, \boldsymbol{\mathcal{M}} = \mathcal{M}))\\ &\quad\approx \ln(P(\boldsymbol{Y} = \boldsymbol{y}|\boldsymbol{\lambda} = \hat{\lambda}, \boldsymbol{\mathcal{M}} = \mathcal{M})) + \ln(P(\boldsymbol{Z} = \boldsymbol{z}|\boldsymbol{Y} = \boldsymbol{y}, \boldsymbol{\lambda} = \hat{\lambda}, \boldsymbol{\mathcal{M}} = \mathcal{M})).\end{aligned}$$

Nous sommes donc amenés à maximiser le critère

$$\mathrm{BIC} + \ln(P(\boldsymbol{Z} = \boldsymbol{z}|\boldsymbol{Y} = \boldsymbol{y}, \boldsymbol{\lambda} = \hat{\lambda}, \boldsymbol{\mathcal{M}} = \mathcal{M})).$$

Pour une valeur λ du paramètre, la quantité $P(\boldsymbol{Z} = \boldsymbol{z}|\boldsymbol{Y} = \boldsymbol{y}, \boldsymbol{\lambda} = \hat{\lambda})$ est calculée par

$$P(\boldsymbol{Z} = \boldsymbol{z}|\boldsymbol{Y} = \boldsymbol{y}, \boldsymbol{\lambda} = \hat{\lambda}) = \frac{P(\boldsymbol{Z} = \boldsymbol{z})P(\boldsymbol{Y} = \boldsymbol{y}|\boldsymbol{Z} = \boldsymbol{z}, \boldsymbol{\lambda} = \hat{\lambda})}{\sum_{\tilde{\boldsymbol{z}}} P(\boldsymbol{Z} = \tilde{\boldsymbol{z}})P(\boldsymbol{Y} = \boldsymbol{y}|\boldsymbol{Z} = \tilde{\boldsymbol{z}}, \boldsymbol{\lambda} = \hat{\lambda})}.$$

Cette quantité peut être approchée comme suit :

$$\ln(P(\boldsymbol{Z} = \boldsymbol{z}|\boldsymbol{Y} = \boldsymbol{y}, \boldsymbol{\lambda} = \hat{\lambda})) \approx \sum_t \ln\left[\frac{P(Z_t = z_t)P(Y_t = y_t|Z_t = z_t, \boldsymbol{\lambda} = \hat{\lambda})}{\sum_j P(Z_t = j)P(Y_t = y_t|Z_t = j, \boldsymbol{\lambda} = \hat{\lambda})}\right].$$

D'autre part, les quantités $p_j = P(Z_t = j)$ doivent être estimées. Leur estimateur empirique est

$$\hat{p}_j = \frac{\mathrm{card}(\{t|z_t = j\})}{n}.$$

De plus,

$$P(Y_t = y_t|Z_t = j, \boldsymbol{\lambda} = \hat{\lambda}) = \sum_k P(Y_t = y_t|S_t = k, Z_t = j, \boldsymbol{\lambda} = \hat{\lambda})P(S_t = k|Z_t = j).$$

Or il est naturel de faire l'hypothèse que les variables aléatoires observées et les covariables sont indépendantes conditionnellement à l'état caché. Dans ce cas,

$$P(Y_t = y_t|Z_t = j, \boldsymbol{\lambda} = \hat{\lambda}) = \sum_k P(Y_t = y_t|S_t = k, \boldsymbol{\lambda} = \hat{\lambda})P(S_t = k|Z_t = j, \boldsymbol{\lambda} = \hat{\lambda})$$

où $P(Y_t = y_t|S_t = k, \boldsymbol{\lambda} = \hat{\lambda}) = P_{\hat{\theta}_k}(y_t)$ et où $P(S_t = k|Z_t = j)$ est estimé par les fréquences empiriques

$$\frac{\mathrm{card}(\{t|z_t = j, \hat{s}_t = k\})}{\mathrm{card}(\{t|\hat{s}_t = k\})}.$$

L'application de cette méthode à notre problème de sélection de modèles passe par le choix d'une partition a priori des données observées. Des expérimentations se basant sur une partition à partir des usages sont en cours.

Extensions du modèle

L'approche présentée dans ce chapitre, consiste, dans un premier temps, à estimer et supprimer les effets fixes (tarif, mois, puissance, heure) par une analyse de variance dont les hypothèses ne sont pas vérifiées (non normalité et dépendance des résidus). Puis les résidus sont modélisés par des chaînes de Markov cachées pour prendre en compte, justement, la dépendance résiduelle (hypothèse de Markov sur la chaîne cachée) et la non normalité (hypothèse d'une loi de mélange). Ainsi les états cachés visent à modéliser les tendances non prises en compte par les effets fixes du fait que nous travaillons orthogonalement à ces effets.

D'autres approches pourraient autoriser l'analyse conjointe des effets fixes et des états cachés en se basant sur des modèles markoviens de changement de régime. Un tel modèle peut par exemple être défini comme suit :

- $(S_t)_t$ est une chaîne de Markov cachée ;
- conditionnellement à $S_t = k$, $\log(G_t)$ suit une loi normale $\mathcal{N}(\mu_{m,h,c,p,k}; \sigma_k^2)$, où par exemple $\mu_{m,h,c,p,k} = \alpha_k + \beta_k^m + \gamma_k^h + \delta_k^c + \eta_k^p + \zeta_k^{c,m}$;
- sachant $\{S_t = s_t\}_t$, les $(\log(G_t))_t$ sont conditionnellement indépendants.

Ainsi, les états cachés sont déterminés conjointement par la consommation électrique globale et par les facteurs contrôlés. On peut également envisager des modèles où les états cachés auraient une signification différente suivant le foyer (effet contrôlé ayant éventuellement une incidence sur la variance) ou suivant un type de foyer, également caché. Il s'agirait alors d'un mélange des modèles ci-dessus. Ceux-ci sont plus riches que celui que nous avons utilisé, mais plus difficiles à estimer de manière fiable et aussi à interpréter.

Chapitre 5

Conclusion générale et perspectives

Bilan

Un objectif de cette thèse était d'unifier des travaux réalisés sur l'inférence concernant les arbres de Markov cachés (voir Durand, Gonçalvès et Guédon, 2002 [46]) et sur la sélection de chaînes de Markov cachées (voir Celeux et Durand, 2002 [22]). Il nous a donc paru intéressant d'approfondir les liens entre ces modèles graphiques et de voir comment les techniques développées pour un type de graphe particulier s'appliquent à des graphes plus généraux. Ces travaux se sont orientés vers le développement de méthodes permettant d'appliquer ces graphes de Markov cachés à la modélisation stochastique de phénomènes réels. Elles concernent essentiellement les algorithmes d'inférence et la sélection de modèles.

L'application des algorithmes génériques existants, comme l'arbre de jonction de Jensen, Lauritzen et Olesen, 1990 [67], utilisés comme boîte noire pour faire du calcul numérique, se heurte à des problèmes de stabilité et d'efficacité. Cet inconvénient est lié à la difficulté de savoir ce que calculent réellement ces algorithmes, et constitue une de leur limitation fondamentale. De plus, ils ne permettent pas le calcul analytique de probabilités ou d'espérances conditionnelles – entre autres – en fonction des paramètres, et ils conduisent à une utilisation peu efficace de la boîte noire, au sens où les calculs effectués sont répétés inutilement lors d'exécutions successives. À ces algorithmes numériques, nous avons préféré l'utilisation de formules analytiques *arrière* et *avant*, qui se révèlent a posteriori plus efficaces et plus stables numériquement. Cela nous a amené à définir une famille $\mathcal{D}$ de modèles de Markov cachés à structure orientée, acyclique et morale, sans arc entre sommets observés.

La sélection de modèles est un problème pour lequel existent finalement assez peu de méthodes génériques, ces dernières étant de plus mal adaptées ou mal justifiées dans le contexte des modèles de Markov cachés. De manière générale, le meilleur critère est celui qui tient compte de l'utilisation qu'on veut faire du modèle, mais dans cette thèse orientée vers les applications, la notion de *meilleur modèle approximant la loi des variables observées* nous a paru une base à la fois générale et pertinente. Elle est le fondement des méthodes basées sur la théorie de l'information. C'est pourquoi nous avons choisi de les approfondir en nous intéressant tout particulièrement à la justification de la validation croisée et à sa mise en œuvre. Cependant, d'autres critères (comme BIC et ICL) ont donné également de bons résultats, dans les expérimentations réalisées sur des données simulées et réelles.

Enfin, il nous a paru utile de montrer les possibilités pratiques des modèles de Markov cachés en illustrant leur application à des problèmes aussi variés que le traitement du signal par les ondelettes, la fiabilité de logiciels et l'estimation de la consommation électrique d'appareils ménagers. Nous souhaitons avant tout que ces applications contribuent à donner une tournure concrète à ces modèles et qu'elles aident à leur conférer du sens. Nous espérons également que ceux qui cherchent à les appliquer à leur tour trouveront ici des pistes pour les aider.

Perspectives

Dans le chapitre dédié à l'inférence, nous avons été amenés à faire des hypothèses fortes sur la structure des modèles considérés, ce qui permet en contrepartie de développer des algorithmes génériques et interprétables. Nombreux sont les modèles couramment utilisés qui ne respectent pas ces contraintes, le premier étant le modèle autorégressif à changements markoviens, dont la structure est représentée figure 5.1. Il s'agit d'un modèle où $\boldsymbol{S}$ est une chaîne de Markov inobservée telle que sachant $\boldsymbol{S} = \boldsymbol{s}$, le processus $\boldsymbol{Y}$ est autorégressif, ses paramètres dépendant de $\boldsymbol{s}$. Dans le cas d'observations discrètes, les modèles dits M1-Mk, qui étendent le M1-M1 de la figure 5.1 à des chaînes de Markov d'ordre k contrôlées par une chaîne cachée, sont très utilisés pour la détection de zones homogènes dans les séquences d'ADN (voir Muri, 1997 [94]).

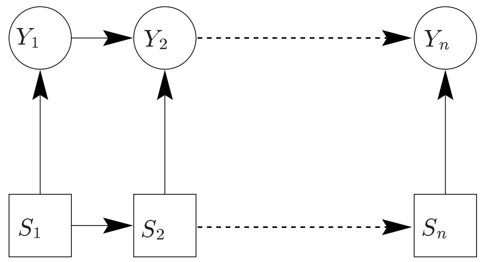


FIG. 5.1 – *La structure des modèles autorégressifs à changements markoviens.*

Un tel modèle admet des algorithmes d'inférence proches de ceux des modèles de Markov cachés de la famille $\mathcal{D}$, ce qui laisse supposer que certains de nos résultats sont applicables à une famille plus vaste. Notons tout d'abord que les algorithmes d'inférence dans des modèles à données observées Y_t supprimées (voir section 3.4.3), sont peu susceptibles d'être applicables à des modèles de Markov cachés admettant des arcs entre variables aléatoires observées. En effet, il est difficile, dans ce cas, d'intégrer la vraisemblance par rapport à y_t. D'autre part, la généricité des modèles traités conduit à une description très lourde des algorithmes, dès qu'il s'agit d'en expliquer les moindres détails. Ceci rendrait sans doute difficile la définition d'une classe contenant $\mathcal{D}$ et admettant des algorithmes d'inférence dont les nôtres seraient un cas particulier.

D'autre part, pour le calcul de probabilités dans les modèles graphiques, l'équivalence entre la propriété de séparation dans le graphe et l'indépendance conditionnelle n'est pas obligatoire. Il suffit que la séparation implique l'indépendance conditionnelle. Ceci revient à ignorer délibérément des relations d'indépendance conditionnelle, ce qui peut sembler peu souhaitable, d'autant que nous montrons, dans cette thèse, comment mettre à profit ces relations pour réaliser le calcul de probabilités de manière efficace. Mais cette technique permet de se ramener à des graphes triangulés, par ajouts d'arcs dans la structure, afin d'assurer l'existence d'un arbre de jonction, comme dans Smyth, Heckerman et Jordan, 1997 [110]. Nous montrons figure 5.2 un graphe triangulé obtenu à partir du modèle autorégressif à changements markoviens de la figure 5.1 par ajouts d'arcs à la structure d'origine afin d'assurer l'existence d'un arbre de jonction $\mathcal{T}$. Cette tech-

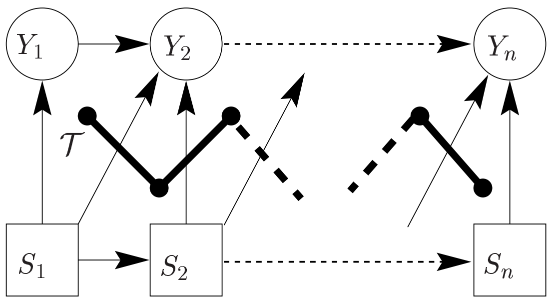


FIG. 5.2 – *Triangulation par ajout d'arcs dans la structure d'un modèle autorégressif à changements markoviens. Cette opération assure l'existence d'un arbre de jonction $\mathcal{T}$.*

nique est en théorie prometteuse, car elle permet le calcul dans des modèles graphiques quelconques. Cependant, elle se heurte, potentiellement, aux deux problèmes suivants :

- la paramétrisation naturelle du modèle n'est plus reflétée par le graphe sur lequel se base le calcul de probabilités. Il devient donc plus difficile, dans les récursions *arrière-avant*, de faire intervenir de manière explicite les véritables paramètres. Cependant, pour certaines classes particulières de modèles, dont celui de la figure 5.1, la prise en compte des paramètres naturels semble possible ;
- le graphe triangulé qui sert de base aux calculs de probabilités peut admettre des cliques d'une taille telle que la complexité des calculs soit exponentielle, comme dans les champs de Markov cachés. Il s'agit de modèles qui ne sont tout simplement pas compatibles avec des récursions *arrière-avant*.

En outre, il est intéressant d'essayer d'adapter notre démarche à une famille différente de modèles de Markov cachés, définie par d'autres contraintes sur la structure. L'étude des répercussions de ces contraintes sur les cliques, les sources et les puits du graphe pourrait permettre de déduire des algorithmes de calcul de probabilités de type *avant-arrière* et des méthodes d'estimation des paramètres basées sur EM. En effet, les propriétés sur lesquelles s'appuient nos techniques découlent de l'hypothèse de moralité de la structure, mais cette hypothèse n'est sûrement pas nécessaire. En particulier, notre démarche semble pouvoir s'adapter à des graphes vérifiant des propriétés “duales” de celles caractérisant la famille $\mathcal{D}$: existence d'un unique puits et de sources multiples au lieu d'une unique source et de multiples puits, existence d'une clique puits au lieu d'une clique source, etc. Le modèle d'arbre de Markov caché orienté vers la racine, obtenu à partir de l'arbre de Markov caché usuel de la section 1.3 en inversant le sens des arêtes entre états cachés, est un modèle d'intérêt pour lequel existent des algorithmes d'inférence identiques aux nôtres (voir Guédon, 2002 [61]). La structure de ce modèle est représentée sur la figure 5.3. Ce modèle admet en particulier un algorithme de lissage de type *ascendant-descendant* plus simple que celui des arbres de Markov cachés usuels. En effet, la phase ascendante se déroule dans le sens des arcs, ce qui rend inutile la phase préliminaire de calcul de la loi marginale des états cachés.

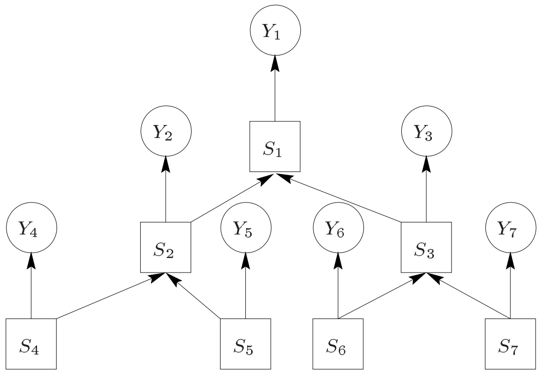


FIG. 5.3 – *Structure des arbres de Markov cachés orientés vers la racine.*

Enfin, concernant la sélection de modèles, la justification usuelle de critères classiques (comme BIC ou les tests) est mise en défaut pour le problème du choix de l'*ordre* (nombre d'états cachés). Ceci tient essentiellement à la non identifiabilité causée par un excès d'états cachés, qui entraîne l'absence de convergence de l'estimateur de maximum de vraisemblance vers une loi normale de plein rang. Il est donc tentant d'explorer les méthodes basées sur une reparamétrisation des modèles de Markov cachés pour parer à ce problème, ou de considérer des approches différentes pour justifier les critères usuels, comme dans Dacunha-Castelle et Gassiat, 1999 [34] ou Gassiat, 2002 [52].

Cependant, la plupart des travaux théoriques réalisés sur les chaînes de Markov cachées reposent sur l'hypothèse que la chaîne est ergodique, voire stationnaire. En effet, il est essentiel que tous les états soient visités par la chaîne quand la longueur de la séquence augmente, si l'on veut pouvoir estimer l'ordre de manière consistante. Or dans de nombreuses applications d'intérêt, comme la reconnaissance de parole (voir Rabiner, 1989 [102]) ou la modélisation en fiabilité de logiciels (voir section 3.8.2), des modèles ayant des états transitoires et des états absorbants ont une utilité. En modélisation statistique pour l'épidémiologie, par exemple, les états transitoires peuvent représenter les différents stades, irréversibles, d'une maladie. Il faudrait donc développer des méthodes de sélection de modèles adaptées à ces hypothèses. Ces dernières ne sont d'ailleurs pas incompatibles avec le fait que tous les états soient presque-sûrement visités au moins une fois, quand la longueur des séquences tend vers l'infini. Néanmoins, l'estimation du nombre d'états repose sur l'existence de motifs réguliers dans la matrice de transition. C'est le cas par exemple des modèles *gauche-droite*, correspondant à une matrice triangulaire supérieure, voire surdiagonale. Éventuellement, on peut étudier le comportement de méthodes de sélection de modèles quand le nombre de réalisations indépendantes d'un même modèle tend vers l'infini, ce qui est réaliste pour certaines applications.

Annexe A

Contributions logicielles : Normixem, Chainxem, Treexem et Valxem

Présentation des logiciels

Nous avons développé un ensemble de logiciels pour l'identification des modèles de chaîne et d'arbre de Markov cachés. Ces logiciels nécessitent MATLAB et reprennent l'interface du logiciel `Normixem` développé par Christophe Biernacki et renommé à présent `MIXMOD`[1], tout en étendant ses fonctionnalités. Le logiciel `Chainxem` est dédié à l'identification des chaînes de Markov cachées, `Treexem` à celle des arbres binaires de Markov cachés et `Valxem` à celle des chaînes de Markov cachés à observations manquantes.

Ces logiciels permettent le calcul de probabilités, la restauration des états cachés et l'estimation des paramètres pour les modèles ci-dessus. Il est également possible de simuler des processus. L'estimation peut se faire par l'algorithme EM, CEM, SEM et EM *à la Gibbs* ou suivant toute combinaison séquentielle de ces algorithmes. L'utilisateur a la possibilité de choisir les lois d'émission parmi les 14 modèles gaussiens de la figure 3.2 ou la loi exponentielle. Il peut également imposer des contraintes sur la matrice de transition (matrice de type diagonale par bandes) ou sur la loi de l'état initial (loi stationnaire).

Remarque A.1 *Dans le cas de modèles stationnaires, l'estimation de la matrice de transition P et de la loi stationnaire π requiert la maximisation de la fonction Q (définie par l'équation (2.5)), utilisée dans l'algorithme EM, sous la contrainte non linéaire $\pi P = \pi$. On ne connaît pas, dans ce cas, de formule explicite pour la réestimation. Nous avons choisi d'estimer P indépendamment de π puis d'estimer π par la solution de l'équation $\pi P = \pi$ ce qui conduit à des valeurs sous-optimales de la fonction Q. Cependant, vu l'oubli géométrique de la condition initiale π (voir Mevel, 1997 [92]), on s'attend à ce que l'influence de π sur la vraisemblance soit négligeable dans le cas d'une chaîne ergodique. Il est possible d'appliquer des méthodes d'optimisation usuelles, disponible par exemple en* MATLAB*, pour maximiser la fonction Q, mais il n'est pas certain que ces méthodes,*

[1]disponible à l'adresse `http ://www-math.univ-fcomte.fr/MIXMOD/index.htm`

itératives, soient capables de trouver une solution admissible pour toute valeur initiale du paramètre.

Les logiciels développés reprennent les mêmes fonctionnalités que `Normixem` pour l'estimation, à savoir :

- choix de la valeur initiale du paramètre ou détermination par un ou plusieurs tirages aléatoires ;
- choix de la condition d'arrêt : croissance relative de la log-vraisemblance, nombre d'itérations ou arrêt lorsque l'une des deux conditions est satisfaite ;
- prise en compte des états connus lorsqu'il y en a ;
- estimation des paramètres par la moyenne (éventuellement en supprimant les valeurs des premières itérations) ou par la valeur maximisant la vraisemblance quand des algorithmes stochastiques sont utilisés.

De plus, plusieurs réalisations indépendantes du modèle (éventuellement avec un nombre de données différent pour chacune) peuvent être utilisées pour identifier un même modèle.

Exemple d'utilisation

L'exemple suivant illustre l'utilisation du logiciel `Chainxem` pour la simulation de deux séquences, de longueur 500 et 600, suivant un modèle de chaîne de Markov cachée stationnaire à deux états cachés, de matrice de transition

$$P = \begin{bmatrix} 0,8 & 0,2 \\ 0,4 & 0,6 \end{bmatrix},$$

de loi stationnaire $\pi = \begin{bmatrix} 0,6667 & 0,3333 \end{bmatrix}$, de lois d'émission gaussiennes de moyennes respectives $\mu_1 = -3$ et $\mu_2 = 3$ et de variances respectives $\sigma_1^2 = 1$ et $\sigma_2^2 = 0,5$.

Les paramètres du modèle sont alors estimés par l'algorithme EM *à la Gibbs* suivi de l'algorithme EM, en utilisant simultanément les deux séquences simulées.

```
% simulation d'une chaine de Markov cachee a deux etats
>> model.fam = 'normal' ;
% lois d'emissions gaussiennes
>> par.A = [ 0.8000 0.2000 ; 0.4000 0.6000 ] ;
% matrice de transition
>> par.p = [ 0.6667 0.3333] ;
% distribution stationnaire de A
>> par.mu = [-3 ; 3] ;
% les deux parametres d'emission : moyenne des lois gaussiennes
>> par.S( :, :,1) = 1 ;
>> par.S( :, :,2) = 0.5 ;
% les deux parametres d'emission : variance des lois gaussiennes
>> [x,z] = chainxrnd([500 600], par,model) ;
>> size(x{1})
ans =
   500    1
```

```
% x{1} est une sequence simulee de longueur 500
% z{1} est la chaine de Markov correspondante
% estimation des parametres
>> model.k=2 ;
% nombre d'etats caches du modele
>> model.model='plkI' ;
% modele stationnaire, gaussien, avec une variance
% dependante de l'etat cache
>> algo = {'gibbs','em'} ;
% algorithmes utilises : EM a la Gibbs puis EM
>> cvg = 'xmlORmaxit' ;
% critere d'arret : stabilisation de la log vraisemblance ou
% depassement du nombre maximal d'iterations
>> maxit = 300 ;
% nombre maximal d'iterations pour l'algorithme EM a la Gibbs
% puis pour EM
>> parinit.A = 'random' ;
>> parinit.p = 'random' ;
>> parinit.mu = 'random' ;
>> parinit.S = 'rand. var.' ;
% initialisation aleatoire de l'algorithme
>> nbxem = 3 ;
% nombre de valeurs initiales pour l'algorithme
>> bestpar = chainxem('x',x,'model',model,'algo',algo,'cvg',cvg,...
'maxit',maxit,'nbxem',nbxem,'parinit',parinit) ;
|-------------------------|
| 2 state(s) - model plkI |
|-------------------------|
| xem 1 : [gibbs.9][em.3]
| xem 2 : [gibbs.100.147][em.3]
| xem 3 : [gibbs.5][em.3]
% estimation des parametres par l'algorithme EM a la Gibbs puis EM
% le nombre d'iterations effectuees est indiquee pour chaque execution
% de EM avec un parametre initial different
>> bestpar.A
ans =
    0.6173    0.3827
    0.1939    0.8061
% estimateur de la matrice de transition
>> bestpar.p
ans =
    0.3363    0.6637
% estimateur de la matrice de la loi stationnaire
>> bestpar.mu
ans =
```

```
    2.9874
   -3.0489
% estimateur des moyennes des parametres d'emission
>> bestpar.S
ans( :, :,1) =
    0.4586
ans( :, :,2) =
    0.9289
% estimateur des variances des parametres d'emission
```

Bibliographie

[1] Abry, P. et Veitch, D. – Wavelet Analysis of Long Range Dependant Traffic. *IEEE Transactions on Information Theory*, vol. 44, n° 1, janvier 1998, pp. 2–15.

[2] Aitkin, M. et Rubin, D.B. – Estimation and Hypothesis Testing in Finite Mixture Models. *Journal of the Royal Statistical Society (Series B)*, vol. 47, 1985, pp. 67–75.

[3] Akaike, H. – Information theory as an extension of the maximum likelihood theory. *In : Second International Symposium on Information Theory*, éd. par Petrov, B.N. et Csaki, F. Akademiai Kiado, Budapest, 1973, pp. 267–281.

[4] Baker, J.K. – The dragon system–An overview. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-23, n° 1, février 1975, pp. 24–29.

[5] Banfield, J.D. et Raftery, A.E. – Model-based Gaussian and non-Gaussian clustering. *Biometrics*, vol. 49, n° 3, 1993, pp. 803–822.

[6] Baum, L.E. et Petrie, T. – Statistical inference for probabilistic functions of finite state Markov chains. *Annals of Mathematical Statistics*, vol. 41, 1966, pp. 1554–1563.

[7] Baum, L.E., Petrie, T., Soules, G. et Weiss, N. – A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov chains. *The Annals of Mathematical Statistics*, vol. 41, n° 1, 1970, pp. 164–171.

[8] Besag, J. – Spatial interaction and the statistical analysis of lattice systems (with discussion). *Journal of the Royal Statistical Society (Series B)*, vol. 35, 1974, pp. 192–236.

[9] Besag, J. – On the statistical analysis of dirty pictures (with discussion). *Journal of the Royal Statistical Society (Series B)*, vol. 48, 1986, pp. 259–302.

[10] Bickel, P.J., Ritov, Y. et Rydén, T. – Asymptotic normality of the maximum-likelihood estimator for general hidden Markov models. *Annals of Statistics*, vol. 26, n° 4, 1998, pp. 1614–1635.

[11] Biernacki, C., Celeux, G. et Govaert, G. – Assessing a Mixture Model for Clustering with the Integrated Completed Likelihood. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, n° 7, 2001, pp. 719–725.

[12] Biernacki, Christophe. – *Choix de modèles en classification.* – Thèse de doctorat, Université de Technologie de Compiègne, septembre 1997.

[13] Bozdogan, H. – *Choosing the Number of Component Clusters in the Mixture-Model Using a New Informational Complexity Criterion of the Inverse-Fisher Information Matrix*, In *Studies in Classification, Data Analysis, and Knowledge Organization*,

pp. 40–54. – O. Opitz, B.Lausen, and R.Klar (eds.), Springer-Verlag, Heidelberg, 1993.

[14] Braffort, Annelies. – *Reconnaissance et compréhension de gestes, application à la langue des signes.* – Thèse de doctorat, Université Paris-XI, juin 1996.

[15] Buneman, P. – A characterization of rigid circuit graphs. *Discrete Mathematics*, vol. 9, 1974, pp. 205–212.

[16] Bunke, O. et Milhaud, X. – Asymptotic behavior of Bayes estimates under possibly incorrect models. *The Annals of Statistics*, vol. 26, n° 2, 1998, pp. 617–644.

[17] Burnham, K.P. et Anderson, D.R. – *Model Selection and Inference.* – Springer-Verlag, 1998.

[18] Celeux, G. et Clairambault, J. – Analyse Discriminante appliquée à l'étude du rythme cardiaque : développements méthodologiques. *La Revue de Modulad*, vol. 8, 1991, pp. 73–80.

[19] Celeux, G. et Clairambault, J. – Estimation de chaînes de Markov cachées : méthodes et problèmes. *In : Actes des journées thématiques Approches markoviennes en signal et images. GDR signal-images CNRS.*, 1992, pp. 5–20.

[20] Celeux, G. et Diebolt, J. – The SEM algorithm : a probabilistic teacher algorithm derived from the EM algorithm for the mixture problem. *Computational Statistics Quaterly*, vol. 2, 1985, pp. 73–82.

[21] Celeux, G. et Diebolt, J. – Une version de type recuit simulé de l'algorithme EM. *Notes aux Comptes Rendus de l'Académie des Sciences de Paris, Série I*, vol. 310, 1990, pp. 119–124. – Également en rapport de recherche de l'INRIA, N. 1123, 1989.

[22] Celeux, G. et Durand, J.-B. – Choosing the order of a hidden Markov chain through cross-validated likelihood. *In : Compstat2002. Berlin (Allemagne)*, 24-28 août 2002.

[23] Celeux, G. et Govaert, G. – A classification EM algorithm for clustering and two stochastic versions. *Computational Statistics and Data Analysis*, vol. 14, n° 3, 1992, pp. 315–332.

[24] Celeux, G. et Govaert, G. – Gaussian Parcimonious Models. *Pattern Recognition*, vol. 28, n° 5, 1995, pp. 781–793.

[25] Celeux, G. et Soromenho, G. – An entropy criterion for assessing the number of clusters in a mixture model. *Classification Journal*, vol. 13, 1996, pp. 195–212.

[26] Chandler, D. – *Introduction to Modern Statistical Mechanics.* – Oxford University Press, 1987.

[27] Chang, R.W. et Hancock, J.C. – On receiver structures for channels having memory. *IEEE Transactions on Information Theory*, vol. IT-12, n° 4, octobre 1966, pp. 263–268.

[28] Chen, Y. et Singpurwalla, N. – Unification of software reliability models by self-exciting point processes. *Advances in Applied Probability*, vol. 29, 1997, pp. 337–352.

[29] Churchill, G.A. – Stochastic Models for Heterogeneous DNA Sequences. *Bulletin of Mathematical Biology*, vol. 51, 1989, pp. 79–94.

[30] Cover, T.M. et Thomas, J.A. – *Elements of Information Theory.* – John Wiley and Sons, Inc., New-York, 1991.

[31] Cowell, R.G., Dawid, A.P., Lauritzen, S.L. et Spiegelhalter, D.J. – *Probabilistic Networks and Expert Systems.* – Springer-Verlag, New York, 1999.

[32] Crouse, M.S., Nowak, R.D. et Baraniuk, R.G. – Wavelet-Based Statistical Signal Processing Using Hidden Markov Models. *IEEE Transactions on Signal Processing*, vol. 46, n° 4, 1998, pp. 886–902.

[33] Cutler, A. et Windham, M.P. – Information-Based Validity Functionals for Mixture Analysis. *In : Proceedings of the first US-Japan Conference on the Frontiers of Statistical Modelling*, éd. par Kluwer (Academic Publishers). Amsterdam, 1993, pp. 149–170.

[34] Dacunha-Castelle, D. et Gassiat, E. – Testing the order of a model using locally conic parametrization : population mixtures and stationary ARMA processes. *Annals of Statistics*, vol. 27, n° 4, 1999, pp. 1178–1209.

[35] Darroch, J.N., Lauritzen, S.L. et Speed, T.P. – Markov fields and log-linear interaction models for contingency tables. *Annals of Statistics*, vol. 8, 1980, pp. 598–617.

[36] Dempster, A.P., Laird, N.M. et Rubin, D.B. – Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society Series B*, vol. 39, 1977, pp. 1–38.

[37] Devijver, P. A. – Baum's forward-backward Algorithm Revisited. *Pattern Recognition Letters*, vol. 3, 1985, pp. 369–373.

[38] Diebolt, J. et Celeux, G. – Asymptotic Properties of a Stochastic EM Algorithm for Estimating Mixing Proportions. *Stochastics Models*, vol. 9, 1993, pp. 599–613.

[39] Diebolt, J. et Robert, C.P. – Estimation of Finite Mixture Distributions Through Bayesian Sampling. *Journal of the Royal Statistical Society (Series B)*, vol. 56, 1994, pp. 363–375.

[40] Diligenti, M., Frasconi, P. et Gori, M. – Image Document Categorization using Hidden Tree Markov Models and Structured Representations. *In : Second International Conference on Advances in Pattern Recognition. Lecture Notes in Computer Science*, éd. par Singh, S., Murshed, N. et Kropatsch, W., 2001.

[41] Douc, R. et Matias, C. – Asymptotics of the maximum-likelihood estimator for general hidden Markov models. *Bernoulli*, vol. 7, n° 3, 2001, pp. 381–420.

[42] Doukhan, P., Massart, P. et Rio, E. – Invariance principle for absolutely regular empirical processes. *Annales de l'institut Poincaré*, vol. 31, n° 2, 1995, pp. 393–427.

[43] Duane, J.T. – Learning curve approach to reliability monitoring. *IEEE Transactions on Aerospace*, vol. AS-2, n° 2, 1964, pp. 563–566.

[44] Durand, J.-B. – Choisir l'ordre d'une chaîne de Markov cachée par half-sampling. *In : XXXIIIèmes Journées de Statistique organisées par l'ENITIAA. Nantes (France)*, 14-18 mai 2001, pp. 325–329.

[45] Durand, J.-B. et Gaudoin, O. – Software reliability modelling and prediction with hidden Markov chains. – 2003. Soumis au *Journal of the Royal Statistical Society (Series C)*. Également en rapport de recherche de l'INRIA, N. 4747, 2003.

[46] Durand, J.-B., Gonçalvès, P. et Guédon, Y. – Statistical Inference for Hidden Markov Tree Models and Application to Wavelet Trees. – 2002. Soumis à *IEEE Transactions on Signal Processing*. Également en rapport de recherche de l'INRIA, N. 4248, 2001.

[47] Ephraim, Y. et Merhav, N. – Hidden Markov processes. *IEEE Transactions on Information Theory*, vol. 48, juin 2002, pp. 1518–1569.

[48] Flandrin, P. – Wavelet Analysis and Synthesis of Fractional Brownian Motion. *IEEE Transactions on Information Theory*, vol. 38, 1992, pp. 910–917.

[49] Forney Jr., G.D. – The Viterbi Algorithm. *In : Proceedings of the IEEE*, mars 1973, pp. 268–278.

[50] Fredkin, D.R. et Rice, J.A. – Bayesian Restoration of Single-Channel Patch Clamp Recordings. *Biometrics*, vol. 48, 1992, pp. 427–448.

[51] Garel, B. – Le choix du nombre de composantes dans un mélange. *In : XXXIVèmes Journées de Statistique organisées par l'Université libre de Bruxelles et l'Université catholique de Louvain. Bruxelles – Louvain-la-Neuve (Belgique)*, 13-17 mai 2002, p. 228.

[52] Gassiat, E. – Likelihood ratio inequalities with application to various mixtures. – 2002. Annales de l'institut Poincaré (À paraître).

[53] Gassiat, E. et Kéribin, C. – The likelihood ratio test for the number of components in a mixture with Markov regime. *ESAIM P & S*, vol. 4, 2000, pp. 25–52.

[54] Gaudoin, O. – *Outils statistiques pour l'évaluation de la fiabilité des logiciels.* – Thèse de doctorat, Joseph Fourier University, Grenoble, juin 1990.

[55] Gaudoin, O. – Software reliability models with two debugging rates. *International Journal of Reliability, Quality and Safety Engineering*, vol. 6, n° 1, 1999, pp. 31–42.

[56] Gaudoin, O., Lavergne, C. et Soler, J.L. – A generalized geometric de-eutrophication software-reliability model. *IEEE Transactions on Reliability*, vol. R-44, n° 4, 1994, pp. 536–541.

[57] Gavril, F. – The intersection graphs of subtrees in trees are exactly the chordal graphs. *Journal of Combinatorial Theory, Series B*, vol. 16, 1974, pp. 47–56.

[58] Geiger, D., Heckerman, D., King, H. et Meek, C. – Stratified exponential families : Graphical models and model selection. *The Annals of Statistics*, vol. 29, n° 2, 2001, pp. 505–529.

[59] Gibbs, W. – *Elementary principles of statistical mechanics.* – Yale University Press, New Haven, Connecticut, 1902.

[60] Goel, A.L. et Okumoto, K. – Time dependent error detection rate model for software reliability and other performance measures. *IEEE Transactions on Reliability*, vol. R-28, n° 1, 1979, pp. 206–211.

[61] Guédon, Y. – Upward-downward algorithm for a directed in-tree. – 2002. (Non publié).

[62] Ito, H., Amari, S.-I. et Kobayashi, K. – Identifiability of hidden Markov information sources and their minimum degrees of freedom. *IEEE Transactions on Information Theory*, vol. 38, 1992, pp. 324–333.

[63] Jaffard, S. – Pointwise Smoothness, two-microlocalization and Wavelet Coefficients. *Publications Mathématiques*, vol. 35, 1991, pp. 155–168.

[64] Jelinek, F. – *Statistical Methods for Speech Recognition.* – MIT Press, 1997.

[65] Jelinek, F., Bahl, L.R. et Mercer, R.L. – Design of a linguistic statistical decoder for the recognition of continuous speech. *IEEE Transactions on Information Theory*, vol. IT-21, 1975, pp. 250–256.

[66] Jelinski, Z. et Moranda, P.B. – *Statistical computer performance evaluation*, In *Software reliability research*, pp. 465–497. – W. Freiberger ed, Academic press, New-York, 1972.

[67] Jensen, F.V., Lauritzen, S.L. et Olesen, K.G. – Bayesian updating in causal probabilistic networks by local computations. *Computational Statistical Quaterly*, vol. 5, n° 4, 1990, pp. 269–282.

[68] Jensen, J.L. et Petersen, N.V. – Asymptotic normality of the maximum likelihood estimator in state space models. *Annals of Statistics*, vol. 27, 1999, pp. 514–535.

[69] Kass, R.E. et Raftery, A.E. – Bayes factors. *Journal of the American Statistical Association*, vol. 90, n° 430, 1995, pp. 773–795.

[70] Kass, R.E., Tierney, L. et Kadane, J.B. – *The Validity of Posterior Asymptotic Expansions Based of Laplace's Method*, In *Bayesian Likelihood Methods in Statistics and Econometrics*, pp. 473–488. – S. Geisser, J.S. Hodges, J.S. Press, and A. Zellner (eds.), New York : North-Holland, 1990.

[71] Kass, R.E. et Wasserman, L. – A Reference Bayesian Test for Nested Hypothesis and its Relationship to the Schwarz Criterion. *Journal of the American Statistical Association*, vol. 90, n° 431, 1995, pp. 928–934.

[72] Keiller, P.A., Littlewood, B., Miller, D.R. et Sofer, A. – Comparison of software reliability predictions. *In : Proceedings of the 13th IEEE International Symposium on Fault Tolerant Computing*, 1983, pp. 128–134.

[73] Koehler, A.B. et Murphee, E.H. – A comparison of the Akaike and Schwarz criteria for selecting model order. *Applied Statistics*, vol. 37, 1988, pp. 187–195.

[74] Künsch, H. R. – *State Space and Hidden Markov Models*, In *Complex Stochastic Systems*, chap. 3, pp. 109–173. – O.E. Barndorff-Nielsen, D.R. Cox and C. Klüppelberg (eds.). Chapman & Hall/CRC Press, Boca raston, 2001.

[75] Lauritzen, S.L. – *Graphical Models.* – Clarendon Press, Oxford, United Kingdom, 1996.

[76] Lauritzen, S.L. – Causal inference from graphical models. – page web `http ://citeseer.nj.nec.com/article/lauritzen99causal.html`, 1999.

[77] Lauritzen, S.L., Dawid, A.P., Larsen, B.N. et Leimer, H.G. – Independence properties of directed Markov fields. *Networks*, vol. 20, 1990, pp. 491–505.

[78] Lauritzen, S.L. et Spiegelhalter, D.J. – Local Computations with Probabilities on Graphical Structures and their Application to Expert Systems. *Journal of the Royal Statistical Society (Series B)*, vol. 50, n° 2, 1988, pp. 157–224.

[79] Le Gland, F. et Mevel, L. – Exponential forgetting and geometric ergodicity in hidden Markov models. *Mathematics of Control, Signals, and Systems*, vol. 13, 2000, pp. 63–93.

[80] Lebart, L., Morineau, A. et Piron, M. – *Statistique exploratoire multidimensionnelle.* – Dunod, 1995.

[81] Leroux, B. – Maximum-likelihood estimation for hidden Markov models. *Stochastic Processes and their Applications*, vol. 40, 1992, pp. 127–143.

[82] Leroux, B.G. et Puterman, M.L. – Maximum-penalized likelihood estimation for independant and Markov dependent mixture models. *Biometrics*, vol. 48, 1992, pp. 545–558.

[83] Levinson, S.E., Rabiner, L.R. et Sondhi, M.M. – An Introduction to the Application of the Theory of Probabilistic Functions of a Markov Process in Automatic Speech Recognition. *Bell System Technical Journal*, vol. 62, 1983, pp. 1035–1074.

[84] Linhart, H. et Zucchini, W. – *Model Selection.* – John Wiley and Sons, 1986.

[85] Littlewood, B. et Verral, J. – A Bayesian reliability growth model for computer software. *Journal of the Royal Statistical Society (Series C)*, vol. 22, 1973, pp. 332–336.

[86] Lo, Y., Mendell, N.R. et Rubin, D.B. – Testing the number of components in a normal mixture. *Biometrika*, vol. 88, n° 3, 2001, pp. 767–778.

[87] Lucke, H. – Which Stochastic Models Allow Baum-Welch Training ? *IEEE Transactions on Signal Processing*, vol. 44, n° 11, novembre 1996, pp. 2746–2756.

[88] Lyu, M.R. (édité par). – *Handbook of software reliability engineering.* – IEEE Computer Society Press and Mc Graw-Hill Book Company, 1996.

[89] Mallat, S. – *A Wavelet Tour of Signal Processing.* – San Diego, California : Academic Press. xxiv, 1998.

[90] McLachlan, G.J. et Peel, D. – *On a Resampling Approach to Choosing the Number of Components in Normal Mixture Models.* – Research report n° 58, Brisbane : Centre for Statistics, the University of Queensland, juin 1996.

[91] McLachlan, G.J. et Peel, D. – *Finite Mixture Models.* – John Wiley and Sons, 2000, *Wiley Series in Probability and Statistics.*

[92] Mevel, Laurent. – *Statistique asymptotique pour les modèles de Markov cachés.* – Thèse de doctorat, Université de Rennes 1, novembre 1997.

[93] Moranda, P.B. – Event altered rate models for general reliability analysis. *IEEE Transactions on Reliability*, vol. R-38, n° 5, 1979, pp. 376–381.

[94] Muri, Florence. – *Comparaison d'algorithmes d'identification de chaînes de Markov cachées et application à la détection de régions homogènes dans les séquences d'ADN.* – Thèse de doctorat, Université Paris-V, octobre 1997.

[95] Musa, J.D. – *Software reliability data.* – Rapport technique, Rome Air Development Center, 1979.

[96] Neal, R.M. et Hinton, G.E. – A View of the EM Algorithm that Justifies Incremental, Sparse and Other Variants. *In : Learning in Graphical Models*, chap. Foundations for Learning. – Dordecht, Kluwer (Academic Publishers), 1988.

[97] Pearl, J. – Reverend Bayes on Inference Engines : A Distributed Hierarchical Approach. *In : AAAI Conference on Artificial Intelligence*, 1982, pp. 133–136.

[98] Pearl, J., Geiger, D. et Verma, T. – *Influence Diagrams, Belief Nets, and Decision Analysis*, In *The logic of influence diagrams*, pp. 67–83. – Oliver, R.M. and Smith, J.Q.(eds.). Chichester, U.K. John Wiley and Sons, 1990.

[99] Peltier, R. et Levy-Vehel, J. – *Multifractional Brownian Motion : Definition and Preliminary Results.* – Rapport technique n° RR-2645, INRIA, 1995. Submitted to Stochastic Processes and their Applications.

[100] Pham, H. – *Software reliability.* – Springer 2000, 2000.

[101] Qian, W. et Titterington, D.M. – Estimation of parameters in hidden Markov models. *Philosophical transactions of the Royal Society of London (Series A)*, vol. 337, 1991, pp. 407–428.

[102] Rabiner, L.R. – A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *In : Proceedings of the IEEE*, février 1989, pp. 257–286.

[103] Redner, R.A. et Walker, H.F. – Mixture densities, maximum likelihood and the EM algorithm. *SIAM Review*, vol. 26, n° 2, 1984, pp. 195–239.

[104] Ripley, B.D. – *Pattern Recognition and Neural Networks.* – Cambridge University Press, janvier 1996.

[105] Robert, C.P. – *L'Analyse statistique bayésienne.* – Paris, Economica, 1992.

[106] Robert, C.P., Celeux, G. et Diebolt, J. – Bayesian estimation of hidden Markov chains : A stochastic implementation. *Statistics and Probability Letters*, vol. 16, n° 1, 1993, pp. 77–83.

[107] Schwarz, G. – Estimating the dimension of a model. *The Annals of Statistics*, vol. 6, 1978, pp. 461–464.

[108] Scott, S.L. – Bayesian Methods for Hidden Markov Models : Recursive Computing in the 21st Century. *Journal of the American Statistical Association*, vol. 97, n° 457, 2002, pp. 337–351.

[109] Shore, J.E. et Johnson, R.W. – Axiomatic derivation of the principle of maximum entropy and the principle of minimum cross-entropy. *IEEE Transactions on Information Theory*, vol. IT-26, n° 1, janvier 1980, pp. 26–37. – cf. commentaires et corrections, *IEEE Transactions on Information Theory*, vol. IT-29, n° 6, novembre 1983, pp. 942–943.

[110] Smyth, P., Heckerman, D. et Jordan, M.I. – Probabilistic Independence Networks for Hidden Markov Probability Models. *Neural Computation*, vol. 9, n° 2, 1997, pp. 227–270.

[111] Soler, J.L. – Modélisation des processus de risque, de défaillance et de correction des systèmes présentant des fautes de conception – application à la fiabilité des logiciels. *In : Proceedings of 6th National Conference On Reliability and Maintainability, Strasbourg*, 1988, pp. 647–650.

[112] Spiegelhalter, D.J., Best, N.G., Gilks, W.R. et Inskip, H. – *Hepatitis B : a case study in MCMC methods*, In *Markov Chain Monte Carlo in Practise*, pp. 21–43. – Chapman & Hall, 1996.

[113] Stone, M. – Cross-validatory choice and assessment of statistical predictions, with discussion. *Journal of the Royal Statistical Society (Series B)*, vol. 36, 1974, pp. 111–147.

[114] Stone, M. – An Asymptotic Equivalence of Choice of Model by Cross-validation and Akaike's Criterion. *Journal of Royal Statistical Society (Series B)*, vol. 39, 1977, pp. 44–47.

[115] Takeuchi, K. – Distribution of informational statistics and a criterion of model fitting. *Suri-Kagaku* (Sciences mathématiques), vol. 153, 1976, pp. 12–18. – (en japonais).

[116] Teicher, H. – Identifiability of mixtures of product measures. *The Annals of Mathematical Statistics*, vol. 38, 1967, pp. 1300–1302.

[117] Viterbi, A.J. – Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, vol. 13, 1967, pp. 260–269.

[118] Walter, J. – *Representations of rigid cycle graphs.* – Thèse de doctorat, Wayne State University, Detroit, MI, 1972.

[119] White, H. – Maximum likelihood estimation of misspecified models. *Econometrica*, vol. 50, n° 1, janvier 1982, pp. 1–25.

[120] Whittaker, J. – *Graphical Models in Applied Multivariate Statistics.* – Chichester, U.K. John Wiley and Sons, 1990.

[121] Wolfe, J.H. – Pattern Clustering by Multivariate Mixture Analysis. *Multivariate Behavioral Research*, vol. 5, 1970, pp. 329–350.

[122] Wornell, G.W. et Oppenheim, A.V. – Estimation of Fractal Signals from Noisy Measurements Using Wavelets. *IEEE Transactions on Signal Processing*, vol. 40, n° 3, mars 1992, pp. 611–623.

[123] Wright, S. – Correlation and causation. *Journal of Agricultural Research*, vol. 20, 1921, pp. 557–585.

[124] Wu, C.J.Jeff. – On the convergence properties of the EM algorithm. *The Annals of Statistics*, vol. 11, 1983, pp. 95–103.

[125] Yamada, S., Ohba, M. et Osaki, S. – S-shaped reliability growth modelling for software error detection. *IEEE Transactions on Reliability*, vol. R-35, n° 5, 1983, pp. 475–478.

[126] Zhang, P. – Model selection via multifold cross validation. *The Annals of Statistics*, vol. 21, n° 1, 1993, pp. 299–313.

TITLE :
Latent structure models : Inference, model selection and applications.

ABSTRACT :
In this study, we present some inference algorithms and selection methods for the analysis of hidden Markov models. Properties of their structure are derived and lead us to define a family of models. These ones can be easily parameterized and interpreted. For these models, we propose inference algorithms based on backward-forward-like recursions which are efficient, numerically stable and which allow analytic calculus. Then, we investigate various order selection methods, among which the multifold cross validation, BIC, AIC and some criteria based on the marginal likelihood penalization. The existence of dependencies between variables complicates the implementation of half-sampling technics and leads to appropriate algorithms. These selection methods are compared through experimentations on simulated and on real data sets, the latter being related to software reliability. The importance of the hidden Markov chains and trees is also illustrated by applications in signal processing.

KEYWORDS :
hidden Markov models, EM algorithm, smoothing algorithm, *backward-forward* recursion, junction tree, model selection, penalized likelihood, cross-validation.

TITRE :
Modèles à structure cachée : inférence, sélection de modèles et applications.

RÉSUMÉ :
L'objet de cette thèse est l'étude d'algorithmes d'inférence et de méthodes de sélection pour les modèles de Markov cachés. L'analyse de propriétés du graphe d'indépendance conditionnelle aboutit à la définition d'une famille de modèles aisément paramétrables et interprétables. Pour ces modèles, nous proposons des algorithmes d'inférence basés sur des récursions de type *arrière-avant* efficaces, numériquement stables et permettant des calculs analytiques. Puis nous étudions différentes méthodes de sélection du nombre d'états cachés, dont le demi-échantillonnage, les critères BIC, AIC, ICL, et la pénalisation de la vraisemblance marginale. L'implémentation de la validation croisée, problématique dans le cas de dépendances entre variables, fait l'objet de développements particuliers. Ces méthodes sont comparées par des expérimentations sur des données simulées puis réelles (fiabilité de logiciels). Nous illustrons l'intérêt des arbres et chaînes de Markov cachés en traitement du signal.

MOTS-CLÉS :
modèles de Markov cachés, algorithme EM, algorithmes de lissage, récursion *arrière-avant*, arbre de jonction, sélection de modèles, vraisemblance pénalisée, validation croisée.

DISCIPLINE : Mathématiques Appliquées

Thèse réalisée au sein du projet IS2 de l'Inria Rhône-Alpes.