



École Doctorale SMPC

Thèse présentée pour l'obtention du diplôme de
Docteur de Télécom & Management SudParis

**Doctorat conjoint Télécom & Management SudParis
et Université Pierre et Marie Curie**

Spécialité : **MATHÉMATIQUES APPLIQUÉES**

Par :

Mohamed BEN MABROUK

Sujet de la thèse:

Modèles de Markov triplets en restauration des signaux

Soutenue le 26 Avril 2011 devant le jury composé de :

Rapporteurs :

**François ROUEFF
Nikolaos LIMNIOS**

**Professeur TELECOM ParisTech.
Professeur Université de Technologie de Compiègne.**

Examineurs :

**Jean-Yves TOURNERET
Michel BRONIATOWSKI
Pascal LARZABAL**

**Professeur INP - ENSEEIHT Toulouse.
Professeur Université Pierre et Marie Curie.
Professeur École Normale Supérieure de Cachan.**

Directeur de thèse :

Wojciech PIECZYNSKI Professeur Télécom Sud-Paris.

Résumé

La restauration statistique non-supervisée de signaux admet d'innombrables applications dans les domaines les plus divers comme économie, santé, traitement du signal, météorologie, finances, biologie, fiabilité, transports, environnement, ... Un des problèmes de base, qui est au cœur de cette thèse, est d'estimer une séquence cachée $(x_n)_{1:N}$ à partir d'une séquence observée $(y_n)_{1:N}$. En traitements probabilistes du problème ces séquences sont considérées comme réalisations, respectivement, des processus $X = (X_n)_{1:N}$ et $Y = (Y_n)_{1:N}$. Plusieurs techniques fondées sur des méthodes statistiques ont été développées pour résoudre ce problème. Le modèle parmi le plus répandu pour le traiter est le modèle dit « modèle de Markov caché » (MMC). Dans ce modèle nous supposons que le processus caché X est markovien et les lois $p(y|x)$ de Y conditionnelles à X sont suffisamment simples pour que la loi $p(x|y)$ soit également markovienne, cette dernière propriété étant nécessaire pour les traitements. Plusieurs extensions de ces modèles ont été proposées depuis 2000. Dans les modèles de Markov couples (MMCouples), plus généraux que les MMC, le couple (X, Y) est markovien, ce qui implique que $p(x|y)$ est également markovienne (alors que $p(x)$ ne l'est plus nécessairement), ce qui permet les mêmes traitements que dans les MMC. Plus récemment (2002) les MMCouples ont été étendus aux « modèles de Markov triplet » (MMT), dans lesquels on introduit un processus auxiliaire U et suppose que le triplet $T = (X, U, Y)$ est markovien. Là encore il est possible, dans un cadre plus général que celui des MMCouples, d'effectuer des traitements avec une complexité raisonnable.

L'objectif de cette thèse est de proposer des nouvelles modélisations faisant partie des MMT et d'étudier leur pertinence et leur intérêt. Nous proposons deux types de nouveautés :

(i) Lorsque la chaîne cachée est discrète et lorsque le couple (X, Y) n'est pas stationnaire, avec un nombre fini de « sauts » aléatoires dans les paramètres, l'utilisation récente des MMT dans lesquels les sauts sont modélisés par un processus discret U a donné des résultats très convaincants (Lanchantin, 2006). Notre première idée est d'utiliser cette démarche avec un processus U continu, qui modéliserait des non-stationnarités "continues" de (X, Y) . Nous proposons des chaînes et des champs triplets et présentons quelques expériences. Les résultats obtenus dans la modélisation de la non-stationnarité continue semblent moins intéressants que dans le cas discret. Cependant, les nouveaux modèles peuvent présenter d'autres intérêts ; en particulier, ils semblent plus efficaces que les modèles « chaînes de Markov cachées » classiques lorsque le bruit est corrélé ;

(ii) Soit un MMT $T = (X, U, Y)$ tel que X et Y sont continu et U est discret fini. Nous sommes en présence du problème de filtrage, ou du lissage, avec des sauts

aléatoires. Dans les modélisations classiques le couple caché (X, U) est markovien mais le couple (U, Y) ne l'est pas, ce qui est à l'origine de l'impossibilité des calculs exacts avec une complexité linéaire en temps. Il est alors nécessaire de faire appel à diverses méthodes approximatives, dont celles utilisant le filtrage particulière sont parmi les plus utilisées. Dans des modèles MMT récents le couple caché (X, U) n'est pas nécessairement markovien, mais le couple (U, Y) l'est, ce qui permet des traitements exacts avec une complexité raisonnable (Pieczynski 2009). Notre deuxième idée est d'étendre ces derniers modèles aux triplets $T = (X, U, Y)$ dans lesquels les couples (U, Y) sont "partiellement" de Markov. Un tel couple (U, Y) n'est pas de Markov mais U est de Markov conditionnellement à Y . Nous obtenons un modèle $T = (X, U, Y)$ plus général, qui n'est plus de Markov, dans lequel le filtrage et le lissage exacts sont possibles avec une complexité linéaire en temps. Quelques premières simulations montrent l'intérêt des nouvelles modélisations en lissage en présence des sauts.

Mot-clefs

Chaînes de Markov cachés, chaînes de Markov couples et triplets, chaînes partiellement de Markov, chaîne caché à saut markovien, dépendance longue, champ de Gaussien-Markov caché, modèle Logit, inférence bayésienne, filtrage, lissage, Espérance Maximisation, Estimation Conditionnelle Itérative.

Abstract

Restoration unsupervised statistical signal admits countless applications in fields as diverse economy, health, signal processing, meteorology, finance, biology, reliability, transport, environment, ... One problem base, which is central to this thesis is to estimate a sequence hidden $(x_n)_{1:N}$ from an observed sequence $(y_n)_{1:N}$. Probabilistic treatment of the problem in these sequences are considered accomplishments, respectively, process $X = (X_n)_{1:N}$ and $Y = (Y_n)_{1:N}$. Several techniques based on statistical methods have been developed for solve this problem. The most common model among the process is the model called of hidden Markov model fg (MMC). In this model we assume that the hidden process X is Markov and laws $p(y|x)$ of Y conditional on X is sufficiently simple so that the law $p(x|y)$ is also Markovian, this last property is necessary for treatment. More Extensions of these models have been proposed since 2000. In Markov models couples (MMCouple), more general than the MMC, the pair (X, Y) is Markovian, implying that $p(x|y)$ is also Markov (when $p(x)$ is no necessarily), which allows the same treatment as in MMC. More recently (2002), were extended to MMCouples « triplet Markov models » (MMT), in which we introduce a auxiliary process U and suppose that the triple $T = (X, U, Y)$ is Markov. Again it is possible, in a more general that of MMCouples, perform treatments with a reasonable complexity.

The objective of this thesis is to propose new modeling part of the MMT and to investigate their relevance and interest. We offer two types of innovations :

(i) When the system is discrete and hidden when the couple (X, Y) is not stationary with a finite number of « jumps » random parameters, the recent use of MMT in where the jumps are modeled by a discrete process U a been very convincing (Lanchantin, 2006). Our first idea is to use this approach with a process U continuous, which models non-steady "continuous" from (X, Y) . We offer chains and fields and triplets present some experiments. The results obtained in the modeling of non-stationarity still seem less interesting that in the discrete case. However, new models may have other interests, in particular, they seem more efficient than of hidden Markov fg classic when the noise is correlated ;

(ii) An MMT $T = (X, U, Y)$ such that X and Y are continuous and U is discrete finite. We are dealing with the problem of filtering, or smoothing, with random jumps. In modeling classic the pair hidden (X, U) is Markov, but the pair (U, Y) is not, what is the cause of the impossibility of Exact calculations with linear complexity in time. It is then necessary to use various approximate methods, which those using particle filtering are among the most used. In recent models MMT the pair hid (X, U) is not necessarily Markovian, but the pair (U, Y) is, what which allows accurate treatment with a reasonable complexity (Pieczynski 2009). Our second idea

is to extend these models to triplets $T = (X, U, Y)$ where the pairs (U, Y) are "partially" Markov. Such a pair (U, Y) is not Markov but U is Markov conditionally on Y . We a model with $T = (X, U, Y)$ more general, which is more Markov, wherein the filtering and smoothing are accurate possible with linear complexity in time. Some preliminary Simulations show interest in new models smoothing in the presence of jumps.

Keywords

Hidden Markov Chain, pairwise and triplet Markov chains, Partially Markov chains, Hidden Chain to Jump-Markov, long dependence, Hidden Gaussian-Markov field, Logit model, Bayesian inférence, filtering, smoothing, Expectation Maximisation, Iterative Conditional Estimation

Remerciements

Je tiens à remercier en premier lieu Wojciech PIECZYNSKI pour avoir accepté de m'accueillir au sein de son département et qui a dirigé cette thèse dans la continuité de mon stage de master. Tout au long de ma thèse, il a su orienter mes recherches aux bons moments. Malgré son emploi chargé, il a toujours été disponible pour d'intenses et rationnelles discussions. Pour tout cela, sa confiance et son soutien, je le remercie vivement et j'espère qu'il trouvera dans ce travail, ma grande reconnaissance et mon respect absolu.

Je remercie aussi les rapporteurs de ma thèse François ROUEFF et Nikolaos LIMNIOS pour leur réactivité et les apports constructifs qu'ils ont ajouté et l'intérêt qu'ils ont porté à mon travail.

Je tiens également à remercier les autres membres du jury : Jean-Yves TOURNET Professeur INP - ENSEEIHT Toulouse, Michel BRONIATOWSKI Professeur Université Pierre et Marie Curie et Pascal LARZABAL Professeur École Normale Supérieure de Cachan. Pour avoir accepté de juger ce travail, et pour leur disponibilité inconditionnelle.

Je remercie, de même, mes collègues de bureau pour leur gentillesse, leur disponibilité et surtout pour la bonne ambiance de travail que nous avons partagée : Noufel ABBASSI et Soumaya SALLEM Je tiens aussi à associer à ces remerciements l'ensemble du département CITI, et en particulier : l'ancien post-doc Hawari, ainsi que Julie BONNET qui m'a facilité toutes les démarches administratives au sein de l'école.

Un travail de thèse représente toujours une continuité. Pour cela je n'oublis pas de remercier tous les anciens thésards qui ont contribué directement ou indirectement à ce sujet par leurs recherches, notamment : JEROME pour son aide précieuse.

Des remerciements particuliers vont à toute ma famille pour leur soutien inconditionnel : A mon père, j'espère qu'il trouvera dans ce travail les valeurs qu'il m'a transmis : la rigueur, la méthode, la patience et la persévérance. A ma mère, j'espère qu'elle trouvera aussi tout ce qu'elle m'a transmis et le résultat de ces efforts, sa générosité, sa créativité et sa persévérance.

A mes chers frères, sœurs, mon cousin Nasser et sa famille, j'adresse ma gratitude pour leurs encouragements, leur soutien et leurs conseils.

Table des matières

<i>Résumé</i>	i
<i>Abstract</i>	iii
<i>Remerciements</i>	v
Table des matières	vii
<i>Notations</i>	ix
Introduction	1
I Modèles de Markov Classiques	5
I.1 Modèle graphique	5
I.1.1 Modèle graphique non orienté	6
I.1.2 Modèle graphique orienté	7
I.1.3 Factorisation d'une loi selon un graphe	10
I.2 Chaîne de Markov cachée	10
I.2.1 CMC à espace d'état discret	11
I.2.2 Inférence Bayésienne dans la CMC discrète	14
I.3 CMC à espace d'états continu	16
I.3.1 Inférence Bayésienne	17
I.3.2 Filtre de Kalman	18
I.3.3 Lissage	19
I.4 Conclusion	22
II Modèle de Markov Couple	25
II.1 Chaîne de Markov Couple	25
II.2 Inférence dans le modèle couple	31
II.2.1 Algorithme de Baum-Welsh	31
II.2.2 Algorithme de Baum-Welsh conditionnel	33
II.2.3 Programmation dynamique : Algorithme de Viterbi	34
II.3 Etude comparative entre CMC-BI et CMCouples	36
II.4 Conclusion	39
III Estimation des paramètres	41
III.1 L'algorithme EM et ses variantes	42
III.1.1 L'algorithme EM	42
III.1.2 Variantes de l'algorithme EM	43
III.2 Estimation Conditionnelle Itérative	45
III.3 Applications	46
III.3.1 Mélange indépendant gaussien fini	46

III.3.2 CMC-BI gaussiennes	48
III.3.3 CMCouple	51
III.3.4 Exemple numérique	53
III.4 Conclusion	55
IV Modèle de Markov Triplet	57
IV.1 Chaîne de Markov Triplet : Cas discret	57
IV.1.1 Chaîne de Markov Cachée M-stationnaire	58
IV.1.2 Chaîne de Markov Couple M-stationnaire	67
IV.1.3 Chaîne Semi-Markovienne Cachée	68
IV.2 Chaîne de Markov Triplet Mixte	75
IV.3 Champ de Markov Triplet Mixte	84
IV.3.1 Champ Gaussien-Markovien	86
IV.3.2 Simulation du Champ Gaussien-Markovien	87
IV.3.3 Champ Gaussien-Markovien caché	90
IV.3.4 Champ Gaussien-Markovien triplet	94
IV.4 Conclusion	98
V Chaînes conditionnellement linéaires à sauts markoviens	99
V.1 Processus à mémoire longue	99
V.2 Chaîne couple partiellement de Markov (CCPM)	101
V.2.1 Présentation des CCPM	101
V.2.2 Chaînes de Markov cachées par du bruit gaussien(CMC-BG)	103
V.3 Modèle conditionnellement linéaire à sauts markoviens	105
V.3.1 Modèle Classique	105
V.3.2 Modèle conditionnellement linéaire à sauts markoviens (CCLSM)	106
V.3.3 CCLSM et bruit gaussien à mémoire longue	108
V.4 Experimentations	111
V.5 Conclusion	119
Conclusion et perspectives	121
A K-means	123
B Approximation de Laplace	125
C Vecteurs gaussiens	127
Bibliographie	129

Notations

Symboles

N	Nombre des observations
K	Nombre des états
\mathcal{S}	Ensemble des indices.
$ \mathcal{S} $	Cardinal ensemble des indices.
\mathcal{X}	Espace des états cachés
\mathcal{Y}	Espace des observations
\mathcal{U}	Espace des états auxiliaires
X_{1n}	Processus caché de longueur n $(X_i)_{1 \leq i \leq n}$.
Y_{1n}	Processus observé de longueur n $(Y_i)_{1 \leq i \leq n}$.
U_{1n}	Processus auxiliaire de longueur n $(U_i)_{1 \leq i \leq n}$.
X_n, U_n, Y_n	Variables aléatoires indicées par n .
x_n, u_n, y_n	Réalisations des v.a X_n, U_n, Y_n .
ν	Mesure de référence.
\mathbb{P}	Mesure de Probabilité.
\mathbb{E}	Espérance associée.
\mathbb{V}	Variance associée.
\mathbb{C}	Covariance associée.
$\mathbb{E}(\cdot \cdot)$	Espérance conditionnelle.
$p(x)$	Pour x discret probabilité $X = x$, pour $x \in \mathcal{X}$ densité de X par rapport à ν .
$p(\cdot \cdot)$	Densité conditionnelle d'une v.a.
$\alpha_n(\cdot)$	Quantité directe d'indice n .
$\beta_n(\cdot)$	Quantité rétrograde d'indice n .
$(\gamma(k))_{k \in \mathbb{Z}}$	Famille de covariance d'un processus.
\perp	relation d'indépendance.
$A \perp B$	A est indépendante de B .
$A \perp B C$	A est indépendante de B conditionnellement à C .
$\mathcal{N}(m, \sigma^2)$	Loi normale de moyenne m et de variance σ^2 .
\mathcal{G}	Graphe d'indépendance conditionnelle non orientée.
$\vec{\mathcal{G}}$	Graphe d'indépendance conditionnelle orientée.
\mathcal{G}^m	Graphe moral.
$\widehat{S}()$	Fonction de classification.
L	Fonction de coût .
\mathcal{L}	Log-vraisemblance.

Acronymes

V.A	Variable Aléatoire.
MAP	Maximum A Posteriori.
MPM	Maximum Marginal a Posteriori.
ML	Maximum de Vraisemblance.
CMC	Chaîne de Markov cachée.
CMCouple	Chaîne de Markov couple.
CMT	Chaîne de Markov triplet.
CMC-MS	Chaîne de Markov cachée M -stationnaire.
CMCouple-MS	Chaîne de Markov Couple M -stationnaire.
CSMC	Chaîne semi-markovienne cachée.
CMC-ML	Chaîne de Markov cachée à mémoire longue.
CMTM	Chaîne de Markov Triplet Mixte.
EM	Expectation Maximization.
ECI	Estimation Conditionnelle Itérative.
BI	Bruit Indépendant.
ML	Mémoire Longue.
CCPM	Chaîne Couple Partiellement de Markov.
CTPM	Chaîne Triplet Partiellement de Markov.
CG	Champ Gaussien.
CGM	Champ Gaussien-Markovien.
RTS	Rauch-Tung-Striebel.
GML	Gaussien à Mémoire Longue.
CCLSM	Chaîne conditionnellement Linéaire à Sauts Markoviens.

Introduction

Notre travail se situe dans le domaine des traitements statistiques des signaux. Parmi ces traitements nous nous intéressons aux méthodes bayésiennes. Le problème général est celui de l'estimation du signal caché à partir du signal observé. Nous considérons que le signal observé est une réalisation $y = (y_n)_{1:N}$ d'un processus aléatoire $Y = (Y_n)_{1:N}$ sur un réseau $\mathcal{S} = \{1, \dots, N\}$ et nous cherchons à estimer la réalisation $x = (x_n)_{1:N}$ d'un processus aléatoire caché $X = (X_n)_{1:N}$. Nous supposons que les variables X_n prennent leurs valeurs dans un espace \mathcal{X} et les variables Y_n prennent leurs valeurs dans un espace \mathcal{Y} . Lorsque le signal caché est discret son estimation sera appelée « segmentation » ; lorsqu'il est continu on parlera de « filtrage » ou de « lissage ».

Une méthode bayésienne de segmentation ou de lissage est caractérisée par une fonction de coût, dite aussi fonction de perte, $L : \mathcal{X}^N \times \mathcal{X}^N \rightarrow \mathbb{R}$. La fonction de coût L permet de mesurer la perte moyenne subie lors de l'application d'une méthode donnée. En notant une méthode bayésienne \widehat{S} , la méthode (on dit souvent la "stratégie") bayésienne vérifie :

$$\mathbb{E}[L(\widehat{S}(Y), X)] = \underset{S}{\operatorname{arg\,min}} \mathbb{E}[L(S(Y), X)] \quad (1)$$

Lorsque l'espace \mathcal{X} est fini les fonctions de perte parmi les plus utilisées sont les suivantes :

□

$$L : \begin{array}{l} \mathcal{X}^N \times \mathcal{X}^N \rightarrow \mathbb{R} \\ (x_{1:N}^1, x_{1:N}^2) \mapsto 1 - \delta(x_{1:N}^1, x_{1:N}^2) \end{array} ; \quad (2)$$

□

$$L : \begin{array}{l} \mathcal{X}^N \times \mathcal{X}^N \rightarrow \mathbb{R} \\ (x_{1:N}^1, x_{1:N}^2) \mapsto 1 - \frac{1}{N} \sum_{n=1}^N \delta(x_n^1, x_n^2) \end{array} \quad (3)$$

avec δ est l'indice de Kronecker $\delta(a, b) = 1$ si $a = b$ et 0 sinon.

La fonction (2) pénalise de la manière toutes les solutions $\widehat{S}(Y)$ différentes de X . La solution bayésienne correspondant à cette fonction de perte, appelée « Maximum a Posteriori » (MAP), est donnée par :

$$\widehat{x}_{MAP}(y) = \underset{x \in \mathcal{X}^N}{\operatorname{arg\,max}} p(x|y) \quad (4)$$

On remarque que l'estimateur du MAP est équivalent à l'estimateur du maximum de vraisemblance lorsque la loi a priori est la loi uniforme.

Contrairement à la fonction (2), la fonction (3) a un aspect local et elle ne pénalise pas de la même manière toutes les estimations de X . La méthode bayésienne

correspondant à cette fonction de perte est dite « Maximum des Marginales a Posteriori » (MPM). Elle est donnée par :

$$\forall n \in \mathcal{S} \quad \hat{x}_n(y) = \underset{x_n \in \mathcal{X}}{\operatorname{argmax}} p(x_n|y) \quad (5)$$

Notre travail concerne les situations dans lesquelles le nombre d'observations N est trop grand pour que l'on puisse envisager d'utiliser les lois du couple (X, Y) dans toute leur généralité et il est nécessaire de faire appel à des formes particulières de ces lois. A titre d'exemple, en traitement des images, le réseau \mathcal{S} est constitué des "pixels", ou des "sites", dont le nombre N peut être de l'ordre d'un million. Il est alors trop grand pour pouvoir envisager le calcul direct de la loi $p(x, y)$ du couple (X, Y) . Les modèles particuliers utilisés classiquement à des fins de segmentation ou de lissage pour N "grand" sont les « modèles de Markov cachés » (MMC). Introduits dans les articles fondateurs [5, 9, 10, 53], ils permettent, d'une part, de modéliser d'une manière simple la loi $p(x, y)$ et de prendre en considération la corrélation spatiale (dans le cas des images) ou temporelle (dans le cas des signaux mono-dimensionnels) des données cachées. D'autre part, ils permettent de calculer (dans le cas des signaux mono-dimensionnels) ou d'approcher efficacement (dans le cas des images) les estimations des données cachées par des méthodes bayésiennes. Ces modèles ont d'innombrables applications dans divers domaines. Ainsi, de manière non exhaustive, nous pouvons citer quelques publications en imagerie médicale [84], bioscience [29, 62], imagerie radar [32, 51, 55, 114], écologie [8, 18], communication [25], acoustique [3].

Dans les modèles de Markov cachés on définit la loi $p(x, y)$ en deux temps. On définit d'abord $p(x)$, qui est supposée markovienne, et ensuite on définit la loi (qui modélise le caractère stochastique du "bruit") $p(y|x)$. L'important dans ces modèles est que la loi $p(x|y)$, dite « a posteriori », est markovienne. Or, une fois que l'on a posé la markovianité de $p(x)$, on ne peut obtenir la markovianité de $p(x|y)$ qu'au prix des bruits $p(y|x)$ simples. C'est une faiblesse inutile de ces modèles car la markovianité de $p(x)$ n'est pas nécessaire. Cette faiblesse a été levée dans les modèles dits « modèles de Markov couples » [91, 93, 104], dans lesquels on suppose directement que le couple (X, Y) est markovien. Comme conséquence nous obtenons que $p(x|y)$ et $p(y|x)$ sont aussi markoviennes. La première propriété implique la possibilité d'utiliser les mêmes méthodes bayésiennes que dans le cas des MMC. La deuxième permet une modélisation « markovienne » des bruits, ce qui est plus complet que les modélisations habituelles. Notons que dans une chaîne de Markov couple la chaîne X n'est pas nécessairement markovienne, mais cela ne nuit aucunement aux traitements car la markovianité de X ne sert, dans les MMC, que pour montrer la markovianité de $p(x|y)$.

Par la suite, les modèles de Markov Couple ont connu des extensions qui sont les modèles de Markov Triplet (MMT) [36, 92, 94]. Dans ces modèles on introduit un troisième processus $U = (U_n)_{n \in \mathcal{S}}$ et on suppose que le triplet $T = (X, U, Y)$ est de Markov. On arrive à une famille de modèles très riche, où le processus U peut modéliser des propriétés très diverses. Il peut modéliser la non-stationnarité de X [64, 67, 69], la semi-markovianité de X [27, 70, 71], ou encore le caractère "évidentiel" X [64, 68, 96]. Sachant que dans une chaîne de Markov $T = (X, U, Y)$ la loi de (X, Y) n'est pas nécessairement markovienne, le modèle MMT est strictement plus

général que le modèle de Markov couple. Pourtant, les mêmes traitements bayésiens sont encore possibles. En effet, un MMT $T = (X, U, Y)$ peut être vu comme un modèle markovien couple $T = (V, Y)$, avec $V = (X, U)$. Il est alors possible d'estimer simultanément X et U .

Notre travail concerne les modèles de Markov triplets. Plus particulièrement nous nous intéressons aux chaînes et aux champs de Markov triplet. Nous proposons trois contributions :

- ◇ Dans le cas des chaînes triplets avec la chaîne cachée discrète, nous proposons un nouveau modèle dans lequel U soit une chaîne de Markov à espace d'état continu [80] ;
- ◇ Dans le cas des champs triplets avec le champ caché discret, nous proposons un nouveau modèle de champ Triplet dans lequel U est champ Gaussien-Markovien ;
- ◇ Un modèle de chaînes de Markov triplets récent (2009) a permis de traiter, par des méthodes directes et optimales, le problème de filtrage et de lissage en présence des sauts. Nous généralisons ce modèle à un modèle original permettant de considérer des bruits à mémoire longue.

Organisation du manuscrit

Notre manuscrit est composé de cinq chapitres, qui sont organisés comme suit :

Le premier chapitre est consacré à des rappels et des définitions concernant les modèles graphiques et les chaînes de Markov cachées. En premier lieu, nous rappelons les définitions des modèles graphiques orientés et non-orientés, qui servent en second lieu à rappeler les modèles des chaînes de Markov cachées à espace d'états discret et à espace d'états continu. Pour chacun de ces modèles, nous donnons divers algorithmes classiques d'inférence Bayésienne permettant de calculer $p(x_n|y)$ et $p(x_n|x_{n-1}, y)$. Nous présentons également des exemples de simulations pour ces modèles.

Dans le deuxième chapitre, nous introduisons les chaînes de Markov couples. Nous détaillons des algorithmes d'inférence Bayésienne : algorithme de Baum-Welsh, algorithme de Baum-Welsh conditionnel et l'algorithme de Viterbi. Ces algorithmes permettent de calculer les solutions bayésiennes MPM et MAP dans le cas de la segmentation des observations $y_{1:N}$. Nous procédons à une étude comparative entre la CMCouple et la CMC-BI en présentant des simulations originales.

Dans les deux premiers chapitres, nous présentons divers algorithmes de traitement correspondant aux divers modèles paramétriques, dont les paramètres sont supposés connus. Le troisième chapitre sera dédié au cas dans lequel ces paramètres ne sont pas connus et doivent être estimés à partir des observations. Nous énonçons d'abord les principes généraux de deux algorithmes généraux d'estimation des paramètres : algorithme Espérance-Maximisation (EM) [16, 20, 22, 34] et Estimation Conditionnelle Itérative (ICE) [7, 33, 89, 95]. Ensuite nous détaillons ces algorithmes pour certains modèles traités dans les chapitres précédents. Enfin, nous présentons une application numérique originale d'estimation des paramètres.

Dans le quatrième chapitre, nous présentons les modèles de Markov triplets. En premier lieu, nous traitons le cas des chaînes de Markov Triplets (CMT) avec processus auxiliaire U discret, et nous détaillons différentes utilisations de ce processus

pour modéliser la M-stationnarité et la semi-markovianité. En second lieu, nous présentons un nouveau modèle de la CMT avec processus auxiliaire continu [80] et nous donnons des exemples d'applications en segmentation d'images. Nous présentons une étude numérique concernant des images simulées et des images artificielles. En troisième lieu, nous présentons un modèle original de champ de Markov triplet, avec un champ auxiliaire U Gaussien-Markovien (CGM) [50, 106, 108].

Dans le cinquième chapitre, nous rappelons tout d'abord la définition d'un processus à mémoire longue. Ensuite, nous présentons le modèle récent des chaînes couples partiellement de Markov (CCPM) qui permet de prendre en considération la mémoire longue d'un processus observé, tout en préservant la possibilité d'étendre l'algorithme de Baum-Welsh, qui permet de calculer $p(x_n|y)$. Puis, nous présentons deux nouveaux modèles prenant en considération la mémoire longue et permettant de faire un filtrage et un lissage exact dans certains modèles cachés à sauts. Le premier, général, est dit « chaîne conditionnellement linéaire à sauts markoviens » (CCLSM). Dans le deuxième, appelé « chaîne conditionnellement linéaire à sauts markoviens et bruit gaussien à mémoire longue » (CCLSM-GML) on considère des bruits gaussiens [49, 99, 100]. Enfin, nous produisons des résultats d'expérimentations informatiques concernant ces nouveaux modèles.

Le manuscrit se termine par une conclusion et des perspectives, une brève annexe, et une bibliographie.

Chapitre I

Modèles de Markov Classiques

Dans ce chapitre, nous rappelons l'un des modèles de Markov cachés le plus utilisé depuis les années quatre-vingt. Il s'agit du modèle des chaînes de Markov cachées (CMC), connu en anglais sous le nom de « Hidden Markov Chain » (HMC). Nous détaillerons les algorithmes de calcul dans deux cas : celui où les données latentes sont à valeurs dans un espace d'état discret et celui où elles sont à valeurs dans un espace d'état continu.

Pour commencer, nous rappelons quelques notions sur les modèles graphiques orientés et non-orientés. Nous rappelons la notion d'indépendance conditionnelle, qui joue un rôle essentiel dans le développement des algorithmes de calcul, et aussi dans la définition des modèles graphiques.

Dans toute la suite, on note par X une collection de variables aléatoires (X_n) indexées par une partie non vide \mathcal{S} de \mathbb{N}^m . Lorsque $m = 1$ il s'agit de chaînes et pour $m = 2$, il s'agit de champs bidimensionnels. Nous noterons $x = (x_n)_{n \in \mathcal{S}}$ les réalisations de X . Pour tout \mathcal{A} sous-ensemble de \mathcal{S} , on notera par ailleurs $X_{\mathcal{A}}$ la restriction du processus X à \mathcal{A} .

I.1 Modèle graphique

La modélisation graphique est un moyen général pour illustrer certains aspects des modèles probabilistes. En effet, elle permet de représenter les variables aléatoires par les sommets d'un graphe et les relations d'indépendance conditionnelle entre ces variables par les arêtes de ce dernier.

Dans toute la suite, nous noterons par $p(\cdot)$ les densités des lois, éventuellement $p(\cdot|\cdot)$ les densités des lois conditionnelles, par rapport à la mesure de comptage dans le cas discret et par rapport à la mesure de Lebesgue dans le cas continu. La notion de base dans la modélisation graphique est l'indépendance conditionnelle.

Définition I.1 *Soit \mathcal{A}, \mathcal{B} et \mathcal{C} trois sous ensembles de \mathcal{S} , deux à deux disjoints. On dit que $X_{\mathcal{A}}$ et $X_{\mathcal{C}}$ sont indépendants conditionnellement à $X_{\mathcal{B}}$ - et on note $X_{\mathcal{A}} \perp\!\!\!\perp X_{\mathcal{C}}|X_{\mathcal{B}}$ - si et seulement si :*

$$p(x_{\mathcal{A}}, x_{\mathcal{C}}|x_{\mathcal{B}}) = p(x_{\mathcal{A}}|x_{\mathcal{B}})p(x_{\mathcal{C}}|x_{\mathcal{B}}) \quad (\text{I.1})$$

En terme d'information, ceci se traduit par : connaissant $X_{\mathcal{B}}$, $X_{\mathcal{C}}$ n'apporte pas d'information supplémentaire sur $X_{\mathcal{A}}$ (resp. $X_{\mathcal{A}}$ sur $X_{\mathcal{C}}$).

Nous détaillons dans la suite les modèles graphiques orientés et non-orientés ; pour plus de précisions, les lecteurs intéressés pourront également consulter le livre de Steffen L. Lauritzen [74], ou l'article de P. Smyth et al. [113].

I.1.1 Modèle graphique non orienté

Soit $\mathcal{S} = \{1, \dots, N\}$ une partie de \mathbb{N} , et soit \mathcal{E} un sous ensemble de $\mathcal{S} \times \mathcal{S}$. Nous appelons « graphe » le couple $\mathcal{G} = (\mathcal{S}, \mathcal{E})$. L'ensemble \mathcal{S} sera nommé ensemble des sommets, et \mathcal{E} l'ensemble des arêtes. Le graphe \mathcal{G} sera nommé graphe non orienté lorsque pour tout couple de sommets $(u, v) \in \mathcal{S}^2$, si $(u, v) \in \mathcal{E}$ alors $(v, u) \in \mathcal{E}$. Soient u et v deux sommets de \mathcal{G} . On dit qu'il y a une arête entre u et v dans \mathcal{G} , et on note $u \leftrightarrow_{\mathcal{G}} v$, si $(u, v) \in \mathcal{E}$. On note $u \not\leftrightarrow_{\mathcal{G}} v$ s'il n'y a pas d'arête.

Pour tout sommet u de \mathcal{S} , on note $\nu(u)$ l'ensemble des voisins de u , cad. l'ensemble des sommets v qui sont reliés par des arêtes à u :

$$\nu(u) = \{v \in \mathcal{S}, u \leftrightarrow_{\mathcal{G}} v\} \quad (\text{I.2})$$

Un sous-ensemble \mathcal{C} de \mathcal{G} est une clique si et seulement si :

- \mathcal{C} est un singleton, ou
- $\forall (u, v)$ deux sommets de \mathcal{C} , u et v sont mutuellement voisins ($u \in \nu(v)$ et $v \in \nu(u)$).

On dit qu'il y a un chemin entre u et v , s'il existe une suite des sommets u_1, \dots, u_n , telle que : $u \in \nu(u_1), \dots, u_{i-1} \in \nu(u_i), \dots, u_n \in \nu(v)$. Soient \mathcal{A}, \mathcal{B} et \mathcal{C} trois sous-ensembles de \mathcal{S} deux à deux disjoints. On dit que \mathcal{B} sépare \mathcal{A} et \mathcal{C} , si seulement si pour tout sommet a dans \mathcal{A} , tout sommet c dans \mathcal{C} , tout chemin entre a et c passe par ou moins un sommet b appartenant à \mathcal{B} .

Sur la figure (I.1), $\mathcal{D} = \{3, 4\}$ sépare $\mathcal{B} = \{1, 2\}$ et $\mathcal{C} = \{5, 6\}$. Un chemin entre $\{1\} \in \mathcal{B}$ et $\{5\} \in \mathcal{C}$ passe par $\{3\} \in \mathcal{D}$.

Définition I.2 Soit \mathcal{G} un graphe. Le processus aléatoire $X = (X_n)_{1:N}$ satisfait la propriété de Markov « par paires » par rapport à \mathcal{G} , si $\forall (u, v)$ couple de sommets tel que $u \leftrightarrow_{\mathcal{G}} v$, X_u et X_v sont indépendants conditionnellement à $X_{t \notin \{u, v\}}$:

$$\forall (u, v), \text{ si } u \leftrightarrow_{\mathcal{G}} v \text{ alors } X_u \perp\!\!\!\perp X_v | X_{t \notin \{u, v\}} \quad (\text{I.3})$$

Définition I.3 Soit \mathcal{G} un graphe. Le processus aléatoire $X = (X_n)_{1:N}$ satisfait la propriété de Markov « locale » par rapport à \mathcal{G} , si pour tout sommet $n \in \{1, \dots, N\}$, X_n est indépendant de ses non-voisins $(X_t)_{t \notin \{n \cup \nu(n)\}}$ conditionnellement à $(X_t)_{t \in \nu(n)}$:

$$\forall n \in \mathcal{S} \quad X_n \perp\!\!\!\perp (X_t)_{t \notin \{n \cup \nu(n)\}} | (X_t)_{t \in \nu(n)} \quad (\text{I.4})$$

Définition I.4 Soit \mathcal{G} un graphe. Le processus aléatoire $X = (X_n)_{1:N}$ satisfait la propriété de Markov « globale » par rapport à \mathcal{G} , si pour tout triplet $(\mathcal{B}, \mathcal{C}, \mathcal{D})$ de sous-ensembles de \mathcal{S} , disjoints deux à deux, tels que \mathcal{D} sépare \mathcal{B} et \mathcal{C} dans \mathcal{G} , on a :

$$X_{\mathcal{B}} \perp\!\!\!\perp X_{\mathcal{C}} | X_{\mathcal{D}} \quad (\text{I.5})$$

On dit alors également que \mathcal{G} est le graphe d'indépendance conditionnelle pour X .

Notons, à titre d'exemple, que sur la figure (I.1), \mathcal{D} sépare \mathcal{B} et \mathcal{C} : $\{X_1, X_2\} \perp\!\!\!\perp \{X_5, X_6\} \mid \{X_3, X_4\}$

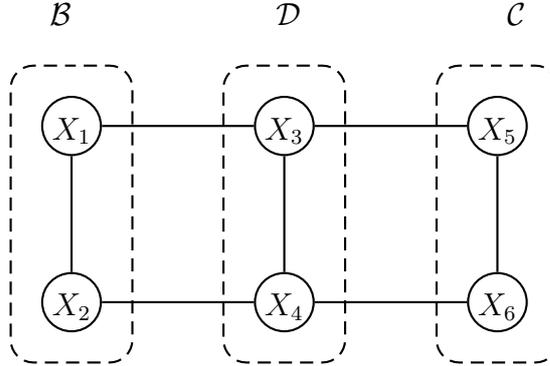


FIGURE I.1 – Exemple d'un graphe d'indépendance conditionnelle non-orienté : $\{X_1, X_2\}$ et $\{X_5, X_6\}$ sont indépendants conditionnellement à $\{X_3, X_4\}$

Remarque I.1 Pour un graphe d'indépendance conditionnelle non-orienté $\mathcal{G} = (\mathcal{S}, \mathcal{E})$, les trois propriétés de Markov (locale, par paire et globale) sont équivalentes. La démonstration de l'équivalence est détaillée dans le livre de Steffen L. Lauritzen [74].

Sachant que ces trois propriétés sont équivalentes, nous choisirons, lorsqu'il s'agira de graphe non orienté, d'utiliser l'une ou l'autre selon nos besoins dans chaque cas d'étude. Par exemple, si nous nous intéressons à la loi $p(x_n | x_{t \neq n})$ dans le cas d'une chaîne de Markov la propriété de Markov locale nous sera la plus utile, et si nous nous intéressons à la factorisation de la loi $p(x)$ nous pourrons utiliser la propriété de Markov globale.

I.1.2 Modèle graphique orienté

On appelle graphe orienté et on note $\vec{\mathcal{G}}$, tout graphe dont les arêtes sont définies par leur origine et leur extrémité, c'est-à-dire dont les arêtes sont orientées, munies d'un sens. Une arête d'un graphe orienté est appelée arc. On note par $\vec{\mathcal{E}} \subset \mathcal{S}^2$ l'ensemble des arcs.

Soit $\vec{\mathcal{G}} = (\mathcal{S}, \vec{\mathcal{E}})$ un graphe orienté, et soit $(u, v) \in \mathcal{S}^2$. On dit que u est un parent de v s'il existe un arc d'origine u et d'extrémité v , ce que l'on note par $u \xrightarrow{\vec{\mathcal{G}}} v$.

Un graphe orienté $\vec{\mathcal{G}}$ est dit acyclique si pour tout u un sommet de \mathcal{S} il n'existe aucun chemin (cycle) orienté qui part de u et revient à u . Un sommet v de \mathcal{S} est un descendant d'un sommet u de \mathcal{S} s'il existe un arc d'origine u et d'extrémité finale v ; sinon v est un non-descendant de u . On note par $pa_{\vec{\mathcal{G}}}(v)$ l'ensemble des parents de v et par $nd_{\vec{\mathcal{G}}}(v)$ l'ensemble des non-descendants de v .

Définition I.5 Soit $\vec{\mathcal{G}}$ un graphe orienté acyclique. Le processus aléatoire X satisfait la propriété de Markov locale orientée par rapport à $\vec{\mathcal{G}}$, si pour tout sommet u

de $\vec{\mathcal{G}}$, X_u est indépendant de ses non descendants $X_{nd_{\vec{\mathcal{G}}}(u)}$ conditionnellement à ses parents $X_{pa_{\vec{\mathcal{G}}}(u)}$:

$$X_u \perp\!\!\!\perp X_{nd_{\vec{\mathcal{G}}}(u)} | X_{pa_{\vec{\mathcal{G}}}(u)} \quad (\text{I.6})$$

On dit que $(X, \vec{\mathcal{G}})$ est un modèle graphique Markovien orienté, si X satisfait la propriété de Markov locale orientée par rapport à $\vec{\mathcal{G}}$. On dit également que $\vec{\mathcal{G}}$ est un graphe d'indépendance conditionnelle orienté de X .

Exemple I.1 Soit $(X_n)_{n \in \mathbb{N}}$ un processus gaussien vérifiant :

$$\forall n \in \mathbb{N}^* \quad X_{n+1} = \rho X_n + \epsilon_n$$

avec X_1 une gaussienne centrée réduite, $\rho \in \mathbb{R}^*$ et $|\rho| < 1$. Les ϵ_n sont indépendants identiquement distribués, $\forall n \in \mathbb{N}^* \quad \epsilon_n \sim \mathcal{N}(0, 1 - \rho^2)$ et chaque ϵ_n est indépendant de X_1, \dots, X_n .

Si on note par $\vec{\mathcal{G}}$ le graphe de sommets $n \in \{1, \dots, N\}$ et d'arcs $(n, n+1)$, $(X, \vec{\mathcal{G}})$ est un modèle graphique orienté.

La définition de la propriété de Markov globale pour les graphes orientés se base sur la notion de séparation des sous ensembles. Nous avons donné la définition dans le cas des graphes non-orientés (voir I.4); dans le cas des graphes orientés la définition de sous ensembles séparables est différente. Pour donner cette définition nous avons besoin de passer d'un graphe orienté à un graphe non-orienté. Cette méthode consiste à passer par un graphe moral. On associe à chaque graphe orienté $\vec{\mathcal{G}}$ un graphe moral $\mathcal{G}^m = (\mathcal{S}, \mathcal{E}^m)$, où \mathcal{E}^m est obtenu à partir de $\vec{\mathcal{E}}$. Pour obtenir \mathcal{E}^m on relie, par une arête non orientée, les parents, qui ont un descendant commun; ensuite on supprime toutes les orientations. Nous avons donc l'équivalence suivante :

$$u \underset{\mathcal{G}^m}{\leftrightarrow} v \Leftrightarrow \left[u \underset{\vec{\mathcal{G}}}{\rightarrow} v \text{ ou } v \underset{\vec{\mathcal{G}}}{\rightarrow} u \text{ ou } \{ \exists k \in \mathcal{S}, u \underset{\vec{\mathcal{G}}}{\rightarrow} k \text{ et } v \underset{\vec{\mathcal{G}}}{\rightarrow} k \} \right]$$

Pour illustrer cette transformation, la figure (I.3) présente un graphe moral obtenu à partir du graphe représenté sur la figure (I.2). Le sommet $\{4\}$ est un descendant commun de $\{3\}$ et $\{2\}$; de manière analogue, $\{6\}$ est descendant commun de $\{5\}$ et $\{4\}$. En suivant la règle générale, on relie $\{2\}$ et $\{3\}$, $\{4\}$ et $\{5\}$, puis on supprime les orientations.

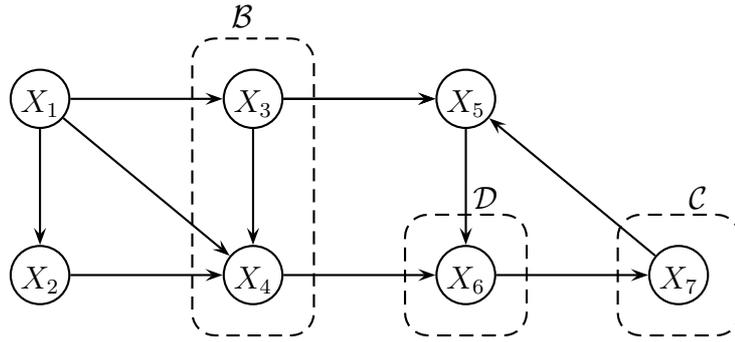


FIGURE I.2 – Exemple d’un graphe d’indépendance conditionnelle orientée : $\{X_7\}$ et $\{X_3, X_4\}$ sont indépendants conditionnellement à $\{X_6\}$

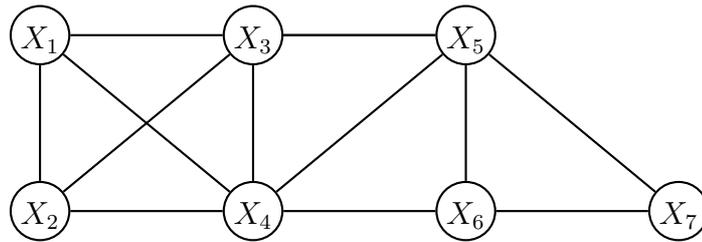


FIGURE I.3 – Graphe moral du graphe (I.2)

Soit \mathcal{A} une partie de \mathcal{S} . On dit que \mathcal{A} est ancestral, si pour tout u appartenant à \mathcal{A} , les parents de u sont dans \mathcal{A} , $pa(u) \subset \mathcal{A}$. Soient \mathcal{B}, \mathcal{C} et \mathcal{D} trois sous ensembles de \mathcal{S} , deux à deux disjoints. On dit que \mathcal{D} sépare \mathcal{B} et \mathcal{C} dans $\vec{\mathcal{G}}$, si et seulement si, \mathcal{D} sépare \mathcal{B} et \mathcal{C} dans le graphe moral \mathcal{G}^m , avec \mathcal{G}^m engendré¹ par le plus petit ensemble ancestral contenant \mathcal{B}, \mathcal{C} et \mathcal{D} .

Définition I.6 Soit $\vec{\mathcal{G}} = (\mathcal{S}, \vec{\mathcal{E}})$ un graphe orienté. Le processus aléatoire $X = (X_n)_{1:N}$ satisfait la propriété de Markov « globale » par rapport à $\vec{\mathcal{G}}$, si pour tout triplet $(\mathcal{B}, \mathcal{C}, \mathcal{D})$ de sous-ensembles de \mathcal{S} , disjoints deux à deux, tel que \mathcal{D} sépare \mathcal{B} et \mathcal{C} dans $\vec{\mathcal{G}}$, on a :

$$X_{\mathcal{B}} \perp\!\!\!\perp X_{\mathcal{C}} | X_{\mathcal{D}} \tag{I.7}$$

Le choix entre les deux modèles graphiques « orientés » et « non-orientés » dépend essentiellement de type de relation que nous cherchons à modéliser. Chaque

1. Soit $\vec{\mathcal{G}} = (\mathcal{S}, \vec{\mathcal{E}})$ un graphe, $\mathcal{S}' \subset \mathcal{S}$, on appelle sous graphe engendré par \mathcal{S}' le graphe $\vec{\mathcal{G}}' = (\mathcal{S}', \vec{\mathcal{E}}')$ où $\vec{\mathcal{E}}'$ est l’ensemble d’arcs dans $\vec{\mathcal{G}}$ ayant les deux extrémités dans \mathcal{S}' .

modèle a des avantages par rapport à l'autre, comme mentionné par Lauritzen [75]. Les modèles graphiques orientés acycliques ont un grand intérêt pour modéliser les relations asymétriques ou dites « relations de causalité ». Les modèles graphiques non-orientés ont un grand intérêt pour modéliser les relations symétriques d'interaction entre les variables aléatoires. Les deux ensembles des modèles graphiques orientés et non-orientés ne sont pas égaux : Smyth et al. [113] donnent un exemple de modèle graphique orienté acycliques qui n'admet pas de modèle graphique non-orienté équivalent (un graphe orienté $\vec{\mathcal{G}} = (\mathcal{S}, \vec{\mathcal{E}})$ et un graphe non orienté $\mathcal{G} = (\mathcal{S}, \mathcal{E})$ sont dits « équivalents » en terme d'indépendance conditionnelle si et seulement si pour tous sous-ensembles \mathcal{B}, \mathcal{C} et \mathcal{D} disjoints deux à deux non-vides de \mathcal{S} , \mathcal{C} sépare \mathcal{B} et \mathcal{D} dans $\vec{\mathcal{G}} \Leftrightarrow \mathcal{C}$ sépare \mathcal{B} et \mathcal{D} dans \mathcal{G}). Il donne également un exemple de graphe non-orienté qui n'admet pas de modèle graphique orienté acyclique équivalent.

I.1.3 Factorisation d'une loi selon un graphe

Il n'est pas toujours possible de définir un graphe d'indépendance conditionnelle pour un processus X . Smyth et al. [113] donnent des exemples de lois de probabilité qui n'admettent une représentation graphique ni par modèle orienté ni par modèle non-orienté. Réciproquement, il est parfois possible d'associer plusieurs lois à un graphe d'indépendance conditionnelle.

Soit $X = (X_n)_{1:N}$ un processus aléatoire. Les X_n sont à valeurs dans un espace \mathcal{X} , et ils sont indexés par un ensemble fini des sommets \mathcal{S} d'un graphe d'indépendance conditionnelle \mathcal{G} . La densité de la loi de X se factorise selon le graphe \mathcal{G} par rapport à la mesure de référence $\nu_{\mathcal{X}} = \bigotimes_{n \in \mathcal{S}} \nu_{\mathcal{X}_n}$, si elle s'écrit :

$$p(x) = \prod_{c \in \mathcal{C}} p_c(x_c) \quad (\text{I.8})$$

où \mathcal{C} est l'ensemble des cliques du graphe \mathcal{G} . $\forall c \in \mathcal{C}$, p_c est une fonction de $\mathcal{X}^{|c|}$ dans \mathbb{R}^+ .

Le développement de la loi d'un processus selon son graphe d'indépendance conditionnelle permet d'aborder des problèmes d'estimation et de segmentation dans certains modèles à données incomplètes. Dans toute la suite, nous utilisons des modèles graphiques non-orientés « minimaux », dans lesquels deux points s et t ne sont pas reliés par une arête si et seulement si les deux variables aléatoires associées X_s et X_t sont indépendantes conditionnellement aux autres. Le graphe minimal prend en compte de l'indépendance conditionnelle « par paires » pour l'ensemble des variables considérées [88].

I.2 Chaîne de Markov cachée

Dans cette section, nous traitons un modèle de Markov caché parmi les plus simples, qui est la chaîne de Markov cachée (CMC). Nous distinguons deux cas : celui où le processus caché X est un processus discret et celui où c'est un processus réel continu.

Les chaînes de Markov cachées sont des modèles graphiques comportant des données observées et des données latentes. On note $y = (y_n)_{1:N}$ les données observées, qui sont issues de la réalisation d'un processus $Y = (Y_n)_{1:N}$ sur un réseau mono-dimensionnel \mathcal{S} , et $x = (x_n)_{1:N}$ les données latentes, qui sont issues de la réalisation d'un processus $X = (X_n)_{1:N}$.

En premier lieu, nous étudions le cas des chaînes mono-dimensionnelles à espace d'état discret : les X_n sont à valeurs dans le même espace discret $\mathcal{X} = \{\omega_1, \dots, \omega_k\}$. En second lieu, nous étudions le cas où les X_n sont à valeurs dans \mathbb{R} .

I.2.1 CMC à espace d'état discret

Définition I.7 Soit $X = (X_n)_{1:N}$ un processus aléatoire, avec les X_n à valeurs dans un espace discret $\mathcal{X} = \{\omega_1, \dots, \omega_k\}$. On dit que X est une chaîne de Markov si et seulement si il admet comme graphe d'indépendance conditionnelle non-orienté, le graphe où chaque site n admet son prédécesseur $n - 1$ (si $n > 0$) et son successeur $n + 1$ (si $n < N$) comme voisins (voir figure (I.4)).

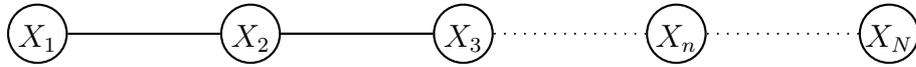


FIGURE I.4 – Graphe d'indépendance conditionnelle non-orienté d'une chaîne de Markov

Selon la remarque (I.1), le processus X satisfait les propriétés suivantes :

- la propriété de Markov globale : $\forall n \geq 1$, $X_{1:n-1}$ et $X_{n+1:N}$ sont indépendants conditionnellement à X_n :

$$p(x_{1:n-1}, x_{n+1:N} | x_n) = p(x_{1:n-1} | x_n) p(x_{n+1:N} | x_n); \quad (\text{I.9})$$

- La propriété de Markov locale : X_n et les $(X_t)_{t \neq \{n, n-1, n+1\}}$ sont indépendantes conditionnellement à (X_{n-1}, X_{n+1}) :

$$p(x_n | x_{t \neq \{n, n-1, n+1\}}) = p(x_n | x_{n-1}, x_{n+1}); \quad (\text{I.10})$$

- La propriété de Markov par paires : si i n'appartient pas au voisinage de n , alors X_n et X_i sont indépendants conditionnellement au reste des variables :

$$p(x_n, x_i | x_{t \neq \{n, i\}}) = p(x_n | x_{t \neq \{n, i\}}) p(x_i | x_{t \neq \{n, i\}}) \quad (\text{I.11})$$

Une chaîne de Markov est dite homogène si les transitions $p(x_{n+1}|x_n)$ ne dépendent pas de n . La loi d'une chaîne de Markov homogène est alors entièrement déterminée par sa loi initiale $p(x_1)$ et les transitions $p(x_{n+1}|x_n)$. On dit qu'une chaîne de Markov est stationnaire si elle est homogène et si la loi marginale $p(x_n)$ ne dépend pas de n . La loi initiale est alors appelée loi invariante ou stationnaire. Notons qu'il est possible de factoriser la loi d'une chaîne de Markov selon son graphe d'indépendance conditionnelle. En effet, si X est une chaîne de Markov admettant pour graphe d'indépendance conditionnelle (I.4), nous obtenons avec la propriété de Markov globale : X_N et $X_{1:N-2}$ sont indépendants conditionnellement à X_{N-1} :

$$p(x_{1:N}) = p(x_{1:N-2}|x_{N-1})p(x_N|x_{N-1}) \quad (\text{I.12})$$

par récurrence sur la chaîne $X_{1:N-2}$ nous obtenons alors :

$$p(x_{1:N}) = p(x_1) \prod_{n=1}^{N-1} p(x_{n+1}|x_n) \quad (\text{I.13})$$

Dans toute la suite, on note p_i pour désigner $p(x_1 = \omega_i)$, et $\mathcal{P} = (p_{i,j})_{k \times k}$ pour désigner la matrice des transitions $p_{i,j} = p(x_{n+1} = \omega_i | x_n = \omega_j)$.

Dans toute la suite nous serons amenés à effectuer des simulations des réalisations des chaînes de Markov. Ces simulations se font directement : il suffit de fixer la loi initiale p_i et la matrice \mathcal{P} . Nous commençons par simuler x_1 , ensuite pour tout n , on effectue un tirage aléatoire de x_n selon sa loi $p(x_n|x_{n-1})$ conditionnelle à x_{n-1} .

Exemple

Considérons l'exemple de simulation suivant. Nous simulons une chaîne X de taille $N = 128 \times 128$ à deux états $\mathcal{X} = \{\omega_1, \omega_2\}$, dont la loi admet pour matrice de transition \mathcal{P} :

$$\mathcal{P} = \begin{pmatrix} 0.98 & 0.02 \\ 0.02 & 0.98 \end{pmatrix}$$

La loi invariante correspond au vecteur propre gauche associé à la valeur propre 1 de la matrice \mathcal{P} . Le vecteur ligne π d'éléments p_i correspondant à la loi invariante et est obtenu en résolvant l'équation matricielle :

$$\pi \mathcal{P} = \pi. \quad (\text{I.14})$$

Dans le cas de la matrice \mathcal{P} , la loi invariante est donnée par $\pi = (\frac{1}{2}, \frac{1}{2})$.

Pour visualiser la chaîne simulée sous forme d'une image 2-D, nous allons représenter la réalisation par le parcours d'Hilbert-Peano. Le parcours d'Hilbert-Peano [7, 83] est souvent utilisé en traitements d'images [110] dans le but de transformer des données bidimensionnelles en données monodimensionnelles. Le parcours est représenté par la figure (I.5). L'avantage de cette transformation est qu'elle permet de garder deux points, voisins dans la chaîne, voisins dans l'image. Cependant, deux points voisins dans l'image peuvent ne pas l'être dans la chaîne [7]. D'autres types de transformations sont connus dans la littérature comme la transformation en ligne ou en colonne, ou encore la transformation de Hilbert, qui ressemble à celui de Hilbert-Peano.

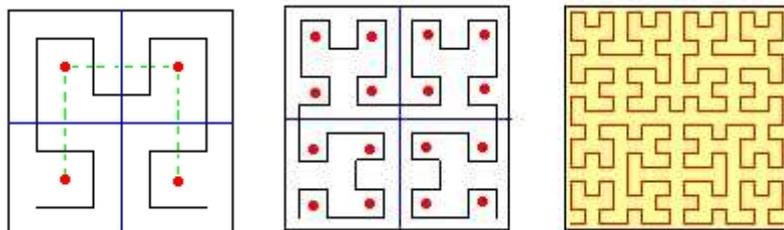


FIGURE I.5 – Construction du parcours d'Hilbert Péano



FIGURE I.6 – Exemple de simulation d'une chaîne de Markov : image obtenue avec transformation d'Hilbert-Péano

Définition I.8 Soit $(X, Y) = (X_n, Y_n)_{1:N}$ un processus aléatoire couple où les X_n prennent leurs valeurs dans un espace discret $\mathcal{X} = \{\omega_1, \dots, \omega_K\}$ et les Y_n sont à valeurs dans $\mathcal{Y} = \mathbb{R}$. (X, Y) est appelée "chaîne de Markov cachée à bruit indépendant" (CMC-BI), si X est une chaîne de Markov, et la densité de la loi du Y conditionnellement à $X = x$ vérifie :

□ les Y_n sont indépendants conditionnellement à $X_{1:N}$:

$$p(y_{1:N}|x_{1:N}) = \prod_{n=1}^N p(y_n|x_{1:N}) \quad (\text{I.15})$$

□ $\forall n \in [1, \dots, N]$

$$p(y_n|x_{1:N}) = p(y_n|x_n) \quad (\text{I.16})$$

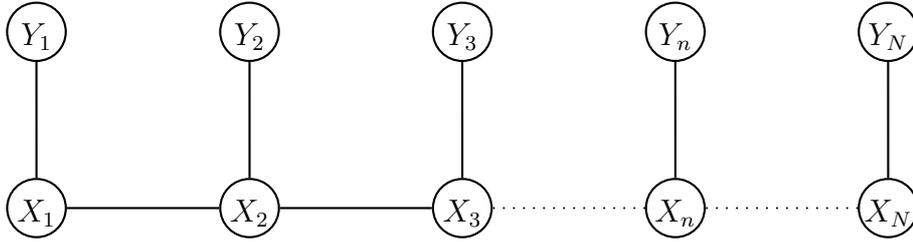


FIGURE I.7 – Graphe d'indépendance conditionnelle d'une chaîne de Markov cachée à bruit indépendant.

Le modèle de la chaîne de Markov cachée à bruit indépendant (CMC-BI) est souvent utilisé pour l'estimation des données latentes. Il existe plusieurs algorithmes d'estimation, qui se basent sur la possibilité du calcul des probabilités $p(x_n|y_{1:N})$ et $p(x_{n+1}|x_n, y_{1:N})$. Nous détaillons dans la suite certains algorithmes qui permettent ces calculs.

I.2.2 Inférence Bayésienne dans la CMC discrète

Pour calculer les probabilités a posteriori $p(x_n|y_{1:N})$, l'un des algorithmes les plus utilisés est celui de Baum-Welsh. Cet algorithme comporte deux étapes : la première est une étape de « filtrage » et la deuxième est une étape de « lissage », termes qui seront précisés par la suite. En utilisant la propriété de Markov globale, la loi $p(x_n, y_{1:N})$ se développe, sous la forme :

$$p(x_n, y_{1:N}) = p(x_n, y_{1:n})p(y_{n+1:N}|x_n, y_{1:n}). \quad (\text{I.17})$$

Si on note :

$$\alpha_n(x_n) = p(x_n, y_{1:n}), \quad (\text{I.18})$$

et

$$\beta_n(x_n) = p(y_{n+1:N}|x_n, y_n), \quad (\text{I.19})$$

nous obtenons $p(x_n|y_{1:N})$ en fonction des α_n et des β_n :

$$p(x_n|y_{1:N}) = \frac{\alpha_n(x_n)\beta_n(x_n)}{\sum_{j=1}^k \alpha_n(x_n = \omega_j)\beta_n(x_n = \omega_j)}. \quad (\text{I.20})$$

De même pour $p(x_{n+1}|x_n, y_{1:N})$, on a :

$$p(x_{n+1}|x_n, y_{1:N}) = \frac{p(x_{n+1}, x_n, y_{1:N})}{p(x_n, y_{1:N})},$$

avec

$$\begin{aligned} p(x_{n+1}, x_n, y_{1:N}) &= p(y_{n+2:N}|x_{n+1}, y_{n+1})p(x_{n+1}, y_{n+1}|x_n, y_n)p(x_n, y_{1:n}) \\ &= \beta_{n+1}(x_{n+1})p(x_{n+1}, y_{n+1}|x_n, y_n)p(x_n, y_{1:n}) \end{aligned}$$

On a $p(x_{n+1}, y_{n+1}|x_n, y_n) = p(x_{n+1}, y_{n+1}|x_n)$ du fait que c'est une chaîne de Markov à bruit indépendant. La loi $p(x_n, y_{1:N})$ se développe comme dans (I.17) et comme $p(y_{n+1:N}|x_n, y_n) = \beta_n(x_n)$ alors $p(x_n, y_{1:N}) = \beta_n(x_n)p(x_n, y_{1:n})$. nous pouvons exprimer $p(x_{n+1}|x_n, y_{1:N})$ en fonction des β_n et des transitions $p(x_{n+1}, y_{n+1}|x_n)$:

$$p(x_{n+1}|x_n, y_{1:N}) = \frac{\beta_{n+1}(x_{n+1})p(x_{n+1}, y_{n+1}|x_n)p(x_n, y_{1:n})}{\beta_n(x_n)p(x_n, y_{1:n})},$$

et après simplification :

$$p(x_{n+1}|x_n, y_{1:N}) = \frac{\beta_{n+1}(x_{n+1})}{\beta_n(x_n)}p(x_{n+1}, y_{n+1}|x_n). \quad (\text{I.21})$$

La densité de la loi du couple (X_{n+1}, X_n) conditionnellement à $y_{1:N}$ s'écrit :

$$p(x_{n+1}, x_n|y_{1:N}) = p(x_n|y_{1:N})p(x_{n+1}|x_n, y_{1:N}) \quad (\text{I.22})$$

En utilisant les équations (I.20) et (I.21) on obtient :

$$p(x_{n+1}, x_n|y_{1:N}) \propto \beta_{n+1}(x_{n+1})\alpha_n(x_n)p(x_{n+1}, y_{n+1}|x_n) \quad (\text{I.23})$$

Le point important est que les probabilités α_n et β_n sont calculables par les récurrences suivantes :

1. calcul des α_n (récurrence directe) :
 - Initialisation : $\alpha_1(x_1) = p(x_1, y_1)$;
 - Supposons α_n calculé dans l'étape précédente (n) :

$$\begin{aligned} \alpha_{n+1}(x_{n+1}) &= p(x_{n+1}, y_{1:n+1}) \\ &= \sum_{j=1}^k p(x_n = \omega_j, x_{n+1}, y_{1:n+1}) \end{aligned}$$

la formule de récurrence est la suivante :

$$\alpha_{n+1}(x_{n+1}) = \sum_{j=1}^k \alpha_n(x_n = \omega_j)p(x_{n+1}, y_{n+1}|x_n = \omega_j) \quad (\text{I.24})$$

2. calcul des β_n (récurrence rétrograde) :
 - Initialisation : $\beta_N(x_N) = 1$;
 - Supposons $\beta_{n+1}(x_{n+1})$ calculé dans l'étape précédente ($n+1$) :

$$\begin{aligned} \beta_n(x_n) &= p(y_{n+1:N}|x_n, y_n) \\ &= \sum_{j=1}^k p(x_{n+1} = \omega_j, y_{n+1:N}|x_n, y_n) \end{aligned}$$

la formule de récurrence est la suivante :

$$\beta_n(x_n) = \sum_{j=1}^k \beta_{n+1}(x_{n+1} = \omega_j)p(x_{n+1}, y_{n+1}|x_n). \quad (\text{I.25})$$

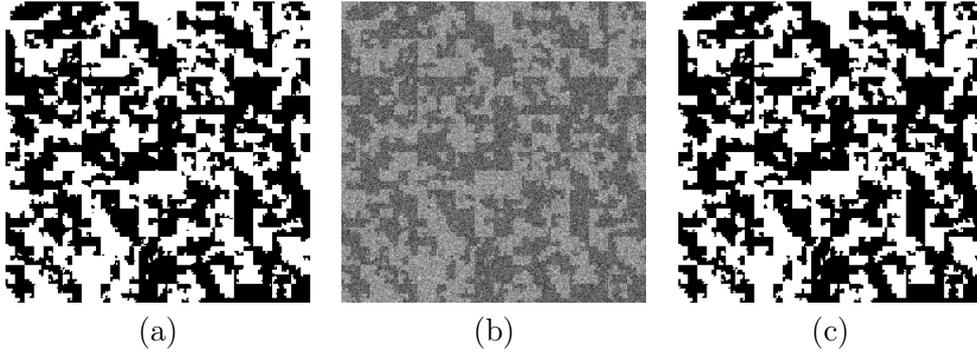


FIGURE I.8 – Exemple : simulation et segmentation d’une chaîne CMC-BI (a) X simulé à 2 classes $\{\omega_1, \omega_2\}$, $p_i = 0.5$ et $p(\omega_i|\omega_i) = 0.99$ (b) Y simulé avec $\mathcal{N}_{\omega_1}(0.0, 1.0)$ et $\mathcal{N}_{\omega_2}(2.0, 1.0)$ (c) X estimé avec les vrais paramètres (taux d’erreur 0.61%)

Lors de l’implémentation de l’algorithme de Baum-Welsh, des problèmes d’ordre numérique peuvent apparaître car les α_n tendent vers 0 lorsque n est suffisamment grand et β_1 tend également vers 0 lorsque la taille N de l’échantillon tend vers l’infini. Pour surmonter ces problèmes, des variations de cet algorithme ont été proposées, comme l’algorithme de Baum-Welsh conditionnel [35, 37].

I.3 CMC à espace d’états continu

Dans cette section, nous nous intéressons au cas des modèles de chaînes de Markov cachées où les données latentes sont réelles. La définition de la markovianité est, bien entendu, identique à celle des chaînes discrètes :

Définition I.9 Soit $X = (X_n)_{1:N}$ un processus aléatoire, dont les marginales X_n sont à valeurs dans \mathbb{R} . X est une chaîne de Markov si et seulement si il admet comme graphe d’indépendance conditionnelle non-orienté le graphe représente par la figure (I.4).

Comme nous l’avons déjà mentionné dans l’exemple (I.1), le processus X qui est un AR(1) est une chaîne de Markov à espace d’état continu. L’écriture des processus AR(1) se généralise à l’écriture suivante

$$\forall n \in \{2, \dots, N\}, \quad X_{n+1} = f(X_n, \epsilon_n), \quad (\text{I.26})$$

où les ϵ_n sont identiquement distribués dans le cas d’une chaîne homogène. Une telle écriture admet souvent des interprétations physiques en termes de la dépendance entre les variables cachées et le « bruit », et est de ce fait souvent utilisée. Cependant, nous continuerons à utiliser les mêmes notations $p(\cdot|x_n)$ pour les transitions : la transition $p(\cdot|x_n)$ est alors la densité de la variable aléatoire $f(x_n, \epsilon_n)$. Comme dans le cas discret, la densité $p(x)$ est factorisée selon :

$$p(x) = p(x_1) \prod_{n=1}^{N-1} p(x_{n+1}|x_n). \quad (\text{I.27})$$

La définition suivante, rappelée pour mémoire, est identique à celle définissant les CMC-BI discrètes :

Définition I.10 Soit $(X, Y) = (X_n, Y_n)_{1:N}$ un processus aléatoire couple à valeurs dans l'espace produit $\mathbb{R} \times \mathbb{R}$. (X, Y) est dit une chaîne de Markov cachée à bruit indépendant, si seulement si X est une chaîne de Markov, et la loi $p(y|x)$ vérifie :

□ les Y_n sont indépendants conditionnellement aux $X_{1:N}$:

$$p(y_{1:N}|x_{1:N}) = \prod_{n=1}^N p(y_n|x_{1:N}), \quad (\text{I.28})$$

□ $\forall n \in \{1, \dots, N\}$:

$$p(y_n|x_{1:N}) = p(y_n|x_n). \quad (\text{I.29})$$

I.3.1 Inférence Bayésienne

Le calcul des probabilités à posteriori de X_n conditionnellement à $y = y_{1:N}$ n'est pas toujours possible d'une manière exacte. Bien que les démarches générales soient les mêmes que dans le cas des CMC-BI, les sommes sont remplacées par des intégrales ne pouvant pas être calculées explicitement dans tous les cas. En effet, si on désigne par $\alpha_n = p(x_n, y_{1:n})$ les probabilités « Forward » et par $\beta_n = p(y_{n+1:N}|x_n, y_{1:n})$ les probabilités « Backward », alors :

$$p(x_1|y) \propto \alpha_1(x_1)\beta_1(x_1) \quad (\text{I.30})$$

et pour tout $n \in \{2, \dots, N\}$

$$p(x_n|y) \propto \alpha_n(x_n)\beta_n(x_n), \quad (\text{I.31})$$

avec α_n et β_n sont obtenus d'une manière recursive :

1. Calcul de $\alpha_n(\cdot)$:

□ Initialisation : $\alpha_1(x_1) = p(x_1, y_1)$,

□ Supposons α_n calculé à l'étape n :

$$\alpha_{n+1}(x_{n+1}) = \int_{\mathbb{R}} \alpha_n(x_n) p(x_{n+1}, y_{n+1}|x_n) dx_n. \quad (\text{I.32})$$

2. Calcul de $\beta_n(\cdot)$

□ Initialisation : $\beta_N(x_N) = 1$,

□ Supposons β_{n+1} calculé à l'étape $n+1$:

$$\beta_n(x_n) = \int_{\mathbb{R}} \beta_{n+1}(x_{n+1}) p(x_{n+1}, y_{n+1}|x_n) dx_{n+1}. \quad (\text{I.33})$$

Nous traitons dans la suite le cas d'une chaîne de Markov cachée gaussienne. Dans ce cas, nous avons les propriétés suivantes :

- Pour tout n , X_n est une variable aléatoire gaussienne ;
- Conditionnellement à $X_n = x_n$, le noyau de transition $p(\cdot|x_n)$ est la densité d'une loi gaussienne.

Dans ce cas, le calcul de α_n et de β_n est possible de manière directe par le filtrage de Kalman et l'algorithme de Rauch-Tung-Striebel (RTS) [105], qui est un algorithme de lissage. Considérons l'écriture suivante du modèle :

$$\begin{cases} X_{n+1} &= F X_n + \epsilon_n, \\ Y_n &= m_y + H X_n + \xi_n \end{cases} \quad (\text{I.34})$$

avec $X_1 \sim \mathcal{N}(m_1, V_1)$, et les $\epsilon_n \sim \mathcal{N}(0, R)$ identiquement distribués et tels que $\epsilon_n \perp x_n$, les $\xi_n \sim \mathcal{N}(0, Q)$ et tels que $\xi_n \perp x_n$. Dans la sous-section suivante, nous détaillons le filtre de Kalman.

I.3.2 Filtre de Kalman

Le filtre de Kalman permet de calculer les probabilités « Forward » $p(x_n|y_{1:n})$ en deux étapes : une étape de « prédiction » qui permet de calculer la loi de X_n conditionnellement à X_{n-1} et $Y_{1:n-1}$, et une étape de « correction », qui permet d'intégrer les informations apportées par l'observation de Y_n .

Prédiction

Le couple (X, Y) est gaussien, donc la loi de X_n conditionnellement à $Y_{1:n-1} = y_{1:n-1}$ est une gaussienne de moyenne $\mathbb{E}[X_n|y_{1:n-1}] = m_n^-$ et de variance $\mathbb{V}[X_n|y_{1:n-1}] = P_n^-$. De même, la loi de X_n conditionnellement à $y_{1:n}$ est une gaussienne de moyenne $\mathbb{E}[X_n|y_{1:n}] = m_n$ et de variance $\mathbb{V}[X_n|y_{1:n}] = P_n$. D'après les équations du système (I.34), nous pouvons calculer m_n^- et P_n^- en fonction de m_{n-1} et P_{n-1} :

$$m_n^- = \mathbb{E}[X_n|y_{1:n-1}] = \mathbb{E}[(FX_{n-1} + \epsilon_{n-1})|y_{1:n-1}],$$

comme $\epsilon_{n-1} \perp\!\!\!\perp Y_{1:n-1}$, et $\mathbb{E}[X_{n-1}|y_{1:n-1}] = m_{n-1}$, nous obtenons :

$$m_n^- = Fm_{n-1}, \quad (\text{I.35})$$

de même pour la variance :

$$P_n^- = F^2P_{n-1} + R. \quad (\text{I.36})$$

Correction

Dans cette étape, nous utilisons les informations apportées par l'observation de Y_n , et nous cherchons la formule de récurrence pour calculer m_n et P_n en fonction de m_n^- et P_n^- . D'après les hypothèses, X_n conditionnellement à $y_{1:n-1}$ est une gaussienne, de même pour $Y_n|y_{1:n-1}$. Nous commençons par calculer la moyenne de $Y_n|y_{1:n-1}$ et sa variance, puis la covariance entre $X_n|y_{1:n-1}$ et $Y_n|y_{1:n-1}$. En utilisant toujours les équations du système (I.34) :

$$\begin{aligned} \mathbb{E}[Y_n|y_{1:n-1}] &= \mathbb{E}[(m_y + HX_n + \xi_n)|y_{1:n-1}] \\ &= m_y + Hm_n^- \\ \mathbb{V}[Y_n|y_{1:n-1}] &= \mathbb{E}[(Y_n - \mathbb{E}[Y_n|y_{1:n-1}])^2|y_{1:n-1}] \\ &= H^2P_n^- + Q \\ \mathbb{C}[(X_n, Y_n)|y_{1:n-1}] &= \mathbb{E}[(X_n - \mathbb{E}[X_n|y_{1:n-1}])(Y_n - \mathbb{E}[Y_n|y_{1:n-1}])] \\ &= HP_n^- \\ &= k_n \end{aligned}$$

Comme conséquence, nous obtenons la loi du couple (X_n, Y_n) conditionnellement à $y_{1:n-1}$, qui est une gaussienne $\mathcal{N}_2(M, V)$ avec :

$$M = \begin{pmatrix} m_n^- \\ m_y + Hm_n^- \end{pmatrix} \quad \text{et} \quad V = \begin{pmatrix} P_n^- & k_n \\ k_n & H^2P_n^- + Q \end{pmatrix}$$

Finalement, nous obtenons la loi du X_n conditionnellement à $y_{1:n}$ à partir de la loi du couple, $X_n|y_{1:n} \sim \mathcal{N}(m_n, P_n)$ avec $k_n = HP_n^-$:

$$m_n = m_n^- + k_n[y_n - m_y - Hm_n^-][H^2P_n^- + Q]^{-1} \quad (\text{I.37})$$

$$P_n = P_n^- - k_n[H^2P_n^- + Q]^{-1}k_n^T \quad (\text{I.38})$$

Une fois calculées les moyennes m_n et les variances P_n , nous pouvons passer à l'étape suivante, qui est l'étape de lissage qui a pour but de calculer les moyennes et les variances de $X_n|y_{1:N}$.

I.3.3 Lissage

Le lissage RTS a pour but de déterminer la loi du $X_n|y_{1:N}$ et la loi de $X_n|x_{n+1}, y_{1:N}$ dans le cas d'un système linéaire gaussien de type (I.34). Ces lois étant des gaussiennes, $X_n|y_{1:N} \sim \mathcal{N}(m_n^*, P_n^*)$ et $X_n|x_{n+1}, y_{1:N} \sim \mathcal{N}(m_n^+, P_n^+)$. Les moyennes m_n^* , m_n^+ et les variances P_n^* , P_n^+ peuvent être calculées par récurrence rétrograde. Comme initialisation, nous prenons les valeurs obtenues avec le filtre de Kalman :

$$m_N^* = m_N^+ = m_N \quad (\text{I.39})$$

$$P_N^* = P_N^+ = P_N \quad (\text{I.40})$$

Sur le graphe d'indépendance conditionnelle non-orienté (I.9), (B) sépare X_n (A) et $Y_{n+1:N}$ (C), donc nous avons :

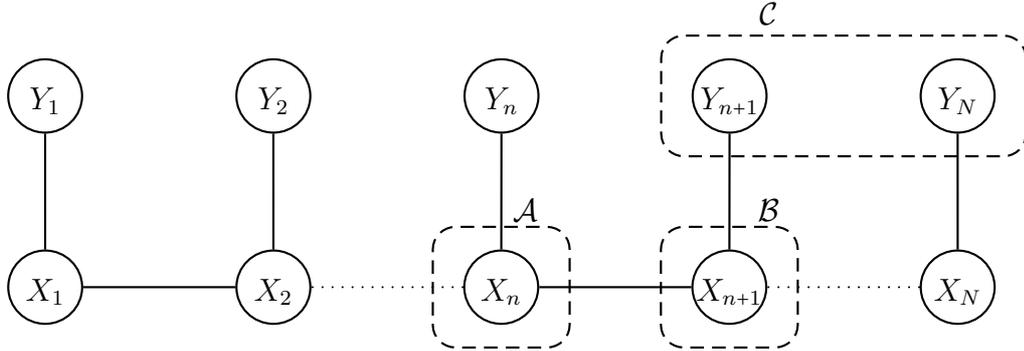


FIGURE I.9 – Graphe moral obtenu à partir du graphe d'indépendance conditionnelle non-orienté

$$p(x_n|x_{n+1}, y_{1:N}) = p(x_n|x_{n+1}, y_{1:n})$$

Le calcul de la covariance du couple (X_n, X_{n+1}) conditionnelle à $Y_{1:n}$, qu'on note C_n ,

donne :

$$\begin{aligned}
 C[(X_n, X_{n+1})|y_{1:n}] &= \mathbb{E}[(X_n - \mathbb{E}[X_n|y_{1:n}])(X_{n+1} - \mathbb{E}[X_{n+1}|y_{1:n}])] \\
 &= \mathbb{E}[(X_n - m_n)(X_{n+1} - m_{n+1}^-)] \\
 &= \mathbb{E}[(X_n - m_n)(FX_n + \epsilon_n - Fm_n)] \\
 &= F\mathbb{E}[(X_n - m_n)^2] \\
 &= FP_n \\
 &= C_n
 \end{aligned}$$

En utilisant la propriété du calcul de loi conditionnelle pour les vecteurs gaussiens, nous pouvons en déduire la moyenne m_n^+ et la variance P_n^+ de $X_n|x_{n+1}y_{1:n}$:

$$m_n^+ = m_n + C_n[x_{n+1} - m_{n+1}^-][P_{n+1}^-]^{-1} \quad (\text{I.41})$$

$$P_n^+ = P_n - C_n[P_{n+1}^-]^{-1}C_n^T \quad (\text{I.42})$$

De même, nous pouvons en déduire la loi du $x_n|y_{1:N}$:

$$L_n^* = FP_nP_{n+1}^{-1} \quad (\text{I.43})$$

$$m_n^* = m_n + L_n^*[m_{n+1}^* - m_{n+1}^-] \quad (\text{I.44})$$

$$P_n^* = P_n - L_n^*[P_{n+1}^* - P_{n+1}^-](L_n^*)^T \quad (\text{I.45})$$

ce qui était notre objectif.

Exemple

Dans cet exemple nous simulons des réalisations d'un système linéaire gaussien qui a comme dynamique le système (I.34), et nous présentons des résultats de filtrage par le filtre de Kalman et de lissage RTS. Les paramètres de simulation sont présentés dans la table 1.1.

Paramètres	Valeur
m_1	0
V_1	1
F	1
R	1
m_y	0
H	2
Q	0.1

TABLE I.1 – Paramètres de simulation

Les réalisations $x_{1:N}$ et $y_{1:N}$ sont représentées sur la figure (I.10) avec $N = 100$. La courbe supérieure est celle de $y_{1:N}$ et la courbe inférieure est celle de $x_{1:N}$. Sur la figure (I.11) on trouve l'estimation $\hat{x}_{1:N}$ à partir de l'observation $y_{1:N}$ (a) est l'estimation avec le filtre de Kalman et (b) l'estimation par le lissage RTS.

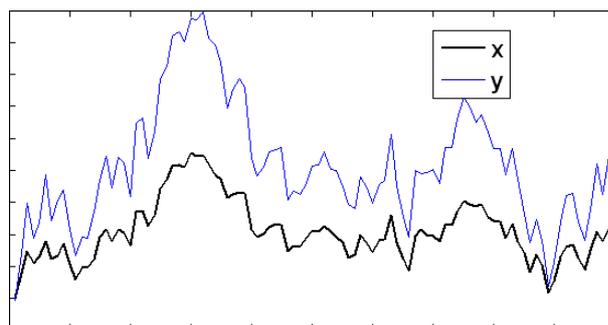


FIGURE I.10 – Simulation système linéaire gaussien

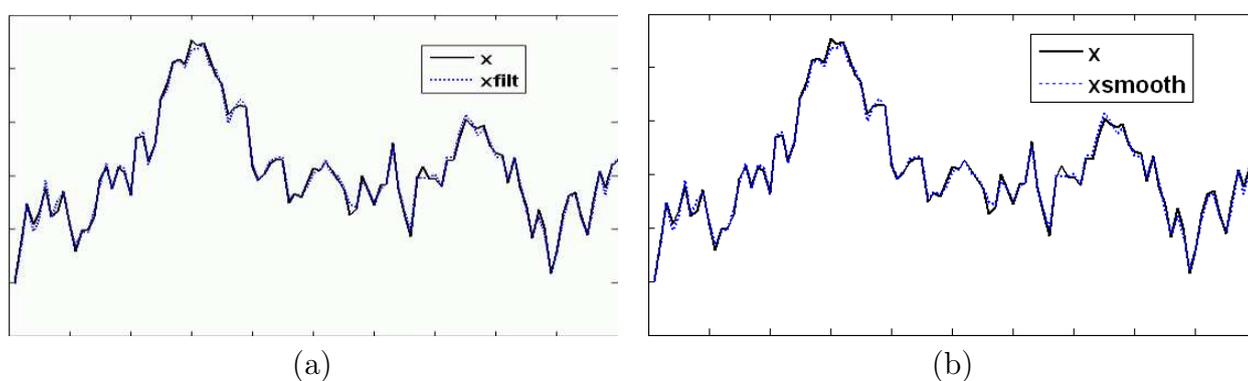


FIGURE I.11 – (a) Estimation par filtrage de Kalman (b) estimation par lissage RTS

On remarque que les trajectoires estimées par le filtre de Kalman et le lissage RTS, sont presque identiques et elles sont très proches de la vraie trajectoire. Cela est dû au fait que la variance de ξ_n est petite $Q = 0.1$; le système est donc peu « bruité ». Dans l'exemple suivant nous considérons $Q = 1$ et nous gardons les mêmes valeurs pour les autres paramètres. Les réalisations $x_{1:N}$ et $y_{1:N}$ sont représentées sur la figure (I.12).

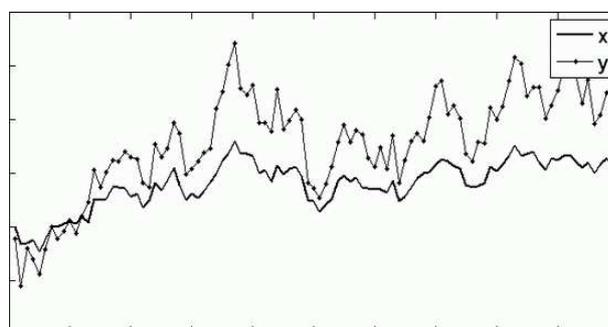


FIGURE I.12 – Simulation système linéaire gaussien

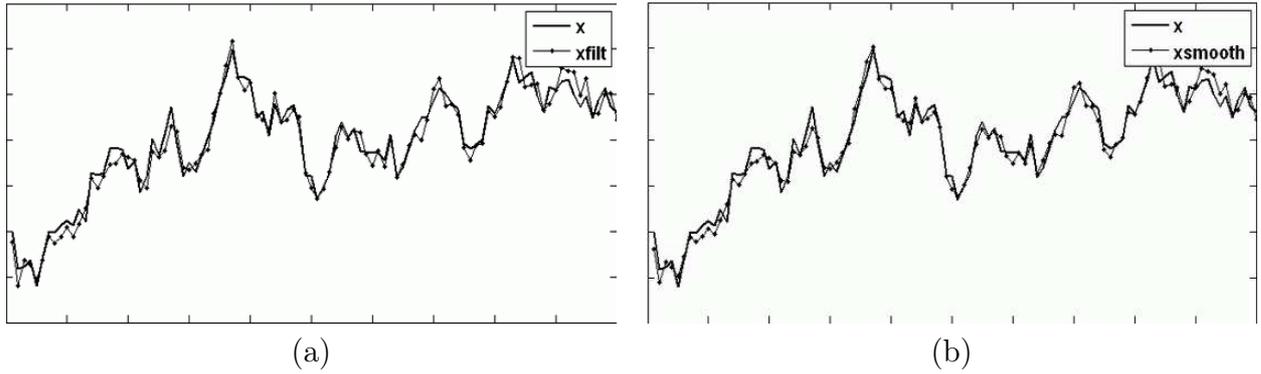


FIGURE I.13 – (a) Estimation par filtrage de Kalman (b) estimation par lissage RTS

Nous pouvons remarquer que l'estimation de la trajectoire $\widehat{x}_{1:N}$ par le lissage RTS (I.13).b est plus proche du signal caché que son estimation par le filtrage de Kalman (I.13).a. Bien entendu, cela est dû au fait que le lissage prend en considération toutes les observations $y_{1:N}$ pour estimer x_n tandis que le filtrage prend en considération que les observations de 1 jusqu'à n i.e $y_{1:n}$.

Considérons une CMC-BI générale, qui peut se mettre sous la forme du système suivant :

$$(\mathcal{E}') : \begin{cases} X_{n+1} &= f(X_n, \epsilon_n) \\ Y_n &= g(X_n, \xi_n) \end{cases} \quad (\text{I.46})$$

avec les fonctions f et g connues. Dans ce cas « non linéaire » et « non gaussien », il est possible de proposer une extension du filtre de Kalman, qui est le filtre de Kalman « étendu » (ou « linéarisé »), qui consiste à linéariser le système en remplaçant les equations par leurs développements de Taylor. Pour tout n dans $[1, \dots, N]$, on développe les equations du système (I.46) à l'ordre 1 au point X_n^0 , et nous obtenons le système sous sa forme linéarisée (I.47), en suite on peut appliquer les mêmes algorithmes déjà présentés sur le nouveau système qui est linéaire en x .

$$(\mathcal{E}'') : \begin{cases} X_{n+1} &= f(X_n^0, \epsilon_n) + (X_n - X_n^0) \frac{\partial f}{\partial x}(X_n^0, \epsilon_n) \\ Y_n &= g(X_n^0, \xi_n) + (X_n - X_n^0) \frac{\partial g}{\partial x}(X_n^0, \xi_n) \end{cases} \quad (\text{I.47})$$

Le filtrage particulière, très utilisé actuellement, est une autre alternative pour traiter le système non linéaire non gaussien ci-dessus.

I.4 Conclusion

Dans ce chapitre, nous avons tout d'abord rappelé diverses définitions classiques de la modélisation graphique. En particulier, nous avons brièvement discuté de relations entre les graphes orientés et les graphes non orientés. Ce type de modélisation permet de décrire d'une manière visuelle les relations d'indépendance conditionnelle entre les variables aléatoires X_n d'un processus X observé sur un réseau $\mathcal{S} = \{1, \dots, N\}$. En second lieu, nous avons utilisé ces modèles pour définir les modèles de chaînes de Markov cachées, dans les deux cas des données cachées discrètes et continues. Dans ces deux cas nous avons détaillé des algorithmes d'inférence bayésienne permettant de calculer la loi a posteriori $p(x_n|y_{1:N})$ et $p(x_{n+1}|x_n, y_{1:N})$. Pour

le modèle de chaîne cachée à états finis nous avons détaillé l'algorithme de Baum-Welsh; pour celui de chaîne cachée à états continus nous avons détaillé le filtre de Kalman et le lissage RTS. Nous avons donné également quelques exemples de simulation pour ces différents modèles et traitements classiques.

Dans le chapitre qui suit nous présentons un autre type de modèles de Markov, dits « modèles de Markov couples », qui est une généralisation du modèle de chaîne cachée classique [91, 92, 101].

Chapitre II

Modèle de Markov Couple

Dans le modèle de chaîne de Markov cachée classique, nous supposons que les données cachées ont une structure markovienne $p(x) = p(x_1) \prod_{n=1}^{N-1} p(x_{n+1}|x_n)$. D'une part, cette hypothèse n'est pas toujours vérifiée dans le cas des données issues du monde réel. D'autre part, il s'avère qu'elle n'est pas nécessaire pour la mise en oeuvre des traitements bayésiens discutés dans le chapitre précédent. Pour alléger cette hypothèse et prendre en considération une situation plus générale Pieczynski [91, 104] a proposé de remplacer l'hypothèse de la markovianité de processus X par la markovianité du couple (X, Y) . Cette hypothèse permet d'assurer la markovianité de X conditionnellement à Y qui est essentielle pour appliquer les algorithmes de calcul pour $p(x_n|y_{1:N})$ et $p(x_{n+1}|x_n, y_{1:N})$. Cependant, le processus caché peut être markovien ou pas, ce qui mène à un modèle plus général. Ce nouveau modèle est dit « modèle de chaîne de Markov Couple » (CMCouple). Dans la suite, nous présentons ce modèle d'une manière détaillée ainsi que les algorithmes d'inférence Bayésienne qui permettent de calculer les probabilités a posteriori d'intérêt. Nous comparons également le CMCouple avec le modèle de CMC classique.

II.1 Chaîne de Markov Couple

Dans toute la suite, $Z = (Z_n)_{n=1:N}$ désigne un processus aléatoire couple à temps discret, où $Z_n = (X_n, Y_n)$, avec X_n et Y_n à valeurs dans un espace respectivement \mathcal{X} et \mathcal{Y} . Dans le cas discret \mathcal{X} est un espace fini $\mathcal{X} = \{\omega_1, \dots, \omega_K\}$, et dans le cas continu réel $\mathcal{X} = \mathbb{R}$. Sauf mention contraire, les Y_n considérés seront réels : $\mathcal{Y} = \mathbb{R}$.

Définition II.1 *Un processus aléatoire couple Z est appelé « Chaîne de Markov Couple » (CMCouple) si et seulement si il admet pour graphe d'indépendance conditionnelle le graphe représenté par la figure (I.4).*

Bien entendu, une CMCouple vérifie alors les propriétés classiques que nous rappelons ci-dessous pour mémoire :

Propriété II.1 *Le processus Z vérifie les propriétés de Markov par rapport au graphe \mathcal{G} :*

- *Propriété de Markov globale : Soit \mathcal{B}, \mathcal{C} et \mathcal{D} trois sous ensembles de \mathcal{S} deux à deux disjoints, si \mathcal{C} sépare \mathcal{B} et \mathcal{D} dans \mathcal{G} alors :*

$$p(z_{\mathcal{B}}, z_{\mathcal{D}}|z_{\mathcal{C}}) = p(z_{\mathcal{B}}|z_{\mathcal{C}})p(z_{\mathcal{D}}|z_{\mathcal{C}}) \quad (\text{II.1})$$

□ *Propriété de Markov locale* : $\forall u \in \mathcal{S}$, de voisinage $\nu(u)$ dans \mathcal{G} .

$$p(z_u | z_{v \in [1:N]/u}) = p(z_u | z_{v \in \nu(u)}) \quad (\text{II.2})$$

□ *Propriété de Markov par paires* : $\forall (u, v)$ deux sommets de \mathcal{S} , si $v \notin \nu(u)$.

$$p(Z_u, Z_v | Z_{t \neq \{u, v\}}) = p(Z_u | Z_{t \neq \{u, v\}}) p(Z_v | Z_{t \neq \{u, v\}}) \quad (\text{II.3})$$

Nous pouvons factoriser la loi de Z selon son graphe d'indépendance conditionnelle non orienté \mathcal{G} :

$$p(z) = p(z_1) \prod_{n=1}^{N-1} p(z_{n+1} | z_n) \quad (\text{II.4})$$

La proposition suivante précise la markovianité du processus caché conditionnellement aux observations, ce qui va rendre possible les traitements bayésiens analogues à ceux pratiqués dans les CMC classiques. Pour des raisons de symétrie, le processus observé est également markovien conditionnellement au processus caché. Cette propriété, qui permet de modéliser le bruit de manière plus complète que dans les CMC classiques, constitue un avantage des CM Couples par rapport à ces derniers.

Proposition II.1 *Soit $Z = (X_n, Y_n)_{1:N}$ une chaîne de Markov couple. Alors on a les propriétés suivantes :*

- *X conditionnellement à $Y = y$ est une chaîne de Markov.*
- *Y conditionnellement à $X = x$ est une chaîne de Markov.*

La preuve de la proposition, qui est proche de celle du résultat analogue valable dans le cas des CMC, peut être consulté dans [17, 18, 92].

La proposition suivante, relativement immédiate montre que les chaînes de Markov cachées sont des chaînes de Markov couples.

Proposition II.2 *Soit $Z = (X, Y)$ une chaîne de Markov cachée à bruit indépendant alors $Z = (Z_n)_{n=1:N}$ est une chaîne de Markov couple.*

Preuve : Pour montrer qu'un CMC-BI est un cas particulier du modèle CM Couple, il suffit de factoriser la densité de la loi du couple (X, Y) comme la densité de la loi d'un CM Couple.

Soit $X = (X_n)_{n=1:N}$ un CMC-BI et soit $Y = (Y_n)_{n=1:N}$ les données observées. Si on note $Z = (X_n, Y_n)_{n=1:N}$ alors :

$$\begin{aligned} p(z) &= p(x, y) \\ &= p(x) p(y | x) \\ &= p(x_1) \prod_{n=1}^{N-1} p(x_{n+1} | x_n) \prod_{n=1}^N p(y_n | x_n) \\ &= p(x_1) p(y_1 | x_1) \prod_{n=1}^{N-1} p(x_{n+1} | x_n) p(y_{n+1} | x_{n+1}) \end{aligned}$$

En utilisant la formule de Bayes $p(x_1, y_1) = p(x_1)p(y_1|x_1)$ et $\forall n \in [1..N-1] p(x_{n+1}, y_{n+1}|x_n, y_n) = p(x_{n+1}|x_n)p(y_{n+1}|x_{n+1})$ du fait que X est une CMC-BI, nous obtenons par regroupement :

$$\begin{aligned} p(z) &= p(x_1, y_1) \prod_{n=1}^{N-1} p(x_{n+1}, y_{n+1}|x_n, y_n) \\ &= p(z_1) \prod_{n=1}^{N-1} p(z_{n+1}|z_n) \end{aligned}$$

Ce qui prouve que $Z = (X, Y)$ est un CMCouple. \blacksquare

La proposition de Pieczynski [93,96] montre que le modèle couple CMCouple est strictement plus général que le modèle classique CMC et précise, dans le cas des chaînes stationnaires réversibles, des conditions sous lesquelles une CMCouple est une CMC. Autrement dit, la proposition suivante donne les conditions nécessaires et suffisantes pour que, dans une CMCouple $Z = (X, Y)$, la chaîne cachée X soit une chaîne de Markov.

Proposition II.3 Soit $Z = (X, Y)$ une chaîne de Markov Couple qui vérifie :

1. $p(z_n, z_{n+1})$ ne dépendant pas de $n \in \{1, \dots, N-1\}$;
2. $p(z_n = a, z_{n+1} = b) = p(z_n = b, z_{n+1} = a)$ pour tout $n \in \{1, \dots, N-1\}$ et pour tout a et b .

Alors les trois conditions suivantes :

1. X est une chaîne de Markov ;
2. $\forall 2 \leq n \leq N-1, p(y_n|x_n, x_{n-1}) = p(y_n|x_n)$;
3. $\forall 1 \leq n \leq N, p(y_n|x) = p(y_n|x_n)$,

sont équivalentes.

Preuve : La démonstration a été présentée par Pieczynski et all. dans les articles [93,96].

$1 \Rightarrow 2$: Z est une CMCouple, nous pouvons développer sa densité comme suit :

$$\begin{aligned} p(z) &= \frac{\prod_{n=1}^{N-1} p(z_n, z_{n+1})}{\prod_{n=2}^{N-1} p(z_n)} \\ &= \frac{\prod_{n=1}^{N-1} p(x_n, x_{n+1}, y_n, y_{n+1})}{\prod_{n=2}^{N-1} p(x_n, y_n)} \\ &= \underbrace{\frac{\prod_{n=1}^{N-1} p(x_n, x_{n+1})}{\prod_{n=2}^{N-1} p(x_n)}}_{a(x)} \underbrace{\frac{\prod_{n=1}^{N-1} p(y_n, y_{n+1}|x_n, x_{n+1})}{\prod_{n=2}^{N-1} p(y_n|x_n)}}_{b(y)} \end{aligned}$$

X étant de Markov nous avons $a(x) = p(x)$. Par ailleurs, nous obtenons la loi de X en intégrant $p(z)$ par rapport à y :

$$p(x) = \int_{\mathcal{Y}^N} p(z) d(y_{1:N}) = a(x)$$

On déduit :

$$\int_{\mathcal{Y}^N} b(y) d(y_{1:N}) = 1$$

et comme pour tout $1 < n \leq N$, (Z_1, \dots, Z_n) est une chaîne de Markov, alors :

$$\int_{\mathcal{Y}^n} b(y_{1:n}) d(y_{1:n}) = 1 \quad (\text{II.5})$$

Considérons $n = 3$, la formule (II.5) elle s'écrit :

$$\int_{\mathcal{Y}^3} b(y_{1:3}) d(y_{1:3}) = 1 \quad (\text{II.6})$$

en intégrant (II.6) par rapport à y_1 et y_3 nous obtenons :

$$\int_{\mathcal{Y}} \frac{p(y_2|x_1, x_2)p(y_2|x_2, x_3)}{p(y_2|x_2)} d(y_2) = 1 \quad (\text{II.7})$$

Fixons x_2 et posons $f_i(y_2) = p(y_2|x_1 = \omega_i, x_2) = p(y_2|x_2, x_3 = \omega_i)$. L'égalité (II.7) définit alors le produit scalaire

$$\langle f_i, f_j \rangle = \int_{\mathcal{Y}} \frac{f_i(y_2)f_j(y_2)}{p(y_2|x_2)} dy_2. \quad (\text{II.8})$$

On a donc K vecteurs f_1, \dots, f_K tels que $\langle f_i, f_j \rangle = 1$ pour tous i, j variant de 1 à K . On a en particulier $\|f_i\| = \sqrt{\langle f_i, f_i \rangle}$, et donc $\|f_i - f_j\|^2 = \|f_i\|^2 + \|f_j\|^2 - 2\langle f_i, f_j \rangle = 0$. Il en résulte que les vecteurs f_1, \dots, f_K sont égaux, ce qui signifie que $f_i(y_2) = p(y_2|x_1 = \omega_i, x_2)$ ne dépend pas de ω_i , d'où le résultat.

2 \Rightarrow 1 et 3 : Supposons $p(y_n|x_{n-1}, x_n) = p(y_n|x_n, x_{n+1}) = p(y_n|x_n)$,

$$\begin{aligned} b(y_{1:N}) &= \frac{\prod_{n=1}^{N-1} p(y_n, y_{n+1}|x_n, x_{n+1})}{\prod_{n=2}^{N-1} p(y_n|x_n)} \\ &= \frac{\prod_{n=1}^{N-1} p(y_n|x_n, x_{n+1})p(y_{n+1}|y_n, x_n, x_{n+1})}{\prod_{n=2}^{N-1} p(y_n|x_n)} \\ &= p(y_1|x_1, x_2) \prod_{n=1}^{N-1} p(y_{n+1}|y_n, x_n, x_{n+1}) \end{aligned}$$

donc l'intégrale de $b(y_{1:N})$ par rapport $y_{1:N}$ vaut 1 :

$$\int_{\mathcal{Y}^N} b(y) d(y_{1:N}) = 1$$

et nous obtenons :

$$p(x) = a(x)$$

la densité de la loi de X est markovienne, d'où 1. Par ailleurs, $\forall n \in [1..N]$ l'intégration de $p(z)$ par $y_{1:n-1}, y_{n+1:N}$ vaut $p(x)p(y_n|x_n)$, en effet

$$\begin{aligned} \int_{\mathcal{Y}^{N-1}} p(z) d(y_{1:n-1})d(y_{n+1:N}) &= \int_{\mathcal{Y}^{N-1}} a(x)b(y_{1:N}) d(y_{1:n-1})d(y_{n+1:N}) \\ &= p(x) \int_{\mathcal{Y}^{N-1}} b(y_{1:N}) d(y_{1:n-1})d(y_{n+1:N}) \end{aligned}$$

en utilisant l'hypothèse $p(y_n|x_{n-1}, x_n) = p(y_n|x_n, x_{n+1}) = p(y_n|x_n)$ nous pouvons calculer l'intégrale de $\int_{\mathcal{Y}^{N-1}} b(y_{1:N}) d(y_{1:n-1})d(y_{n+1:N})$:

$$\begin{aligned} \int_{\mathcal{Y}^{N-1}} b(y_{1:N}) d(y_{1:n-1})d(y_{n+1:N}) &= \int_{\mathcal{Y}^{N-1}} \frac{\prod_{n=1}^{N-1} p(y_n, y_{n+1}|x_n, x_{n+1})}{\prod_{n=2}^{N-1} p(y_n|x_n)} d(y_{1:n-1})d(y_{n+1:N}) \\ &= \int_{\mathcal{Y}^{n-1}} \frac{\prod_{i=1}^{n-1} p(y_i, y_{i+1}|x_i, x_{i+1})}{\prod_{i=2}^{n-1} p(y_i|x_i)} d(y_{1:n-1}) \\ &\quad \times \int_{\mathcal{Y}^{N-n-1}} \frac{\prod_{i=n}^{N-1} p(y_i, y_{i+1}|x_i, x_{i+1})}{\prod_{i=n}^{N-1} p(y_i|x_i)} d(y_{n+1:N}) \end{aligned}$$

$$\begin{aligned} \text{on a : } \int_{\mathcal{Y}^{n-1}} \frac{\prod_{i=1}^{n-1} p(y_i, y_{i+1}|x_i, x_{i+1})}{\prod_{i=2}^{n-1} p(y_i|x_i)} d(y_{1:n-1}) &= \int_{\mathcal{Y}^{n-2}} \left[\int_{\mathcal{Y}} p(y_1, y_2|x_1, x_2) dy_1 \right] \frac{\prod_{i=2}^{n-1} p(y_i, y_{i+1}|x_i, x_{i+1})}{p(y_2|x_2) \prod_{i=3}^{n-1} p(y_i|x_i)} dy_{2:n-1} = \\ \int_{\mathcal{Y}^{n-2}} \frac{\prod_{i=2}^{n-1} p(y_i, y_{i+1}|x_i, x_{i+1})}{\prod_{i=3}^{n-1} p(y_i|x_i)} dy_{2:n-1} &= \dots = 1, \text{ et} \\ \int_{\mathcal{Y}^{N-n-1}} \frac{\prod_{i=n}^{N-1} p(y_i, y_{i+1}|x_i, x_{i+1})}{\prod_{i=n}^{N-1} p(y_i|x_i)} d(y_{n+1:N}) &= \\ \int_{\mathcal{Y}^{N-n-2}} \frac{\prod_{i=n}^{N-2} p(y_i, y_{i+1}|x_i, x_{i+1})}{\prod_{i=n}^{N-2} p(y_i|x_i) p(y_{N-1}|x_{N-1})} \left[\int_{\mathcal{Y}} p(y_{N-1}, y_N|x_{N-1}, x_N) dy_N \right] d(y_{n+1:N-1}) &= \dots = p(y_n|x_n, x_{n+1}) \\ \int_{\mathcal{Y}^{N-1}} p(z) d(y_{1:n-1})d(y_{n+1:N}) &= p(x, y_n) = p(x)p(y_n|x_n) \end{aligned}$$

En utilisant la formule de Bayes et le résultat précédent $p(y_n|x) = \frac{p(x, y_n)}{p(x)} = \frac{p(x)p(y_n|x_n)}{p(x)}$, d'où 3.
 $3 \Rightarrow 2$: nous avons :

$$\begin{aligned} p(y_n|x_n, x_{n-1}) &= \int_{\mathcal{X}^{N-2}} p(y_n, x_{1:n-2}, x_{n+1:N}|x_n, x_{n-1}) d(x_{1:n-2}, x_{n+1:N}) \\ &= \int_{\mathcal{X}^{N-2}} \frac{p(y_n, x_{1:N})}{p(x_n, x_{n-1})} d(x_{1:n-2}, x_{n+1:N}) \\ &= \int_{\mathcal{X}^{N-2}} \frac{p(y_n|x_{1:N})p(x_{1:N})}{p(x_n, x_{n-1})} d(x_{1:n-2}, x_{n+1:N}) \end{aligned}$$

d'après 3, $p(y_n|x_{1:N}) = p(y_n|x_n)$ donc :

$$\begin{aligned} p(y_n|x_n, x_{n-1}) &= \int_{\mathcal{X}^{N-2}} \frac{p(y_n|x_{1:N})p(x_{1:N})}{p(x_n, x_{n-1})} d(x_{1:n-2}, x_{n+1:N}) \\ &= p(y_n|x_n) \int_{\mathcal{X}^{N-2}} \frac{p(x_{1:N})}{p(x_n, x_{n-1})} d(x_{1:n-2}, x_{n+1:N}) \\ &= p(y_n|x_n) \end{aligned}$$

d'où 2. ■

D'après cette proposition, le fait que X soit une chaîne de Markov sachant que $Z = (X, Y)$ est un CMCouple, est équivalent à $p(y_n|x) = p(y_n|x_n)$ pour tout $n \in \{1, \dots, N\}$. Prenons un exemple concret de la nature pour interpréter cette propriété. Soit une image numérique où chaque pixel peut être "eau" ou "forêt", et supposons que la

chaîne couple est une ligne de cette image. L'observation est alors le niveau de gris. Dans le modèle classique CMC nous avons $p(y_n|x) = p(y_n|x_n)$, ce qui signifie que l'aspect visuel de la forêt (par exemple) ne peut pas dépendre des pixels ("eau" ou "forêt") se trouvant à côté, alors que cette possibilité existe dans les CMCCouples. Ainsi les CMCCouples permettent de prendre en compte les différences éventuelles des aspects des pixels se trouvant près des frontières.

Soit $Z = (X, Y)$ une chaîne de Markov couple stationnaire et réversible (rappelez que Z est dite stationnaire si $p(z_{n+1}, z_n)$ ne dépendent pas de n pour tout $n \in \{1, \dots, N-1\}$). La loi d'une chaîne de Markov couple stationnaire est complètement déterminée par la loi du couple $(z_2, z_1) = (x_2, y_2, x_1, y_1)$. Pour exploiter la proposition (II.3), nous pouvons développer la loi de $(Z_{1:2})$ sous plusieurs formes particulières donnant lieu à autant de sous-modèles de CMCCouple. Les différents graphes d'indépendance conditionnelle de ces modèles sont représentés sur la figure (II.1). Nous commençons par développer la loi $p(z_{1:2})$ sous la forme suivante :

$$p(z_{1:2}) = p(x_{1:2})p(y_{1:2}|x_{1:2})$$

CMC-BI

Dans ce premier modèle, nous supposons que les y_n sont indépendants conditionnellement aux $(x_n)_{1:N}$ et que pour toute n , y_n ne dépend que de x_n . Le graphe d'indépendance conditionnelle de ce modèle est (a) sur la figure (II.1). D'après la proposition (II.3), X est une chaîne de Markov. La loi de transition est donnée par :

$$p(z_2|z_1) = p(x_2|x_1)p(y_2|x_2) \quad (\text{II.9})$$

CMC

Dans le modèle précédent, nous avons supposé que les y_n sont indépendants conditionnellement aux $(x_n)_{1:N}$ ce qui se traduit par l'absence des arêtes entre les y_n sur le graphe (a). Pour généraliser le modèle de CMC-BI nous supposons cette fois que les y_n sont corrélés et nous gardons l'hypothèse $p(y_n|x_{n-1}, x_n) = p(y_n|x_n)$. Pour cela nous ajoutons des arêtes entre les y_n afin de modéliser cette dépendance et nous obtenons le graphe (b). X est une chaîne de Markov et les transitions et la loi initiale de la chaîne couple sont données par :

$$p(z_2|z_1) = p(x_2|x_1)p(y_2|y_1, x_2) \quad (\text{II.10})$$

$$p(z_1) = p(x_1)p(y_1|x_1) \quad (\text{II.11})$$

Avec le modèle CMC, nous pouvons modéliser la corrélation des observations avec un nombre réduit de paramètres.

CMCCouple-BI

Dans ce modèle nous supposons que les Y_n sont indépendantes conditionnellement à X mais l'égalité $p(y_n|x_{n-1:n}) = p(y_n|x_n)$ n'est pas vérifiée. D'après la proposition (II.3), X n'est pas une chaîne de Markov. En utilisant la formule de Bayes pour développer la densité $p(y_{n:n+1}|x_{n:n+1})$, on obtient :

$$p(y_{1:2}|x_{1:2}) = p(y_1|x_{1:2})p(y_2|x_{1:2}) \quad (\text{II.12})$$

les transitions de la chaîne couple sont alors :

$$p(z_2|z_1) = p(x_2|z_1)p(y_2|x_{1:2}) \quad (\text{II.13})$$

CMCouple

Dans le modèle CMCouple général nous avons :

$$p(y_{1:2}|x_{1:2}) = p(y_1|x_{1:2})p(y_2|z_1, x_2) \quad (\text{II.14})$$

et la forme la plus générale des transitions est :

$$p(z_2|z_1) = p(x_2|z_1)p(y_2|z_1, x_2) \quad (\text{II.15})$$

Nous pouvons remarquer sur la figure (II.1) que les arêtes entre les sommets sont en nombre croissant d'un graphe à l'autre, ce qui illustre la richesse et la complexité croissante des modèles considérés.

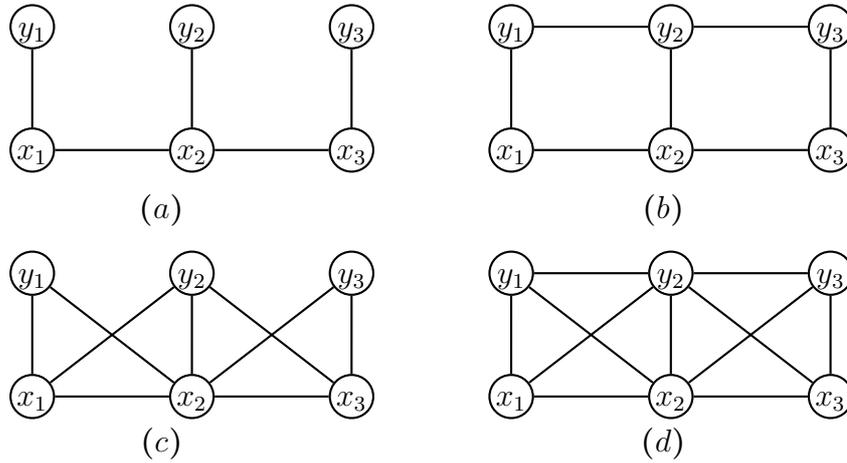


FIGURE II.1 – Graphe d'indépendance conditionnelle des chaînes de Markov couple. (a) : CMC-BI; (b) : CMC; (c) : CMCouple-BI; (d) : CMCouple

II.2 Inférence dans le modèle couple

Nous présentons dans cette sous-section l'extension de l'algorithme de Baum-Welsh, qui permet de calculer les probabilités a posteriori $p(x_n|y)$ et $p(x_{n+1}|x_n, y)$ (avec $y = y_{1:N}$), aux modèles CMCouples. Lorsque le CMCouple est une CMC l'extension redonne l'algorithme classique. Comme nous l'avons mentionné, ce calcul est de complexité comparable à celle de l'algorithme classique dans les CMC.

II.2.1 Algorithme de Baum-Welsh

Soit $Z = (X, Y)$ une chaîne de Markov couple (II.1), dont nous disposons des observations $y = y_{1:N}$ du processus Y sur une grille de taille N . La loi de Z s'écrit

sous sa forme factorisée :

$$p(z) = p(z_1) \prod_{n=1}^{N-1} p(z_{n+1}|z_n) \quad (\text{II.16})$$

Comme dans le cas classique des CMC, l'algorithme de Baum-Welsh permet d'estimer une réalisation du processus caché X à partir des observations au sens de la solution MPM (5).

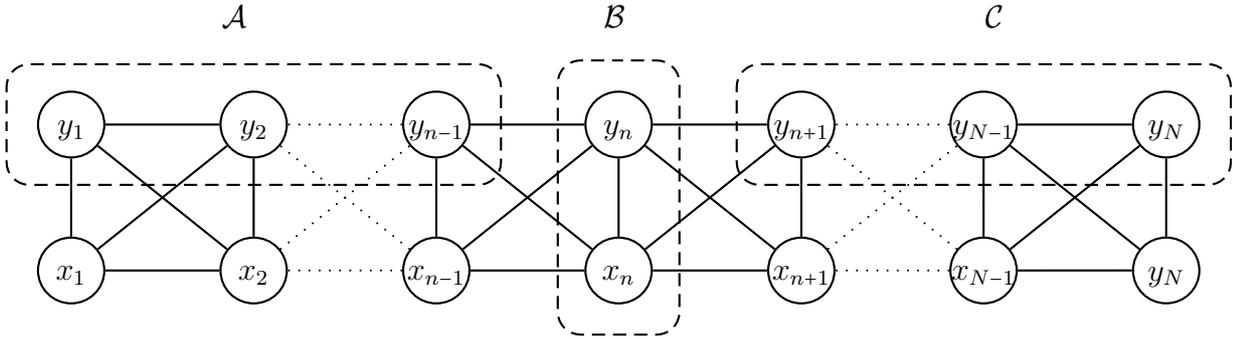


FIGURE II.2 – Graphe d'indépendance conditionnelle d'une chaîne Markov couple

Nous reprenons ci-après le calcul des probabilités « directes » α_n et les probabilités « rétrogrades » β_n , qui sont des extensions des quantités analogues utilisées dans les CMC classiques. Nous avons :

$$p(x_n = \omega_i | y_{1:N}) = \frac{p(x_n = \omega_i, y_{1:N})}{\sum_{j=1}^K p(x_n = \omega_j, y_{1:N})} \quad (\text{II.17})$$

Comme nous pouvons remarquer sur la figure (II.2) $\mathcal{B} = (x_n, y_n)$ sépare $\mathcal{A} = (y_{1:n-1})$ et $\mathcal{C} = (y_{n+1:N})$ (tout chemin entre \mathcal{A} et \mathcal{C} passe par \mathcal{B} voir (I.1.1)) d'où $p(x_n, y_{1:N})$ peut s'écrire comme un produit :

$$p(x_n, y_{1:N}) = p(x_n, y_{1:n})p(y_{n+1:N}|x_n, y_{1:n})$$

En notant par $\alpha_n(x_n) = p(x_n, y_{1:n})$ la probabilité directe, et par $\beta_n(x_n) = p(y_{n+1:N}|x_n, y_n)$ la probabilité rétrograde, nous avons :

$$p(x_n = \omega_i | y_{1:N}) = \frac{\alpha_n(x_n = \omega_i)\beta_n(x_n = \omega_i)}{\sum_{j=1}^K \alpha_n(x_n = \omega_j)\beta_n(x_n = \omega_j)} \quad (\text{II.18})$$

De même nous pouvons déduire la probabilité de X_{n+1} conditionnellement à $X_n, Y_{1:N}$ en fonction des β_n et les transitions $p(z_{n+1}|z_n)$:

$$p(x_{n+1} | x_n, y_{1:N}) = \frac{\beta_{n+1}(x_{n+1})}{\beta_n(x_n)} p(z_{n+1}|z_n) \quad (\text{II.19})$$

Les α_n et les β_n sont calculés par récurrence :

1. Étape directe :

- Initialisation : $\alpha_1(x_1) = p(x_1, y_1)$
- à l'étape $(n + 1)$, supposons α_n calculé à l'étape précédente (n) :

$$\alpha_{n+1}(x_{n+1}) = \sum_{j=1}^K \alpha_n(x_n = \omega_j) p(z_{n+1}|z_n)$$

2. Étape rétrograde :

- Initialisation : $\beta_N(x_N) = 1$
- à l'étape n , supposons β_{n+1} calculé à l'étape précédente :

$$\beta_n(x_n) = \sum_{j=1}^K \beta_{n+1}(x_{n+1} = \omega_j) p(z_{n+1}|z_n)$$

Lors de l'implémentation de l'algorithme de Baum-Welsh, des problèmes d'ordre numérique peuvent apparaître à cause des α_n qui seront considérés comme nuls si n est très grand, de même pour les β_n si n est petit, et N grand. Pour surmonter ces problèmes, P. A. Devijver a proposé l'algorithme de Baum-Welsh conditionnel dans le cas de CMC-BI [37]. Dans la section suivante nous présentons l'algorithme de Baum-Welsh conditionnel dans le cas du chaîne de Markov couple présenté par S. Derrode et W. Pieczynski [35], qui peut être vu comme une généralisation de celui de P. A. Devijver au cas CMCouple.

II.2.2 Algorithme de Baum-Welsh conditionnel

Dans le cas de CMC-BI Devijver (1985) a proposé un algorithme fondé sur la factorisation des probabilités de lissage $p(x_n|y_{1:N})$ pour assurer la stabilité numérique lors du calcul des α_n et des β_n . Pour cela, il propose de les diviser par des quantités ayant le même ordre de grandeur, afin de garder les $p(x_n|y_{1:N})$ et $p(x_{n+1}|x_n, y_{1:N})$ invariantes.

On pose :

$$\tilde{\alpha}_n(x_n) = \frac{\alpha_n(x_n)}{p(y_{1:n})} = p(x_n|y_{1:n}) \quad (\text{II.20})$$

(on peut remarquer que ceci est équivalent à l'étape de correction dans le filtre de Kalman). On a :

$$p(x_n, y_{1:N}) = \alpha_n(x_n) \beta_n(x_n)$$

En remplaçant α_n par $p(y_{1:n}) \tilde{\alpha}_n(x_n)$ nous obtenons :

$$p(x_n, y_{1:N}) = p(y_{1:n}) \tilde{\alpha}_n(x_n) \beta_n(x_n) \quad (\text{II.21})$$

En divisant (II.21) par $p(y_{1:N})$ nous obtenons $p(x_n|y_{1:N})$ en fonction de $\tilde{\alpha}_n$:

$$\begin{aligned} p(x_n|y_{1:N}) &= \frac{p(y_{1:n}) \tilde{\alpha}_n(x_n) \beta_n(x_n)}{p(y_{1:N})} \\ &= \frac{\tilde{\alpha}_n(x_n) \beta_n(x_n)}{p(y_{n+1:N}|y_{1:n})} \end{aligned}$$

En posant :

$$\tilde{\beta}_n(x_n) = \frac{\beta_n(x_n)}{p(y_{n+1:N}|y_{1:n})} \quad (\text{II.22})$$

La densité de la loi du X_n conditionnellement aux $y_{1:N}$ est donnée en fonction de $\tilde{\alpha}_n$ et $\tilde{\beta}_n$ par :

$$p(x_n|y_{1:N}) = \tilde{\alpha}_n(x_n)\tilde{\beta}_n(x_n) \quad (\text{II.23})$$

et la loi de X_{n+1} conditionnellement aux x_n et $y_{1:N}$ est donnée par :

$$p(x_{n+1}|x_n, y_{1:N}) = \frac{\tilde{\beta}_{n+1}(x_{n+1})}{\tilde{\beta}_n(x_n)} p(z_{n+1}|z_n) \quad (\text{II.24})$$

Les probabilités $\tilde{\alpha}_n$ et $\tilde{\beta}_n$ sont calculées d'une manière récursive comme dans l'algorithme de Baum-Welsh classique. En effet, nous avons :

1. Calcul des $\tilde{\alpha}_n$:

- Initialisation $\tilde{\alpha}_1(x_1) = p(x_1|y_1)$;
- à l'étape $n + 1$, supposons $\tilde{\alpha}_n$ calculé à l'étape n :

$$\tilde{\alpha}_{n+1}(x_{n+1}) = \frac{1}{p(y_{n+1}|y_{1:n})} \sum_{x_n} \tilde{\alpha}_n(x_n) p(z_{n+1}|z_n) \quad (\text{II.25})$$

avec $p(y_{n+1}|y_{1:n}) = \sum_{x_{n+1}} \sum_{x_n} \tilde{\alpha}_n(x_n) p(z_{n+1}|z_n)$.

2. Calcul des $\tilde{\beta}_n$:

- Initialisation $\tilde{\beta}_N(x_N) = 1$;
- à l'étape n supposons $\tilde{\beta}_{n+1}(x_{n+1})$ calculé à l'étape précédente :

$$\tilde{\beta}_n(x_n) = \frac{\sum_{x_{n+1}} \tilde{\beta}_{n+1}(x_{n+1}) p(z_{n+1}|z_n)}{\sum_{x_{n+1}} \sum_{x_n} \tilde{\alpha}_n(x_n) p(z_{n+1}|z_n)} \quad (\text{II.26})$$

Notons que la plus grande généralité théorique des CM Couples par rapport aux CMC classiques se traduit par leur plus grande efficacité en segmentation non supervisée - les paramètres étant estimés par la méthode "ICE" présentée dans le chapitre suivant - des données. Comme présenté dans [35] dans certaines situations l'erreur de classification est systématiquement divisée par deux. Nous présentons également quelques résultats dans la sous-section (II.3) ci-après.

II.2.3 Programmation dynamique : Algorithme de Viterbi

Dans la section précédente, nous avons présenté les deux versions de l'algorithme de Baum-Welsh, qui permettent de calculer la solution MPM (5), qui correspond à la fonction de perte locale (3) .

$$\hat{x}_{MPM} = (\hat{x}_n^{MPM})_{1:N} \quad (\text{II.27})$$

avec $\hat{x}_n^{MPM} = \underset{x_n \in \Omega}{\operatorname{argmax}} p(x_n|y_{1:N})$ pour tout $n \in \{1, \dots, N\}$.

Dans cette section, nous présentons l'algorithme qui permet de calculer la solution du maximum a posteriori MAP (4) qui correspond à la fonction de perte globale (2). Cette solution est définie par :

$$\hat{x}_{MAP}(y_{1:N}) = \underset{x \in \Omega^N}{\operatorname{argmax}} p(x|y_{1:N}) \quad (\text{II.28})$$

Le calcul du MAP est possible avec l'algorithme de programmation dynamique, appelé algorithme de Viterbi.

Nous pouvons factoriser $\max_x p(z)$ de la façon suivante :

$$\max_x p(z) = \max_{x_n} \{ \max_{x_{n+1:N}} p(z_{n+1:N} | z_n) \max_{x_{1:n-1}} p(z_{1:n}) \} \quad (\text{II.29})$$

Cette factorisation permet d'avoir la récursion de type « avant » basée sur les probabilités $p(z_{1:n})$.

Si on définit $\delta_n(z_n)$ par :

$$\delta_n(z_n) = \max_{x_{n+1:N} \in \Omega^{N-n}} p(z_{n+1:N} | z_n) \quad (\text{II.30})$$

Les $\delta_n(z_{1:n})$ vérifient la récurrence arrière suivante :

$$\delta_{n-1}(z_{n-1}) = \max_{x_n \in \Omega} p(z_n | z_{n-1}) \delta_n(z_n) \quad (\text{II.31})$$

en effet :

$$\begin{aligned} \delta_{n-1}(z_{n-1}) &= \max_{x_{n:N} \in \Omega^{N-n+1}} p(z_{n:N} | z_{n-1}) \\ &= \max_{x_{n:N} \in \Omega^{N-n+1}} p(z_{n+1:N} | z_n) p(z_n | z_{n-1}) \\ &= \max_{x_n \in \Omega} p(z_n | z_{n-1}) \max_{x_{n+1:N} \in \Omega^{N-n}} p(z_{n+1:N} | z_n) \\ &= \max_{x_n \in \Omega} p(z_n | z_{n-1}) \delta_n(z_n) \end{aligned}$$

Pour stocker les valeurs qui maximisent les $\delta_{n-1}(z_{1:n-1})$ dans le sens rétrograde, d'une étape à l'autre, on utilise l'égalité suivante :

$$\psi_n(x_{n-1}) = \operatorname{argmax}_{x_n \in \Omega} p(z_n | z_{n-1}) \delta_n(z_n) \quad (\text{II.32})$$

Finalement, l'algorithme de Viterbi se déroule de la manière suivante :

□ Initialisation :

$$\psi_N(x_{N-1}) = \operatorname{argmax}_{x_N \in \Omega} p(z_N | z_{N-1})$$

et

$$\delta_{N-1}(z_{N-1}) = \max_{x_N \in \Omega} p(z_N | z_{N-1})$$

□ Calcul dans le sens rétrograde : $\forall n = N - 1, \dots, 1$

$$\psi_n(x_{n-1}) = \operatorname{argmax}_{x_n \in \Omega} p(z_n | z_{n-1}) \delta_n(z_n)$$

et

$$\delta_{n-1}(z_{n-1}) = \max_{x_n \in \Omega} p(z_n | z_{n-1}) \delta_n(z_n)$$

□ Calcul dans le sens direct : Initialisation : $\widehat{x}_{1,MAP} = \operatorname{argmax}_{x_1 \in \Omega} p(x_1, y_1) \delta_1(z_1)$ et

$\forall n = 2, \dots, N$

$$\widehat{x}_{n+1,MAP} = \psi_{n+1}(\widehat{x}_{n,MAP})$$

La méthode MAP a été programmée et comparée avec la méthode MPM dans le cadre des CMCouples [35]. Malgré leur différence les résultats obtenus, de manière relativement surprenante, se sont avérés d'être toujours très proches en qualité.

II.3 Etude comparative entre CMC-BI et CMCouples

Dans la suite, nous procédons à une étude comparative entre le modèle de chaîne de Markov Couple et celui de chaîne de Markov cachée à bruit indépendant. En un premier lieu, nous considérons des données issues du modèle CMC-BI classique et en second lieu des données issues d'un modèle CMCouple. Dans un troisième temps, nous segmentons avec les deux modèles des données de synthèse obtenues par différents bruits introduits sur une image artificielle. Pour finir la comparaison nous segmentons une image réelle.

Dans toutes les séries d'expériences nous considérons des chaînes de même taille $N = 256 \times 256$ et pour visualiser les réalisations de la chaîne cachée et de la chaîne observée sous forme bi-dimensionnelle nous utilisons la transformation d'Hilbert-Peano (I.5). Nous commençons par simuler une chaîne de Markov à deux classes ω_1 et ω_2 dont la loi initiale $p(x_1) = \frac{1}{2}$ et la matrice de transitions $\mathcal{P} = p(x_{n+1}|x_n)$ donne par :

$$\mathcal{P} = \begin{pmatrix} 0.99 & 0.01 \\ 0.01 & 0.99 \end{pmatrix}.$$

La loi $p(y_n|x_n)$ est une gaussienne de variance égale à 1 pour les deux classes et de moyenne 0 si x_n vaut ω_1 et de moyenne égale à 2 si x_n vaut ω_2 . La réalisation $x_{1:N}$ est représentée par (a) sur la figure (II.3) et la réalisation $y_{1:N}$ par (b) de la même figure. (c), (d) et (e) sont les estimées obtenues avec la méthode MPM (5). L'estimation des paramètres par l'algorithme ICE sera traitée dans le chapitre suivant.

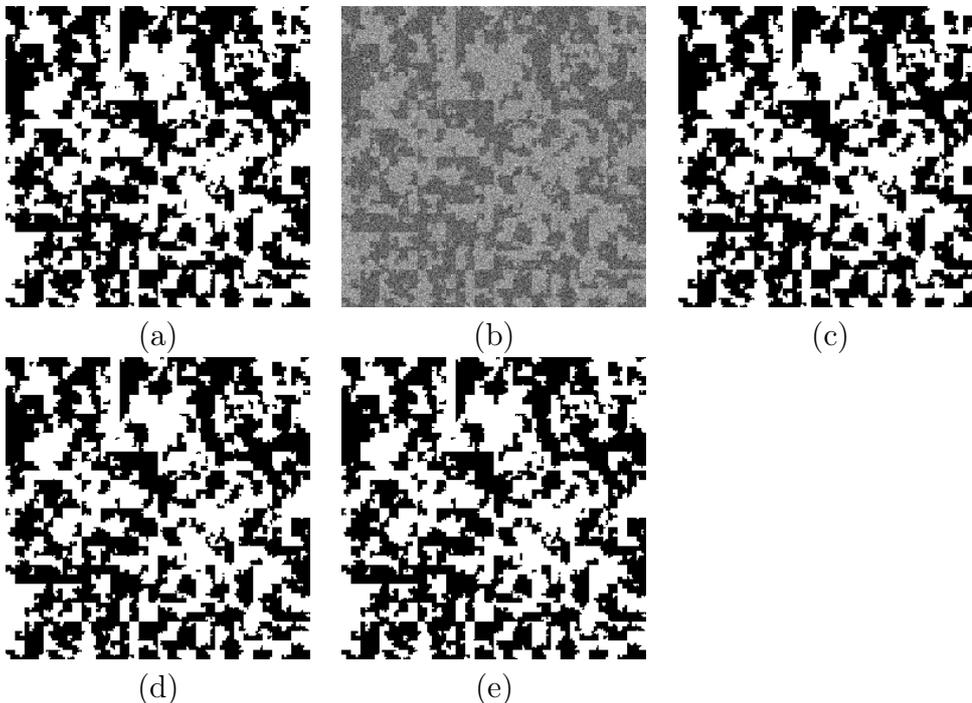


FIGURE II.3 – Simulation et Segmentation d'une CMC-BI (les pourcentages désignent les taux d'erreur). (a) : réalisation $x_{1:N}$; (b) : réalisation $y_{1:N}$; (c) : segmentation par CMC-BI avec les vrais paramètres 0.62% ; (d) : segmentation par CMC-BI paramètres estimés par ICE 0.62% ; (e) : segmentation par CMCouple paramètres estimés par ICE 0.70%

Nous remarquons que les taux d'erreur sont très proches : 0.62% pour le CMC-BI qui est le vrai modèle et 0.70% pour le CMCouple. Cela montre que la méthode ICE appliquée au modèle CMCouple est capable de lui faire prendre en considération la markovianité de données cachées. Dans l'expérience suivante, nous considérons des données issues du modèle CMCouple qui n'est pas une CMC-BI. La loi du couple (Z_{n+1}, Z_n) est donnée par la loi jointe $\mathcal{P} = p(x_{n+1}, x_n)$:

$$\mathcal{P} = \begin{pmatrix} 0.49 & 0.01 \\ 0.01 & 0.49 \end{pmatrix}.$$

et la loi $p(y_{n+1}, y_n | x_{n+1}, x_n)$ est une gaussienne $\mathcal{N}_{\mathbb{R}^2}(\Gamma, \Sigma)$ avec : $\Gamma(x_{n+1}, x_n) = \vec{0}$ pour tout $(x_{n+1}, x_n) \in \mathcal{X}^2$ et

$$\Sigma(\omega_1, \omega_1) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \Sigma(\omega_1, \omega_2) = \begin{pmatrix} 50 & 0 \\ 0 & 1 \end{pmatrix}, \Sigma(\omega_2, \omega_1) = \begin{pmatrix} 1 & 0 \\ 0 & 50 \end{pmatrix} \text{ et } \Sigma(\omega_2, \omega_2) = \begin{pmatrix} 50 & 45 \\ 45 & 50 \end{pmatrix}$$

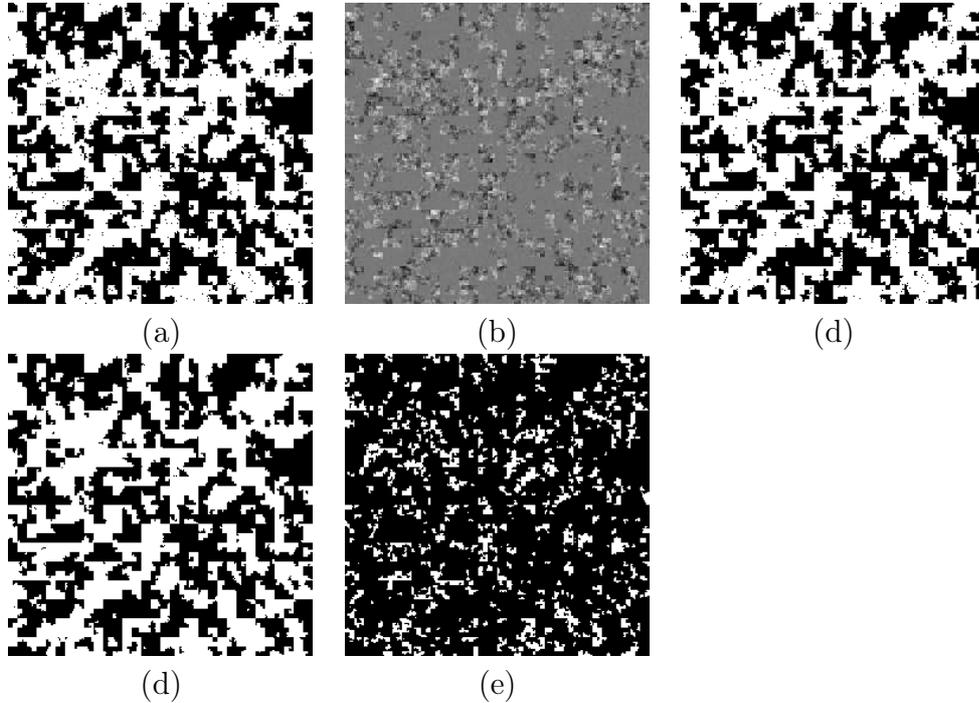


FIGURE II.4 – Simulation et segmentation d'une CMCouple (les pourcentages désignent les taux d'erreur). (a) : réalisation $x_{1:N}$; (b) : réalisation $y_{1:N}$; (c) : segmentation par CMCouple avec les vrais paramètres 1.05%; (d) : segmentation par CMCouple avec paramètres estimés par ICE 2.09%; (e) : segmentation par CMC-BI avec paramètres estimés par ICE 35.51%

Nous constatons que le modèle CMC-BI (35.51% taux d'erreur) a une grande difficulté à segmenter des données issues du modèle CMCouple considéré. Le modèle CMCouple a un taux d'erreur de 2.09% bien que le bruit soit assez important (variance égale à 50). Dans l'expérience suivante, nous appliquons les modèles CMC-BI et CMCouple sur des données issues de deux bruitages d'une image artificielle (a)

sur la figure (II.5). La première image bruitée (b) est obtenue avec un bruit gaussien indépendant ; la deuxième image bruitée (c) est obtenue avec un bruit à moyenne mobile. Le bruit indépendant est $\mathcal{N}(1.0, 5.0)$ si $x_n = \omega_1$ et $\mathcal{N}(2.0, 5.0)$ si $x_n = \omega_2$. Le bruit à moyenne mobile sont données par $y_n = \delta_{\omega_1}(x_n) + \alpha\epsilon_n + \beta[\epsilon_{n+1} + \epsilon_{n-1}]$ avec $\delta_{\omega_1}(x_n) = 1$ si $x_n = \omega_1$ et 0 sinon, les ϵ_n sont des $\mathcal{N}(0, 1)$ indépendants, $\alpha = 0.96$ et $\beta = 0.2$.

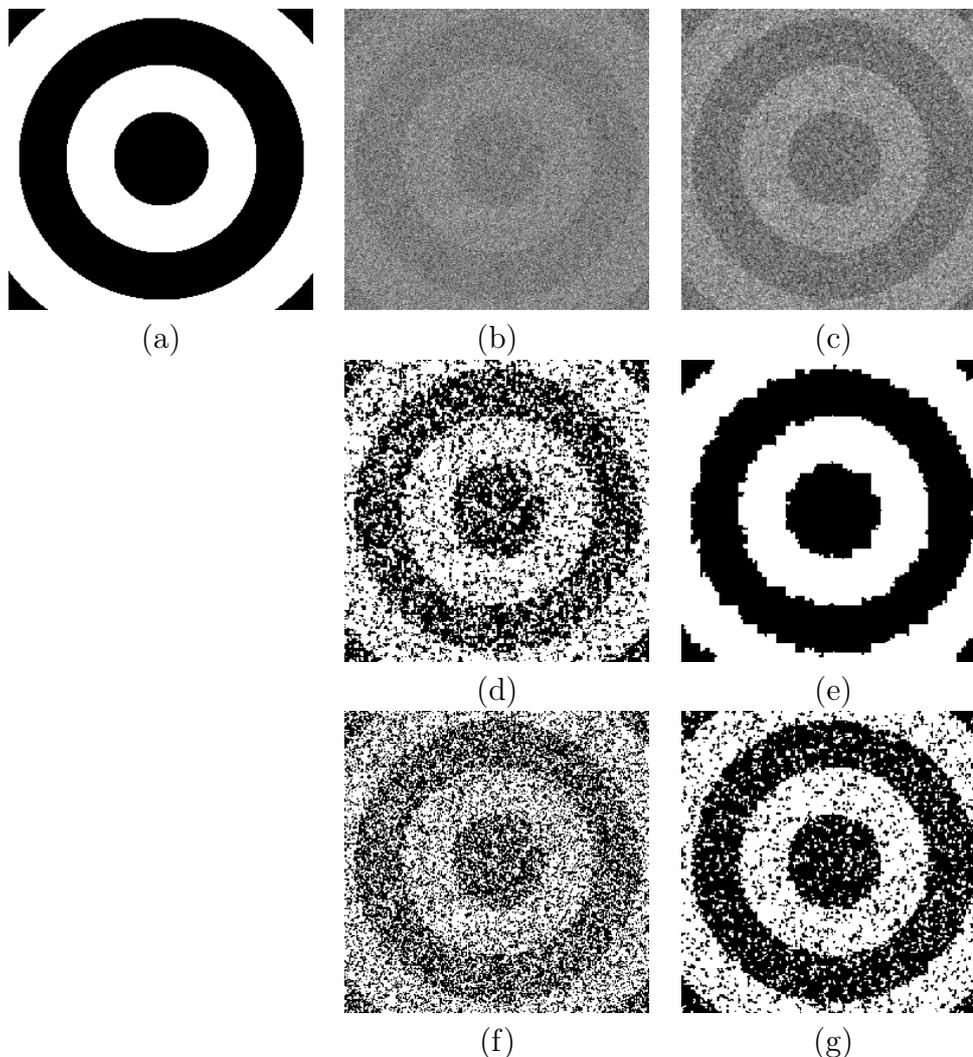


FIGURE II.5 – Segmentation d’une image artificielle (+ estimation des paramètres par ICE). (a) : réalisation $x_{1:N}$; (b) : réalisation $y_{1:N}^1$ avec bruit gaussien indépendant ; (c) : réalisation $y_{1:N}^2$ avec bruit à moyenne mobile ; (d) : segmentation avec CMCouple de $y_{1:N}^1$ 27.14% ; (e) : segmentation avec CMCouple de $y_{1:N}^2$ 4.06% ; (f) : segmentation avec CMC-BI de $y_{1:N}^1$ 34.81% ; (g) : segmentation avec CMC-BI de $y_{1:N}^2$ 16.32%

Les résultats de la segmentation avec le modèle CMCouple sont nettement supérieurs à ceux obtenus avec le modèle CMC-BI dans les deux cas : bruit indépendant et bruit à moyenne mobile. Ceci montre, dans les conditions de l’expérience, que le modèle CMCouple peut considérablement améliorer, en segmentation non supervisée, les résultats donnés par le modèle CMC-BI classique.

Dans la dernière expérience, nous segmentons une image réelle SAR (Synthetic Aperture Radar¹), présentée sur la figure (II.6), (a). Le résultat obtenu avec CM-Couple est présenté sur la même figure en (b), et celui obtenu avec CMC-BI est présenté sur la même figure en (c). Il est relativement difficile d’apprécier les différences entre les deux segmentations de manière rigoureuse; cependant, le résultat obtenu avec CM-Couples semble visuellement bien plus précis. En particulier, les frontières des zones sont mieux retrouvées et les objets de petites tailles sont mieux détectés.

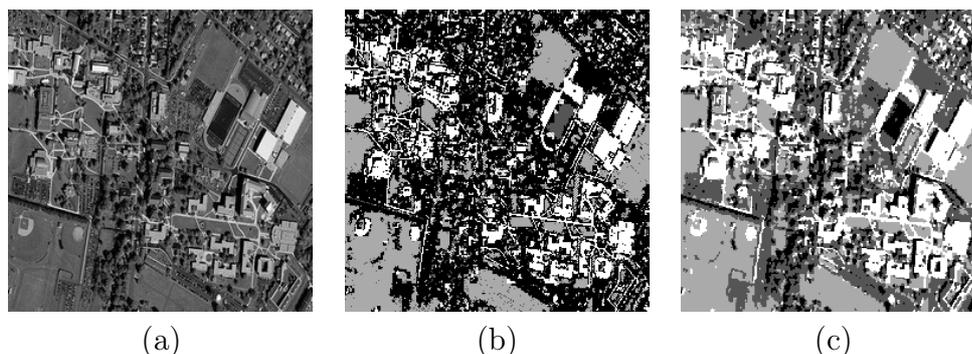


FIGURE II.6 – Segmentation d’une image SAR (+ estimation des paramètres par ICE)(a) image réelle (b) segmentation par CM-Couple (b) segmentation par CMC-BI

II.4 Conclusion

Dans ce chapitre, nous avons présenté en premier lieu le modèle de chaîne de Markov couple, qui généralise le modèle de chaîne de Markov cachée présenté dans le chapitre précédent. En second lieu, nous avons détaillé les différences entre les modèles CM-Couple et les chaînes de Markov cachées en rappelant les conditions nécessaires et suffisantes sur le couple (X, Y) pour que X soit une chaîne de Markov. En troisième lieu, nous avons détaillé les algorithmes de calcul inférentiel permettent de calculer $p(x_n|y_{1:N})$ et $p(x_{n+1}|x_n, y_{1:N})$ dans le cas général d’une CM-Couple (généralisations aux modèles couples des algorithmes classiques de Baum-Walsh, Baum-Welsh conditionnel, Viterbi). Pour finir nous avons procédé à une étude comparative entre le modèle classique de CMC et son extension CM-Couple, en les appliquant sur des données simulées selon ces deux modèles, artificielles et des réelles. Les résultats de cette étude expérimentale montre une nouvelle fois l’intérêt du modèle CM-Couple par rapport au modèle CMC classique.

Dans le chapitre suivant, nous nous intéressons à l’estimation des paramètres, qui est d’une grande importance pour le traitement des problèmes réels.

1. <http://www.onera.fr/images-science/observation-imagerie/image-radar-sar.php> : Technique radar qui exploite le déplacement de l’antenne pour obtenir une résolution angulaire bien supérieure à celle d’une antenne immobile. La résolution angulaire d’une antenne est inversement proportionnelle à sa taille, nécessairement réduite à bord d’un avion.

Chapitre III

Estimation des paramètres

Nous avons présenté, jusqu'à maintenant, des algorithmes permettant de calculer les probabilités a posteriori $p(x_n|y_{1:N})$ pour plusieurs modèles. Nous avons également détaillé les règles de décision pour segmenter un signal ou une image cachés à partir des données observées $y_{1:N}$. Le problème de l'estimation des paramètres de ces modèles, qui peut être très important dans les applications réelles, n'a pas encore été abordé. Notons $\theta = (\theta_1, \dots, \theta_M)$ l'ensemble des paramètres d'un modèle considéré. L'objet de ce chapitre est de présenter diverses méthodes de leur estimation à partir de $y_{1:N}$. On note \mathcal{P}_Θ l'ensemble des lois paramétriques de $p(y|\theta)$

$$\mathcal{P}_\Theta = \{p(y|\theta), \theta \in \Theta\} \quad (\text{III.1})$$

L'estimateur du maximum de vraisemblance consiste en recherche des paramètres qui maximisent la vraisemblance $p(y|\theta)$:

$$\theta_{MVL}(y) = \underset{\theta \in \Theta}{\operatorname{argmax}} p(y|\theta) \quad (\text{III.2})$$

Pour des raisons pratiques, la vraisemblance $p(y|\theta)$ est généralement remplacée par la log-vraisemblance $\mathcal{L}(y|\theta) = \log[p(y|\theta)]$; la fonction logarithme étant croissante, l'estimateur du maximum de vraisemblance est également donné par :

$$\theta_{MVL}(y) = \underset{\theta \in \Theta}{\operatorname{argmax}} \mathcal{L}(y|\theta) \quad (\text{III.3})$$

Dans le cadre de notre étude des modèles markoviens cachés ou couples, la maximisation de l'équation (III.3) est généralement difficile à réaliser directement du fait que y n'est que la partie observée des données du modèle $Z = (X, Y)$. Des algorithmes numériques itératifs ont été proposés pour résoudre ce problème en utilisant les données complètes et la log-vraisemblance complète :

$$\mathcal{L}_c(x, y|\theta) = \log[p(x, y|\theta)]. \quad (\text{III.4})$$

Parmi ces algorithmes, nous pouvons citer l'algorithme Espérance-Maximisation (en anglais Expectation-Maximisation algorithm, souvent abrégé en « EM ») qui a été proposé par Dempster et *al.* en 1977 [34], ainsi que ces variantes : GEM, CEM, SEM, SAEM [16, 20, 22]. L'algorithme EM est composé de deux étapes :

- ◇ Étape (E) : à la quelle nous cherchons à évaluer l'espérance conditionnelle de la log-vraisemblance complète $\mathbb{E}_{\theta^*}[\mathcal{L}_c(X, Y|\theta)|Y = y]$

◇ Étape (M) : une fois que nous avons évalué l'espérance, nous cherchons à la maximiser par rapport à θ .

On part d'une initialisation et on applique les deux étapes ci-dessus, ce qui donne le paramètre suivant. De proche en proche, on obtient une suite des estimées que l'on arrête selon divers critères. Ainsi que nous allons le voir dans la suite, l'étape de la maximisation est parfois malaisée ; de plus, l'algorithme EM est relativement sensible à l'initialisation.

Un autre algorithme itératif d'estimation a été proposé par M. Pieczynski dans [7, 89]. Il s'agit de l'algorithme « Estimation Conditionnelle Itérative » (Iterative Conditional Estimation, en abrégé « ICE »). Cet algorithme est fondé sur l'existence d'un estimateur $\hat{\theta}(x, y)$ de θ , à partir des données complètes, et sur la possibilité de simulation de X selon sa loi conditionnelle aux observations $y_{1:N}$ et utilisant la valeur courante θ^q de θ . Ainsi ICE est très générale et souvent plus facile à utiliser que EM. Par ailleurs, une étude de la convergence de l'algorithme ICE a été récemment présentée par M. Pieczynski [95]. Une étude sur le lien entre EM et ICE a été proposée par M. Delmas [33] montrant leur équivalences dans certains cas des modèles exponentiels.

Nous allons présenter dans un premier temps ces algorithmes itératifs d'estimation, et dans un deuxième temps, nous les appliquerons aux modèles présentés dans les chapitres précédents.

Dans toute la suite $X = (X_n)_{1:N}$ est processus aléatoire discret et $Y = (Y_n)_{1:N}$ un processus aléatoire continu. On note $x = (x_n)_{1:N}$ - respectivement $y = (y_n)_{1:N}$ - une réalisation de X - respectivement de Y . Comme précédemment, on note $Z = (Z_n)_{1:N}$ le couple (X, Y) .

III.1 L'algorithme EM et ses variantes

L'algorithme EM ainsi que ses variantes sont des algorithmes itératifs. Nous trouvons dans la littérature plusieurs critères d'arrêt, qui peuvent être liés soit au temps d'exécution ou au nombre d'itérations de l'algorithme. D'autres critères d'arrêt peuvent être liés à la stabilité de la solution obtenue, comme par exemple la norme de la différence entre θ^{q+1} obtenu à l'itération $q + 1$ et θ^q obtenu à l'itération q . Mentionnons également que le temps d'exécution dépend essentiellement de la taille du processus considéré, du nombre d'états du processus caché, ainsi que du type d'algorithme d'optimisation choisi dans l'étape de maximisation.

III.1.1 L'algorithme EM

L'idée de l'algorithme EM est d'utiliser la log-vraisemblance complète qui est facile à calculer et à maximiser. Nous avons :

$$\mathcal{L}_c(z|\theta) = \log p(z|\theta) = \log p(y|\theta) + \log p(x|y, \theta) = \mathcal{L}(y|\theta) + \log p(x|y, \theta) \quad (\text{III.5})$$

d'ou la relation entre la log-vraisemblance et la log-vraisemblance complète :

$$\mathcal{L}(y|\theta) = \mathcal{L}_c(z|\theta) - \log p(x|y, \theta). \quad (\text{III.6})$$

L'algorithme EM procède d'une manière itérative pour maximiser la log-vraisemblance complète (III.4) en utilisant l'espérance conditionnelle de (X, Y) . Supposons que à l'itération q , nous disposons de la valeur courante θ^q de θ alors l'accroissement est donné par :

$$\mathcal{L}(y|\theta) - \mathcal{L}(y|\theta^q) = \mathcal{L}_c(z|\theta) - \mathcal{L}_c(z|\theta^q) + \log\left[\frac{p(x|y, \theta^q)}{p(x|y, \theta)}\right], \quad (\text{III.7})$$

et l'espérance de l'accroissement, qui est une constante car elle ne dépend pas de X , est :

$$\mathcal{L}(y|\theta) - \mathcal{L}(y|\theta^q) = \mathbb{E}_{\theta^q}[\mathcal{L}_c(z|\theta) - \mathcal{L}_c(z|\theta^q)|Y = y] + E_{\theta^q}\left[\log\left[\frac{p(x|y, \theta^q)}{p(x|y, \theta)}\right]|Y = y\right] \quad (\text{III.8})$$

Le terme $\mathbb{E}_{\theta^q}\left[\log\left[\frac{p(x|y, \theta^q)}{p(x|y, \theta)}\right]\right]$ s'appelle distance de Kullback de $p(x|y, \theta^q)$ par rapport à $p(x|y, \theta)$, qui toujours positive ou nulle d'après l'inégalité de Jensen¹.

$$\mathbb{E}_{\theta^q}\left[\log\left[\frac{p(x|y, \theta^q)}{p(x|y, \theta)}\right]|Y = y\right] \geq 0 \quad (\text{III.9})$$

Si on pose $Q(\theta, \theta^q) = \mathbb{E}_{\theta^q}[\mathcal{L}_c(Z|\theta)|Y = y]$, nous obtenons l'inégalité suivante, du fait que (III.9) est positive ou nulle :

$$\mathcal{L}(y|\theta) - \mathcal{L}(y|\theta^q) \geq Q(\theta, \theta^q) - Q(\theta^q, \theta^q) \quad (\text{III.10})$$

il en résulte que toute valeur de θ qui fait croître $Q(\theta, \theta^q) - Q(\theta^q, \theta^q)$ fait aussi croître $\mathcal{L}(y|\theta)$. Et puisque $Q(\theta^q, \theta^q)$ est une constante par rapport à θ , il suffit de maximiser $Q(\theta, \theta^q)$ par rapport à θ , pour obtenir θ^{q+1} :

$$\theta^{q+1} = \underset{\theta \in \mathbb{R}^M}{\text{argmax}} Q(\theta, \theta^q) \quad (\text{III.11})$$

Pour obtenir l'étape de maximisation nous pouvons utiliser les algorithmes de maximisation tel que la descente du gradient, le gradient conjugué ou encore l'algorithme de Newton Raphson dans le cas vectoriel. Finalement, le déroulement de l'algorithme EM est le suivant :

Algorithme III.1 *Choix d'une valeur θ^0 comme valeur initiale ;*

Répéter jusqu'à critère d'arrêt :

□ *Étape (E) :*

$$Q(\theta, \theta^q) = \mathbb{E}[\mathcal{L}_c(X, Y|\theta)|Y = y, \theta^q]$$

□ *Étape (M) : choisir θ^{q+1} tel que :*

$$\theta^{q+1} = \underset{\theta \in \mathbb{R}^M}{\text{argmax}} Q(\theta, \theta^q)$$

III.1.2 Variantes de l'algorithme EM

La mise en oeuvre de l'algorithme EM pose parfois des problèmes et pour simplifier son implémentation des variantes de ce dernier ont été proposées.

1. Inégalité de Jensen : soit f une fonction convexe sur $]a, b[$ et X une v.a d'espérance fini, à valeurs dans $]a, b[$ alors $f(\mathbb{E}[X]) \leq \mathbb{E}[f(X)]$

L'algorithme GEM

L'algorithme EM généralisé (GEM) qui a été proposé par Dempster et al. (1977) [34]. Le principe de GEM consiste à choisir à l'itération $q + 1$ de l'étape (M) n'importe quelle valeur de θ^{q+1} à condition qu'elle permette d'accroître Q , ce qui assure sous certaines hypothèses ($Q(\cdot, \cdot)$ est une fonction continue par rapport aux deux variables) la convergence monotone de la log-vraisemblance $\mathcal{L}(y|\theta)$ vers une valeur stationnaire (maximum local). Une étude sur les conditions et la nature de la convergence de l'algorithme EM et l'algorithme GEM été présentée par Wu(1983) [118].

$$Q(\theta, \theta^{q+1}) \geq Q(\theta, \theta^q)$$

L'algorithme GEM se déroule comme suit :

Algorithme III.2 *Choix d'une valeur θ^0 comme valeur initiale ;*

Répéter jusqu'à critère d'arrêt :

□ *Étape (E) :*

$$Q(\theta, \theta^q) = E[\mathcal{L}_c(X, Y|\theta)|Y = y, \theta^q];$$

□ *Étape (M) : choisir θ^{q+1} tel que*

$$Q(\theta, \theta^{q+1}) \geq Q(\theta, \theta^q) :$$

Version stochastique de EM

Des versions stochastiques de l'algorithme EM ont été proposées, aussi bien afin d'améliorer son efficacité qu'afin de résoudre certains problèmes de calcul. En particulier, à cause d'une mauvaise initialisation, l'algorithme EM risque de se bloquer dans des minima locaux. Broniatowski et al. [16, 20, 21] ont proposé d'introduire une phase stochastique (S) entre l'étape (E) et l'étape (M), qui permet de perturber le système, pour éviter son blocage dans des minima locaux. Cet algorithme est appelé « stochastique EM » (SEM). Il se présente comme suit [23] :

Algorithme III.3 *Choix d'une valeur θ^0 comme valeur initiale.*

Répéter jusqu'à critère d'arrêt :

□ *Étape (E) : Calculer les densités conditionnelles $p(x_n|y_{1:N}, \theta^q)$;*

□ *Étape (S) : Simuler aléatoirement un échantillon $x_{1:N}^q$ selon la loi multinomiale de paramètres $\mathcal{M}(1, p_{1,n}^q, \dots, p_{K,n}^q)$ avec $\forall i = 1, \dots, K$ $p_{i,n}^q = p(x_n = \omega_i|y_{1:N}, \theta^q)$;*

□ *Étape (M) : Calculer l'estimateur du maximum de vraisemblance θ^{q+1} :*

$$\theta^{q+1} = \underset{\theta \in \Theta}{\operatorname{argmax}} \mathcal{L}_c(x_{1:N}^q, y_{1:N}|\theta)$$

Une autre version stochastique appelé SAEM (*Stochastic Approximation EM*), qui permet de combiner la version originale de EM et la version stochastique SEM, a été proposée par Celeux et al. [22]. En effet, à chaque itération q on considère deux processus d'estimation qui se déroulent séparément et parallèlement : EM standard qui permet d'estimer θ_{EM}^{q+1} et SEM pour estimer θ_{SEM}^{q+1} , et on pose comme le paramètre suivant :

$$\theta^{q+1} = \left(1 - \frac{1}{T^{q+1}}\right)\theta_{EM}^{q+1} + \frac{1}{T^{q+1}}\theta_{SEM}^{q+1}$$

avec T^q est une suite croissante. Le SAEM se comporte comme SEM au départ pour éviter les minima locaux, et il se comporte comme EM à la fin.

Parfois le calcul analytique de la fonction $Q(\theta, \theta^q)$ est impossible ; pour pallier à ce problème dans l'étape (E) de l'algorithme EM, Wei et *al.* [115] ont proposé le « Monte Carlo EM » (MCEM), qui consiste à tirer aléatoirement, à chaque itération q , τ réalisations du processus caché X , $\{x_1^q, \dots, x_\tau^q\}$ selon sa loi conditionnelle $p(x|y, \theta^q)$ et à poser

$$Q(\theta, \theta^q) = \frac{1}{\tau} \sum_{i=1}^{\tau} \log p(x_i^q, y, \theta)$$

Il est difficile de comparer ces différentes variantes dans le cas général. On peut cependant noter que l'algorithme EM est très sensible à l'initialisation et il risque de se bloquer dans des minima locaux non significatifs contrairement à ces versions stochastiques. Par contre la vitesse de convergence de l'algorithme SEM est plus faible que celui de EM [22]. Notons également que dans certains cas l'efficacité de l'algorithme SEM peut être supérieure à celle de l'algorithme EM [19].

III.2 Estimation Conditionnelle Itérative

L'algorithme « Estimation Conditionnelle Itérative » (en anglais « Iterative Conditional Estimation », ICE) a été proposé par Pieczynski [7, 89]. Comme EM, c'est un algorithme itératif qui permet d'estimer les paramètres d'un modèle à structure cachée. Cet algorithme est fondé sur : (i) l'existence d'un estimateur $\widehat{\theta}(X, Y)$ de paramètres θ à partir des données complètes, et (ii) la possibilité de simuler X conditionnellement aux paramètres courants et aux observations $y_{1:N}$. ICE produit alors une suite : à chaque itération, la prochaine valeur θ^{q+1} est l'espérance conditionnelle de $\widehat{\theta}$ sachant $Y = y$ et la valeur courante du paramètre θ^q . L'idée de l'ICE vient du fait que l'espérance conditionnelle est la meilleure approximation, au sens de l'erreur quadratique moyenne, d'une variable aléatoire par une autre.

$$\theta^{q+1}(y) = \mathbb{E}_{\theta^q}[\widehat{\theta}(X, Y)|Y = y] \quad (\text{III.12})$$

Dans le cas où l'espérance conditionnelle n'est pas calculable pour une composante θ_m du vecteur θ , on l'approxime par sa moyenne empirique. Et pour cela, on génère τ échantillons (x^1, \dots, x^τ) de X selon la loi $p(x|y, \theta^q)$, avec $x^i = (x_1^i, \dots, x_N^i)$

$$\theta_m^{q+1} = \frac{1}{\tau} \sum_{i=1}^{\tau} \widehat{\theta}_m(x^i, y) \quad (\text{III.13})$$

Algorithme III.4 *Choix d'une valeur θ^0 comme valeur initiale.*

Répéter jusqu'à critère d'arrêt :

□ *Calculer $\theta^{q+1} = (\theta_1^{q+1}, \dots, \theta_M^{q+1})$.*

• *Pour tout $m = 1, \dots, M$*

◇ *Si l'espérance mathématique est calculable, alors :*

$$\theta_m^{q+1} = \mathbb{E}_{\theta^q}[\widehat{\theta}_m(X, Y)|Y = y, \theta^q]$$

- ◊ *S'il existe des composantes θ_m pour lesquelles l'espérance ci-dessus n'est pas calculable, on simule τ réalisations de X selon $p(x|y, \theta^q)$ et on pose :*

$$\theta_m^{q+1} = \frac{1}{\tau} \sum_{i=1}^{\tau} \widehat{\theta}_m(x^i, y)$$

pour toutes ces composantes.

Dans certains cas, comme le cas des CMC-BI classiques avec du bruit gaussien [7], celui des arbres de Markov cachés, avec bruit gaussien ou pas [86], ou encore mélanges gaussiens simples [87], les algorithmes EM, SEM et ICE peuvent donner des résultats très proches. Au plan théorique, les liens entre EM et ICE ont été étudiés par Delmas [33]. L'avantage de l'ICE est qu'il est facile à programmer et nous n'avons pas des problèmes de maximisation sous contraintes contrairement à l'algorithme EM. En général, nous avons plusieurs choix pour les estimateurs de paramètres et parmi ces choix l'estimateur de maximum de vraisemblance. La vitesse de convergence de l'algorithme ICE dépend essentiellement du choix de ces estimateurs. Une étude sur la convergence de ICE était présentée récemment par M. Pieczynski [95].

III.3 Applications

Nous présentons dans cette section des applications de l'estimation des paramètres pour quelques types de modèle à données incomplètes. Soit $X = (X_n)_{n=1:N}$ un processus aléatoire discret à valeurs dans un espace fini $\Omega = \{\omega_1, \dots, \omega_K\}$. X est inobservé, et les données que nous disposons sont représentées par le processus $Y = (Y_n)_{n=1:N}$, qui est un processus réel. On note par $Z = (Z_n)_{n=1:N}$ le couple (X, Y) . La première modélisation classique que nous étudions est le cas d'un mélange indépendant gaussien fini.

III.3.1 Mélange indépendant gaussien fini

Les hypothèses de ce modèle sont les suivantes : les X_n sont indépendants, et les Y_n sont indépendants conditionnellement à X , Y_n et $(X_t)_{t \neq n}$ sont indépendants conditionnellement à X_n .

$$p(x) = \prod_{n=1}^N p(x_n) \tag{III.14}$$

$$p(y|x) = \prod_{n=1}^N p(y_n|x) = \prod_{n=1}^N p(y_n|x_n) \tag{III.15}$$

Dans le cas de mélange des gaussiennes considéré, la loi conditionnelle de Y_n sachant $x_n = \omega_k$ est une gaussienne de moyenne μ_k et de variance σ_k^2 $\mathcal{N}(\mu_k, \sigma_k^2)$. La log-vraisemblance est donnée par :

$$\mathcal{L}_c(z|\theta) = \log p(x, y|\theta) = \log p(x|\theta) + \log p(y|x, \theta) \tag{III.16}$$

$$= \sum_{n=1}^N \log p(x_n) + \sum_{n=1}^N \log p(y_n|x_n) \tag{III.17}$$

Les paramètres du modèle sont les probabilités $\pi_k = p(x_n = \omega_k)$ et les paramètres de la loi conditionnelle de Y_n sachant X_n , qui sont les moyennes μ_k et les variances σ_k^2 . Nous avons $3K$ paramètres à estimer, et comme $\sum_{k=1}^K \pi_k = 1$ le nombre des paramètres sera réduit à $3K - 1$.

$$\theta = (\pi_1, \dots, \pi_k, \mu_1, \dots, \mu_K, \sigma_1^2, \dots, \sigma_K^2) \quad (\text{III.18})$$

L'espérance conditionnelle de $\mathcal{L}_c(z|\theta)$ sachant $y_{1:N}$ et la valeur courante de θ notée θ^q est :

$$\begin{aligned} Q(\theta, \theta^q) &= \mathbb{E}[\mathcal{L}(z|\theta)|y, \theta^q] \\ &= \sum_{n=1}^N \sum_{k=1}^K \log(\pi_k) E[1(x_n = \omega_k)|y, \theta^q] \\ &+ \sum_{n=1}^N \sum_{k=1}^K \log p(y_n|x_n = \omega_k, \theta) E[1(x_n = \omega_k)|y, \theta^q] \end{aligned}$$

Si on note $\gamma_n(k, \theta^q)$ la probabilité $p(x_n = \omega_k|y, \theta^q) = \mathbb{E}[1(x_n = \omega_k)|y, \theta^q]$, la fonction à maximiser est :

$$\begin{aligned} Q(\theta, \theta^q) &= \sum_{n=1}^N \sum_{k=1}^K \gamma_n(k, \theta^q) \log(\pi_k) \\ &+ \sum_{n=1}^N \sum_{k=1}^K \gamma_n(k, \theta^q) \log p(y_n, \mu_k, \sigma_k^2) \end{aligned} \quad (\text{III.19})$$

L'étape (E) de l'algorithme EM consiste à calculer $\gamma_n(k, \theta^q)$, qui est donnée par :

$$\gamma_n(k, \theta^q) = p(x_n|y, \theta^q) = \frac{\pi_k^q \mathcal{N}(y_n, \mu_k^q, (\sigma_k^2)^q)}{\sum_{k=1}^K \pi_k^q \mathcal{N}(y_n, \mu_k^q, (\sigma_k^2)^q)} \quad (\text{III.20})$$

L'étape de maximisation de la fonction $Q(\theta, \theta^q)$ par rapport à π_k , sous la contrainte $\sum_{k=1}^K \pi_k = 1$, donne comme solution :

$$\pi_k^{q+1} = \frac{1}{N} \sum_{n=1}^N \gamma_n(k, \theta^q), \quad (\text{III.21})$$

et les paramètres de la loi du Y_n sachant $x_n = \omega_k$ sont donnés par :

$$\mu_k^{q+1} = \frac{\sum_{n=1}^N y_n \gamma_n(k, \theta^q)}{\sum_{n=1}^N \gamma_n(k, \theta^q)} \quad (\text{III.22})$$

$$(\sigma_k^2)^{q+1} = \frac{\sum_{n=1}^N (y_n - \mu_k^{q+1})^2 \gamma_n(k, \theta^q)}{\sum_{n=1}^N \gamma_n(k, \theta^q)} \quad (\text{III.23})$$

Nous pouvons également appliquer l'algorithme ICE pour estimer les paramètres de ce modèle. En effet, à chaque étape $q + 1$ les (π_1, \dots, π_K) sont estimables par l'espérance mathématique de leurs estimateurs. Choisissons comme estimateur à partir des données complètes $\widehat{\pi}_k(x, y)$ comme estimateur

$$\widehat{\pi}_k(x, y) = \frac{\sum_{n=1}^N 1(x_n = \omega_k)}{N}$$

alors sachant la valeur courante θ^q et les observations $y = (y_n)_{1:N}$, nous pouvons calculer π_i^{q+1} par :

$$\pi_i^{q+1} = \mathbb{E}_{\theta^q}[\widehat{\pi}_i(x, y)|y] = \frac{1}{N} \sum_{n=1}^N p(x_n = \omega_i | y_{1:N}, \theta^q) = \frac{1}{N} \sum_{n=1}^N \gamma_n(i, \theta^q) \quad (\text{III.24})$$

et nous retrouvons le même estimateur pour π_i^{q+1} que par l'algorithme EM donné par l'équation (III.34).

Les estimateurs classiques pour les moyennes et les variances à partir des données complètes sont les suivants :

$$\widehat{\mu}_k^{q+1} = \frac{\sum_{n=1}^N y_n \mathbf{1}(x_n = \omega_k)}{\sum_{n=1}^N \mathbf{1}(x_n = \omega_k)} \quad (\text{III.25})$$

$$(\widehat{\sigma}_k^2)^{q+1} = \frac{\sum_{n=1}^N (y_n - \mu_k^{q+1})^2 \mathbf{1}(x_n = \omega_k)}{\sum_{n=1}^N \mathbf{1}(x_n = \omega_k)} \quad (\text{III.26})$$

Les espérances mathématiques de ces estimateurs ne sont pas calculables explicitement ; conformément au principe de l'ICE, on effectue des tirages des données cachées selon leurs lois conditionnelles aux observations et on applique les estimateurs ci-dessus aux données simulées.

III.3.2 CMC-BI gaussiennes

Rappelons que dans les CMC-BI X est une chaîne de Markov, les Y_n sont indépendants conditionnellement à X , et Y_n et $(X_t)_{t \neq n}$ sont indépendants conditionnellement à X_n . De plus, nous supposons que Y_n sachant X_n qui sont des lois gaussiennes. Nous avons donc

$$p(x) = p(x_1) \prod_{n=1}^{N-1} p(x_{n+1} | x_n) \quad (\text{III.27})$$

$$p(y|x) = \prod_{n=1}^N p(y_n | x) = \prod_{n=1}^N p(y_n | x_n) \quad (\text{III.28})$$

Les paramètres de ce modèle sont regroupés en deux ensembles : le premier est constitué des paramètres de la loi du processus X , que nous supposons homogène, donné par sa loi initiale $\pi = (\pi_1, \dots, \pi_K)$ et la matrice des transitions $A = (a_{i,j})_{i=1:K, j=1:K}$ avec $a_{i,j} = p(x_{n+1} = \omega_i | x_n = \omega_j)$. Le deuxième est celui des paramètres des loi de Y_n conditionnellement à X_n qui sont choisies comme des gaussiennes dans notre exemple. Le vecteur des paramètres θ est donc :

$$\theta = (\pi_1, \dots, \pi_K, a_{1,1}, \dots, a_{K,K}, \mu_1, \dots, \mu_K, \sigma_1^2, \dots, \sigma_K^2) \quad (\text{III.29})$$

Le nombre des paramètres du modèle peut être réduit du fait qu'il y a des contraintes, comme $\sum_{k=1}^K \pi_k = 1$ et $\forall j = 1, \dots, K \sum_{i=1}^K a_{i,j} = 1$. On suppose par ailleurs que la chaîne X est réversible ($p(X_{n+1} = \omega_i | X_n = \omega_j) = p(X_{n+1} = \omega_j | X_n = \omega_i)$), ce qui implique $a_{i,j} = a_{j,i}$

La log-vraisemblance est donnée par :

$$\mathcal{L}_c(z|\theta) = \log p(x_1) + \sum_{n=1}^{N-1} \log p(x_{n+1}, x_n) + \sum_{n=1}^N \log p(y_n | x_n) \quad (\text{III.30})$$

et son espérance conditionnelle sachant y et la valeur courante des paramètres θ^q est :

$$Q(\theta, \theta^q) = \sum_{k=1}^K \log(\pi_k) \mathbb{E}[1(x_1 = \omega_k) | y, \theta^q] + \sum_{n=1}^{N-1} \sum_{i=1}^K \sum_{j=1}^K \log(a_{i,j}) \mathbb{E}[1(x_{n+1} = \omega_i, x_n = \omega_j) | y, \theta^q] \\ + \sum_{n=1}^N \sum_{i=1}^K \log p(y_n | x_n = \omega_i, \theta) \mathbb{E}[1(x_n = \omega_i) | y, \theta^q]$$

En notant $\gamma_n(i, \theta^q) = p(x_n = \omega_i | y, \theta^q)$ et $\varphi_n(i, j, \theta^q) = p(x_n = \omega_i, x_{n+1} = \omega_j | y, \theta^q)$, nous obtenons la fonction à maximiser :

$$Q(\theta, \theta^q) = \sum_{i=1}^K \gamma_1(i, \theta^q) \log(\pi_i) + \sum_{n=1}^{N-1} \sum_{i=1}^K \sum_{j=1}^K \xi_n(i, j, \theta^q) \log(a_{i,j}) \\ + \sum_{n=1}^N \sum_{i=1}^K \gamma_n(i, \theta^q) \log p(y_n, \mu_i, \sigma_i) \quad (\text{III.31})$$

L'étape (E) consiste à calculer les $\gamma_n(i, \theta^q)$ et les $\varphi_n(i, j, \theta^q)$, qui sont calculables par l'algorithme de Baum-Welsh sachant les paramètres courants θ^q . Nous avons

$$\gamma_n(i, \theta^q) = p(x_n = \omega_i | y, \theta^q) = \frac{\alpha_n(i) \beta_n(i)}{\sum_{j=1}^K \alpha_n(j) \beta_n(j)} \quad (\text{III.32})$$

par ailleurs nous avons :

$$p(x_{n+1} = \omega_j | x_n = \omega_i, y) = \frac{\beta_{n+1}(j)}{\beta_n(i)} p(x_{n+1} | x_n) p(y_{n+1} | x_{n+1})$$

en utilisant la formule de Bayes :

$$p(x_n = \omega_i, x_{n+1} = \omega_j | y) = p(x_n = \omega_i | y) p(x_{n+1} = \omega_j | x_n = \omega_i, y)$$

nous obtenons :

$$\varphi_n(i, j, \theta^q) = p(x_n = \omega_i, x_{n+1} = \omega_j | y, \theta^q) = \frac{\alpha_n(i) a_{i,j} p(y_{n+1}, \phi_j) \beta_{n+1}(j)}{\sum_{i=1}^K \sum_{j=1}^K \alpha_n(i) a_{i,j} p(y_{n+1}, \phi_j) \beta_{n+1}(j)} \quad (\text{III.33})$$

Une fois les quantités $\gamma_n(i, \theta^q)$ et $\varphi_n(i, j, \theta^q)$ calculées, nous pouvons passer à l'étape de maximisation (M) de la fonction $Q(\theta, \theta^q)$ par rapport aux paramètres du modèle. La maximisation de cette fonction fait intervenir le multiplicateur de Lagrange pour prendre en considération les contraintes du modèle. La première contrainte est $\sum_{i=1}^K \pi_i = 1$ et la deuxième est $\forall j = 1, \dots, K \sum_{i=1}^K a_{i,j} = 1$. Le premier terme fait intervenir la première contrainte et nous obtenons après résolution :

$$\pi_i^{q+1} = \frac{1}{N} \sum_{n=1}^N \gamma_n(i, \theta^q) \quad (\text{III.34})$$

Le deuxième terme fait intervenir la contrainte $\forall j = 1, \dots, K \sum_{i=1}^K a_{i,j} = 1$, et $a_{i,j}^{q+1}$ est donnée par :

$$a_{i,j}^{q+1} = \frac{\sum_{n=1}^{N-1} \varphi_n(i, j, \theta^q)}{\sum_{n=1}^{N-1} \gamma_n(i, \theta^q)} \quad (\text{III.35})$$

Le troisième terme de l'équation (III.31) donne :

$$\mu_i^{q+1} = \frac{\sum_{n=1}^N \gamma_n(i, \theta^q) y_n}{\sum_{n=1}^N \gamma_n(i, \theta^q)} \quad (\text{III.36})$$

$$\sigma_i^{q+1} = \frac{\sum_{n=1}^N \gamma_n(i, \theta^q) (y_n - \mu_i^{q+1})(y_n - \mu_i^{q+1})^T}{\sum_{n=1}^N \gamma_n(i, \theta^q)} \quad (\text{III.37})$$

Les paramètres d'une CMC-BI sont également estimables par l'algorithme ICE. En effet, soit X une chaîne cachée stationnaire ($p(x_{n+1}, x_n)$ ne dépend pas de n) les estimateurs des paramètres de la loi de X sont calculables à chaque itération de l'algorithme ICE. En utilisant l'algorithme de Baum-Welsh et sachant la valeur courante θ^q de θ et les observations $y_{1:N}$ nous pouvons calculer les quantités $\varphi_n(i, j, \theta^q)$ et $\gamma_n(i, \theta^q)$. Si nous choisissons $\widehat{\pi}_i(x, y)$ comme estimateur pour π_i à partir des données complètes :

$$\widehat{\pi}_i(x, y) = \frac{\sum_{n=1}^N 1(x_n = \omega_i)}{N},$$

alors l'estimation de π_i à l'itération $q+1$ est donné par son espérance conditionnelle aux observations. Elle est ici calculable et nous avons :

$$\pi_i^{q+1} = \mathbb{E}_{\theta^q}[\widehat{\pi}_i(x, y)|Y = y] = \frac{1}{N} \sum_{n=1}^N \gamma_n(i, \theta^q) \quad (\text{III.38})$$

Le calcul des transitions $a_{i,j}$ fait intervenir les probabilités $c_{i,j} = p(x_{n+1} = \omega_i, x_n = \omega_j)$,

$$a_{i,j} = p(x_{n+1}|x_n) = \frac{p(x_{n+1}, x_n)}{p(x_n)} = \frac{c_{i,j}}{p(x_n)}.$$

L'estimateur empirique des probabilités $c_{i,j}$ sachant les observations complètes $(x_{1:N}, y_{1:N})$ est :

$$\widehat{c}_{i,j} = \frac{\sum_{n=1}^{N-1} 1(x_{n+1} = \omega_i, x_n = \omega_j)}{N-1}. \quad (\text{III.39})$$

L'espérance conditionnelle aux observations de cet estimateur de $c_{i,j}$ est à nouveau calculable ; en effet, à l'itération $q+1$ elle est donnée par :

$$c_{i,j}^{q+1} = \mathbb{E}_{\theta^q}[\widehat{c}_{i,j}(x, y)|Y = y] = \frac{1}{N-1} \sum_{n=1}^{N-1} \varphi_n(i, j, \theta^q) \quad (\text{III.40})$$

Finalement, nous obtenons les coefficients de la matrice de transition $a_{i,j}^{q+1}$ par :

$$a_{i,j}^{q+1} = \frac{\sum_{n=1}^{N-1} \varphi_n(i, j, \theta^q)}{\sum_{n=1}^{N-1} \gamma_n(i, \theta^q)}. \quad (\text{III.41})$$

Il est intéressant de noter que nous obtenons, par une démarche différente, une formule identique à celle donnée par l'algorithme EM. Le deuxième sous ensemble des paramètres sont les paramètres de la loi de Y_n sachant X_n . Les estimateurs de ces paramètres à partir des données complètes, rappelés ci-dessous, sont les mêmes que ceux utilisés dans le cas indépendant dans la sous-section précédente. Comme dans le cas indépendant, le calcul de l'espérance conditionnelle de ces estimateurs n'est

pas possible, donc nous faisons recours à l'approximation par le biais des tirages de X selon sa loi conditionnelle à Y . La différence avec le cas indépendant est qu'ici cette loi est markovienne. Comme dans le cas indépendant, les estimateurs à partir des complètes sont :

$$\widehat{\mu}_k = \frac{\sum_{n=1}^N y_n 1(x_n = \omega_k)}{\sum_{n=1}^N 1(x_n = \omega_k)} \quad (\text{III.42})$$

$$\widehat{\sigma}_k^2 = \frac{\sum_{n=1}^N (y_n - \widehat{\mu}_k)^2 1(x_n = \omega_k)}{\sum_{n=1}^N 1(x_n = \omega_k)} \quad (\text{III.43})$$

Notons que dans la pratique on effectue souvent un seul tirage à chaque itération de l'ICE (on prend $\tau = 1$, voir la description de l'ICE).

III.3.3 CMCouple

Nous traitons dans cette sous-section l'estimation des paramètres pour le modèle de la chaîne de Markov couple, qui généralise les modèles classiques comme nous l'avons vu dans le deuxième chapitre. Soit Z une chaîne de Markov couple, dont la densité de loi s'écrit :

$$p(z) = \frac{\prod_{n=1}^{N-1} p(z_n, z_{n+1})}{\prod_{n=2}^{N-1} p(z_n)} \quad (\text{III.44})$$

Cette densité peut être développée sous une autre forme. En effet, la loi du couple (Z_n, Z_{n+1}) peut s'écrire $p(z_n, z_{n+1}) = p(x_n, x_{n+1})p(y_n, y_{n+1}|x_n, x_{n+1})$, ce qui permet de mettre (III.44) sous la forme :

$$p(z) = \frac{\prod_{n=1}^{N-1} p(x_n, x_{n+1}) \prod_{n=1}^{N-1} p(y_n, y_{n+1}|x_n, x_{n+1})}{\prod_{n=2}^{N-1} p(z_n)} \quad (\text{III.45})$$

Considérons d'abord l'extension aux CMCouples de l'algorithme EM.

La log-vraisemblance des données complètes est :

$$\mathcal{L}_c(z|\theta) = \sum_{n=1}^{N-1} \log p(x_n, x_{n+1}) - \sum_{n=2}^{N-1} \log p(z_n) + \sum_{n=1}^{N-1} \log p(y_n, y_{n+1}|x_n, x_{n+1}) \quad (\text{III.46})$$

Posons :

$$\begin{aligned} \gamma_n(k, \theta^q) &= p(x_n = \omega_k | y, \theta^q) \\ \varphi_n(i, j, \theta^q) &= p(x_n = \omega_i, x_{n+1} = \omega_j | y, \theta^q) \\ c_{i,j} &= p(x_n = \omega_i, x_{n+1} = \omega_j) \\ f_{i,j}(y_n, y_{n+1}) &= p(y_n, y_{n+1} | x_n = \omega_i, x_{n+1} = \omega_j) \\ b_k(y_n) &= p(x_n = \omega_k, y_n) \end{aligned}$$

Nous obtenons alors la fonction à maximiser suivante :

$$\begin{aligned}
 Q(\theta, \theta^q) &= \sum_{n=1}^{N-1} \sum_{i=1}^K \sum_{j=1}^K \varphi_n(i, j, \theta^q) \log c_{i,j} \\
 &\quad - \sum_{n=2}^{N-1} \sum_{k=1}^K \gamma_n(k, \theta^q) \log b_k(y_n) \\
 &\quad + \sum_{n=1}^{N-1} \sum_{i=1}^K \sum_{j=1}^K \varphi_n(i, j, \theta^q) \log f_{i,j}(y_n, y_{n+1})
 \end{aligned} \tag{III.47}$$

L'étape (E) de l'algorithme EM consiste à calculer les quantités $\gamma_n(k, \theta^q)$ et $\varphi_n(i, j, \theta^q)$; elles sont calculables par l'algorithme de Baum-Welsh que nous l'avons présenté dans le deuxième chapitre. L'étape (M) fait intervenir le multiplicateur de Lagrange pour inclure les contraintes suivantes :

1. $\sum_{i=1}^K \sum_{j=1}^K c_{i,j} = 1$
2. $\forall j = 1 : K, \sum_{i=1}^K \int_{u \in \mathcal{Y}} c_{i,j} f_{i,j}(u, v) = b_j(v)$
3. $\forall (i, j) \in [1 : K]^2, \iint f_{i,j}(u, v) du dv = 1$

La solution de ce problème d'optimisation est (l'algorithme EM a été étendu aux chaînes de Markov Couple par P. Lanchantin ; les détails des calculs peuvent être consultés dans sa thèse [64]) :

$$c_{i,j}^{q+1} = \frac{\sum_{n=1}^{N-1} \varphi_n(i, j, \theta^q)}{N-1}, \tag{III.48}$$

$$f_{i,j}^{q+1}(u, v) = \frac{\sum_{n=1}^{N-1} [(y_n, y_{n+1}) = (u, v)] \varphi_n(i, j, \theta^q)}{\sum_{n=1}^{N-1} \varphi_n(i, j, \theta^q)}. \tag{III.49}$$

Dans le cas gaussien c'est à dire $(Y_{n:n+1} | x_{n:n+1} = (\omega_i, \omega_j)) \sim \mathcal{N}_2(m_{i,j}, \Gamma_{i,j})$ (rappelons que malgré la gaussianité de $(Y_{n:n+1} | x_{n:n+1} = (\omega_i, \omega_j))$, dans les CMCouples la loi de Y conditionnelle à X n'est pas nécessairement gaussienne), les vecteurs moyens et les matrices de co-variance sont re-estimés par :

$$m_{i,j}^{q+1} = \frac{\sum_{n=1}^{N-1} (y_n, y_{n+1})^T \varphi_n(i, j, \theta^q)}{\sum_{n=1}^{N-1} \varphi_n(i, j, \theta^q)} \tag{III.50}$$

$$\Gamma_{i,j}^{q+1} = \frac{\sum_{n=1}^{N-1} (y_{n:n+1} - m_{i,j}^{q+1})(y_{n:n+1} - m_{i,j}^{q+1})^T \varphi_n(i, j, \theta^q)}{\sum_{n=1}^{N-1} \varphi_n(i, j, \theta^q)} \tag{III.51}$$

La deuxième méthode que nous pouvons utiliser afin d'estimer les paramètres d'une CMCouple est l'algorithme ICE. Si nous supposons que (X, Y) est stationnaire, le premier sous-ensemble des paramètres est la loi $p(x_{n+1}, x_n)$ (rappelons que ce paramètre ne définit pas nécessairement la loi de X car X n'est pas nécessairement markovien). Comme dans le cas des CMC, on prend pour l'estimateur à partir des données complètes l'estimateur :

$$\widehat{c}_{i,j} = \frac{\sum_{n=1}^{N-1} 1[x_{n:n+1} = (\omega_i, \omega_j)]}{N-1}, \tag{III.52}$$

alors l'espérance mathématique de cet estimateur est calculable à chaque itération et nous obtenons $c_{i,j}^{q+1}$ par :

$$c_{i,j}^{q+1} = \frac{\sum_{n=1}^{N-1} \varphi_n(i, j, \theta^q)}{N-1}. \quad (\text{III.53})$$

Le deuxième sous ensembles des paramètres est constitué par les paramètres de la loi conditionnelle de (Y_n, Y_{n+1}) sachant (X_{n+1}, X_n) . Dans le cas gaussien, ces paramètres correspondent au vecteur moyen conditionnel $m_{i,j}$ et au matrice de co-variance conditionnelle $\Gamma_{i,j}$. Comme estimateur empirique de vecteur moyen nous choisissons :

$$\widehat{m}_{i,j} = \frac{\sum_{n=1}^{N-1} (y_{n:n+1}) \mathbf{1}[x_{n:n+1} = (\omega_i, \omega_j)]}{\sum_{n=1}^{N-1} \mathbf{1}[x_{n:n+1} = (\omega_i, \omega_j)]}, \quad (\text{III.54})$$

et pour la matrice de co-variance :

$$\widehat{\Gamma}_{i,j} = \frac{\sum_{n=1}^{N-1} (y_{n:n+1} - \widehat{m}_{i,j})(y_{n:n+1} - \widehat{m}_{i,j})^T \mathbf{1}[x_{n:n+1} = (\omega_i, \omega_j)]}{\sum_{n=1}^{N-1} \mathbf{1}[x_{n:n+1} = (\omega_i, \omega_j)]}. \quad (\text{III.55})$$

Comme dans les cas précédents, le calcul des espérances conditionnelles de ces estimateurs n'est pas possible et on utilise les simulations des réalisations de la chaîne cachée selon sa loi conditionnelle aux observations.

III.3.4 Exemple numérique

Comme application, nous utilisons l'algorithme ICE pour estimer les paramètres d'une chaîne de Markov cachée à bruit indépendant de taille $N = 128 \times 128$. Nous appliquons l'algorithme dans une série d'expériences suivante. On fait varier d'une expérience à l'autre les paramètres de la chaîne ainsi que les paramètres du bruit. Les résultats de l'estimation sont représentés dans la table (III.1). Pour initialiser les paramètres du modèle nous utilisons l'algorithme des K-moyennes pour obtenir x^0 , et sachant ce x^0 et $y_{1:N}$ nous estimons θ^0 qui prise comme la valeur initiale pour θ . Pour chaque essai nous itérons 100 fois. Nous pouvons noter que l'estimation des paramètres du bruit reste de bonne qualité, même dans l'exemple 3 où le niveau du bruit est important. L'estimation de la loi $p(x_{n+1}, x_n)$ semble moins robuste au bruit.

	Vrais paramètres						Paramètres estimés					
	p_1	p_2	μ_1	σ_1^2	μ_2	σ_2^2	\widehat{p}_1	\widehat{p}_2	$\widehat{\mu}_1$	$\widehat{\sigma}_1^2$	$\widehat{\mu}_2$	$\widehat{\sigma}_2^2$
1	0.95	0.05	0.0	0.5	2.0	0.5	0.94	0.04	0.01	0.5	2.0	0.5
2	0.95	0.05	0.0	1.0	2.0	1.0	0.91	0.08	0.01	1.0	1.99	1.01
3	0.98	0.02	1.0	1.0	2.0	1.0	0.91	0.08	1.01	0.97	2.03	0.98

TABLE III.1 – Estimation des paramètres par l'algorithme ICE dans le cas de modèle CMC-BI [$p_1 = p(x_{n+1} = \omega_1 | x_n = \omega_1)$ $p_2 = p(x_{n+1} = \omega_2 | x_n = \omega_1)$]

La table (III.2) représente les taux d'erreur en pourcentage de la restauration de la chaîne X à partir de Y . La première colonne contient les taux d'erreur obtenus avec restauration en mode supervisé i.e sachant les vrais paramètres du modèle. Dans

la deuxième colonne nous présentons les taux obtenus en utilisant la restauration MPM en mode non supervisé, i.e en estimant les paramètres avec ICE.

Expériences	Mode Supervisé	Mode Non Supervisé
1	1.3	1.28
2	3.0	3.02
3	5.62	5.64

TABLE III.2 – Taux d'erreur(%)

Finalement, pour comparer les algorithmes d'estimation ICE et EM sur des données qui ne suivent pas le modèle exactement, nous considérons une image artificielle à deux classes $\{\omega_1, \omega_2\}$ bruitée avec un bruit gaussien indépendant. Les résultats sont représentés sur la figure (III.1).

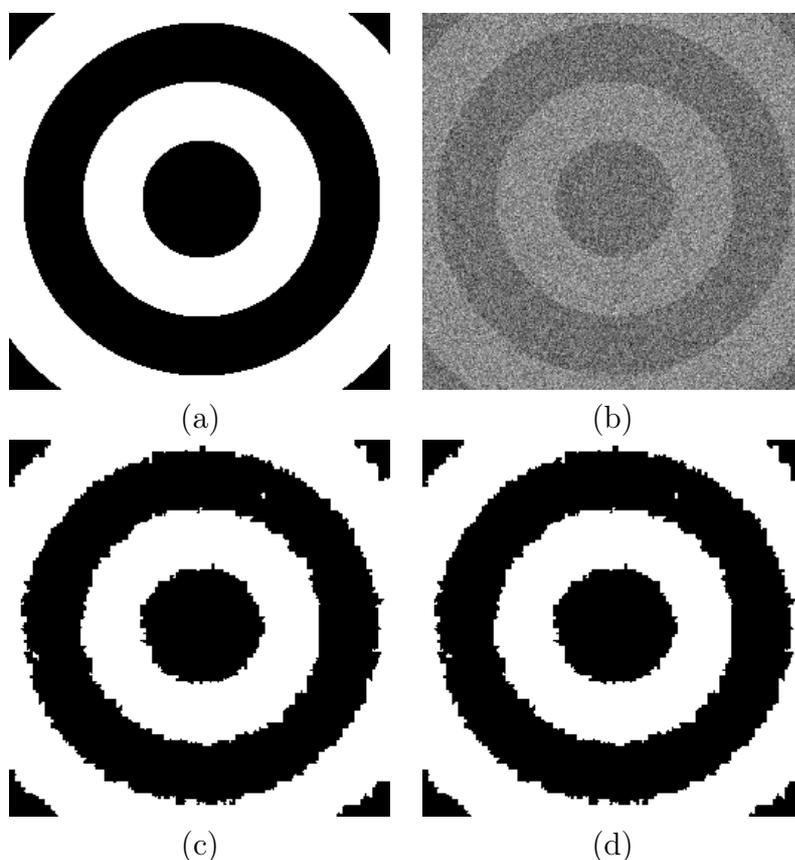


FIGURE III.1 – Exemple de segmentation non-supervisé avec estimation des paramètres par ICE et EM : (a) Image originale (b) Image bruitée (c) MPM+ICE 3.04%(d)MPM+EM 3.04%

Les valeurs des paramètres réelles (le paramètre "vrai" p est celui estimé à partir de l'image des classes X) sont données dans la table (III.3) ainsi que les paramètres estimés par les algorithmes ICE et EM.

	ω_1			ω_2			taux d'erreur (%)
	p	m	σ^2	p	m	σ^2	
Vrais paramètres	0.49	0.00	1.00	0.51	1.00	1.00	-
ICE	0.49	0.01	1.01	0.51	0.99	1.01	3.04
EM	0.48	0.01	1.01	0.51	0.99	1.01	3.04

TABLE III.3 – Estimation des paramètres par ICE et EM

Notons que les estimées de p donnent pour les estimées des matrices des transitions $\mathcal{P} = p(x_{n+1}|x_n)$

$$\mathcal{P}_{rel} = \begin{pmatrix} 0.98 & 0.02 \\ 0.02 & 0.98 \end{pmatrix}, \widehat{\mathcal{P}}_{ICE} = \begin{pmatrix} 0.99 & 0.01 \\ 0.00 & 1.00 \end{pmatrix} \text{ et } \widehat{\mathcal{P}}_{EM} = \begin{pmatrix} 0.99 & 0.01 \\ 0.01 & 0.99 \end{pmatrix}$$

Nous notons que les deux méthodes d'estimation ICE et EM présentent, dans le cadre de l'expérience, des qualités comparables.

III.4 Conclusion

L'estimation des paramètres d'un modèle statistique est une phase fondamentale pour segmenter des données à variables latentes. Dans ce chapitre nous avons présenté différents algorithmes de l'estimation itérative : l'algorithme EM, certaines de ces variantes stochastiques, ainsi que l'algorithme ICE. Nous avons détaillé les calculs pour les modèles CMC et CMCouple, et nous avons présenté quelques résultats numériques originaux d'application de EM et de ICE sur ces modèles. L'équivalence entre les performances de EM et ICE, déjà connue dans les CMC classiques avec du bruit gaussien, est montrée une nouvelle fois dans les exemples numériques traités. Dans la suite, nous choisissons d'utiliser l'algorithme ICE.

Chapitre IV

Modèle de Markov Triplet

Dans ce chapitre nous présentons un modèle récent, qui est une extension des modèles Markov cachés (MMC) et des modèles de Markov couples (MMCouple). Proposé par Pieczynski et all. dans [92, 101, 103] ce modèle est appelé « modèle de Markov Triplet » (MMT). Le principe de ce modèle consiste à introduire un troisième processus U , qui n'a pas nécessairement une interprétation physique. On note par V le processus couple (X, U) , et par T le processus triplet $(X, U, Y) = (V, Y)$. On suppose alors que le processus T est un processus de Markov. Dans un premier temps, on va s'intéresser dans ce chapitre aux chaînes de Markov triplet et on distinguera deux cas, selon les valeurs de (X, U) qui peuvent être discrètes ou mixtes [80]. Dans un deuxième temps on va introduire les champs de Markov triplets.

On rappelle que $T = (X, U, Y)$ est une chaîne de Markov triplet si et seulement si T admet pour graphe d'indépendance conditionnel le graphe \mathcal{G} (I.4) (c'est-à-dire pour tout $1 < n < N$, t_n admet pour voisins t_{n-1} et t_{n+1}). La loi de T s'écrit alors :

$$p(t) = p(t_1) \prod_{n=1}^{N-1} p(t_{n+1}|t_n) \quad (\text{IV.1})$$

Notons \mathcal{X} et \mathcal{U} les deux espaces d'états de X et U respectivement. Nous allons détailler dans les paragraphes qui suivent deux cas :

- Cas discret : les deux espaces sont discrets finis.
- Cas Mixte : l'un est discret et l'autre est réel.

Nous commençons par étudier le cas discret, nous traitons particulièrement la modélisation de la M-stationnarité et la semi-markovianité de X à l'aide du processus auxiliaire U . Ensuite, nous mettons en évidence nos apports originaux dans le cas mixte. Notre contribution dans cette partie consiste à modéliser la non stationnarité par un processus à espace d'états continu.

IV.1 Chaîne de Markov Triplet : Cas discret

Considérons deux processus $X = (X_n)_{n=1:N}$ et $U = (U_n)_{n=1:N}$ discrets, les X_n et les U_n prenant leurs valeurs dans les espaces finis respectifs $\mathcal{X} = \{\omega_1, \dots, \omega_K\}$, $\Lambda = (\lambda, \dots, \lambda_M)$. Le processus $Y = (Y_n)_{n=1:N}$ est réel ; c'est à dire les Y_n sont à valeurs dans $\mathcal{Y} = \mathbb{R}$. On note par $T = (T_n)_{n=1:N}$ le triplet (X, U, Y) . On suppose que T est une chaîne de Markov. En posant $V = (X, U)$, nous pouvons alors utiliser les

résultats présentés dans le chapitre (II) du fait que le couple $T = (V, Y)$ est une CMCouple. Les marginales a posteriori $p(x_n|y_{1:N})$ sont obtenues par :

$$p(x_n|y_{1:N}) = \sum_{u_n \in \Lambda} p(x_n, u_n|y_{1:N}), \quad (\text{IV.2})$$

avec les marginales a posteriori $p(x_n, u_n|y_{1:N})$ calculées en utilisant les probabilités progressives α_n et les probabilités rétrogrades β_n obtenues en appliquant les récursions suivantes, valables dans les CMCouples, à $v_n = (x_n, u_n)$:

$$\alpha_1(v_1) = p(t_1) \text{ et } \alpha_{n+1}(v_{n+1}) = \sum_{v_n \in \mathcal{X} \times \Lambda} \alpha_n(v_n) p(t_{n+1}|t_n) \quad \forall 2 \leq n \leq N \quad (\text{IV.3})$$

$$\beta_N(v_N) = 1 \text{ et } \beta_n(v_n) = \sum_{v_{n+1} \in \mathcal{X} \times \Lambda} \beta_{n+1}(v_{n+1}) p(t_{n+1}|t_n) \quad \forall 1 \leq n \leq N-1 \quad (\text{IV.4})$$

Le modèle CMT regroupe plusieurs modèles particuliers qui généralisent les modèles présentés dans le chapitre (I) et le chapitre (II). En effet, rappelons la généralisation récente suivante [66] de la proposition (II.3). Soit $W = (H, F)$ une chaîne couple, avec $W_n = (H_n, F_n)$ à valeurs dans l'espace produit $\mathcal{H} \times \mathcal{F}$. Soit η une mesure σ -additive sur \mathcal{H} , et soit μ une mesure σ -additive sur \mathcal{F} . Supposons que W est une chaîne de Markov et on note par la même lettre p les densités des variables par rapport aux mesures η et μ . En pratique, les mesures utilisées sont celle de comptage ou celle de Lebesgue. Cependant, des mesures mixtes faisant intervenir des masses de Dirac et la mesure de Lebesgue peuvent être envisagées comme dans [109]. Nous avons le résultat suivant :

Proposition IV.1 *Soit $W = (H, F) = (H_n, F_n)_{1:N}$ une chaîne de Markov vérifiant :*

- i) $p(w_n, w_{n+1})$ ne dépend pas de $1 \leq n \leq N-1$;*
- ii) $p(w_n = a, w_{n+1} = b) = p(w_n = b, w_{n+1} = a) \quad \forall 1 \leq n \leq N-1$ et $\forall (a, b) \in (\mathcal{H} \times \mathcal{F})^2$.*

Alors les trois conditions suivantes :

- a) H est une chaîne de Markov (c'est-à-dire $W = (H, F)$ est une chaîne de Markov cachée) ;*
- b) $\forall 2 \leq n \leq N, p(f_n|h_n, h_{n-1}) = p(f_n|h_n)$;*
- c) $\forall 1 \leq n \leq N, p(f_n|h) = p(f_n|h_n)$,*

sont équivalentes.

Preuve : La démonstration, donnée dans [66], suit le même schéma général que la démonstration de la proposition II.3.

IV.1.1 Chaîne de Markov Cachée M-stationnaire

Parmi les applications du modèle de la chaîne de Markov Triplet est la modélisation de la M-stationnarité du processus caché X . Une étude et des applications de ce modèle ont été présentées récemment dans la thèse de P. Lanchantin [64, 66]. La deuxième application est la modélisation de la M-stationnarité du couple (X, Y) , ce qui permet de généraliser le modèle de la CMC-MS.

Définition IV.1 Soit $X = (X_n)_{n=1:N}$ une chaîne de Markov, les X_n prenant leur valeurs dans un espace fini $\mathcal{X} = \{\omega_1, \dots, \omega_K\}$. X est dite une chaîne de Markov stationnaire (CM-S) si $p(x_n = \omega_i, x_{n+1} = \omega_j)$ ne dépend pas de n . Dans le cas contraire X est dite une chaîne de Markov non-stationnaire (CM-NS). On dira que X est une chaîne de Markov M -stationnaire (CM-MS) s'il existe un processus $U = (U_n)_{n=1:N}$ tel que les U_n prennent leur valeurs dans un espace fini, $\Lambda = (\lambda_1, \dots, \lambda_M)$, et tel que le couple (X, U) est une chaîne de Markov stationnaire.

Définition IV.2 Soit $Z = (X, Y)$ un processus couple, dont X est caché et Y observé. Z est dit chaîne de Markov cachée M -stationnaire à bruit indépendant (CMC-MS-BI) si :

- X est une CM-MS.
- Y et U sont indépendant conditionnellement à X :

$$Y \perp\!\!\!\perp U | X \tag{IV.5}$$

- Les Y_n sont indépendants conditionnellement à X , et pour toute n , $1 \leq n \leq N$, Y_n ne dépend que de X_n .

$$p(y|x) = \prod_{n=1}^N p(y_n|x) = \prod_{n=1}^N p(y_n|x_n) \tag{IV.6}$$

Dans les modèles classiques nous supposons que le processus caché est stationnaire tandis que dans la pratique ce dernier peut ne pas l'être. Ainsi en situations réelles l'hypothèse de la stationnarité peut conduire à une mauvaise estimation des paramètres et par suite à une mauvaise restauration du processus caché. Ainsi la m -stationnarité permet de modéliser la non stationnarité du processus caché par une chaîne triplet stationnaire.

Exemple

Soit $T = (X, U, Y)$ une CMT dont la loi est donnée par $p(x_1, u_1, y_1)$ et les transitions définies par :

$$p(t_{n+1}|t_n) = p(v_{n+1}|v_n)p(y_{n+1}|x_{n+1}) \tag{IV.7}$$

La loi initiale de la chaîne (X, U) est $p(u_1)p(x_1|u_1)$ et les transitions sont de la forme :

$$p(v_{n+1}|v_n) = p(x_{n+1}, u_{n+1}|x_n, u_n) = p(u_{n+1}|u_n)p(x_{n+1}|u_{n+1}, x_n) \tag{IV.8}$$

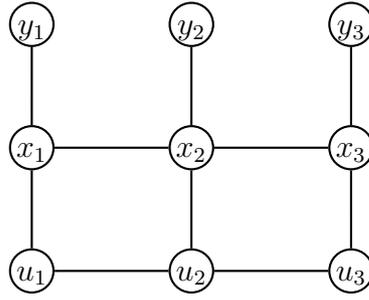


FIGURE IV.1 – Graphe d'indépendance conditionnelle d'une chaîne de Markov cachée M-stationnaire à bruit indépendant (CMC-MS-BI)

Les paramètres d'une CMC-MS-BI à bruit indépendant sont classiquement regroupés en deux ensembles : le premier est celui de la loi du $p(y_n|x_n)$ et le deuxième celui de la chaîne $V = (X, U)$. Nous avons détaillé dans le chapitre (III) comment estimer le premier ensemble de paramètres. Pour le deuxième ensemble nous détaillons le cas où V est une chaîne stationnaire réversible ($p(v_n = a, v_{n+1} = b) = p(v_n = b, v_{n+1} = a)$) et que la loi $p(v_n, v_{n+1})$ se développe comme dans l'équation (IV.8) alors les paramètres à estimer sont : $p(u_{n+1}|u_n)$ et $p(x_{n+1}|u_{n+1}, x_n)$. Ces paramètres sont obtenus à partir de la loi $p(x_{n+1}, u_{n+1}, x_n, u_n)$ que nous pouvons les estimer avec l'algorithme ICE (III.2) en se donnant un estimateur \widehat{p} à partir des données complètes. Une nouvelle fois, nous choisissons comme estimateur l'estimateur empirique donné par :

$$\begin{aligned} \widehat{p}(x_{n+1} = \omega_i, u_{n+1} = \lambda_k, x_n = \omega_j, u_n = \lambda_l) &= \frac{1}{2(N-1)} \sum_{n=1}^{N-1} \\ &[1(x_{n+1} = \omega_i, u_{n+1} = \lambda_k, x_n = \omega_j, u_n = \lambda_l) \\ &+ 1(x_{n+1} = \omega_j, u_{n+1} = \lambda_l, x_n = \omega_i, u_n = \lambda_k)] \end{aligned}$$

A chaque itération $q + 1$ de l'algorithme ICE et sachant la valeur ancienne des paramètres θ^q et les observations $y_{1:N}$, l'espérance mathématique de cet estimateur est calculable et est donnée par :

$$\begin{aligned} \widehat{p}(x_{n+1} = \omega_i, u_{n+1} = \lambda_k, x_n = \omega_j, u_n = \lambda_l, \theta^{q+1}) &= \frac{1}{2(N-1)} \sum_{n=1}^{N-1} \\ &[p(x_{n+1} = \omega_i, u_{n+1} = \lambda_k, x_n = \omega_j, u_n = \lambda_l | \theta^q, y_{1:N}) \\ &+ p(x_{n+1} = \omega_j, u_{n+1} = \lambda_l, x_n = \omega_i, u_n = \lambda_k | \theta^q, y_{1:N})] \end{aligned}$$

cela donne en particulier :

$$p(u_{n+1} = \lambda_k | u_n = \lambda_l, \theta^{q+1}) = \frac{\sum_{i,j} \widehat{p}(x_{n+1} = \omega_i, u_{n+1} = \lambda_k, x_n = \omega_j, u_n = \lambda_l, \theta^{q+1})}{\sum_{i,j,k} \widehat{p}(x_{n+1} = \omega_i, u_{n+1} = \lambda_k, x_n = \omega_j, u_n = \lambda_l, \theta^{q+1})} \quad (\text{IV.9})$$

$$p(x_{n+1} = \omega_i | u_{n+1} = \lambda_k, x_n = \omega_j, \theta^{q+1}) = \frac{\sum_l \widehat{p}(x_{n+1} = \omega_i, u_{n+1} = \lambda_k, x_n = \omega_j, u_n = \lambda_l, \theta^{q+1})}{\sum_{i,l} \widehat{p}(x_{n+1} = \omega_i, u_{n+1} = \lambda_k, x_n = \omega_j, u_n = \lambda_l, \theta^{q+1})} \quad (\text{IV.10})$$

Nous procédons dans la suite à une série d'expériences pour étudier le modèle de la chaîne de Markov cachée M-stationnaire et le comparer avec le modèle de la chaîne de Markov cachée.

Nous commençons par simuler une chaîne M-stationnaire à bruit gaussien indépendant à deux classes, deux stationnarités, et de taille $N = 256 \times 256$ ($K = 2$, $M = 2$). La loi du couple (X, U) est donnée par (IV.8). Nous commençons par simuler le processus U de loi initiale $p(u_1) = \frac{1}{2}$ et de matrice de transition $\mathcal{M} = (p_{i,j})$ avec $p_{i,j} = p(u_{n+1} = i | u_n = j)$:

$$\mathcal{M} = \begin{pmatrix} 0.999 & 0.001 \\ 0.001 & 0.999 \end{pmatrix} \quad (\text{IV.11})$$

Conditionnellement à U les matrices de transitions de X sont données par \mathcal{M}_1 et \mathcal{M}_2 qui ont respectivement comme coefficients $p(x_{n+1} | u_{n+1} = u_1, x_n)$ et $p(x_{n+1} | u_{n+1} = u_2, x_n)$:

$$\mathcal{M}_1 = \begin{pmatrix} 0.99 & 0.01 \\ 0.01 & 0.99 \end{pmatrix} \text{ et } \mathcal{M}_2 = \begin{pmatrix} 0.7 & 0.3 \\ 0.3 & 0.7 \end{pmatrix} \quad (\text{IV.12})$$

Dans un premier exemple, conditionnellement à X , les Y_n sont des gaussiennes de moyennes $m_1 = 0$ si $x_n = \omega_1$, $m_2 = 2$ si $x_n = \omega_2$ et de même variance $\sigma^2 = 1$. Pour visualiser les processus simulés, présentés à la figure (IV.2), sous forme d'images nous utilisons la transformation d'Hilbert-Peano (I.5).

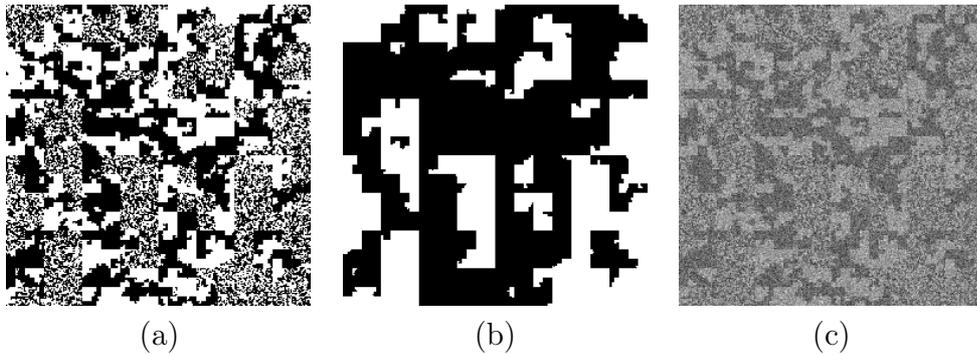


FIGURE IV.2 – Premier exemple de simulation d'une chaîne M-stationnaire à bruit indépendant. (a) : réalisation $x_{1:N}$; (b) : réalisation $u_{1:N}$ (c) : réalisation $y_{1:N}$.

Dans le second exemple, on fait augmenter le bruit, au lieu de $m_2 = 2$ pour le second classe nous choisissons $m_2 = 1$ et nous gardons les valeurs des autres variables sans modifications. Les réalisations des processus simulés sont représentées sur la figure (IV.3).

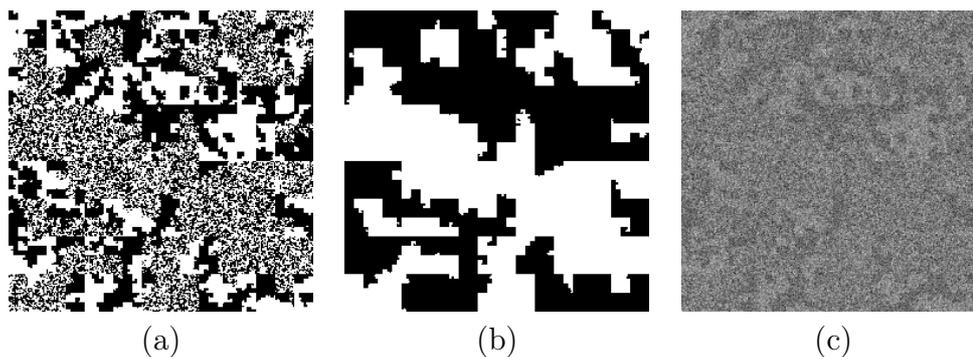


FIGURE IV.3 – Second exemple de simulation d’une chaîne M-stationnaire à bruit indépendant. (a) : réalisation $x_{1:N}$; (b) : réalisation $u_{1:N}$ (c) : réalisation $y_{1:N}$.

Pour le premier exemple, l’estimation supervisée du processus X à partir du processus observé $y_{1:N}$ nous donne un taux d’erreur de 5.84% avec le modèle de CMC-MS-BI et un taux de 7.54% avec le modèle CMC-BI classique (pour utiliser ce dernier, les paramètres de la chaîne X sont estimés à partir de la réalisation $x_{1:N}$).

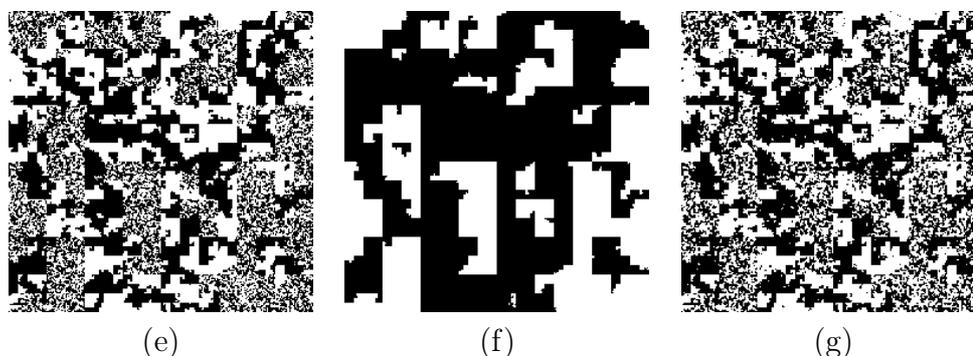


FIGURE IV.4 – Segmentation supervisée à partir de $y_{1:N}$ (IV.2). (e) : $\hat{x}_{CMC-MS-BI}$ restauré avec CMC-MS-BI (5.84%) ; (f) : \hat{u} restauré avec CMC-MS-BI (0.58%) ; (g) : \hat{x}_{CMC-BI} restauré avec CMC-BI (7.54%)

Dans le second exemple où le bruit est plus important, nous remarquons bien la différence entre les résultats obtenus par la CMC-MS-BI et la CMC-BI classique, Nous obtenons un taux d’erreur de 16.13% avec la première méthode et un taux plus élevé avec la deuxième méthode qui est de 20.30%.

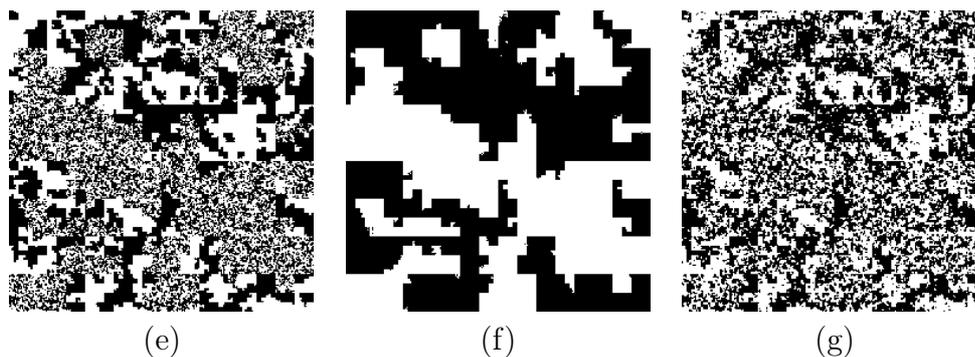


FIGURE IV.5 – Segmentation supervisée à partir de $y_{1:N}$ (IV.3). (e) : $\hat{x}_{CMC-MS-BI}$ restauré avec CMC-MS-BI (16.13%); (f) : \hat{u} restauré avec CMC-MS-BI (2.86%); (g) : \hat{x}_{CMC-BI} restauré avec CMC-BI (20.30%)

Cette expérience, ainsi que d'autres expériences du même type que nous avons effectuées, montre que lorsque les données suivent le modèle CMC-MS-BI, les résultats obtenus avec le modèle CMC-BI sont bien inférieurs à ceux obtenus avec le vrai modèle. Cela montre que les deux modèles sont assez différents, ou encore que la généralisation de CMC-BI à CMC-MS-BI est significative. Dans la deuxième expérience nous testons si cette plus grande généralité est utile lorsque les données ne suivent aucun de deux modèles.

La deuxième expérience est la segmentation non supervisée d'un processus réel à deux classes représenté par (a) sur la figure (IV.6), sur laquelle nous introduisons des bruits gaussiens indépendants. Les bruits ont la même variance égal à 1 et ont des moyennes différentes : nulle pour la première classe et égal à 2 pour la seconde. La version bruitée est présentée par (b) sur la figure (IV.6). Lors de la segmentation avec le modèle CMC-MS-BI nous supposons que le processus réel à trois stationnarités ($M = 3$).

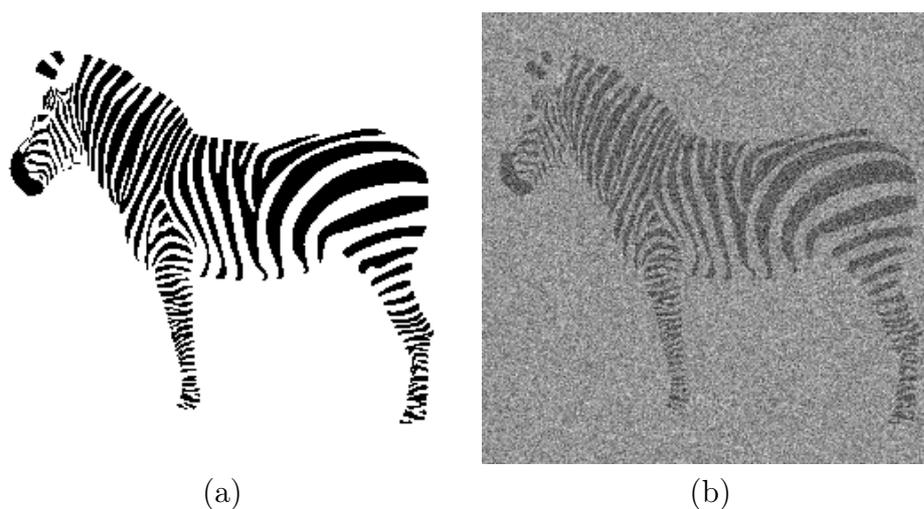


FIGURE IV.6 – (a) : processus réel; (b) : version bruitée

Les paramètres estimés sont les suivantes sont présentés dans la table (IV.1).

modèle	Moyenne		Variance		Taux d'erreur
	ω_1	ω_2	ω_1	ω_2	
CMC-BI	0.48	1.99	1.68	1.00	6.48%
CMC-MS	0.06	1.99	1.07	0.99	3.13%
Vrais Paramètres	0.00	2.00	1.00	1.00	-

TABLE IV.1 – Estimation des paramètres du bruit et taux d'erreur de la segmentation non supervisée

L'estimation de la matrice des transitions $\mathcal{P} = p(x_{n+1}|x_n)$ du modèle CMC-BI donne :

$$\mathcal{P} = \begin{pmatrix} 0.992 & 0.008 \\ 0.030 & 0.970 \end{pmatrix}$$

L'estimation de la matrice des transitions $\mathcal{M} = p(u_{n+1}|u_n)$ en supposant 3 stationnarités pour le CMC-MS donne :

$$\mathcal{M} = \begin{pmatrix} 0.999244 & 0.000065 & 0.000691 \\ 0.000145 & 0.994690 & 0.005165 \\ 0.002772 & 0.009194 & 0.988035 \end{pmatrix}$$

et l'estimation des transitions $\mathcal{M}_i = p(x_{n+1}|u_{n+1} = \lambda_i, x_n)$ donne :

$$\mathcal{M}_1 = \begin{pmatrix} 0.998 & 0.001 \\ 0.000 & 1.000 \end{pmatrix} \quad \mathcal{M}_2 = \begin{pmatrix} 0.866 & 0.133 \\ 0.194 & 0.805 \end{pmatrix} \quad \mathcal{M}_3 = \begin{pmatrix} 0.641 & 0.358 \\ 0.009 & 0.990 \end{pmatrix}$$



(c)



(d)



(e)

FIGURE IV.7 – Segmentation non supervisée du zèbre bruité présenté à la figure (IV.6) (c) : CMC-MS-BI (3.13%), (d) : U estimé ; (e) CMC-BI : (6.48%)

Nous pouvons remarquer une nette amélioration de la qualité de la segmentation lorsque l'on utilise le modèle CMC-MS-BI à la place du modèle CMC-BI. En effet, les taux d'erreur sont respectivement de 3.13% et 6.48%. Les petites taches (zone grise sur la figure IV.7.d) sont mieux détectées en supposant l'existence de 3 stationnarités : λ_1 correspond au fond de l'image qui à une faible variation de couleur, λ_2 qui correspond aux frontières de zones homogènes qui sont caractérisées par une variation forte, et λ_3 qui correspond aux zones qui ont une variation moyenne.

La dernière expérience consiste à segmenter une image réelle SAR (Synthetic Aperture Radar¹) représentée par (a) sur la figure (IV.8). Comme ci-dessus, on utilise le modèle CMC-MS-BI et le modèle CMC-BI. Nous supposons l'existence de quatre classes et de trois stationnarités. Les résultats des différentes segmentations

1. <http://www.onera.fr/images-science/observation-imagerie/image-radar-sar.php> : Technique radar qui exploite le déplacement de l'antenne pour obtenir une résolution angulaire bien supérieure à celle d'une antenne immobile. La résolution angulaire d'une antenne est inversement proportionnelle à sa taille, nécessairement réduite à bord d'un avion.

non supervisées sont présentés sur la figure (IV.8). Le résultat de la segmentation non supervisée avec CMC-MS-BI est l'image (b), l'image estimée des différentes stationnarités est donnée en (c), et l'image des classes est estimée par CMC-BI est donnée en (d).

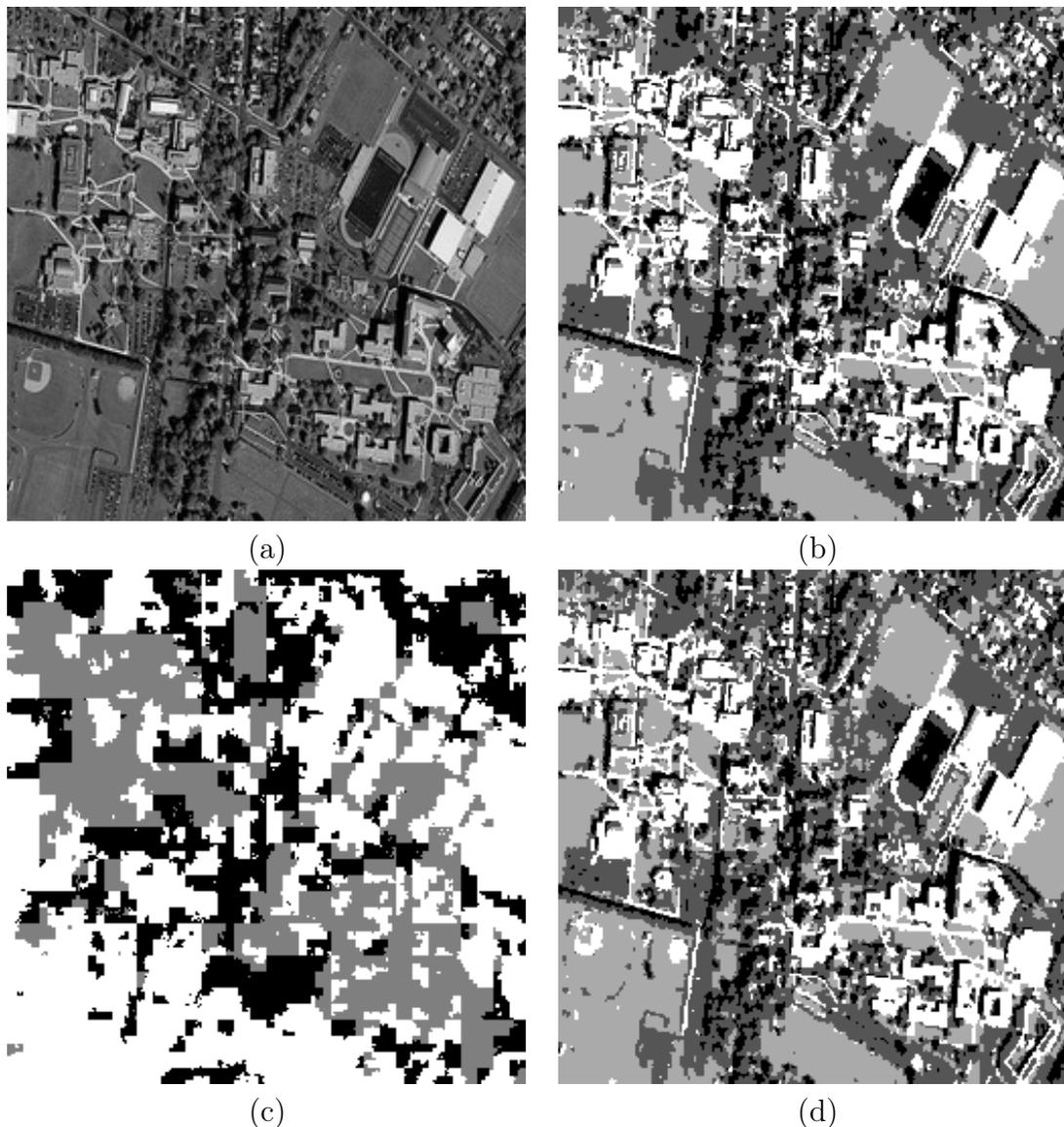


FIGURE IV.8 – Segmentation non supervisée d'une image réelle SAR. (a) : image réelle; (b) : \hat{x} estimée par CMC-MS-BI; (c) : \hat{u} stationnarités estimées; (d) : \hat{x} estimée par CMC-BI

Nous pouvons remarquer que les frontières des zones sur la figure (b) sont plus fines que celles sur la figure (d); ce qui est probablement dû au fait que les frontières de zones représentent une discontinuité en intensité et une variation forte des couleurs. Cette hypothèse est prise en compte par le modèle de la CMC-MS-BI en supposant l'existence de trois zones de stationnarité différentes : zone homogène de

faible variation de couleur (couleur blanche), zone de forte variation de couleur (couleur noir) et une zone de variation moyenne (couleur grise). Le modèle CMC-MS-BI permet donc de mieux détecter les détails fins sur une image qui comporte des objets de tailles différentes.

IV.1.2 Chaîne de Markov Couple M-stationnaire

Nous présentons dans cette sous-section un modèle original qui généralise celui présenté dans la sous-section précédente. Il s'agit du modèle de la Chaîne de Markov Couple M-stationnaire. Dans ce modèle ni X ni le couple $Z = (X, Y)$ n'est nécessairement une CM stationnaire, ce qui permet de prendre en considération une gamme de cas plus large.

Définition IV.3 Soit $Z = (X, Y)$ une chaîne couple, où $X = (X_n)_{n=1:N}$ est tel que les X_n sont à valeurs dans un espace fini et $Y = (Y_n)_{n=1:N}$ est tel que les Y_n sont aux valeurs dans un espace \mathcal{Y} . Z est dite chaîne de Markov couple M-stationnaire (CMCouple-MS) s'il existe un processus $U = (U_n)_{n=1:N}$, où les U_n prennent leurs valeurs dans un espace fini de cardinal M , $\Lambda = (\lambda_1, \dots, \lambda_M)$, tel que le triplet $T = (X, U, Y)$ est une chaîne de Markov stationnaire.

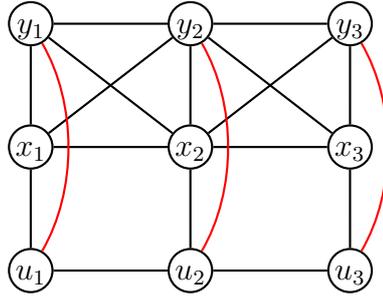


FIGURE IV.9 – Graphe d'indépendance conditionnelle d'une chaîne de Markov couple M-stationnaire

Notons que dans une CMCouple-MS la chaîne $Z = (X, Y)$ n'est pas nécessairement markovienne. Bien entendu, le modèle CMCouple-MS est strictement plus général que le modèle CMC-MS-BI et le modèle du CMCouple stationnaire. En particulier, il permet de prendre en compte l'évolution éventuelle des paramètres du bruit. Notons un cas particulier du modèle CMCouple-MS dans lequel le processus des "sauts" est une chaîne de Markov et dont la loi s'écrit :

$$p(t_{n+1}|t_n) = p(u_{n+1}|u_n)p(z_{n+1}|z_n, u_{n+1}) \quad (\text{IV.13})$$

On peut remarquer que dans ce modèle la chaîne $Z = (X, Y)$ n'est toujours pas nécessairement markovienne.

IV.1.3 Chaîne Semi-Markovienne Cachée

Une autre application pour les CMT est la modélisation de la semi-Markovianité du processus caché X . D'une part, les CMT permettent des généralisations aisées des modèles par chaînes semi-markoviennes cachées classiques. D'autre part, les CMT sont à l'origine de l'introduction d'une nouvelle famille de chaînes semi-markoviennes cachées par Lapuyade et all. [73]. En effet, une loi semi-markovienne de la chaîne X peut être vue comme la loi marginale d'une chaîne de Markov couple (X, U) , où le processus U gère le temps de séjour des X_n dans un état donné. Nous commençons par donner quelques définitions concernant la semi-markovianité. Plus de précisions sur les modèles classiques peuvent être trouvés dans [4].

Définitions

Nous définissons une chaîne semi-markovienne à espace d'états fini comme la marginale d'une chaîne de Markov.

Définition IV.4 Soit \mathcal{X} un ensemble fini de cardinal K , et soient :

- la probabilité π sur \mathcal{X} ;
- la transition $q(\cdot|\cdot)$ sur \mathcal{X}^2 vérifiant $\forall x \in \mathcal{X} \quad q(x|x) = 0$;
- pour tout $x \in \mathcal{X}$, les densités de probabilité $d(x, \cdot)$ sur \mathbb{N}^* .

Soit $X = (X_n)_{n>1}$ un processus aléatoire discret dont chaque X_n prend ses valeurs dans $\mathcal{X} = \{\omega_1, \dots, \omega_K\}$. X est une chaîne semi-markovienne homogène de loi initiale π , de transition q , et de loi de durée d s'il existe un processus $U = (U_n)_{n>1}$, chaque U_n prenant ses valeurs dans \mathbb{N}^* , tel que $V = (X, U)$ soit une chaîne de Markov de loi donnée par :

$$p(x_1, u_1) = \pi(x_1)d(x_1, u_1), \quad (\text{IV.14})$$

et les transitions :

$$p(x_{n+1}, u_{n+1}|x_n, u_n) = p(x_{n+1}|u_n, x_n)p(u_{n+1}|x_{n+1}, u_n), \quad (\text{IV.15})$$

avec :

$$p(x_{n+1}|u_n, x_n) = \begin{cases} \delta_{x_n}(x_{n+1}) & \text{si } u_n > 1, \\ q(x_{n+1}|x_n) & \text{si } u_n = 1, \end{cases} \quad (\text{IV.16})$$

et

$$p(u_{n+1}|u_n, x_{n+1}) = \begin{cases} \delta_{u_n-1}(u_{n+1}) & \text{si } u_n > 1, \\ d(x_{n+1}, u_{n+1}) & \text{si } u_n = 1, \end{cases} \quad (\text{IV.17})$$

où $\delta_a(b) = 1$ si $b = a$ et 0 sinon .

La chaîne $V = (X, U)$ est dite homogène du fait que les transitions ne dépendent pas de n . En effet, ni $q(\cdot|\cdot)$ ni $d(\cdot, \cdot)$ ne dépend de n . U_n représente alors le temps de séjour restant pour la chaîne X dans l'état ω_i .

Proposition IV.2 Soit (X, U) une chaîne vérifiant (IV.15), (IV.16) et (IV.17). Soit $D(x, n) = \sum_{k \geq n} d(x, k)$ la probabilité que le temps de séjour dans l'état x soit supérieur ou égal à n .

La loi de la chaîne à temps fini $X = (X_n)_{1:N}$ est la loi marginale de la chaîne $(X_{1:N}, W_{1:N})$ à valeurs dans $\mathcal{X} \times \{1, \dots, N\}$ dont la loi est donnée par :

– *Loi initiale* :

$$p(x_1, w_1) = \pi(x_1)p(w_1|x_1), \quad (\text{IV.18})$$

avec :

$$p(w_1|x_1) = \begin{cases} d(x_1, w_1) & \text{si } 1 \leq w_1 \leq N-1, \\ D(x_1, N) & \text{si } w_1 = N, \end{cases} \quad (\text{IV.19})$$

– *Transition* :

$$p(x_{n+1}, w_{n+1}|x_n, w_n) = p(x_{n+1}|w_n, x_n)p(w_{n+1}|x_{n+1}, w_n), \quad (\text{IV.20})$$

avec :

$$p(x_{n+1}|w_n, x_n) = \begin{cases} \delta_{x_n}(x_{n+1}) & \text{si } w_n > 1, \\ q(x_{n+1}|x_n) & \text{si } w_n = 1, \end{cases} \quad (\text{IV.21})$$

et :

$$p(w_{n+1}|x_{n+1}, w_n) = \begin{cases} \delta_{w_n-1}(w_{n+1}) & \text{si } w_n > 1, \\ d(x_{n+1}, w_{n+1}) & \text{si } w_n = 1 \text{ et } 1 \leq w_{n+1} \leq N-1, \\ D(x_{n+1}, N) & \text{si } w_n = 1 \text{ et } w_{n+1} = N, \end{cases} \quad (\text{IV.22})$$

où $\delta_a(b) = 1$ si $b = a$ et 0 sinon .

Preuve : voir [70]

Nous pouvons conclure de cette proposition que la loi de toute chaîne semi-markovienne à temps fini $X = (X_n)_{1:N}$ peut être considérée comme la loi marginale d'une chaîne de Markov à espace d'états fini $(X, W) = (X_n, U_n)_{1:N}$ dont les (X_n, U_n) sont à valeurs dans $\mathcal{X} \times \{1, \dots, N\}$. Avec cette considération, nous pouvons appliquer les algorithmes d'estimation tel que l'algorithme de Baum-Welsh pour une chaîne de Markov couple à temps fini $(X_n, W_n)_{1:N}$. Les chaînes semi-markoviennes représentent une généralisation des chaînes de Markov [77, 78]. Le temps du séjour d'une chaîne de markov dans un état est modélisé par une loi géométrique. La proposition suivante donne les conditions nécessaires et suffisantes pour qu'une CSM soit une CM [4, 28, 78].

Proposition IV.3 *Soit $X = (X_n)_{n \geq 1}$ une chaîne semi-markovienne à valeurs dans un espace d'état fini \mathcal{X} vérifiant (IV.20), (IV.21) et (IV.22). Alors X est une chaîne de Markov de matrice de transitions $Q(x', x) = p(x_{n+1} = x'|x_n = x)$ si et seulement si :*

$$d(x, u) = Q(x, x)^{u-1} \times (1 - Q(x, x)) \quad \forall x \in \mathcal{X} \text{ et } \forall u \geq 1. \quad (\text{IV.23})$$

Si cette condition est vraie, les transitions $q(x'|x)$ vérifient :

$$q(x'|x) = \frac{Q(x', x)}{1 - Q(x, x)} \quad \forall x, x' \text{ tel que } x' \neq x \quad (\text{IV.24})$$

La loi d'une chaîne semi-markovienne cachée $(X_n, Y_n)_{1:N}$ tel que $(X_n)_{1:N}$ peut ainsi être considérée comme la loi marginale d'une chaîne de Markov triplet tel que le couple $(X_n, U_n)_{1:N}$ est une chaîne de Markov à espace d'états fini et les (X_n, U_n) sont à valeurs dans $\mathcal{X} \times \{1, \dots, N\}$. Un tel triplet (X, U, Y) sera encore appelé chaîne semi-markovienne cachée (CSMC). Un cas particulier de ce modèle est la chaîne semi-markovienne cachée à bruit indépendant, qui admet pour graphe d'indépendance conditionnelle le graphe représenté sur la figure (IV.10).

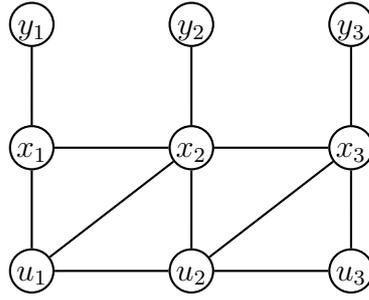


FIGURE IV.10 – Graphe d'indépendance conditionnelle non-orienté d'une chaîne semi-markovienne cachée à bruit indépendant CSMC-BI

Pour une chaîne X de taille N et à K classes, l'espace d'états du couple (X_n, U_n) à un cardinal de $N \times K$, ce qui peut poser le problème de complexité algorithmique élevée pour l'algorithme de Baum-Welsh utilisé - grâce au fait que (X, U, Y) est une chaîne de Markov triplet - dans l'estimation des paramètres et la segmentation. Pour surmonter ce problème de complexité algorithmique un nouveau modèle particulier de CSMC a été récemment présenté par J. Lapuyade et W. Pieczynski [73]. Dans ce nouveau modèle, les U_n sont à valeurs dans un espace d'états fini, dont le cardinal est indépendant de N . Ce modèle fait ainsi partie des modèles triplets discutés précédemment.

Chaîne semi-markovienne particulière

Ce nouveau modèle diffère du modèle de la chaîne semi-markovienne classique par le fait que les transitions $q(x|x)$ peuvent être non nulles; c'est-à-dire que le processus X peut rester dans la même état x malgré que le "temps de séjour" dans cet état est écoulé. Une autre différence par rapport au modèle de la CSM classique est que les U_n sont à valeurs dans un espace fini indépendant de N .

Soit $(X_n, U_n)_{1:N}$ une chaîne tel que les X_n et les U_n sont à valeurs respectivement dans $\mathcal{X} = \{\omega_1, \dots, \omega_K\}$ et $\Lambda = \{1, \dots, L\}$. Soit :

- π une distribution de probabilité sur \mathcal{X} ;
- pour tout $x \in \mathcal{X}$, soit $\bar{d}(x, \cdot)$ une distribution de probabilité sur Λ ;
- $r(\cdot, \cdot)$ transitions sur \mathcal{X}^2 .

Supposons que la loi de la chaîne $(X_n, U_n)_{1:N}$ est définie de la façon suivante. La loi initiale de la chaîne est donnée par :

$$p(x_1, u_1) = \pi(x_1)\bar{d}(x_1, u_1), \quad (\text{IV.25})$$

et les transitions sont définies par :

$$p(x_{n+1}, u_{n+1}|x_n, u_n) = p(x_{n+1}|x_n, u_n)p(u_{n+1}|x_{n+1}, u_n), \quad (\text{IV.26})$$

où :

$$p(x_{n+1}|x_n, u_n) = \begin{cases} \delta_{x_n}(x_{n+1}) & \text{si } u_n > 1, \\ r(x_{n+1}|x_n) & \text{si } u_n = 1, \end{cases} \quad (\text{IV.27})$$

et :

$$p(u_{n+1}|x_{n+1}, u_n) = \begin{cases} \delta_{u_n-1}(u_{n+1}) & \text{si } u_n > 1, \\ \bar{d}(x_{n+1}, u_{n+1}) & \text{si } u_n = 1. \end{cases} \quad (\text{IV.28})$$

Il est alors possible de montrer [70, 73] que ce modèle est une chaîne semi-markovienne classique dont les transitions et la loi du temps de séjour sont données par :

$$q(x'|x) = \frac{r(x'|x)}{1 - r(x|x)} \quad \forall x \neq x'; \quad (\text{IV.29})$$

$$d(x, u) = (1 - r(x|x)) \sum_{k=1}^u (r(x|x))^{k-1} \sum_{v_1+\dots+v_k=u} \bar{d}(x, v_1) \dots \bar{d}(x, v_k) \quad \forall u \geq 1. \quad (\text{IV.30})$$

Nous avons ainsi une chaîne semi-markovienne dont la loi de séjour n'est pas donnée explicitement. Par ailleurs, les conditions nécessaires et suffisantes pour que X soit une chaîne de Markov de matrice des transitions $Q(\omega_i, \omega_j) = p(x_{n+1} = \omega_i | x_n = \omega_j)$ sont les suivantes :

$$\begin{cases} \bar{d}(\omega_i, 1) = 1 \quad \forall i = 1, \dots, K \text{ et} \\ r(\omega_i | \omega_j) = Q(\omega_i, \omega_j). \end{cases} \quad (\text{IV.31})$$

Dans la suite, nous présentons deux expériences pour tester les différences entre les chaînes semi-markovienne cachées et les chaînes de Markov cachées classiques. La première et la deuxième expérience consistent à segmenter des données issues respectivement d'un modèle de chaîne de Markov cachée à bruit indépendant et d'un modèle de chaîne semi-markovienne à bruit indépendant. Le but de la première expérience est de vérifier que l'utilisation, en non supervisé, des chaînes semi-markoviennes ne dégrade pas les résultats obtenus avec les chaînes markoviennes. Le but de la deuxième expérience est de regarder si les chaînes markoviennes classiques peuvent "approcher" les chaînes semi-markoviennes.

Nous considérons des données de taille $N = 256 \times 256$, qui sont représentées sous forme bi-dimensionnelle en utilisant parcours d'Hilbert-Peano (I.5). Pour l'estimation des paramètres de chaque modèle nous choisissons d'utiliser l'algorithme ICE et nous itérons 100 fois pour chaque modèle. Concernant l'initialisation de l'ICE nous utilisons, comme précédemment, l'algorithme des K-moyennes (voir Annexe (A)). Ce dernier nous donne une "réalisation" x^0 du processus X que l'on utilise pour estimer la valeur initiale des paramètres.

Les données de la première expérience sont obtenues par la simulation d'une réalisation $x = (x_n)_{1:N}$ d'une chaîne $X = (X_n)_{1:N}$ markovienne dont les X_n sont à valeurs dans $\mathcal{X} = \{\omega_1, \omega_2\}$. La matrice des transitions $\mathcal{P} = p(x_{n+1}|x_n)$ de la chaîne X est :

$$\mathcal{P} = \begin{pmatrix} 0.99 & 0.01 \\ 0.01 & 0.99 \end{pmatrix}, \quad (\text{IV.32})$$

la loi de $p(y_n|x_n)$ est une gaussienne $\mathcal{N}_1(0, 3)$ si $x_n = \omega_1$ et $\mathcal{N}_2(1, 3)$ si $x_n = \omega_2$ et nous supposons que les Y_n sont indépendantes conditionnellement aux X_n . La réalisation $(x_n)_{1:N}$ de X est représentée par (a) de la figure (IV.11) et la réalisation $(y_n)_{1:N}$ de Y par (b) de la même figure.

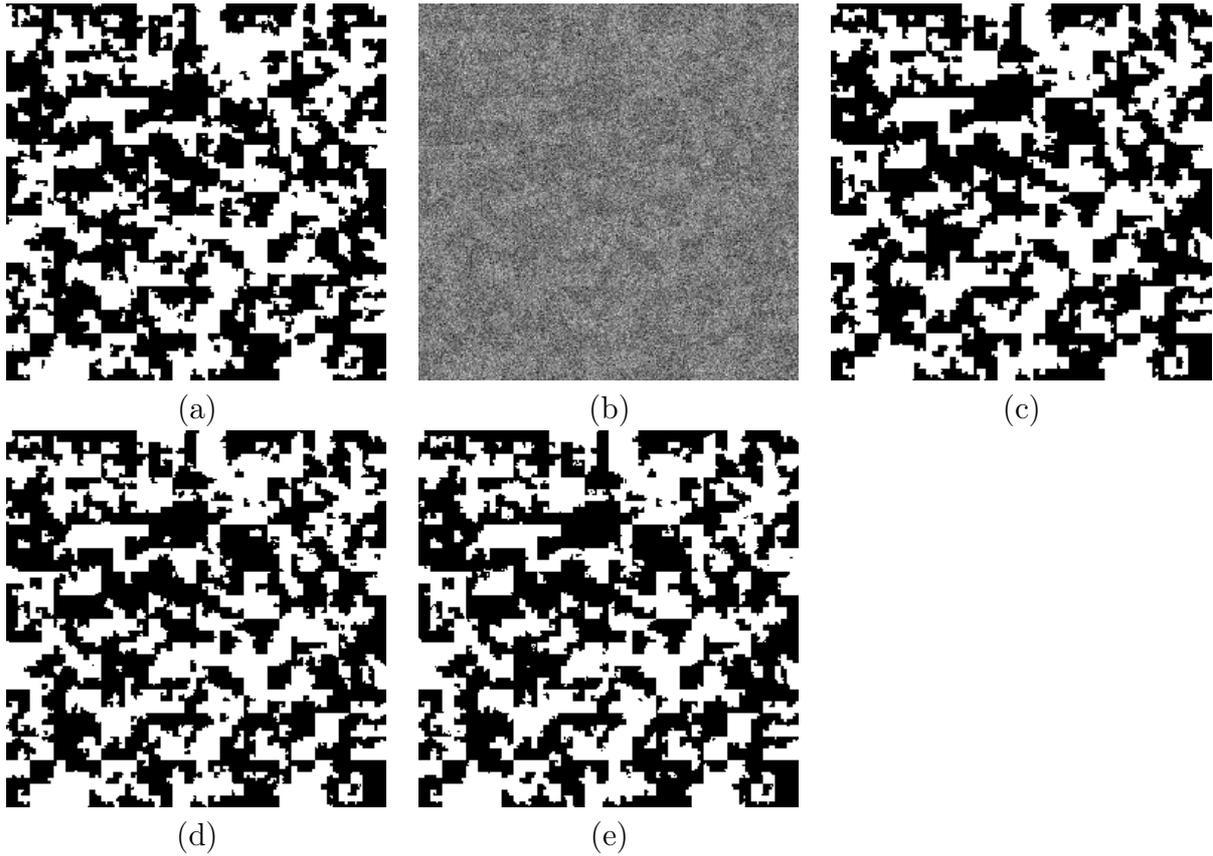


FIGURE IV.11 – Réalisation et segmentation des données issues du modèle CMC-BI. (a) : réalisation $x_{1:N}$; (b) : réalisation $y_{1:N}$; (c) : segmentation supervisée par CMC-BI, taux d'erreur de 7.25%; (d) : segmentation non supervisée par CMC-BI, taux d'erreur de 7.56%; (e) : segmentation non supervisée par CSMC-BI, taux d'erreur de 7.84%

La segmentation des observations $y_{1:N}$ (b) figure (IV.11) par le vrai modèle donne un taux d'erreur de 7.25% avec les vrais paramètres de simulation et un taux de 7.56% en non-supervisé. Les deux taux d'erreur sont très proches bien que les bruits ont des moyennes proches et des variances importantes. Ceci montre l'efficacité de l'algorithme ICE pour l'estimation des paramètres. Les paramètres estimés sont :

- Pour la loi $p(x_{n+1}|x_n)$

$$\widehat{\mathcal{P}} = \begin{pmatrix} 0.98 & 0.02 \\ 0.01 & 0.99 \end{pmatrix},$$

- Pour la loi $p(y_n|x_n)$

	ω_1		ω_2		taux d'erreur (%)
	m	σ^2	m	σ^2	
Vrais paramètres	0.00	3.00	1.00	3.00	7.25
Paramètres estimés	-0.01	2.99	1.01	2.90	7.56

FIGURE IV.12 – Paramètres de la loi $p(y_n|x_n)$ estimés avec ICE.

En supposant un temps de séjour maximal de $L = 10$, le taux d'erreur de la segmentation par CSMC est de 7.84%. Les paramètres estimés sont :

- Loi du couple (X, U) :
- la loi $\mathcal{R} = p(x_{n+1}|x_n, u_n = 1)$

$$\widehat{\mathcal{R}} = \begin{pmatrix} 0.94 & 0.06 \\ 0.05 & 0.95 \end{pmatrix}$$

- la loi $\bar{d}(x_n, u_n)$:

	1	2	3	4	5	6	7	8	9	10
$\bar{d}(\omega_1, u_n)$	0.10	0.05	0.13	0.12	0.04	0.04	0.14	0.15	0.16	0.06
$\bar{d}(\omega_2, u_n)$	0.05	0.09	0.06	0.13	0.12	0.08	0.15	0.08	0.12	0.11

TABLE IV.2 – Paramètres estimés de la loi $\bar{d}(\omega_n, u_n)$

- les lois $p(y_n|x_n)$

	ω_1		ω_2		taux d'erreur (%)
	m	σ^2	m	σ^2	
Vrais paramètres	0.00	3.00	1.00	3.00	7.25
Paramètres estimés	0.00	2.99	1.00	2.94	7.84

TABLE IV.3 – Paramètres de la loi $p(y_n|x_n)$

Nous remarquons que le taux d'erreur de la CSMC est comparable à celui de la CMC, bien que les données soient issues du modèle CMC. Dans l'expérience suivante, nous utilisons des données issues de CSMC à bruit indépendant. Nous considérons une chaîne semi-markovienne $(X_n)_{1:N}$ à deux classes ω_1 et ω_2 dont la matrice de transitions $\mathcal{R} = p(x_{n+1}|x_n, u_n = 1)$ est :

$$\mathcal{R} = \begin{pmatrix} 0.8 & 0.2 \\ 0.2 & 0.8 \end{pmatrix},$$

Comme précédemment, nous prenons $L = 10$. La loi de $p(u_n|x_n)$ est pour tout x_n : $\bar{d}(x_n, u_n) = \frac{1}{45}$ pour $u_n \neq 10$ et $\bar{d}(x_n, 10) = \frac{4}{5}$. La loi $p(y_n|x_n)$ est une gaussienne $\mathcal{N}_1(0, 2)$ si $x_n = \omega_1$ et $\mathcal{N}_2(1, 2)$ si $x_n = \omega_2$. Les réalisations $x_{1:N}$ et $y_{1:N}$ sont représentées respectivement par (a) et (b) de la figure (IV.13).

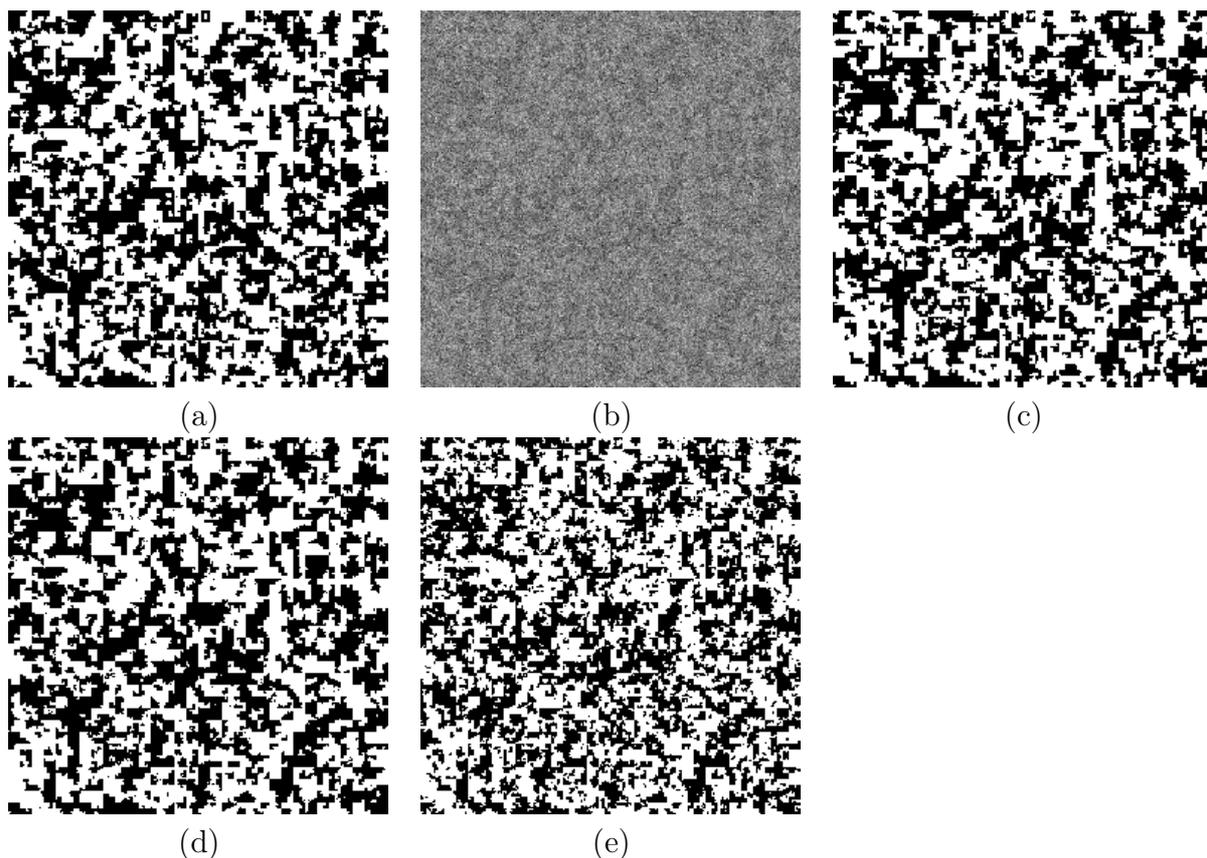


FIGURE IV.13 – Réalisation et segmentation des données issues d'un modèle CSMC-BI. (a) : réalisation $x_{1:N}$; (b) : réalisation $y_{1:N}$; (c) : segmentation supervisée par CSMC-BI, taux d'erreur de 12.01%; (d) : segmentation non-supervisée par CSMC-BI, taux d'erreur de 12.06%; (e) : segmentation non-supervisée par CMC-BI, taux d'erreur de 14.95%.

L'estimation des paramètres avec l'algorithme ICE pour le modèle CMC-BI est :

- Loi du couple (X, U) :
- la loi $\mathcal{P} = p(x_{n+1}|x_n)$:

$$\widehat{\mathcal{P}} = \begin{pmatrix} 0.92 & 0.08 \\ 0.09 & 0.91 \end{pmatrix},$$

- les lois $p(y_n|x_n)$:

	ω_1		ω_2		taux d'erreur (%)
	m	σ^2	m	σ^2	
Vrais paramètres	0.00	2.00	1.00	2.00	-
Paramètres estimés	-0.09	1.91	1.08	1.87	14.95

TABLE IV.4 – Paramètres de la loi $p(y_n|x_n)$ estimés avec ICE

Le taux d'erreur de la segmentation par CSMC-BI est de 12.01% en supervisé et de 12.06% en non-supervisé. Les valeurs des paramètres estimées par ICE sont les suivantes :

- Loi du couple (X, U) :
- la loi $\mathcal{R} = p(x_{n+1}|x_n, u_n = 1)$

$$\widehat{\mathcal{R}} = \begin{pmatrix} 0.88 & 0.12 \\ 0.12 & 0.88 \end{pmatrix}$$

- la loi $\bar{d}(x_n, u_n)$:

	1	2	3	4	5	6	7	8	9	10
vrai	$\frac{1}{45}$	$\frac{4}{5}$								
$\bar{d}(\omega_1, u_n)$	0.13	0.18	0.08	0.02	0.15	0.11	0.13	0.14	0.06	0.00
$\bar{d}(\omega_2, u_n)$	0.18	0.16	0.1	0.11	0.1	0.08	0.07	0.06	0.08	0.06

TABLE IV.5 – Paramètres de la loi $\bar{d}(\omega_n, u_n)$

- les lois $p(y_n|x_n)$:

	ω_1		ω_2		taux d'erreur (%)
	m	σ^2	m	σ^2	
Vrais paramètres	0.00	2.00	1.00	2.00	12.01
Paramètres estimés	0.00	2.02	1.00	1.96	12.06

TABLE IV.6 – Paramètres de la loi $p(y_n|x_n)$ estimés avec ICE

On note que les valeurs estimé par ICE de loi $\bar{d}(x_n, u_n)$ sont assez différentes des vraies valeurs; une des raisons possibles est le fait qu'elles ont été initialisées avec $\bar{d}(x_n, u_n) = \frac{1}{L}$. Par contre les paramètres des lois $p(y_n|x_n)$ sont bien estimés.

Remarquons que la différence entre les taux d'erreur de la segmentation supervisée et la segmentation non supervisée par le modèle CSMC-BI est très petite, ce qui montre la bonne performance de l'algorithme ICE dans le contexte considéré.

IV.2 Chaîne de Markov Triplet Mixte

Dans cette partie nous proposons un nouveau modèle étudié dans le cadre de la présente thèse. Il s'agit du modèle de chaîne de Markov Triplet Mixte (CMT Mixte), c'est-à-dire le processus auxiliaire $U = (U_n)_{1:N}$ est un processus dont les U_n sont à valeurs dans $\mathcal{U} = \mathbb{R}$, et le processus caché $X = (X_n)_{1:N}$ est à valeurs dans un espace fini. Dans la thèse de P. Lanchantin [64], le processus U est discret et une des applications d'une telle CMT, avec des résultats particulièrement intéressants, est la modélisation de la M-stationnarité de la chaîne X ou du couple (X, Y) mentionné précédemment. La question qui se pose alors est l'extension de la M-stationnarité à une « non-stationnarité continue », qui serait modélisée par un processus auxiliaire continu. Dans ce travail, nous cherchons donc à étudier les possibilités des applications de la CMT Mixte à la modélisation de la non-stationnarité "continue" du processus X . Dans toute la suite, on note $V = (X, U)$ la chaîne couple dont les V_n sont à valeurs dans l'espace produit $\mathcal{X} \times \mathbb{R}$ avec $\mathcal{X} = \{\omega_1, \dots, \omega_K\}$. La chaîne $T = (V, Y)$

est ainsi une chaîne couple où les T_n sont à valeurs dans l'espace produit $\mathcal{X} \times \mathbb{R}^2$. En supposant T markovienne nous avons :

$$p(t) = p(t_1) \prod_{n=1}^{N-1} p(t_{n+1}|t_n) \quad (\text{IV.33})$$

Par ailleurs, la loi marginale a posteriori $p(x_n|y_{1:N})$ du processus caché X est obtenue à partir de celle du processus couple (X, U) par :

$$p(x_n|y_{1:N}) = \int_{\mathbb{R}} p(v_n|y_{1:N}) du = \int_{\mathbb{R}} p(x_n, u_n|y_{1:N}) du_n \quad (\text{IV.34})$$

Étant donné que $T = (V, Y)$ est une chaîne de Markov couple ses lois marginales a posteriori $p(v_n|y_{1:N})$ peuvent être obtenues par des algorithmes analogues à ceux présentés dans le chapitre (II), avec cependant certaines sommes remplacées par des intégrales. Si on note par $\alpha_n(v)$ la probabilité (pour simplifier, on continuera à l'appeler « probabilité » sachant que c'est densité par rapport au produit de la mesure de comptage par la mesure de Lebesgue) progressive et par $\beta_n(v)$ les probabilités rétrogrades alors $p(v_n|y_{1:N})$ est donné par :

$$p(v_n|y_{1:N}) \propto \alpha_n(v_n)\beta_n(v_n) \quad (\text{IV.35})$$

avec les $\alpha_n(v) = p(v_n = v, y_{1:n})$ et $\beta_n(v) = p(y_{n+1:N}|v_n = v, y_n)$ calculables par récurrence :

$$\alpha_1(v_1) \propto p(t_1) \text{ et } \alpha_{n+1}(v_{n+1}) = \sum_{\mathcal{X}} \int_{\mathbb{R}} \alpha_n(v_n) p(t_{n+1}|t_n) du_n, \quad (\text{IV.36})$$

et

$$\beta_N(v_N) = 1 \text{ et } \beta_n(v_n) = \sum_{\mathcal{X}} \int_{\mathbb{R}} \beta_{n+1}(v_{n+1}) p(t_{n+1}|t_n) du_{n+1} \quad (\text{IV.37})$$

Le modèle CMTM est très riche sous sa forme générale et, comme dans le cas de la CMT classique, on obtient un grand nombre de modèles particuliers. Par exemple, en considérant le triplet $T = (X, U, Y)$ comme étant un couple $T = (V, Y)$ à bruit indépendant, les transitions $p(t_{n+1}|t_n)$ s'écrivent :

$$p(t_{n+1}|t_n) = p(y_{n+1}|x_{n+1})p(x_{n+1}|u_{n+1}, x_n)p(u_{n+1}|u_n, x_n) \quad (\text{IV.38})$$

Le graphe d'indépendance conditionnelle non orienté d'un tel modèle est alors le suivant :

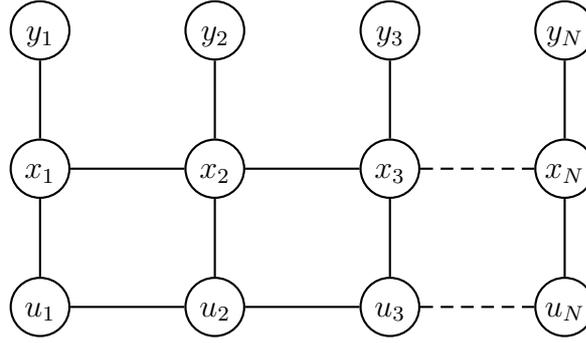


FIGURE IV.14 – Graphe d'indépendance conditionnelle d'une chaîne non-stationnaire cachée à bruit indépendant.

Nous avons la même expression que dans le cas de la chaîne M-stationnaire cachée étudiée précédemment ; cependant, le calcul explicite des α_n et des β_n n'est pas toujours possible à cause des intégrales présentées dans les formules de récurrence.

Considérons le modèle suivant, qui est un cas particulier du modèle précédent et qui est sans doute parmi les plus simples CMTM possibles. Les transitions $p(t_{n+1}|t_n)$ s'écrivent :

$$p(t_{n+1}|t_n) = p(u_{n+1}|u_n)p(x_{n+1}|u_{n+1})p(y_{n+1}|u_{n+1}) \quad (\text{IV.39})$$

Un tel modèle peut également être introduit à partir des propriétés suivantes :

a) U est un processus Markovien :

$$p(u) = p(u_1) \prod_{n=1}^{N-1} p(u_{n+1}|u_n); \quad (\text{IV.40})$$

b) X et Y sont indépendants conditionnellement à U

$$X \perp\!\!\!\perp Y|U \quad (\text{IV.41})$$

c) Les Y_n sont indépendants conditionnellement à U et $Y_n \perp\!\!\!\perp U_{p \neq n}|U_n$

$$p(y|u) = \prod_{n=1}^N p(y_n|u) = \prod_{n=1}^N p(y_n|u_n) \quad (\text{IV.42})$$

d) Les X_n sont indépendants conditionnellement à U et $X_n \perp\!\!\!\perp U_{p \neq n}|U_n$

$$p(x|u) = \prod_{n=1}^N p(x_n|u) = \prod_{n=1}^N p(x_n|u_n) \quad (\text{IV.43})$$

Comme X et Y sont indépendants conditionnellement à U nous pouvons écrire la loi $p(t)$ sous la forme suivante :

$$p(t) = p(x, u, y) = p(u)p(x|u)p(y|u) \quad (\text{IV.44})$$

Comme T est une CM sa loi est donnée par la loi initiale et les transitions. La loi initiale s'écrit :

$$p(t_1) = p(u_1)p(x_1|u_1)p(y_1|u_1), \quad (\text{IV.45})$$

et les transitions sont :

$$p(t_{n+1}|t_n) = p(u_{n+1}|u_n)p(x_{n+1}|u_{n+1})p(y_{n+1}|u_{n+1}), \quad (\text{IV.46})$$

ce qui donne bien la forme requise.

Le graphe d'indépendance conditionnelle non orienté d'un tel modèle est alors le suivant :

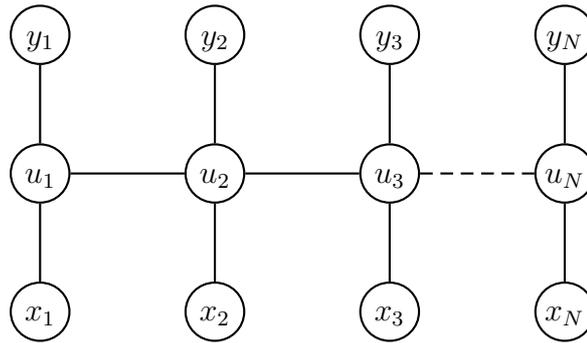


FIGURE IV.15 – Graphe d'indépendance conditionnelle non orienté d'une CMT mixte

Nous sommes en face d'un modèle original très simple ; cependant, là encore les calculs explicites des lois marginales a posteriori $p(x_n, u_n|y_{1:N})$ et $p(x_n|y_{1:N})$ ne sont pas toujours possibles. Ils le sont dans un cas particulier étudié ci-après : le cas gaussien.

Exemple : (U, Y) gaussien

Soit (U, Y) chaîne de Markov cachée à bruit indépendant gaussienne et stationnaire. De plus, on supposera le processus U centré. Classiquement, les transitions de la loi de (U, Y) s'écrivent sous forme du système suivant :

$$(\mathcal{E}) : \begin{cases} U_{n+1} &= \rho U_n + \epsilon_n \\ Y_n &= m_y + cU_n + \xi_n \end{cases} \quad (\text{IV.47})$$

Nous considérons $U_1 \sim \mathcal{N}(0, 1)$, $\rho \in \mathbb{R}$ tel que $|\rho| < 1$ et les ϵ_n Gaussiennes i.i.d de moyenne nulle et de variance $\sigma_\epsilon^2 = 1 - \rho^2$. Pour tout $n \geq 1$, U_n et ϵ_n sont indépendantes. X est un processus discret à trois classes $\mathcal{X} = \{\omega_1, \omega_2, \omega_3\}$.

La loi du couple (U, Y) étant donnée, il reste à définir la loi de X conditionnelle à U . Nous allons considérer la loi dite "Logit multinomial" de la forme suivante :

$$p(x_n = \omega_i | u_n) = \frac{\exp(a_i u_n + b_i)}{\sum_{j=1}^3 \exp(a_j u_n + b_j)} \quad (\text{IV.48})$$

La courbe représentant l'allure générale des probabilités en fonction des coefficients a, b , dans le cas de deux classes, est donnée à la figure (IV.16)

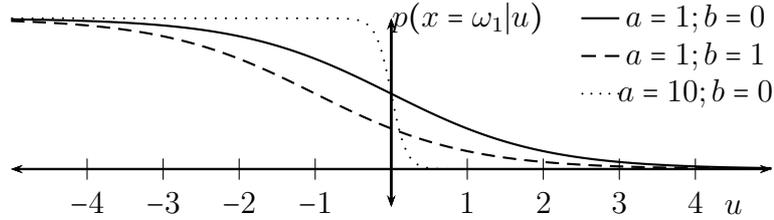


FIGURE IV.16 – Courbe de probabilité Logit à deux classes

La loi $p(x_n|y_{1:N})$ est obtenue à partir de $p(u_n|y_{1:N})$ en utilisant l'approximation de Laplace (voir annexe B). En effet la loi $p(x_n|y_{1:N})$ s'obtient par l'intégrale suivante :

$$p(x_n|y_{1:N}) = \int_{\mathbb{R}} p(x_n|u_n)p(u_n|y_{1:N}) du_n$$

Par ailleurs, nous pouvons écrire $p(x_n|u_n)p(u_n|y_{1:N}) = \exp[\log(p(x_n|u_n)) + \log(p(u_n|y_{1:N}))] = \exp[f(u)]$, avec : $p(u_n|y_{1:N}) \sim \mathcal{N}(m_n, \sigma_n^2)$ et m_n, σ_n^2 calculés par l'algorithme RTS (I.3.3), donc

$$f(u) = a_i u + b_i - \log\left(\sum_{j=1}^3 \exp(a_j u + b_j)\right) - \frac{(u - m_n)^2}{2\pi\sigma_n^2} - \log(\sqrt{2\pi\sigma_n^2})$$

La fonction f admet un développement de Taylor en u^* (tel que $f'(u^*) = 0$) à l'ordre 2 :

$$f(u) \simeq \frac{1}{2}f''(u^*)(u - u^*)^2 + f(u^*)$$

L'approximation de $p(x_n|y_{1:N})$ est alors donnée par :

$$p(x_n|y_{1:N}) \simeq \exp[f(u^*)] \sqrt{\frac{2\pi}{|f''(u^*)|}}$$

Applications numériques

Nous présentons ci-après les résultats des quatre expériences. Dans la première les données sont simulées par la nouvelle CMTM, qui sera notée NCMTM, et on considère trois segmentations : supervisée avec NCMTM, partiellement non supervisée avec NCMTM, et non supervisée avec le modèle classique CMC-BI. Dans la deuxième expérience les données sont simulées par une CMC-BI et segmentées par NCMTM et CMC-BI. Dans la troisième série on considère un bruit corrélé, de manière à ce que les données ne correspondent à aucun de deux modèles. Enfin, dans

la quatrième expérience on choisit une image de deux classes dessinée à la main et bruitée par du bruit corrélé.

Nous commençons par simuler le processus $U = (U_n)_{n=1:N}$. La première variable U_1 est une Gaussienne de moyenne 0 et de variance 1, $U_1 \sim \mathcal{N}(0,1)$, et pour $1 < n \leq N$, la loi de U_n conditionnelle à $U_{n-1} = u_{n-1}$ est une Gaussienne de moyenne ρu_{n-1} et de variance $\sigma_u^2 = 1 - \rho^2$, avec $\rho = 9.88$ (les lois marginales de tous les U_n sont identiques). Ensuite sachant U nous simulons les processus X et Y . Les réalisations x_n sont simulées selon la loi $p(x_n|u_n)$ ci-dessus, avec les paramètres a, b donnés dans la table (IV.7) Pour tout $1 \leq n \leq N$, y_n est simulée selon une loi Gaussienne de moyenne $m_y + cu_n$ et de variance V_y , avec $m_y = 0, V_y = 1$ et $c = 0.85$: $p(y_n|u_n) \sim \mathcal{N}(m_y + cu_n, V_y)$. Les réalisations des trois processus ainsi obtenues sont représentées à la figure (IV.17).

	ω_1	ω_2	ω_3
a	0	-29	47
b	0	-33	-26

TABLE IV.7 – Paramètres des lois $p(x_n|u_n)$

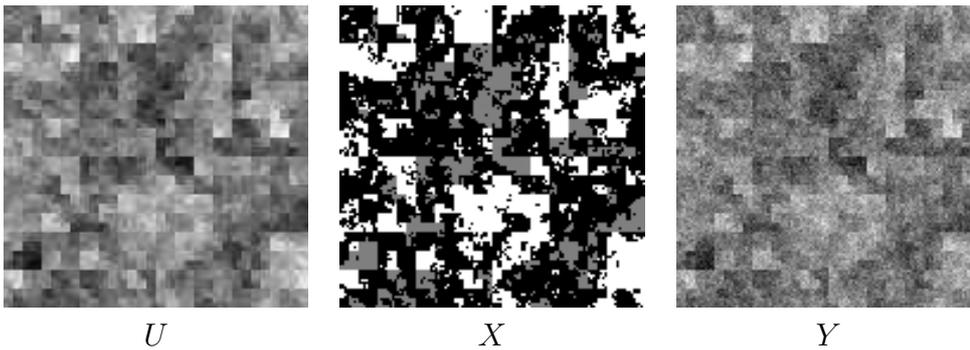


FIGURE IV.17 – Simulation d'une chaîne de Markov Triplet Mixte à trois classes

Les données $y = (y_n)_{n=1:N}$ sont alors segmentées par trois méthodes : la première est fondée sur le nouveau modèle de NCMTM avec les vrais paramètres. La deuxième méthode utilise le même modèle mais en estimant les paramètres de la loi de $p(u_n|y_{1:N})$ avec l'algorithme ICE (ceux de la loi de $p(x_n|u_n)$ sont supposés connus). La troisième méthode est fondée sur le modèle classique CMC-BI, avec les paramètres estimés par ICE. Les résultats de cette série sont présentés à la figure (IV.18) On constate que le modèle classique CMC-BI ne peut pas rivaliser avec NCMTM en mode supervisé ; cependant, il procure des résultats proches en mode non supervisé. Cela montre une très bonne robustesse, connue par ailleurs, du modèle CMC-BI dans la situation considérée.

Pour calculer la loi $p(x_n|y_{1:N})$, nous utilisons l'approximation de Laplace (voir annexe B). En premier lieu, nous estimons les paramètres de la loi $p(u_n|y_{1:N})$ (dans le cas non-supervisé) par ICE. En second lieu, sachant ces paramètres et les paramètres de la loi $p(x_n|u_n)$ qui sont supposés connus, la loi $p(x_n|y_{1:N})$ est une

intégrale qui peut être approchée et son approximation est donnée par (B.3). Enfin, la segmentation de $y_{1:N}$ est donnée par la solution MPM (5).



FIGURE IV.18 – Restauration du processus X à partir de Y par trois méthodes. (1) : NCMTM mode supervisé, (taux d'erreur de 14%); (2) : NCMTM mode non-supervisé (taux d'erreur de 17%); (3) : CMC mode non-supervisé (taux d'erreur de 18%)

Dans la deuxième expérience nous simulons une chaîne de Markov $X = (X_n)_{n=1:N}$ à deux classes $\mathcal{X} = \{\omega_1, \omega_2\}$, avec $p(\omega_1, \omega_1) = p(\omega_2, \omega_2) = 0.46$. $p(y_n | x_n = \omega_1)$ est une Gaussienne $\mathcal{N}(0, 1)$ et $p(y_n | x_n = \omega_2)$ est une Gaussienne $\mathcal{N}(1, 1)$. Nous restaurons X à partir de Y par NCMTM (taux d'erreur de 3.80%), et par CMC (taux d'erreur de 3.07%). Les résultats sont présentés à la figure (IV.19). Nous pouvons noter que les résultats obtenus avec NCMTM sont logiquement moins bons; cependant, malgré un niveau de bruit élevé, ils restent acceptables. Ces résultats sont intéressants dans la mesure où ils montrent que le modèle NCMTM peut être très flexible et peut "s'adapter" relativement bien aux données simulées par les CMC-BI classiques.

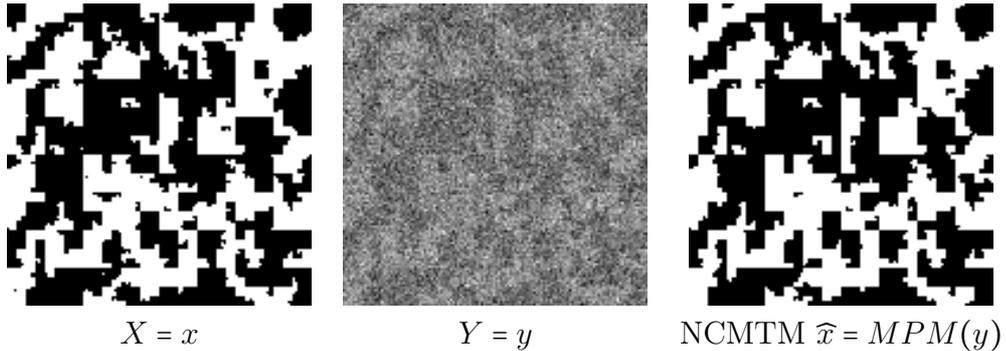


FIGURE IV.19 – X : chaîne de Markov ; Y : processus observé ; \hat{x} : processus restauré par NCMTM (taux d'erreur de 3.80%)

Dans la troisième série, la chaîne X obtenue dans la deuxième expérience est bruitée avec un bruit corrélé. On pose $v_n = \alpha u_n + \sum_{m \in \nu(n)} \beta u_m$, avec $\alpha = 0.85$, $\beta = 0.15$, $u_n \sim \mathcal{N}(0, 1)$, où $\nu(n)$ est le système de voisinage du pixel n . Les résultats sont présentés à la figure (IV.20). Ainsi que nous pouvons le constater le modèle

classique CMC-BI résiste assez mal à la corrélation du bruit. Par contre, malgré le fait que la loi de X correspond à CMC-BI, la restauration par NCMTM est nettement meilleure que la restauration par CMC-BI. En effet, le taux d'erreur est de 8.95% pour NCMTM contre 19.09% pour CMC-BI.

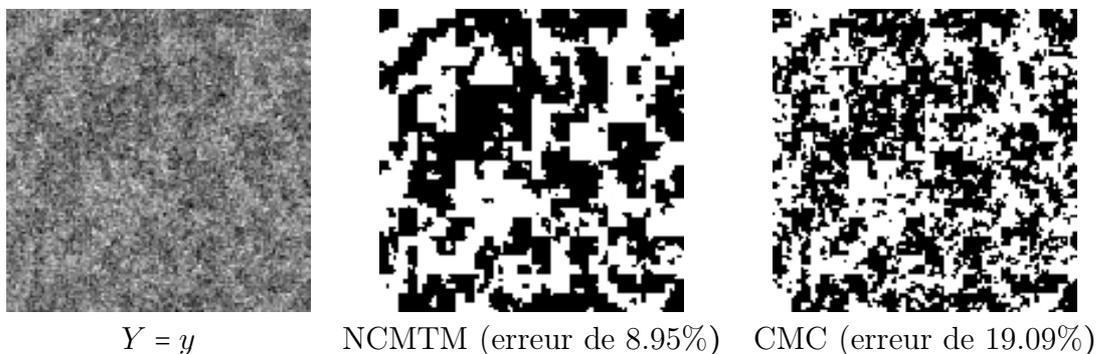


FIGURE IV.20 – Restauration de X bruité par du bruit corrélé par NCMTM et CMC-BI.

Finalement, nous restaurons une image de synthèse bruitée avec un bruit corrélé avec NCMTM (taux d'erreur de 14.49%) et par CMC (taux d'erreur de 20.12%). Les résultats sont présentés sur la figure (IV.21). Nous pouvons remarquer que la qualité de la restauration avec NCMTM est meilleure, aussi bien en terme de qualité visuelle qu'en terme des taux d'erreur, que celle utilisant le modèle CMC-BI.

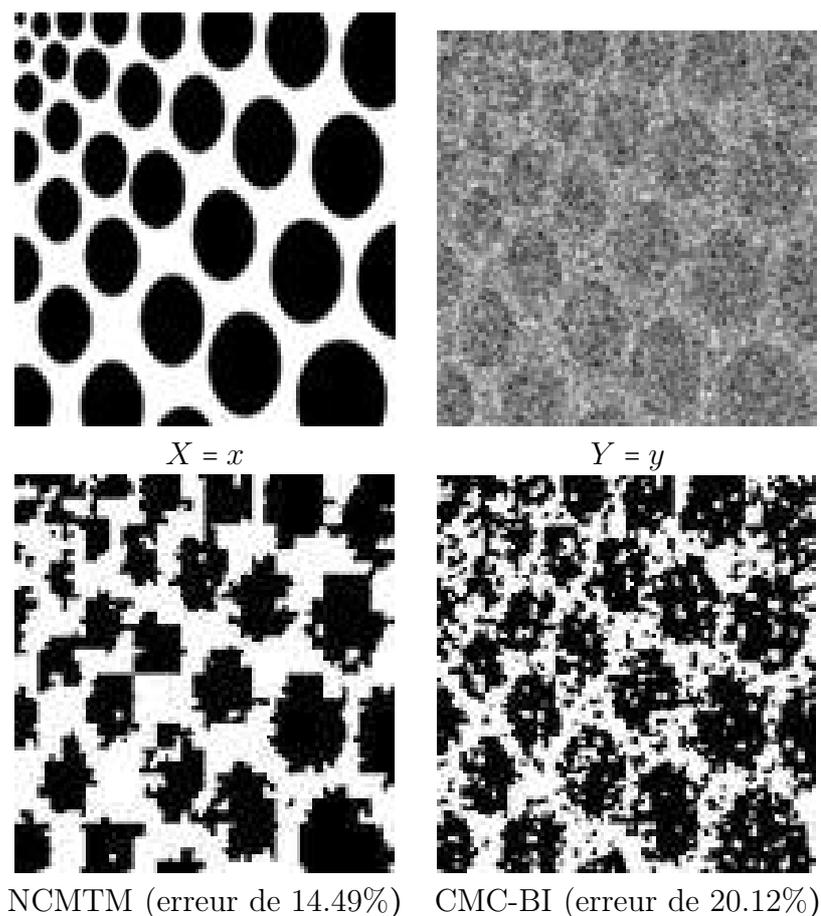


FIGURE IV.21 – Restauration d'image de synthèse bruitée avec du bruit corrélé NCMTM et CMC-BI.

Globalement, il s'avère qu'il existe des situations dans lesquelles le nouveau modèle est compétitif par rapport au modèle CMC-BI classique. Cependant, le problème de l'estimation des paramètres n'est pas entièrement résolu et nous avons proposé des traitements "partiellement" non supervisés. D'autres études plus poussées sont nécessaires pour bien cerner l'intérêt du modèle NCMTM en mode entièrement non-supervisée. Notons également que ce modèle est très simple et l'intérêt pratique de ses différentes extensions possibles constitue un autre axe de recherche possible. Enfin, au départ ce modèle a été proposé pour modéliser la non stationnarité "continue" du processus X . De ce point de vue nous n'avons pas obtenu des résultats aussi frappants que ceux obtenus par Pierre Lanchantin dans le cas des discontinuités "discrètes". D'autres études et simulations sont nécessaires pour apprécier l'intérêt du nouveau modèle NCMTM dans ce contexte.

Dans la partie suivante, nous traitons un nouveau modèle développé dans le cadre de notre thèse qui est le modèle de Champs de Markov Triplet cas mixte, c'est-à-dire (Y, U) sont deux processus continus et X est discret.

IV.3 Champ de Markov Triplet Mixte

Le modèle par champ de Markov aléatoire caché fait partie des outils de base en traitement statistique d'images depuis les années quatre-vingt [53]. L'application de ce modèle est devenue possible et intéressante avec les progrès techniques des calculateurs qui ont rendu le temps de calcul exploitable. L'hypothèse de la markovianité permet de prendre en considération l'interaction spatiale qui peut exister entre les pixels d'une image satellitaire ou d'image médicale [14, 24, 117]. Plusieurs travaux ont été menés dans le cadre de ce modèle, mettant en oeuvre des algorithmes de calcul exact ou approximatif, visant la restauration ainsi que l'estimation de paramètres [50, 106, 108]. Dans cette section nous présentons un modèle original dit « champ de Markov triplet mixte », qui est l'équivalent bi-dimensionnel du modèle « chaîne de Markov triplet mixte » étudié dans la section précédente. Plus particulièrement nous étudions le cas du triplet mixte où dans le couple caché $V = (X, U)$ le champ U est un champ continu Gaussien-Markovien et le champ X est un champs discret. Dans toute la suite, on désigne par S une partie de $\mathbb{N} \times \mathbb{N}$ de taille N . Chaque element $s \in S$ sera désigné par son rang $1 \leq n \leq N$. Soit $U = (U_n)_{n=1:N}$ un champ aléatoire sur S , dont les U_n sont à valeurs dans un espace \mathcal{U} . On note \mathcal{C} l'ensemble des cliques c (IV.22), sachant qu'une clique est soit un singleton soit un ensemble des pixels mutuellement voisins, associé au système de voisinage ν défini sur le réseau S (I.1.1).

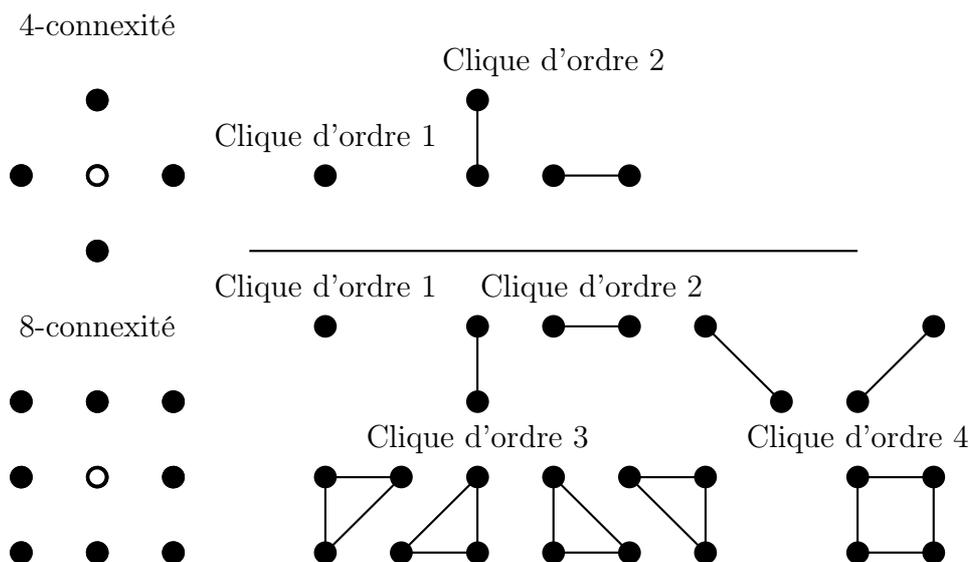


FIGURE IV.22 – Cliques associées à un système de voisinage en 4-connexité et en 8-connexité

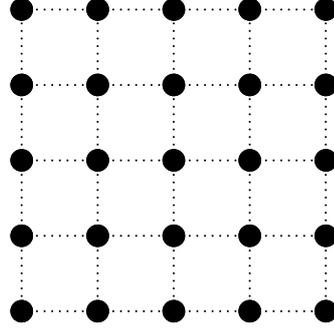


FIGURE IV.23 – Grille \mathcal{S}

Rappelons les définitions classiques ainsi que l'important théorème d'Hammersley-Clifford. Afin d'introduire la markovianité nous considérons un champ $U = (U_n)_{1:N}$ discret. Le cas continu est traité dans la sous-section suivante directement dans le cas gaussien.

Définition IV.5 Soit $U = (U_n)_{1:N}$ un champ aléatoire, chaque U_n prenant ses valeurs dans un espace \mathcal{U} .

U est dit « champ de Markov relativement à un système de voisinage ν » si et seulement si sa loi vérifie :

$$\forall 1 \leq n \leq N, \quad p(u_n | u_t, t \neq n) = p(u_n | u_{t \in \nu(n)}) \quad (\text{IV.49})$$

Ainsi la loi de U_n en un site n conditionnellement à toutes les autres variables du champ ne dépend que des variables sur son voisinage $\nu(n)$.

Définition IV.6 $U = (U_n)_{1:N}$ est dit « champ de Gibbs relativement à un système de voisinage ν » si et seulement si sa loi vérifie :

$$p(u) = \frac{1}{Z} \exp[-\mathbf{U}(u)] \quad (\text{IV.50})$$

avec \mathcal{C} l'ensemble des cliques c définie par le système de voisinage ν et $\mathbf{U}(u) = \sum_{c \in \mathcal{C}} \phi_c(u_c)$. \mathbf{U} est dite « fonction d'énergie », les fonctions ϕ_c sont dites « fonctions potentiel », et Z est la constante de normalisation $Z = \int_{u \in \mathcal{U}^N} \exp[-\mathbf{U}(u)]$. La probabilité $p(u)$ est appelée "mesure de Gibbs".

Un champ de Gibbs est ainsi caractérisé par sa loi globale dite mesure de Gibbs. Le théorème de Hammersley-Clifford suivant donne les conditions d'équivalence entre un champ de Markov et un champ de Gibbs :

Théorème IV.1 Hammersley-Clifford :

Soit \mathcal{S} un réseau muni d'un système de voisinage ν , et soit \mathcal{C} l'ensemble des cliques défini par ν .

$U = (U_n)_{1:N}$ est un champ de Markov relativement à ν vérifiant $\forall u \in \mathcal{U}^N \quad p(u) > 0$ si et seulement si U est un champ de Gibbs avec la fonction d'énergie $\mathbf{U}(u) = \sum_{c \in \mathcal{C}} \phi_c(u_c)$.

La démonstration de ce théorème est donnée dans [9]. L'intérêt de ce théorème qu'il nous permet d'accéder à la loi locale conditionnelle $p(x_n|x_t, t \in \nu(n))$ à partir du système de voisinage et les fonctions « potentiel ». La loi locale conditionnelle $p(u_n|u_t, t \in \nu(n))$ est donnée par :

$$p(u_n|u_{t \in \nu(n)}) = \frac{\exp[-\phi_c(u_n, u_{c-\{n\}})]}{\int_{u_n \in \mathcal{U}} \exp[-\phi_c(u_n, u_{c-\{n\}})]} = \Phi_c(u_n), \quad (\text{IV.51})$$

et la densité globale est un produit :

$$p(u) = \prod_{c \in \mathcal{C}} \Phi_c(u_c). \quad (\text{IV.52})$$

La simulation des réalisations d'un champ de Markov est alors possible par itération en utilisant la loi locale conditionnelle (IV.51), dont la forme est généralement simple et qui est généralement calculable. Cette méthode de simulation, que nous allons détailler par la suite, s'appelle « échantillonneur de Gibbs ».

IV.3.1 Champ Gaussien-Markovien

Soit $U = (U_n)_{n=1:N}$ un vecteur aléatoire Gaussien défini sur S . On suppose que S est muni d'un système de voisinage ν . On note par μ le vecteur moyen de U et par $\Sigma = Q^{-1}$ sa matrice de covariance. La densité de la loi de U s'écrit :

$$p(u) = (2\pi)^{-\frac{N}{2}} |Q|^{\frac{1}{2}} \exp\left[-\frac{1}{2}(u - \mu)^T Q (u - \mu)\right] \quad (\text{IV.53})$$

Le champ U est un champ markovien si et seulement si $p(x_n|x_t, t \neq n)$ vérifie (IV.49). En terme de covariance, cela se traduit par : si t n'appartient pas au voisinage de n , la covariance entre U_n et U_t conditionnellement au reste du champ est nulle.

$$\text{Cov}(U_n, U_t | u_p, p \notin \{n, t\}) = 0 \quad (\text{IV.54})$$

Cette propriété se traduit également par la propriété suivante de la structure de la matrice de covariance inverse $Q = \Sigma^{-1} = (Q_{n,t})_{1 \leq n \leq N, 1 \leq t \leq N}$:

Propriété IV.1 *La matrice Q vérifie :*

$$U_n \perp\!\!\!\perp U_t | U_p, p \notin \{n, t\} \Leftrightarrow Q_{n,t} = 0$$

La loi du champ U s'écrit donc également sous la forme de mesure de Gibbs et $p(u)$ est une mesure de Gibbs donnée par :

$$p(u) = \frac{1}{Z} \exp\left[-\sum_{c \in \mathcal{C}} \phi_c(u_c)\right] \quad (\text{IV.55})$$

Comme U est un vecteur gaussien, nous pouvons calculer la loi locale conditionnelle. C'est une gaussienne de moyenne conditionnelle :

$$\mathbb{E}[U_n | u_t, t \neq n] = \mu_n - \frac{1}{Q_{n,n}} \sum_{t \in \nu(n)} Q_{n,t} (u_t - \mu_t), \quad (\text{IV.56})$$

et de variance conditionnelle :

$$\mathbb{V}[U_n|u_t, t \neq n] = \frac{1}{Q_{n,n}}. \quad (\text{IV.57})$$

La covariance conditionnelle est donnée par :

$$\mathbb{C}[U_n, U_t|u_p, p \notin \{n, t\}] = -\frac{Q_{n,t}}{\sqrt{Q_{n,n}Q_{t,t}}} \quad (\text{IV.58})$$

La connaissance de la loi globale et la loi conditionnelle permettent de simuler le champ U par plusieurs méthodes que nous précisons ci-après.

IV.3.2 Simulation du Champ Gaussien-Markovien

La simulation d'un champ Gaussien-Markovien est possible par plusieurs méthodes. Des logiciels informatiques tels que LAPACK (Linear Algebra Package) et ATLAS (Automatically Tuned Linear Algebra Software) permettent d'aborder ce type de problèmes avec une taille énorme de données ($N > 10^8$ pour une image de taille 128×128).

Nous présentons dans la suite des méthodes indirectes de simulation et une méthode directe qui a été présentée par Rue dans [106].

Soit $U = (U_n)_{n=1:N}$ un champ Gaussien-Markovien de moyenne μ et de matrice de covariance $\Sigma = Q^{-1}$.

Échantillonneur de Gibbs

L'échantillonneur de Gibbs est un algorithme itératif de simulation. Nous commençons par une initialisation arbitraire $U = u^0$ pour le champ U . A chaque itération nous balayons la grille S pour mettre à jour les sommets n . En utilisant les équations (IV.56) et (IV.57), nous obtenons les paramètres de loi conditionnelle de U_n sachant son voisinage ($u_t, t \in \nu(n)$), et selon cette loi on fait un tirage pour mettre à jour le site n avec la nouvelle valeur obtenue.

Le déroulement de l'algorithme est le suivant :

Algorithme IV.1 Échantillonneur de Gibbs :

- i) Initialiser le champ U d'une manière arbitraire u^0 ;*
- ii) à l'iteration q , balayer le réseau S et pour chaque site s :*
 - calculer les paramètres de $p(u_n|u_t, t \in \nu(n))$;*
 - tirer aléatoirement selon $p(u_n|u_t, t \in \nu(n))$ une valeur u_n ;*
 - mettre à jour le site n avec la nouvelle valeur u_n .*

La suite aléatoire U^0, \dots, U^q ainsi obtenue est une chaîne de Markov qui converge, sous des conditions assez générales, vers la loi donnée par l'équation (IV.55) [117]. Nous pouvons distinguer deux types d'échantillonneur de Gibbs selon le parcours utilisé pour balayer S . Le premier utilise un parcours séquentiel dans un ordre prédéfini comme par exemple $1, 2, \dots, N$; le deuxième type utilise un balayage aléatoire des sites.

Méthode directe

La méthode directe décrite par Rue [106] utilise les propriétés de la matrice de covariance Q , qui est une matrice symétrique définie positive, et les propriétés des vecteurs gaussiens. En utilisant la factorisation de Cholesky, il existe une matrice L triangulaire inférieure telle que :

$$Q = LL^T. \quad (\text{IV.59})$$

La transformation d'un vecteur Gaussien centré réduit I par une transformation de type $x \mapsto Ax + B$ est un vecteur Gaussien de moyenne B et de matrice de covariance AA^T . En utilisant ces deux propriétés, une transformation pour le vecteur $U \sim \mathcal{N}_N(\mu, \Sigma)$, $U \mapsto L^T(U - \mu) = Z_U$ permet d'avoir un vecteur Gaussien centré réduit $Z_U \sim \mathcal{N}_N(0_N, I_{N \times N})$. En effet :

$$\begin{aligned} \mathbb{E}[Z_U] &= \mathbb{E}[L^T(U - \mu)] = L^T \mathbb{E}[U - \mu] = 0 \\ \mathbb{V}[Z_U] &= \mathbb{V}[L^T(U - \mu)] = L^T \Sigma_U L = L^T Q^{-1} L = I_{N \times N} \end{aligned}$$

Donc pour simuler U , il suffit de simuler Z_U et de résoudre l'équation :

$$Z_U = L^T(U - \mu) \quad (\text{IV.60})$$

Supposons, pour simplifier, que $\mu = 0_N$. La résolution se fait d'une manière itérative :

$$\begin{aligned} U_N &= \frac{Z_N}{L_{N,N}} \\ U_n &= \frac{Z_n}{L_{n,n}} - \frac{1}{L_{n,n}} \sum_{j < n+1}^N L_{j,n} U_j \quad \forall n = N-1, \dots, 1 \end{aligned} \quad (\text{IV.61})$$

Nous avons la propriété suivante :

Propriété IV.2 (Cholesky) :

Soit Q est une matrice symétrique définie positive. Il existe au moins une matrice réelle triangulaire inférieure L telle que :

$$Q = LL^T$$

Si l'on impose que les éléments diagonaux de la matrice L soient tous positifs alors la factorisation correspondante est unique.

Posons

$$L = \begin{pmatrix} L_{11} & & & & \\ L_{21} & L_{22} & & & \\ \vdots & \vdots & \ddots & & \\ L_{N1} & L_{N2} & \cdots & L_{NN} & \end{pmatrix}$$

La factorisation de Cholesky se déroule de la manière suivante :

Algorithme IV.2 (Factorisation de Cholesky) :

La factorisation de Cholesky d'une matrice Q , qui permet d'obtenir la matrice $L = (L_{i,j})_{N \times N}$, se fait de la manière itérative "colonne par colonne" suivante (on note $c = 1, \dots, N$ les colonnes de L) :

□ Pour $c=1$

$$L_{1,1} = \sqrt{Q_{1,1}}$$

et

$$L_{k,1} = \frac{Q_{1,k}}{L_{1,1}}$$

$k = 2, \dots, N$

□ On suppose la colonne $c = i - 1$ connue. La colonne $c = i$ est donnée par

◇

$$L_{i,i} = \sqrt{Q_{i,i} - \sum_{k=1}^{i-1} L_{i,k}^2}$$

◇

$$L_{j,i} = \frac{Q_{i,j} - \sum_{k=1}^{i-1} L_{i,k} L_{j,k}}{L_{i,i}}$$

Notons qu'il existe des logiciels, comme ATLAS ou LAPACK, qui utilisent cette factorisation et fournissent des fonctions rapides et exactes pour simuler des vecteurs aléatoires et pour résoudre des systèmes linéaires de grande taille ($N \geq 10^6$).

Simulation parallèle

La simulation parallèle consiste à subdiviser la grille S en plusieurs sous-ensembles deux à deux disjoints et tels que les restrictions du processus $U = (U_n)_{1:N}$ à certaines paires de ces sous-ensembles soient indépendantes conditionnellement au reste du champ. Par exemple, soient trois sous ensembles $\mathcal{A} \subset S, \mathcal{B} \subset S$ et $\mathcal{C} \subset S$:

$$\mathcal{A} \cap \mathcal{B} = \emptyset, \mathcal{A} \cap \mathcal{C} = \emptyset, \mathcal{B} \cap \mathcal{C} = \emptyset \quad (\text{IV.62})$$

$$\mathcal{A} \cup \mathcal{B} \cup \mathcal{C} = S \quad (\text{IV.63})$$

(voir la figure (IV.24)). Les processus $U_{|\mathcal{A}}$ et $U_{|\mathcal{C}}$ sont indépendants conditionnellement à $U_{|\mathcal{B}}$:

$$U_{|\mathcal{A}} \perp\!\!\!\perp U_{|\mathcal{C}} | U_{|\mathcal{B}} \quad (\text{IV.64})$$

Nous commençons par simuler $U_{|\mathcal{B}}$, et comme $U_{|\mathcal{A}}$ et $U_{|\mathcal{C}}$ sont indépendants conditionnellement à $U_{|\mathcal{B}}$, nous pouvons simuler parallèlement $U_{|\mathcal{A}}|U_{|\mathcal{B}}$ et $U_{|\mathcal{C}}|U_{|\mathcal{B}}$ par la méthode directe présentée précédemment.

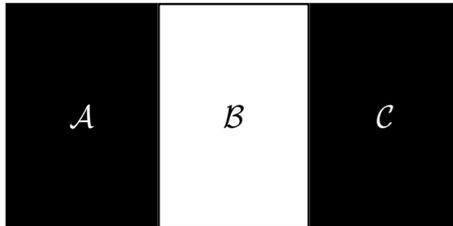


FIGURE IV.24 – $S = \mathcal{A} \cup \mathcal{B} \cup \mathcal{C}$

Lorsque la taille du processus à simuler n'est pas trop grande, la simulation directe est la plus exacte et la plus rapide parmi ces méthodes de simulation. Cependant, lorsque la taille du processus à simuler est trop grande, il peut être préférable, afin d'éviter des problèmes numériques, de recourir à une méthode indirecte comme l'échantillonneur de Gibbs ou la simulation parallèle.

IV.3.3 Champ Gaussien-Markovien caché

Le modèle du champ Gaussien-Markovien caché fait partie de modèles de Markov cachés (MMC). On suppose qu'il existe un champ Gaussien-Markovien $U = (U_n)_{1:N}$ non observé, qui est le processus d'intérêt défini sur une grille S . Les données dont nous disposons $y = (y_n)_{1:N}$ sont la réalisation d'un autre champ aléatoire $Y = (Y_n)_{1:N}$. Nous nous intéressons dans cette partie au cas du champ Gaussien-Markovien à bruit indépendant (CGMC-BI).

Présentation du modèle CGMC-BI

On appelle « champ Gaussien-Markovien caché à bruit indépendant » (CGMC-BI) le couple $(U, Y) = (U_n, Y_n)_{1:N}$ dont la loi est définie de la manière suivante. Le champ inobservable U est Gaussien-Markovien ($U \sim \mathcal{N}_N(\mu, \Sigma = Q^{-1})$). Rappelons que sa loi s'écrit alors :

$$p(u) = (2\pi)^{-\frac{N}{2}} |Q|^{\frac{1}{2}} \exp\left[-\frac{1}{2}(u - \mu)^T Q (u - \mu)\right] \quad (\text{IV.65})$$

avec la matrice de covariance Σ telle que pour tout $n \in \{1, \dots, N\}$, la variable U_n est indépendante de l'ensemble des variables $(U_t, t \notin \nu(n))$ conditionnellement aux $(U_t, t \in \nu(n))$:

$$p(u_n | u_t, t \neq n) = p(u_n | u_t, t \in \nu(n)) \quad (\text{IV.66})$$

La loi du champ observé Y conditionnelle au champ caché U est une loi quelconque (qui vérifie certaines conditions données dans la suite) : sachant $U = u$, les Y_n sont indépendantes et pour tout $n \in \{1, \dots, N\}$, la loi de Y_n conditionnelle à $U = u$ ne dépend que de U_n :

$$p(y|u) = \prod_{n=1}^N p(y_n|u) = \prod_{n=1}^N p(y_n|u_n) \quad (\text{IV.67})$$

Le champ U conditionnellement à $Y = y$ est alors également un champ Gaussien-Markovien [11], ce qui permet d'aborder les problèmes de restauration de U à partir de Y . En effet, selon la formule de Bayes :

$$p(u|y) = \frac{p(u, y)}{p(y)} \propto p(y|u)p(u), \quad (\text{IV.68})$$

nous avons donc :

$$p(u|y) \propto \exp\left[-\frac{1}{2}(u - \mu)^T Q (u - \mu) + \sum_{n=1}^N \log p(y_n|u_n)\right] \quad (\text{IV.69})$$

Si $\log[p(y_n|u_n)]$ admet un développement de Taylor à l'ordre 2 en μ_u par rapport à u_s qui est donné par :

$$\log p(y_n|u_n) \propto -\frac{1}{2}c_n(u_n - \mu)^2 + b_n(u_n - \mu) + a_n \quad (\text{IV.70})$$

soit $C = \text{diag}(c_1, \dots, c_N)$ la matrice formée par les c_s sur le diagonale et 0 ailleurs et $B = (b_1, \dots, b_N)^T$ le vecteur formé par les b_s :

$$p(u|y) \propto \exp\left[-\frac{1}{2}(u - \mu)^T(Q + C)(u - \mu) + B^T u\right] \quad (\text{IV.71})$$

La densité de la loi du $U|Y = y$ est proportionnelle à la densité d'une gaussienne de moyenne $\mu^* = \mu + [Q + C]^{-1}B$ et de matrice de covariance $Q^* = Q + C$. La matrice Q^* est une matrice symétrique définie positive ce qui permet d'utiliser la méthode directe pour simuler U sachant $Y = y$.

$$\mu^* = \mu + [Q + C]^{-1}B \quad (\text{IV.72})$$

$$Q^* = Q + C \quad (\text{IV.73})$$

Exemples

Dans cette sous-section, nous présentons quelques exemples d'applications de champ de Markov-Gauss caché. Les résultats d'estimation de paramètres et de la segmentation sont obtenus par le package **R-INLA**² développé par H. Rue et S. Martino [107].

Dans le premier exemple, nous considérons un processus auto-régressif d'ordre 1 de taille $N = 100$:

$$\forall 2 \leq n \leq N, U_n = \rho U_{n-1} + \epsilon_n, \quad (\text{IV.74})$$

avec U_1 est une gaussienne de moyenne nulle et de variance $[\tau(1 - \rho^2)]^{-1}$ et les ϵ_n sont des gaussiennes indépendantes de moyenne nulle et de variance τ^{-1} telle que $\epsilon_n \perp U_{n-1}$. $U = (U_n)_{1:N}$ est un champ Gaussien-Markovien de vecteur de moyenne $\vec{0}_N$ et de matrice de $\Sigma = Q^{-1}$:

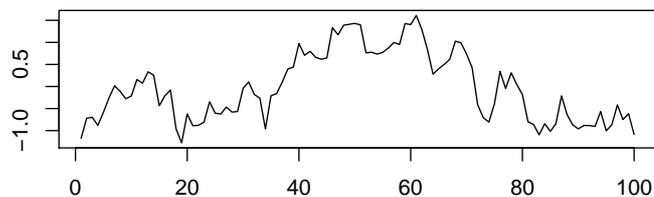
$$\Sigma = \frac{1}{\tau(1 - \rho^2)} \begin{pmatrix} 1 & \rho & \rho^2 & \dots & \rho^{N-1} \\ \rho & 1 & \rho & \dots & \rho^{N-2} \\ \rho^2 & \rho & 1 & \dots & \rho^{N-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \rho^{N-1} & \rho^{N-2} & \rho^{N-3} & \dots & 1 \end{pmatrix} \quad Q = \tau \begin{pmatrix} 1 & -\rho & 0 & \dots & 0 \\ -\rho & 1 + \rho^2 & -\rho & \ddots & \vdots \\ 0 & -\rho & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & 1 + \rho^2 & -\rho \\ 0 & \dots & 0 & -\rho & 1 \end{pmatrix} \quad (\text{IV.75})$$

La loi de $p(y_n|x_n)$ est un Poisson de paramètre $\lambda_n = E_n \exp(u_n)$ avec les E_n sont connus :

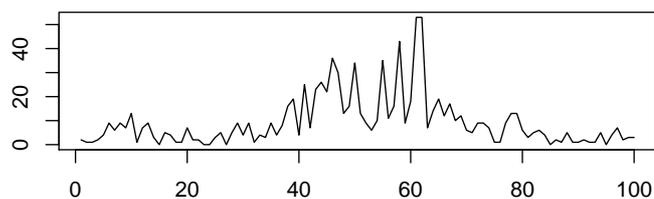
$$p(y_n = k|E_n, u_n) = \frac{\lambda_n^k}{k!} \exp(-\lambda_n) \quad (\text{IV.76})$$

Les paramètres du modèle sont $\theta = (\tau, \rho)$. Pour la simulation, $\rho = 0.9$ et $\tau = 10$ et les réalisations des champs U et Y sont représentés sur la figure (IV.25).

2. Integrated nested Laplace approximation



(a)



(b)

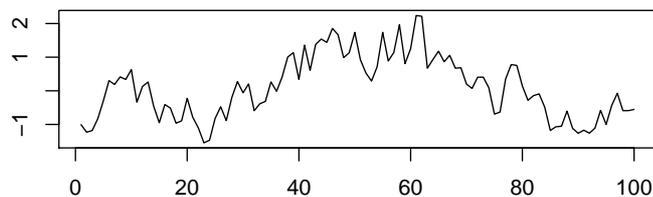
FIGURE IV.25 – Simulation d’un champ Gaussien Markovien caché (a) réalisation $u_{1:N}$ (b) réalisation $y_{1:N}$.

L’estimation des paramètres par la méthode INLA (Integrated nested Laplace approximation) [107] en utilisant le package R-INLA est la suivante :

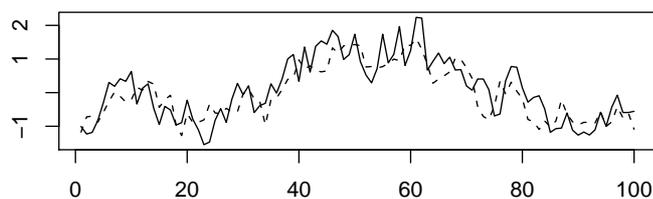
Paramètre	valeur réelle	valeur estimée
ρ	0.9	0.8
τ	10	15.6

TABLE IV.8 – Estimation des paramètres par R-INLA

L’estimation de la trajectoire $u_{1:N}$ conditionnellement à $(\hat{\theta}, y_{1:N})$ est représentée par (a) sur la figure (IV.26) ainsi qu’une comparaison entre la vraie trajectoire et la trajectoire estimée (b).



(a)



(b)

FIGURE IV.26 – Estimation de trajectoire (a) $\hat{u}_{1:N}$ estimée (b) trajectoire réelle et estimée.

Nous pouvons remarquer que la trajectoire estimée est proche de la trajectoire réelle en général mais sur une petite échelle elle s'écarte un peu de la vraie trajectoire et cela est dû au fait qu'elle est estimée par $p(u_n|\hat{\theta}, y_{1:N})$ et non pas par $p(u_n|u_t, t \in \nu(n), \hat{\theta}, y_{1:N})$.

Dans l'exemple suivant, nous considérons un champ gaussien-markovien dont l'inverse de sa matrice de covariance Q est donnée par :

$$Q = \tau \left(I_{N \times N} - \frac{\beta}{\lambda_{max}} C \right),$$

avec C est une matrice défini positive connue et qui décrit les interactions entre les différents sites n , λ_{max} est la valeur propre la plus grande de C , $\beta \in [0, 1[$ et τ un scalaire positif. Les paramètres de la loi de U sont (τ, β) :

Paramètres	Vraie valeur	Valeur estimée
τ	1.0	1.02
β	0.95	0.97

TABLE IV.9 – Paramètres du champ gaussien-markovien

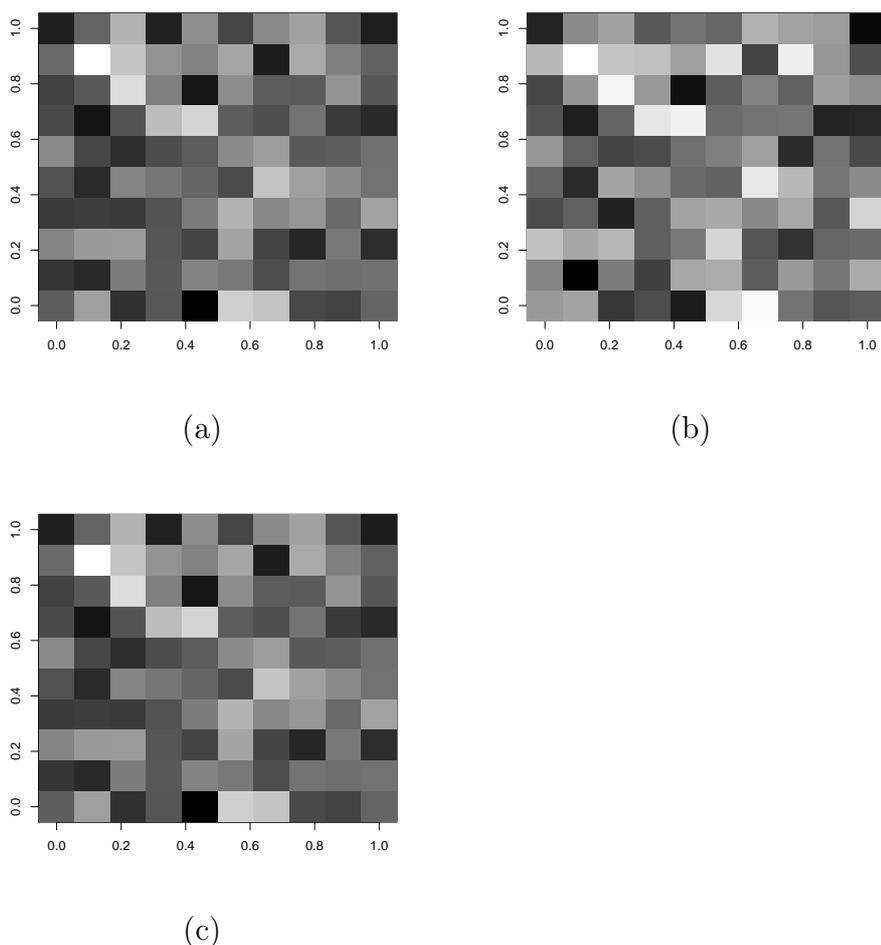


FIGURE IV.27 – Simulation et restauration d'un champ gaussien markovien

La loi $p(y_n|u_n)$ est une gaussienne de moyenne u_n et de variance $\sigma^2 = \frac{1}{\tau_2\omega}$ avec les ω sont connu égale à 1 et $\tau_2 = 100$. Les résultats de la simulation sont représentés sur la figure (IV.27) (a) champ U et (b) champ Y . L'estimation de la trajectoire $u_{1:N}$ est représenté par (c) de la même figure.

Le traitement de champs gaussiens markoviens qui ont une matrice de covariance générale nécessite des ressources informatiques très importantes pour le calcul et le stockage des résultats pour cela nous avons traité un cas ou $N = 100$ c'est-à-dire une image de taille 10×10 .

IV.3.4 Champ Gaussien-Markovien triplet

Dans cette partie, nous présentons un nouveau modèle semblable à celui que nous avons présenté dans le cas des chaînes. Soit $T = (X, U, Y)$ un champ triplet défini sur une grille S . On suppose que pour tout $n \in \{1, \dots, N\}$, la variable aléatoire $T_n = (X_n, U_n, Y_n)$ est à valeur dans l'espace produit $\mathcal{T} = \mathcal{X} \times \mathcal{U} \times \mathcal{Y}$. Pour simplifier, on utilisera parfois la notation V pour le couple des champs cachés (X, U) .

Présentation du modèle

Soit $T = (T_n)_{1:N}$ un champ Markovien de graphe d'indépendance conditionnelle représenté par le graphe sur la figure (IV.28).

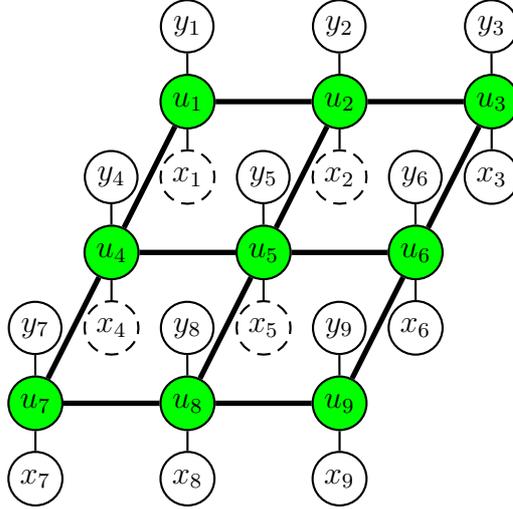


FIGURE IV.28 – Graphe d'indépendance conditionnelle du champ T

La loi de notre modèle vérifie les propriétés suivantes :

Propriété IV.3 *Les propriétés du modèle sont les suivantes :*

a) U est un champ gaussien-markovien :

$$p(u) = \prod_{n=1}^N p(u_n | u_t, t \in \nu(n)) \quad (\text{IV.77})$$

b) X et Y sont indépendants conditionnellement à U

$$X \perp\!\!\!\perp Y | U \quad (\text{IV.78})$$

c) Les Y_s sont indépendants conditionnellement à U et $Y_s \perp\!\!\!\perp U_{t \neq s} | U_s$

$$p(y|u) = \prod_{n=1}^N p(y_n|u) = \prod_{n=1}^N p(y_n|u_n) \quad (\text{IV.79})$$

d) Les X_n sont indépendants conditionnellement à U et $X_n \perp\!\!\!\perp U_{t \neq n} | U_n$

$$p(x|u) = \prod_{n=1}^N p(x_n|u) = \prod_{n=1}^N p(x_n|u_n) \quad (\text{IV.80})$$

Inférence Bayésienne

Le but de ce modèle est de trouver X sachant Y pour cela nous utilisons le processus auxiliaire U . La loi du couple $V = (X, U)$ sachant Y s'écrit comme suit :

$$p(x, u|y) = p(u|y)p(x|u, y) \quad (\text{IV.81})$$

en utilisant le fait que X et Y sont indépendants conditionnellement à U nous obtenons :

$$p(x, u|y) = p(u|y)p(x|u) \quad (\text{IV.82})$$

Nous avons détaillé précédemment comment nous pouvons approximer la loi du $U|Y$ dans le cas où U est CGM et comment nous pouvons le simuler. Supposons que $U \sim \mathcal{N}_N(\mu, \Sigma = Q^{-1})$ alors $U|Y \sim \mathcal{N}_N(\mu^*, \Sigma^* = [Q^*]^{-1})$ avec μ^* et Q^* sont obtenus par les formules (IV.72) et (IV.73). La loi $p(x_n|y)$ est obtenue en intégrant par rapport à u_n :

$$p(x_n|y) = \int p(x_n|u_n = u)p(u_n = u|y) du \quad (\text{IV.83})$$

Pour modéliser la loi du X sachant U , nous pouvons utiliser le modèle Logit. Supposons que les X_s sont à valeur dans un espace fini $\mathcal{X} = \Omega = \{\omega_1, \dots, \omega_K\}$, alors $p(x_s|u_s)$ est donné par :

$$p(x_n = \omega_i|u_n) = \frac{\exp(a_i u_n + b_i)}{\sum_{j=1}^K \exp(a_j u_n + b_j)} \quad (\text{IV.84})$$

Exemple

Le champ U est un champ gaussien Markovien de moyenne nulle et de matrice de covariance $\Sigma = Q^{-1}$ avec $Q = \tau(I_{N \times N} + \phi H)$.

$$H_{s,t} = \begin{cases} h_s & \text{si } s = t \\ -h_{s,t} & \text{si } t \in \nu(s) \\ 0 & \text{sinon} \end{cases}$$

Avec $h_{s,t} > 0$ mesure de similarité entre le site s et le site t du réseau S , $h_{s,t} = h_{t,s}$, et $h_s = \sum_{t \in \nu(s)} h_{s,t}$.

Sur la figure (IV.29) on fait varier les valeurs $\phi = 0, 1, 2, 10$ et nous prenons des valeurs fixes pour $h_{s,t} = 1$ de $\tau = 1$. Nous concéderons que le système de voisinage est celui de 4-plus proche voisin. Et pour le champ Y , pour tout n $y_n \sim \mathcal{N}(0, 1)$ et de covariance $cov(u_n, y_n) = 0.95$. Le champ X est un champ discret à deux classes obtenus par filtre Logit $a = 1$ et $b = 1$. Les 4 lignes sur la figure (IV.29) représentant les différentes valeurs de ϕ . les colonnes sont les processus : X , U , Y et Xr la restauration de X à partir de Y en mode supervisé.

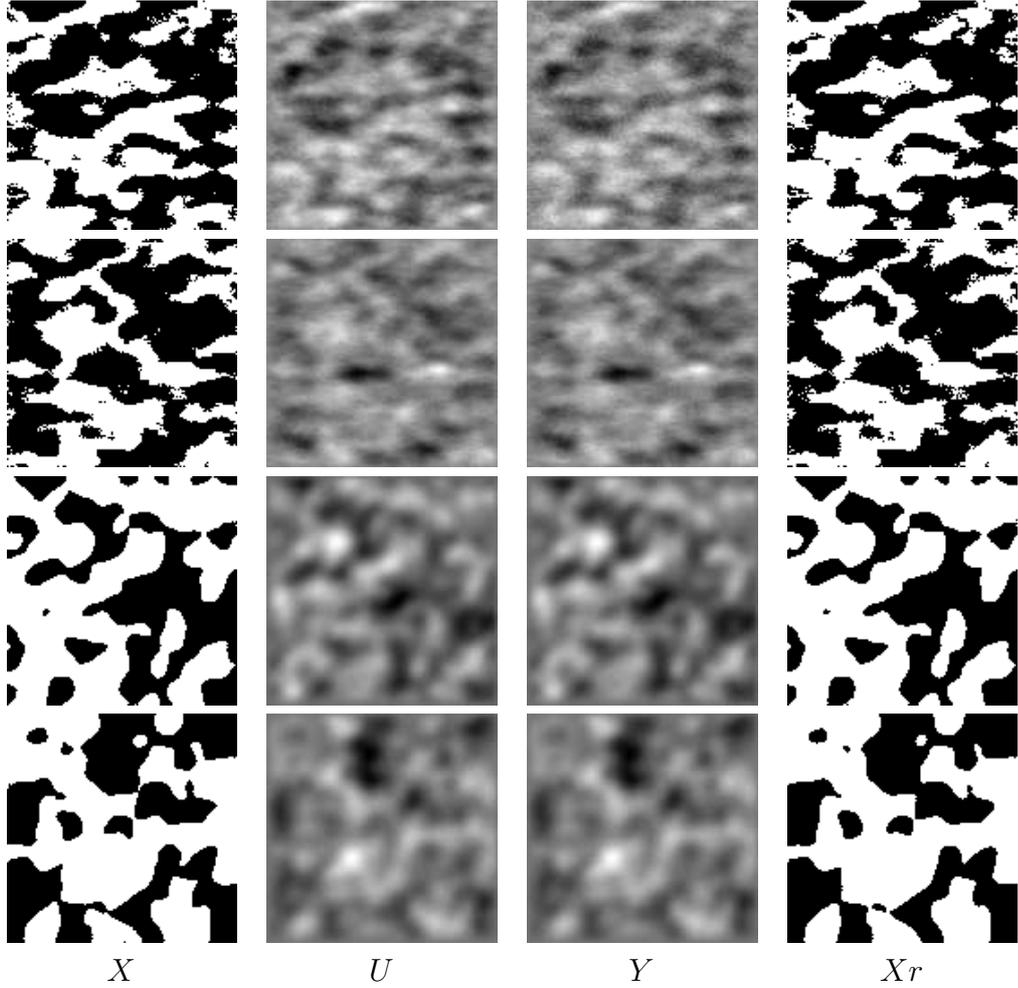


FIGURE IV.29 – Simulation du champ triplet (X, U, Y) et restauration en mode supervisé

Pour calculer la loi $p(x_n|y_{1:N})$, nous calculons en premier lieu le vecteur moyen de $U_{1:N}$ conditionnellement à $y_{1:N}$ ainsi que l'inverse de la matrice de covariance, qui sont donnés par les équations (IV.72) et (IV.73). Une fois nous avons le vecteur μ^* et la matrice $\Sigma^{-1,*} = Q^*$, la loi $p(u_n|y_{1:N})$ est une gaussienne dont la moyenne et la variance sont respectivement μ_n^* et $\Sigma_{n,n}^*$. En second lieu, nous utilisons l'approximation de Laplace (voir annexe (B)) pour approcher l'intégrale sur u de $p(x_n|u_n)p(u_n|y_{1:N})$ qui à la forme suivante :

$$I = \int_{\mathbb{R}} \exp[Mf(u)] du,$$

avec $M = 1$ et $f(u)$ est donné par :

$$f(u) = a_i u + b_i - \log\left(\sum_{j=1}^2 \exp(a_j u + b_j)\right) - \frac{(u - \mu_n^*)^2}{2\pi \Sigma_{n,n}^*} - \log(\sqrt{2\pi \Sigma_{n,n}^*})$$

La fonction f admet un développement de Taylor en u^* (qui est un optimum calculé à partir des paramètres a_i, b_i, μ_n^* et $\Sigma_{n,n}^*$) à l'ordre 2 :

$$f(u) \simeq \frac{1}{2} f''(u^*) (u - u^*)^2 + f(u^*)$$

L'approximation de $p(x_n|y_{1:N})$ est donnée par :

$$p(x_n|y_{1:N}) \simeq \exp[f(u^*)] \sqrt{\frac{2\pi}{|f''(u^*)|}}$$

Enfin, pour estimer X nous utilisons la solution MPM.

IV.4 Conclusion

Dans ce chapitre, nous avons traité des modèles de Markov triplets avec le processus caché X discret.

Dans un premier temps nous avons rappelé les modèles récents avec U également discret ; plus particulièrement, nous avons exposé les modèles M-stationnaires et les modèles semi-markoviens cachés. Nous avons proposé des simulations originales avec ces modèles et avons appliqué un modèle M-stationnaire à la segmentation d'une image radar réelle. Toutes ces expériences confirment l'intérêt pratique de ces modélisations.

Dans un deuxième temps nous avons présenté deux nouveaux modèles triplets avec U continu. Sachant que X est discret nous les appelons "modèles de Markov triplets mixtes". L'idée était d'étendre la "M-stationnarité" à la non stationnarité "continue". Nous avons proposé deux modèles originaux : chaîne de Markov triplet mixte et champ de Markov Gaussien triplet mixte. Dans le cadre des expériences menées ces modèles se sont révélés moins efficaces pour traiter la non stationnarité que les modèles avec U discret et d'autres études sont nécessaires pour cerner pleinement leur intérêt pratique dans ce domaine. Cependant, les chaînes de Markov triplets mixtes se sont montrées plus efficaces que les chaînes de Markov cachées classiques pour traiter des données avec du bruit fortement corrélé.

Dans le chapitre suivant, nous proposons des nouveaux modèles permettant de faire du filtrage et du lissage exacts dans le cas des certains modèles cachés à sauts aléatoires. Les natures de U et X seront ainsi inversées : U est discret et X est continu. Plus particulièrement, ces modèles permettront de prendre en considération des bruits à mémoire longue.

Chapitre V

Chaînes conditionnellement linéaires à sauts markoviens

Dans ce chapitre, nous proposons un nouveau modèle de chaîne triplet « partiellement » de Markov qui permet de calculer, en présence des sauts, le filtrage et le lissage optimaux avec une complexité linéaire en nombre d'observations. Dans ce modèle, nous considérons un triplet $T = (X, R, Y)$ où X et Y sont deux processus à espace d'état continu aux valeurs respectivement dans \mathcal{X} et \mathcal{Y} . Le processus R est un processus discret qui modélisera les sauts du système considéré. La nouveauté de ce travail, fait en collaboration avec Noufel Abbassi, est de considérer pour la loi de Y conditionnellement à R une loi à mémoire longue. Nous proposons un modèle général dans lequel le filtrage et le lissage exacts sont possibles avec une complexité linéaire en nombre d'observations. Nous discutons également sa position par rapport aux modélisations classiques dans lesquelles les calculs avec une telle complexité ne sont pas possibles et donc on doit faire des approximations. Parmi les méthodes d'approximation celle de type « filtrage particulière » est actuellement sans doute la plus utilisée.

Nous commençons par donner différentes définitions concernant les processus à longue mémoire ainsi qu'un exemple classique d'un tel processus. Ensuite nous présentons le modèle de la chaîne couple partiellement de Markov qui permet de prendre en considération la mémoire longue. Enfin, nous présentons le modèle à saut avec la mémoire longue et précisons les formules du filtre optimal.

V.1 Processus à mémoire longue

Dans cette sous-section, $Y = (Y_n)_{n \in \mathbb{Z}}$ est un processus aléatoire dont les composantes Y_n sont à valeurs dans \mathbb{R} .

Rappelons que l'on définit $L^2(\Omega, P)$ comme l'ensemble des v.a $Y : \Omega \rightarrow \mathbb{R}$ telle que $\|Y\|_2 = [\mathbb{E}(Y^2)]^{\frac{1}{2}}$ est finie. On dit alors que Y est de carré intégrable. $L^2(\Omega, P)$ muni du produit scalaire $\langle X, Y \rangle = \mathbb{E}[XY]$ est un espace d'Hilbert. Pour X et $Y \in L^2(\Omega, P)$ on note :

- $\sigma_X^2 = \mathbb{E}(|X - \mathbb{E}(X)|^2)$ la variance de X ;
- $\Gamma(X, Y) = \mathbb{E}([X - \mathbb{E}(X)][Y - \mathbb{E}(Y)])$ la covariance de X et Y ;
- $\rho(X, Y) = \frac{\Gamma(X, Y)}{\sigma_X \sigma_Y}$ le coefficient de corrélation de X et Y .

On dit que $Y = (Y_n)_{n \in \mathbb{Z}}$ est un processus aléatoire réel du second ordre si pour tout $n \in \mathbb{Z}$, Y_n est de carré intégrable.

Définition V.1 Soit $Y = (Y_n)_{n \in \mathbb{Z}}$ un processus réel. Y est stationnaire du second ordre si Y est un processus de second ordre dont la moyenne et la covariance sont invariantes par translation :

$$\mu(n) = \mathbb{E}(Y_n) = \mu$$

$$\Gamma(Y_p, Y_n) = \gamma(n - p)$$

La famille $(\gamma(h))_{h \in \mathbb{Z}}$ est appelée « la famille des covariances ».

Définition V.2 Une fonction C de \mathbb{R} dans \mathbb{R} est à variation lente si pour tout $x > 0$, $C(tx)/C(t)$ tend vers 1 quand t tend vers l'infini.

L'exemple classique de fonctions à variation lente sont les fonctions constantes $C(h) \mapsto C$.

Définition V.3 Soit $Y = (Y_n)_{n \in \mathbb{Z}}$ un processus réel stationnaire du second ordre. Y est dit à "dépendance longue" (ou à "mémoire longue") s'il existe un $\alpha \in]0, 1[$ et une fonction à variation bornée $C(h)$ telle que la famille des covariances $(\gamma(h))_{h \in \mathbb{Z}}$ est équivalente à $C(h).h^{-\alpha}$ quand h tend vers $+\infty$ et on note $\gamma(h) \underset{+\infty}{\sim} C(h).h^{-\alpha}$

$$\lim_{h \rightarrow +\infty} h^\alpha \gamma(h) = C(h) \tag{V.1}$$

Nous pouvons remarquer que dans un processus à mémoire longue nous avons $\sum_{h \in \mathbb{N}} |\gamma(h)| = +\infty$. Dans les cas où cette somme est finie le processus Y est dit à "mémoire courte". En particulier, les chaînes de Markov sont des processus à mémoire courte.

L'exemple le plus connu dans la littérature sur les processus à mémoire longue sont les processus « Auto-régressifs-moyennes mobiles fractionnaire » ($FARIMA(p, d, q)$) [56, 58, 81], qui généralisent les processus $ARIMA(p, d, q)$ [15]. On montre que si $Y = (Y_n)_{n \in \mathbb{Z}}$ est un processus $FARIMA(0, d, 0)$ alors sa famille de covariance est équivalente à :

$$\gamma(h) \underset{h \rightarrow +\infty}{\sim} \frac{\Gamma(1 - 2d)}{\Gamma(d)\Gamma(1 - d)} \sigma_\varepsilon^2 h^{2d-1} \tag{V.2}$$

Pour un processus $FARIMA$ on a donc : $\alpha = 1 - 2d$.

Un exemple de simulation d'un processus FARIMA est présenté à la figure (V.1).

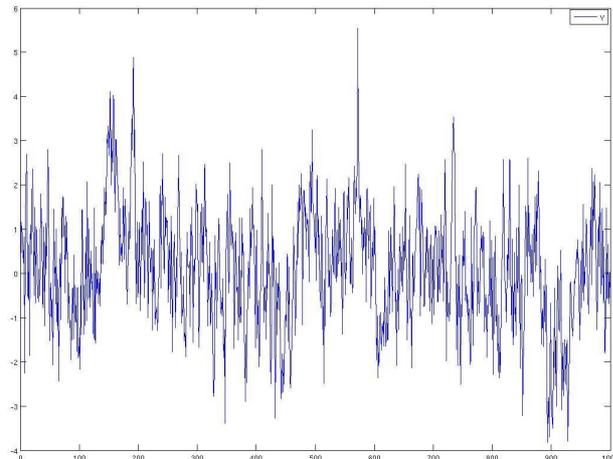


FIGURE V.1 – Simulation d'un $FARIMA(0, 0.49, 0)$, $N = 1000$

Beaucoup de phénomènes aléatoires doivent être modélisés par des processus à mémoire longue ; en particulier, ces phénomènes apparaissent en économie, informatique, ... [41]. Bien entendu, les processus à mémoire longue peuvent présenter des « non-stationnarités », ou des « sauts », modélisés par un processus aléatoire discret. Sachant qu'un processus à mémoire longue n'est pas un processus de Markov, les modèles par chaînes de Markov couple ne peuvent s'appliquer dans ce contexte. Le problème peut être modélisé par une chaîne « partiellement » de Markov récemment introduite et étudiée dans [65], que nous présentons dans la section suivante.

V.2 Chaîne couple partiellement de Markov (CCPM)

Dans cette section, nous présentons le modèle de chaînes couples partiellement de Markov qui a été proposé dans l'article de Pieczynski [94] ; voir également les thèses [64, 70]. Dans le cas particulier où le processus caché discret, modélisant les sauts, est markovien et les lois du processus observé conditionnellement au processus caché sont gaussiens, il est possible de calculer les marginales a posteriori avec une complexité linéaire en nombre d'observations. De plus, la méthode ICE a été étendue à ce cas et les traitements bayésiens peuvent être faits en non supervisés.

V.2.1 Présentation des CCPM

Soient $X = (X_n)_{1:N}$ et $Y = (Y_n)_{1:N}$ deux processus aléatoires, avec les X_n et les Y_n à valeurs respectivement dans un espace fini $\mathcal{X} = \{\omega_1, \dots, \omega_K\}$ et $\mathcal{Y} = \mathbb{R}$. On note par $Z = (Z_n)_{1:N}$ le processus couple (X, Y) , avec que pour tout $1 \leq n \leq N$, $Z_n = (X_n, Y_n)$.

Définition V.4 *On dit que $Z = (Z_n)_{1:N}$ est une « chaîne couple partiellement de Markov » (CCPM) si sa distribution $p(z_{1:N})$ vérifie :*

$$p(z_{1:N}) = p(z_1) \prod_{n=2}^N p(z_n | x_{n-1}, y_{1:n-1}). \quad (\text{V.3})$$

Notons que dans une CCPM nous avons l'indépendance conditionnelle suivante :

$$Z_{n+1:N} \perp\!\!\!\perp X_{1:n-1} | (Z_n, Y_{1:n-1}) \quad (\text{V.4})$$

Par ailleurs, les transitions $p(z_n | z_{1:n-1})$ peuvent s'écrire :

$$p(z_n | z_{1:n-1}) = p(x_n | x_{n-1}, y_{1:n-1}) p(y_n | x_n, x_{n-1}, y_{1:n-1}) \quad (\text{V.5})$$

Notons également que dans le cas général, aucun des processus X , Z n'est ainsi nécessairement de Markov. Cependant, l'important est que la loi de X conditionnelle à Y est markovienne. Cette propriété fait l'objet de la proposition suivante.

Proposition V.1 *Soit $Z = (X_n, Y_n)_{1:N}$ une CCPM. Alors la distribution de X conditionnelle à $Y = y_{1:N}$ est markovienne.*

Preuve : Nous avons :

$$\begin{aligned} p(x_n | x_{1:n-1}, y) &= \sum_{x_{n+1:N}} p(x_n, x_{n+1:N} | x_{1:n-1}, y) \\ &= \frac{\sum_{x_{n+1:N}} p(z)}{\sum_{x_{n:N}} p(z)} \\ &= \frac{\sum_{x_{n+1:N}} p(z_{1:n-1}) p(z_n, z_{n+1:N} | z_{1:n-1})}{\sum_{x_{n:N}} p(z_{1:n-1}) p(z_{n:N} | z_{1:n-1})} \\ &= \frac{\sum_{x_{n+1:N}} p(z_{1:n-1}) p(z_n | z_{1:n-1}) p(z_{n+1:N} | z_{1:n})}{\sum_{x_{n:N}} p(z_{1:n-1}) p(z_{n:N} | z_{1:n-1})} \\ &= p(z_n | z_{1:n-1}) \frac{\sum_{x_{n+1:N}} p(z_{n+1:N} | z_{1:n})}{\sum_{x_{n:N}} p(z_{n:N} | z_{1:n-1})} \end{aligned}$$

D'après (V.5) on a $p(z_n | z_{1:n-1}) = p(z_n | z_{n-1}, y_{1:n-2})$ et $p(z_{n:N} | z_{1:n-1}) = p(z_{n:N} | z_{n-1}, y_{1:n-2})$, d'où :

$$\begin{aligned} p(x_n | x_{1:n-1}, y) &= p(z_n | z_{n-1}, y_{1:n-2}) \frac{\sum_{x_{n+1:N}} p(z_{n+1:N} | z_n, y_{1:n-1})}{\sum_{x_{n:N}} p(z_{n:N} | z_{n-1}, y_{1:n-2})} \\ &= p(z_n | z_{n-1}, y_{1:n-2}) \frac{p(y_{n+1:N} | z_n, y_{1:n-1})}{p(y_{n:N} | z_{n-1}, y_{1:n-2})} \end{aligned}$$

Par ailleurs :

$$\begin{aligned} p(x_{n-1}, y_{1:N}) &= p(x_{n-1}, y_{1:n-1}) p(y_{n:N} | x_{n-1}, y_{1:n-1}) \\ &= p(x_{n-1}, y_{1:n-1}) p(y_{n:N} | z_{n-1}, y_{1:n-2}) \end{aligned}$$

et en utilisant $y_{n+1:N} \perp\!\!\!\perp x_{n-1} | z_n, y_{1:n-1}$:

$$\begin{aligned} p(x_n, x_{n-1}, y_{1:N}) &= p(x_{n-1}, y_{1:n-1}) p(x_n, y_{n:N} | x_{n-1}, y_{1:n-1}) \\ &= p(x_{n-1}, y_{1:n-1}) p(x_n, y_n | x_{n-1}, y_{1:n-1}) p(y_{n+1:N} | x_{n-1:n}, y_{1:n}) \\ &= p(x_{n-1}, y_{1:n-1}) p(z_n | z_{n-1}, y_{1:n-2}) p(y_{n+1:N} | x_{n-1:n}, y_{1:n}) \\ &= p(x_{n-1}, y_{1:n-1}) p(z_n | z_{n-1}, y_{1:n-2}) p(y_{n+1:N} | z_n, x_{n-1}, y_{1:n-1}) \\ &= p(x_{n-1}, y_{1:n-1}) p(z_n | z_{n-1}, y_{1:n-2}) p(y_{n+1:N} | z_n, y_{1:n-1}) \end{aligned}$$

par identification nous obtenons :

$$p(x_n|x_{1:n-1}, y_{1:N}) = p(x_n|x_{n-1}, y_{1:N}) \quad (\text{V.6})$$

d'où le résultat ■

Le fait que la loi a posteriori soit celle d'une chaîne de Markov permet d'utiliser les algorithmes classiques de restauration tels que le filtrage de Kalman ou l'algorithme de Baum-Welsh. Cependant, une condition supplémentaire liée à la mémoire longue apparaît : les quantités $p(z_n|z_{n-1}, y_{1:n-2})$ doivent être calculables. En effet, nous avons :

$$p(x_n|y_{1:N}) \propto \tilde{\alpha}_n(x_n)\tilde{\beta}_n(x_n) \quad (\text{V.7})$$

$$p(x_n|x_{n-1}, y_{1:N}) = p(z_n|z_{n-1}, y_{1:n-2}) \frac{\tilde{\beta}_n(x_n)}{\tilde{\beta}_{n-1}(x_{n-1})} \quad (\text{V.8})$$

avec les $\tilde{\alpha}_n(x_n) = p(x_n, y_{1:n})$ et les $\tilde{\beta}_n(x_n) = p(y_{n+1:N}|x_n, y_{1:n})$ calculables par récurrence :

$$\tilde{\alpha}_1(x_1) = p(x_1, y_1) \text{ et}$$

$$\forall n \geq 2, \tilde{\alpha}_n(x_n) = \sum_{x_{n-1}} \tilde{\alpha}_{n-1}(x_{n-1})p(z_n|z_{n-1}, y_{1:n-2}) \quad (\text{V.9})$$

$$\tilde{\beta}_N(x_N) = 1 \text{ et}$$

$$\forall n \leq N-1, \tilde{\beta}_n(x_n) = \sum_{x_{n+1}} \tilde{\beta}_{n+1}(x_{n+1})p(z_{n+1}|z_n, y_{1:n-1}) \quad (\text{V.10})$$

Ainsi toutes quantités sont calculables dès que $p(z_n|z_{n-1}, y_{1:n-2})$ apparaissant dans le calcul des $\tilde{\alpha}_n(x_n)$ et les $\tilde{\beta}_n(x_n)$ sont calculables. Dans le cas où Y est un processus gaussien conditionnellement à X , que nous précisons ci-après, ces quantités sont calculables grâce aux propriétés des vecteurs gaussiens (voir Annexe.C). Dans la suite nous nous intéressons au cas où $p(y|x)$ est la loi d'un processus gaussien à mémoire longue.

V.2.2 Chaînes de Markov cachées par du bruit gaussien(CMC-BG)

Plusieurs types de modèles ont été proposés par J. Lapuyade dans [70], qui permettent de prendre en considération les données à mémoire longue et d'appliquer les algorithmes de restauration des données manquantes. Parmi ces derniers nous pouvons citer le modèle de la chaîne de Markov cachée par du bruit gaussien (CMC-BG). Les hypothèses de ce modèle, dans lequel les calculs des quantités d'intérêt sont possibles avec une complexité linéaire en N sont les suivantes :

Définition V.5 *Soit une chaîne couple $Z = (X_n, Y_n)_{1:N}$, dont les X_n sont à valeurs dans $\mathcal{X} = \{\omega_1, \dots, \omega_K\}$ et les Y_n sont à valeurs dans $\mathcal{Y} = \mathbb{R}$. Soient q^1, \dots, q^K distributions gaussiennes sur \mathbb{R}^N , chaque q^i définissant $N-1$ transitions $q^i(y_2|y_1), \dots, q^i(y_N|y_{N-1})$. La chaîne Z est dite « chaîne de Markov cachée par du bruit gaussien » (CMC-BG) si sa loi vérifie :*

1. $p(x_{n+1}|x_n, y_{1:n}) = p(x_{n+1}|x_n)$;
2. $p(y_{n+1}|x_n, x_{n+1}, y_{1:n}) = p(y_{n+1}|x_{n+1}, y_{1:n})$;
3. $p(y_{n+1}|x_{n+1} = \omega_i, y_{1:n}) = q^i(y_{n+1}|y_{1:n})$.

Notons que dans une CMC-BC la chaîne cachée X est markovienne. La loi d'une CMC-BG est ainsi donnée par la loi de X et K lois des K processus gaussiens définies par les lois $p(y_{1:n+1}|x_{n+1} = \omega_1), \dots, p(y_{1:n+1}|x_{n+1} = \omega_K)$, ces dernières étant données par les K lois gaussiennes, correspondant aux K classes, du vecteur $Y_{1:N}$. Notons également que le point 2. n'est pas indispensable; cependant, on simplifie légèrement le modèle pour le rendre plus "parlant" (dans la version simplifiée le nombre des bruitages gaussiens différents correspond au nombre des classes).

Les quantités $p(z_n|z_{n-1}, y_{1:n-2})$ sont alors calculables dans une récurrence préalable progressive. En effet, les moyennes $\tilde{m}_{x_{n+1}}$ et les variances $\tilde{\gamma}_{x_{n+1}}$ des lois $p(y_{n+1}|x_{n+1}, y_{1:n})$ peuvent être calculées par les récurrences suivantes. On pose $y_1|x_1 \sim \mathcal{N}(m_{x_1}, \gamma_{x_1}(0))$ et pour tout $1 \leq n \leq N-1$, $y_{n+1}|x_{n+1} \sim \mathcal{N}(m_{x_{n+1}}, \gamma_{x_{n+1}}(0))$. Les matrices d'auto-corrélation correspondant aux K lois gaussiennes du vecteur $Y_{1:N}$ sont donc supposées connues :

$$\Gamma_{x_{n+1}}^{n+1} = \begin{pmatrix} \gamma_{x_{n+1}}(0) & \gamma_{x_{n+1}}(1) & \cdots & \gamma_{x_{n+1}}(n) \\ \gamma_{x_{n+1}}(1) & \gamma_{x_{n+1}}(0) & \cdots & \gamma_{x_{n+1}}(n-1) \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_{x_{n+1}}(n) & \gamma_{x_{n+1}}(n-1) & \cdots & \gamma_{x_{n+1}}(0) \end{pmatrix} = \begin{pmatrix} \Gamma_{x_{n+1}}^n & \Gamma_{x_{n+1}}^{2,1} \\ \Gamma_{x_{n+1}}^{1,2} & \gamma_{x_{n+1}}(0) \end{pmatrix}$$

En utilisant les propriétés de vecteurs gaussiens (voir Annexe C), nous obtenons :

$$\tilde{m}_{x_{n+1}} = m_{x_{n+1}} + \Gamma_{x_{n+1}}^{2,1} (\Gamma_{x_{n+1}}^n)^{-1} (y_{1:n} - m_{x_{n+1}}^n) \quad (\text{V.11})$$

$$\tilde{\gamma}_{x_{n+1}} = \gamma_{x_{n+1}} - \Gamma_{x_{n+1}}^{2,1} (\Gamma_{x_{n+1}}^n)^{-1} \Gamma_{x_{n+1}}^{1,2} \quad (\text{V.12})$$

avec $m_{x_{n+1}}^n = \underbrace{(m_{x_{n+1}}, \dots, m_{x_{n+1}})}_{n \text{ fois}}$. Nous pouvons également calculer les lois gaussiennes

$p(y_{1:n+1}|x_{1:n+1}) \sim \mathcal{N}(M_{x_{1:n+1}}, \Gamma_{x_{1:n+1}})$ par récurrence :

- Initialisation :

$$M_{x_1} = m_{x_1} \text{ et } \Gamma_{x_1} = \gamma_{x_1}(0)$$

- $\forall 2 \leq n \leq N$

$$M_{x_{1:n+1}} = \begin{pmatrix} M_{x_{1:n}} \\ m_{x_{n+1}} + \Gamma_{x_{n+1}}^{2,1} (\Gamma_{x_{n+1}}^n)^{-1} [M_{x_{1:n}} - m_{x_{n+1}}^n] \end{pmatrix} \text{ et} \quad (\text{V.13})$$

$$\Gamma_{x_{1:n+1}} = \begin{pmatrix} \Gamma_{x_{1:n}} & \Gamma_{x_{1:n}} (\Gamma_{x_{n+1}}^n)^{-1} \Gamma_{x_{n+1}}^{1,2} \\ \Gamma_{x_{n+1}}^{1,2} (\Gamma_{x_{n+1}}^n)^{-1} \Gamma_{x_{1:n}} & \gamma_{x_{n+1}}(0) + \Gamma_{x_{n+1}}^{2,1} (\Gamma_{x_{n+1}}^n)^{-1} [\Gamma_{x_{1:n}} (\Gamma_{x_{n+1}}^n)^{-1} \Gamma_{x_{n+1}}^{1,2} - \Gamma_{x_{n+1}}^{1,2}] \end{pmatrix} \quad (\text{V.14})$$

Le modèle CMC-BG est général dans la mesure où les matrices des covariance ci-dessus sont quelconques. Lorsque ces matrices définissent des distributions gaussiennes « à mémoire longue », on obtient le modèle « chaîne de Markov cachée par du bruit gaussien à mémoire longue » (CMC-GML) :

Définition V.6 Une CMC-BG est dite « chaîne de Markov cachée par du bruit gaussien à mémoire longue » (CMC-GML) si les K lois gaussiennes, correspondant aux K classes, du vecteur $Y_{1:N}$ sont à mémoire longue.

Les deux modèles sont des « chaînes de Markov cachées » car l'hypothèse 1. implique immédiatement la markovianité du processus caché X .

Ainsi dans le modèle CMC-BG nous pouvons calculer les quantités $p(z_{n+1}|z_n, y_{1:n-1})$ nécessaires pour faire de la restauration des données manquantes. Notons que ces transitions se présentent sous forme simplifiée :

$$p(z_{n+1}|z_n, y_{1:n-1}) = p(x_{n+1}|x_n)p(y_{n+1}|x_{n+1}, y_{1:n}) \quad (\text{V.15})$$

Dans certains cas où les corrélations sont paramétrisées, il est possible d'appliquer une version étendue de l'ICE pour estimer les paramètres. Cependant, cette adaptation n'est pas immédiate [65, 73]. Nous reviendrons à cette question dans la dernière section de ce chapitre, où l'ICE sera utilisée pour proposer des méthodes "semi-supervisées" de filtrage.

Notons que le modèle CMC-BG présente un certain nombre de propriétés inhabituelles; en particulier, la loi de Y_n conditionnelle à X_1, \dots, X_n dépend de tous les X_1, \dots, X_n .

V.3 Modèle conditionnellement linéaire à sauts markoviens

Dans cette section, nous proposons des nouveaux modèles aléatoires triplets, qui permettent de faire un calcul exact avec une complexité linéaire en nombre d'observations de $\mathbb{E}[X_n|y_{1:n}]$ et $\mathbb{E}[X_n|y_{1:N}]$ en présence de sauts aléatoires. La nouveauté consiste dans la prise en compte, au sein de ces modèles triplets, des modèles CMC-BG brièvement rappelés dans la sous-section précédente. En particulier, nous pourrions donc utiliser les CMC-GML.

Soient $X = (X_n)_{1:N}$ et $Y = (Y_n)_{1:N}$ deux processus aléatoires. Pour tout $1 \leq n \leq N$, X_n et Y_n sont à valeurs dans l'ensemble des réels \mathbb{R} . Soit $R = (R_n)_{1:N}$ un processus discret dont les composantes R_n prennent leurs valeurs dans un espace fini $\Omega = \{\omega_1, \dots, \omega_K\}$. Nous supposons que $Y_{1:N}$ sont observées et que $X_{1:N}$ et $R_{1:N}$ ne le sont pas. Le problème est alors d'estimer les réalisations des $X_{1:N}$ et $R_{1:N}$ à partir des réalisations de $Y_{1:N}$.

Dans la première sous-section ci-après nous rappelons le modèle classique et précisons les difficultés du calcul exact des quantités $\mathbb{E}[X_n|y_{1:n}]$ et $\mathbb{E}[X_n|y_{1:N}]$. La deuxième sous-section est consacrée aux modèles récents, par chaînes de Markov triplets, permettant ce calcul avec une complexité raisonnable. Notre modèle est présenté dans la troisième sous-section.

V.3.1 Modèle Classique

Le modèle classique consiste à considérer que le processus des sauts est markovien, et conditionnellement aux sauts le couple (X, Y) est un système linéaire. Afin

de faciliter les comparaisons, supposons que le système est gaussien. Cela s'écrit de la façon suivante :

$$R_{1:N} \text{ est une chaîne de Markov;} \quad (\text{V.16})$$

$$X_n = F_n(R_n)X_{n-1} + V_n(R_n); \quad (\text{V.17})$$

$$Y_n = G_n(R_n)X_n + W_n(R_n), \quad (\text{V.18})$$

où $V_{1:N}$ et $W_{1:N}$ sont deux vecteurs gaussiens, et $X_1, V_1, \dots, V_N, W_1, \dots, W_N$ sont indépendants et dépendent de R_n . Pour tout $1 \leq n \leq N$, F_n et G_n dépendent également de R_n .

Considérons le problème de filtrage optimal (au sens de l'erreur quadratique moyenne), qui est de calculer $\mathbb{E}[X_{n+1}|y_{1:n+1}]$ et $\mathbb{V}[X_{n+1}|y_{1:n+1}]$. Nous avons :

$$\mathbb{E}[X_{n+1}|y_{1:n+1}] = \sum_{r_{n+1}} p(r_{n+1}|y_{1:n+1})\mathbb{E}[X_{n+1}|r_{n+1}, y_{1:n+1}] \quad (\text{V.19})$$

on cherche alors une relation de récurrence entre $p(r_{n+1}|y_{1:n+1})$ et $p(r_n|y_{1:n})$. Nous avons :

$$p(r_{n+1}|y_{1:n+1}) = \sum_{r_n} p(r_{n+1}, r_n|y_{1:n+1}) = \frac{\sum_{r_n} p(r_{n+1}, y_{n+1}|r_n, y_{1:n})p(r_n|y_{1:n})}{p(y_{n+1}|y_{1:n})} \quad (\text{V.20})$$

et pour le calcul de $p(x_{n+1}|r_{n+1}, y_{1:n+1})$ nous avons les relations suivantes :

$$p(x_{n+1}|r_{n+1}, y_{1:n}) = \int_{\mathbb{R}} p(x_{n+1}|x_n, r_{n+1})p(x_n|r_{n+1}, y_{1:n}) dx_n \quad (\text{V.21})$$

$$p(x_{n+1}|r_{n+1}, y_{1:n+1}) = \frac{p(x_{n+1}|r_{n+1}, y_{1:n})p(y_{n+1}|x_{n+1}, r_{n+1})}{p(y_{n+1}|r_{n+1}, y_{1:n})} \quad (\text{V.22})$$

On résout alors le problème classiquement « à sauts fixés » ; en effet, dans le cas où $R_{1:N}$ est connu, des versions du filtre de Kalman sont applicables à ce modèle pour calculer les quantités, $\mathbb{E}[X_n|r_n, y_{1:n}]$ dans le cas du filtrage [18, 43, 44, 46] et $\mathbb{E}[X_n|r_n, y_{1:N}]$ dans le cas du lissage [18, 47]. Cependant, lorsque $R_{1:N}$ ne sont pas observés, ces quantités ne sont pas calculables directement et doivent être approchées [1, 31, 40]. Les difficultés sont dues au fait que les distributions $p(r_n|y_{1:n})$, $p(y_{n+1}|y_{1:n})$, et $p(y_{n+1}|r_{n+1}, y_{1:n})$ ne sont pas calculables avec une complexité linéaire en N . Par contre, si le couple $(R_{1:N}, Y_{1:N})$ est une chaîne de Markov classique ou une chaîne partiellement de Markov ces calculs sont possibles. Dans la suite, nous présentons deux modèles à saut Markovien dans le premier nous supposons que le couple est une chaîne de Markov et le deuxième qu'elle est partiellement de Markov.

V.3.2 Modèle conditionnellement linéaire à sauts markoviens (CCLSM)

L'idée de ce modèle est de modifier les hypothèses (V.16) et (V.18) par le fait que le couple $(R_{1:N}, Y_{1:N})$ est une chaîne de Markov. De manière générale, dans tous les modèles classiques la loi du triplet (X, R, Y) est définie par la loi (markovienne) de (X, R) et les lois de Y conditionnelles à (X, R) . Le couple (R, Y) n'est alors pas nécessairement markovien, ce qui implique les difficultés mentionnées dans

la sous-section précédente. Dans les modèles récemment proposés [49, 98, 99] la loi du triplet (X, R, Y) est définie par la loi (markovienne) de (R, Y) et les lois de X conditionnelles à (R, Y) .

Définition V.7 $T = (X, R, Y)$ une chaîne triplet est dite une chaîne « conditionnellement linéaire à sauts markoviens » (CCLSM) si :

1. (R, Y) est une chaîne de Markov ;
2. pour tout $2 \leq n \leq N$, $X_n = F_n(R_n, Y_n)X_{n-1} + G_n(R_n, Y_n)V_n + H_n(R_n, Y_n)$

où $F_n(R_n, Y_n)$, $G_n(R_n, Y_n)$ et $H_n(R_n, Y_n)$ des réels dépendant de (R_n, Y_n) . V_1, \dots, V_N sont centrés, indépendants, identiquement distribués et vérifient : pour tout $1 \leq n \leq N$, $V_n \perp (R_{1:N}, Y_{1:N})$

Notons que le mot "conditionnellement" dans CCLSM concerne aussi bien la linéarité du processus caché que la markovianité du processus des sauts.

Le filtrage et le lissage sont possibles dans CCLSM avec une complexité linéaire en nombre d'observations. Nous avons :

Filtrage

Proposition V.2 Soit $T = (X, R, Y)$ une chaîne conditionnellement linéaire à sauts markoviens. Alors $p(r_{n+1}|y_{1:n+1})$ et $\mathbb{E}[X_{n+1}|y_{1:n+1}]$ sont calculables avec une complexité linéaire en n .

Preuve : (R, Y) est une chaîne de Markov de transition $p(r_{n+1}, y_{n+1}|r_n, y_n)$. La probabilité conditionnelle

$$p(r_{n+1}|y_{1:n+1}) = \frac{p(r_{n+1}, y_{1:n+1})}{\sum_{r_{n+1}} p(r_{n+1}, y_{1:n+1})}$$

est calculable par récurrence :

$$\begin{aligned} p(r_{n+1}, y_{1:n+1}) &= \sum_{r_n} p(r_{n+1}, r_n, y_{1:n+1}) \\ &= \sum_{r_n} p(r_n, y_{1:n}) p(r_{n+1}, y_{n+1}|r_n, y_{1:n}) \\ &= \sum_{r_n} p(r_n, y_{1:n}) p(r_{n+1}, y_{n+1}|r_n, y_n) \end{aligned}$$

Nous pouvons écrire $\mathbb{E}[X_{n+1}|y_{1:n+1}]$ sous la forme suivante :

$$\mathbb{E}[X_{n+1}|y_{1:n+1}] = \sum_{r_{n+1}} \mathbb{E}[X_{n+1}|r_{n+1}, y_{1:n+1}] p(r_{n+1}|y_{1:n+1})$$

$\mathbb{E}[X_{n+1}|r_{n+1}, y_{1:n+1}]$ est alors calculable par récurrence. Nous avons :

$$\begin{aligned} \mathbb{E}[X_{n+1}|r_{n+1}, y_{1:n+1}] &= \sum_{r_n} \mathbb{E}[X_{n+1}|r_n, r_{n+1}, y_{1:n+1}] p(r_n|r_{n+1}, y_{1:n+1}) \\ &= \sum_{r_n} F_{n+1}(r_{n+1}, y_{n+1}) \mathbb{E}[X_n|r_n, y_{1:n}] p(r_n|r_{n+1}, y_{1:n+1}) \\ &\quad + \sum_{r_n} H_{n+1}(r_{n+1}, y_{n+1}) p(r_n|r_{n+1}, y_{1:n+1}) \end{aligned}$$

Par ailleurs, la probabilité $p(r_n|r_{n+1}, y_{1:n+1})$ est obtenue à partir du $p(r_n, r_{n+1}, y_{1:n+1})$ par :

$$\begin{aligned} p(r_n|r_{n+1}, y_{1:n+1}) &= \frac{p(r_n, r_{n+1}, y_{1:n+1})}{p(r_{n+1}, y_{1:n+1})} \\ &= \frac{p(r_n, y_{1:n})p(r_{n+1}, y_{n+1}|r_n, y_{1:n})}{p(r_{n+1}, y_{1:n+1})} \\ &= \frac{p(r_n, y_{1:n})}{p(r_{n+1}, y_{1:n+1})}p(r_{n+1}, y_{n+1}|r_n, y_{1:n}) \\ &= \frac{p(r_n, y_{1:n})}{p(r_{n+1}, y_{1:n+1})}p(r_{n+1}, y_{n+1}|r_n, y_n) \end{aligned}$$

Lissage

Proposition V.3 *Soit $T = (X, R, Y)$ une chaîne conditionnellement linéaire à sauts markoviens. Alors $p(r_{n+1}|y_{1:N})$ et $\mathbb{E}[X_{n+1}|y_{1:N}]$ sont calculables avec une complexité linéaire en N .*

Preuve : (R, Y) est une chaîne de Markov de transition $p(r_{n+1}, y_{n+1}|r_n, y_n)$. On a :

$$p(r_{n+1}|y_{1:N}) = \frac{p(r_{n+1}, y_{1:N})}{\sum_{r_{n+1}} p(r_{n+1}, y_{1:N})}$$

$$p(r_{n+1}, y_{1:N}) = p(r_{n+1}, y_{1:n+1})p(y_{n+2:N}|r_{n+1}, y_{1:n+1}) = \alpha_{n+1}(r_{n+1})\beta_{n+1}(r_{n+1})$$

avec $\alpha_{n+1}(r_{n+1})$ et $\beta_{n+1}(r_{n+1})$ calculables par récurrence avec une complexité algorithmique de l'ordre $N \times K^2$.

L'espérance conditionnelle $\mathbb{E}[X_{n+1}|r_{n+1}, y_{1:N}]$ est alors calculable par la récurrence suivante :

$$\begin{aligned} \mathbb{E}[X_{n+1}|r_{n+1}, y_{1:N}] &= \sum_{r_n} \mathbb{E}[X_{n+1}|r_n, r_{n+1}, y_{1:N}]p(r_n|r_{n+1}, y_{1:N}) \\ &= \sum_{r_n} F_{n+1}(r_{n+1}, y_{n+1})\mathbb{E}[X_n|r_n, y_{1:N}]p(r_n|r_{n+1}, y_{1:N}) \\ &\quad + \sum_{r_n} H_{n+1}(r_{n+1}, y_{n+1})p(r_n|r_{n+1}, y_{1:N}) \end{aligned}$$

avec $p(r_n|r_{n+1}, y_{1:N}) = \frac{\alpha_n(r_n)}{\alpha_{n+1}(r_{n+1})}p(r_{n+1}, y_{n+1}|r_n, y_n)$, ce qui termine la démonstration.

V.3.3 CCLSM et bruit gaussien à mémoire longue

Nous proposons dans cette sous-section notre modèle original, développé en collaboration avec Noufel Abbassi. L'idée générale est de remarquer que pour rendre les calculs possibles il est suffisant de supposer que $(R_{1:N}, Y_{1:N})$ est une chaîne partiellement de Markov. De plus, comme précisé dans la sous-section 5.2.2 ci-dessus, ces calculs sont simples dans le cas gaussien. Globalement, notre modèle ressemble, en ce qui concerne la loi de $(X_{1:N}, R_{1:N})$ conditionnelle à $(Y_{1:N})$, au modèle « chaîne conditionnellement linéaire à sauts markoviens » (CCLSM) considérée dans la sous-section précédente ; cependant, la loi du triplet $(X_{1:N}, R_{1:N}, Y_{1:N})$, qui n'est pas markovien, est ici plus complexe. Cependant, l'important est que (R, Y) étant une chaîne

partiellement de Markov, les calculs d'intérêt sont faisables dès que les transitions $p(z_n|x_{n-1}, y_{1:n-1})$ sont calculables.

Il y a ainsi trois niveaux de généralité dans les nouveaux modèles.

Au premier niveau, le plus général, nous avons un modèle identique à CCLSM défini dans la sous-section précédente, sauf qu'ici (R, Y) est une chaîne partiellement de Markov et non de Markov. Ce modèle sera encore noté CCLSM ici ; cependant, si cette différence doit être spécifiée nous dirons que le nouveau modèle est un "CCLSM avec (R, Y) chaîne couple partiellement de Markov (CCPM)" et nous le noterons CCLSM-CCPM. Un tel modèle est très général ; cependant, pour qu'il soit exploitable les transitions $p(z_n|x_{n-1}, y_{1:n-1})$ doivent être calculables.

Au deuxième niveau le processus (R, Y) est une CMC-BG. Les transitions $p(z_n|x_{n-1}, y_{1:n-1})$ sont alors calculables et un tel modèle, qui sera au besoin appelé CCLSM-CMC-BG, est exploitable.

Au troisième niveau le processus (R, Y) est une CMC-GML. C'est un cas particulier où l'on considère des bruits gaussiens paramétrés "à mémoire longue". Avec certaines paramétrisations il est alors possible d'estimer la loi de (R, Y) et proposer des méthodes de filtrage et de lissage "semi-supervisées". Un tel modèle sera noté appelé CCLSM-CMC-GML.

Précisons la définition des CCLSM-CMC-BG :

Définition V.8 Soit $T = (X, R, Y)$ une chaîne Triplet. T est une « chaîne conditionnellement linéaire à sauts markoviens et bruit gaussien » (CCLSM-BG) :

1. (R, Y) est une chaîne Gaussienne partiellement de Markov (CGPM) ;
2. pour tout $2 \leq n \leq N$, $X_n = F_n(R_n, Y_n)X_{n-1} + G_n(R_n, Y_n)V_n + H_n(R_n, Y_n)$,

où $F_n(R_n, Y_n)$, $G_n(R_n, Y_n)$ et $H_n(R_n, Y_n)$ des réels dépendent de (R_n, Y_n) . V_1, \dots, V_N sont centrés, indépendants et identiquement distribués et pour tout $1 \leq n \leq N$, $V_n \perp\!\!\!\perp (R_{1:N}, Y_{1:N})$

L'important est que (R, Y) étant une chaîne Gaussienne partiellement de Markov, les transitions $p(z_n|x_{n-1}, y_{1:n-1})$ sont calculables. Nous pouvons donc énoncer le résultat suivant :

Proposition V.4 Soit $T = (X, R, Y)$ une chaîne conditionnellement linéaire à sauts markoviens et bruit gaussien (CCLSM-BG). Alors le lissage (resp. le filtrage) est réalisable avec une complexité linéaire en N (resp. n).

Preuve : La preuve est analogue aux preuves des propositions (V.2) et (V.3). En effet, pour le lissage nous avons :

$$\begin{aligned} \mathbb{E}[X_{n+1}|r_{n+1}, y_{1:N}] &= \sum_{r_n} \mathbb{E}[X_{n+1}|r_n, r_{n+1}, y_{1:N}]p(r_n|r_{n+1}, y_{1:N}) \\ &= \sum_{r_n} F_{n+1}(r_{n+1}, y_{n+1})\mathbb{E}[X_n|r_n, y_{1:N}]p(r_n|r_{n+1}, y_{1:N}) \\ &\quad + \sum_{r_n} H_{n+1}(r_{n+1}, y_{n+1})p(r_n|r_{n+1}, y_{1:N}) \end{aligned}$$

et pour le filtrage :

$$\begin{aligned}\mathbb{E}[X_{n+1}|r_{n+1}, y_{1:n+1}] &= \sum_{r_n} \mathbb{E}[X_{n+1}|r_n, r_{n+1}, y_{1:n+1}]p(r_n|r_{n+1}, y_{1:n+1}) \\ &= \sum_{r_n} F_{n+1}(r_{n+1}, y_{n+1})\mathbb{E}[X_n|r_n, y_{1:n}]p(r_n|r_{n+1}, y_{1:n+1}) \\ &\quad + \sum_{r_n} H_{n+1}(r_{n+1}, y_{n+1})p(r_n|r_{n+1}, y_{1:n+1})\end{aligned}$$

les probabilités $p(r_n|r_{n+1}, y_{1:N})$ et $p(r_n|r_{n+1}, y_{1:n+1})$ sont calculables dès que $p(r_n, y_{n+1}|r_{n-1}, y_{1:n-1})$ sont calculables, nous avons :

$$\begin{aligned}p(r_n|r_{n+1}, y_{1:N}) &= \frac{p(r_n, r_{n+1}, y_{1:N})}{p(r_{n+1}, y_{1:N})} \\ &= \frac{p(r_n, y_{1:n})p(y_{n+1:N}, r_{n+1}|r_n, y_{1:n})}{\tilde{\alpha}_{n+1}(r_{n+1})\tilde{\beta}_{n+1}(r_{n+1})} \\ &= \frac{p(r_n, y_{1:n})p(y_{n+2:N}|r_{n+1}, y_{1:n+1})p(r_{n+1}, y_{n+1}|r_n, y_{1:n})}{\tilde{\alpha}_{n+1}(r_{n+1})\tilde{\beta}_{n+1}(r_{n+1})} \\ &= \frac{\tilde{\alpha}_n(r_n)}{\tilde{\alpha}_{n+1}(r_{n+1})}p(r_{n+1}, y_{n+1}|r_n, y_{1:n})\end{aligned}$$

et

$$\begin{aligned}p(r_n|r_{n+1}, y_{1:n+1}) &= \frac{p(r_n, r_{n+1}, y_{1:n+1})}{p(r_{n+1}, y_{1:n+1})} \\ &= \frac{p(r_n, y_{1:n})p(r_{n+1}, y_{n+1}|r_n, y_{1:n})}{p(r_{n+1}, y_{1:n+1})} \\ &= \frac{p(r_n, y_{1:n})}{p(r_{n+1}, y_{1:n+1})}p(r_{n+1}, y_{n+1}|r_n, y_{1:n})\end{aligned}$$

L'hypothèse que (R, Y) est une chaîne Gaussienne partiellement de Markov permet ainsi de prendre en considération des chaînes à bruit de mémoire longue et de faire des calculs exacts dans le cas d'un modèle à sauts markoviens. On peut considérer le cas particulier des CCLSM-BG, le le bruit gaussien est "à mémoire longue". Nous appellerons ces modèles "chaîne conditionnellement linéaire à sauts markoviens et bruit gaussien à mémoire longue" (CCLSM-GML). En particulier, lorsque ce bruit est paramétré, il est possible d'estimer la loi de (R, Y) à partir de Y , ce qui permet de proposer des méthodes de filtrage "semi-supervisées".

Proposition V.5 *Soit $T = (X, R, Y)$ une chaîne conditionnellement linéaire à sauts markoviens et bruit gaussien à mémoire longue (CCLSM-GML). Alors le lissage (resp. le filtrage) est réalisable avec une complexité linéaire en N (resp. en n).*

Preuve : même preuve que la preuve de la proposition (V.4).

Nous présentons dans la sous-section suivante quelques premiers résultats numériques obtenus avec les modèles CCLSM-GML.

V.4 Experimentations

Nous présentons deux séries d'expérimentations.

Dans la première on s'intéresse à l'estimation du processus R (la recherche des sauts). On simule deux séries de données : une première selon le modèle classique CMC-BI et une deuxième selon le modèle CCPM. Chacune d'elles est segmentée par la méthode bayésienne MPM, avec les vrais paramètres ou en non supervisé, utilisant, respectivement, chacun des deux modèles.

Dans la deuxième série on s'intéresse au filtrage exact fondé sur notre modèle CCLSM-GML. Nous comparons la qualité du filtrage obtenu "à R connu" avec celle obtenu dans le cas général où R est caché.

Nous commençons par simuler une chaîne de Markov R à deux classes et de taille $N = 500$. La chaîne est stationnaire avec la loi définie par $p(\omega_1, \omega_1) = 0.495$, $p(\omega_1, \omega_2) = 0.05$. La trajectoire obtenue est présentée à la figure (V.2).

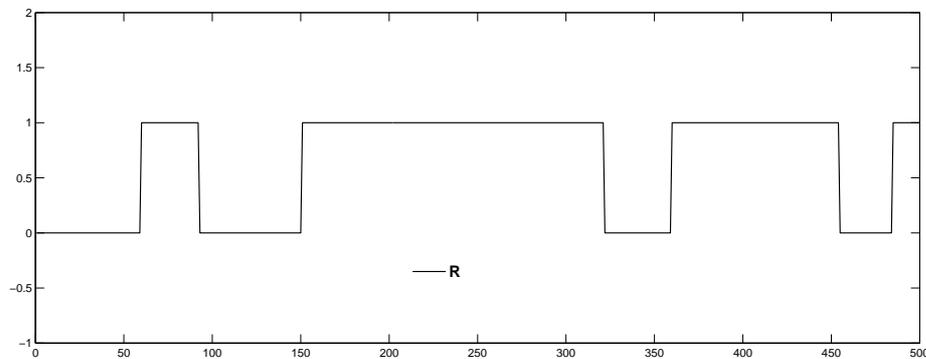


FIGURE V.2 – R : Chaîne de Markov $p(\omega_1, \omega_1) = 0.495$ et $p(\omega_1, \omega_2) = 0.05$

Ensuite, nous appliquons trois types de bruits sur la chaîne R : un bruit gaussien indépendant et deux bruits gaussiens à mémoire longue. Les trajectoires $y_{1:N}^1$, $y_{1:N}^2$ et $y_{1:N}^3$ obtenues ainsi que les paramètres des bruits sont présentés à la figure (V.3) Chacune de ces trajectoires est segmentée par deux méthodes bayésiennes MPM : la première utilise le modèle CMC-BI, et la deuxième utilise le modèle CCLSM-GML. Les segmentations se font en non supervisé, les paramètres étant estimés par la méthode ICE. Pour le modèle du CMC-BI, nous utilisons les estimateurs que nous avons détaillés dans le chapitre (III). Dans notre cas, où le bruit est stationnaire du second ordre avec une famille de covariance $\gamma(h)$ de la forme :

$$\forall h \in \mathbb{N}, \gamma(h) = \sigma^2(1+h)^{-\alpha}, \quad (\text{V.23})$$

les paramètres du modèle CCLSM-GML sont :

- Pour la loi de la chaîne X : les K^2 paramètres $p(x_n = \omega_i, x_{n+1} = \omega_j)$;
- Pour la chaîne Y conditionnellement à X :
 - Les K moyennes m_{ω_i} ,
 - Les K variances $\sigma_{\omega_i}^2 = \gamma_{\omega_i}(0)$,
 - Les K paramètres α_{ω_i} de la famille de covariance.

Nous utilisons comme estimateur pour $p_{i,j} = p(x_n = \omega_i, x_{n+1} = \omega_j)$ à partir des données complètes l'estimateur classique donné par :

$$\widehat{p}_{i,j}(x, y) = \frac{1}{(N-1)} \sum_{n=1}^{N-1} 1(x_n = \omega_i, x_{n+1} = \omega_j)$$

À chaque itération (q) de l'algorithme ICE, l'espérance conditionnelle de l'estimateur $\widehat{p}_{i,j}$ est calculable et elle est donnée par :

$$p_{i,j}^{(q+1)} = \frac{1}{(N-1)} \sum_{n=1}^{N-1} p(x_n = \omega_i, x_{n+1} = \omega_j | y_{1:N}, \theta^{(q)})$$

Le fait que la loi de y_n dépend de tous les x_k $1 \leq k \leq n$ rend la tâche d'estimation des paramètres m_{ω_i} et $\gamma_{\omega_i}(0)$ à partir des données complètes non triviale. Pour illustrer cette difficulté des exemples sont donnés dans [66, 70], ainsi que une description de l'algorithme ICE pour différents types de modèles (bruit stationnaire de second ordre, bruits gaussiens fractionnaires, FARIMA). Dans le cas du bruit stationnaire de second ordre avec une famille de covariances considérée de la forme donnée par (V.23), l'algorithme ICE, pour estimer m_{ω_i} , σ_{ω_i} et α_{ω_i} , se déroule de la manière suivante [66, 70] :

Algorithme V.1 Estimation des paramètres m_{ω_i} , σ_{ω_i} et α_{ω_i} par ICE cas de bruit stationnaire de second ordre (V.23) :

1. Sous $\theta^{(q)}$, calculer les densités $p(y_{1:n+1} | x_{1:n+1})$ par la récurrence progressive (décrite par les équations (V.13) et (V.14)) et en utilisant $m_{\omega_i} = m_{\omega_i}^{(q)}$ et $\Gamma_{\omega_i}^n = \Gamma_{\omega_i}^{n,(q)}$;
2. Considérer $J(i) = \{n_1, \dots, n_r\}$ sous-ensemble de $\{1, \dots, N\}$ tel que $\forall 1 \leq l \leq r$, $x_{n_l} = x_{n_l+1} = \omega_i$. Considérer les r échantillons correspondants $(y_{n_1}, y_{n_1+1}), \dots, (y_{n_r}, y_{n_r+1})$;
3. soient $\forall 1 \leq l \leq r$, $M_{x_{n_l:n_l+1}}^{(q)}$ et $\Gamma_{x_{n_l:n_l+1}}^{(q)}$ les vecteurs moyenne et matrices de covariance de $p(y_{n_l:n_l+1} | x_{1:n_l+1})$ calculées dans l'étape (1). Calculer les transformations de Cholesky (IV.2) $\Gamma_{x_{n_l:n_l+1}}^{(q)} = C_l C_l^T$ et $\Gamma_{\omega_i}^{2,(q)} = D_l D_l^T$ et construire les r nouveaux échantillons, $\forall 1 \leq l \leq r$:

$$\begin{pmatrix} \widetilde{y}_{n_l} \\ \widetilde{y}_{n_l+1} \end{pmatrix} = D_l C_l^{-1} \left[\begin{pmatrix} y_{n_l} \\ y_{n_l+1} \end{pmatrix} - M_{x_{n_l:n_l+1}}^{(q)} \right] (C_l^T)^{-1} D_l^T + m_{\omega_i}^{2,(q)} ;$$

4. estimer les paramètres m_{ω_i} , $\sigma_{\omega_i}^2$ et α_{ω_i} à partir de r nouveaux échantillons obtenus l'étape dans (3) par :

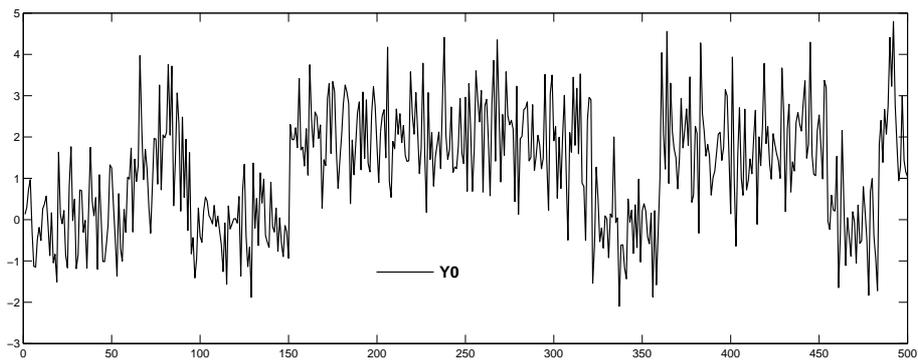
$$\begin{aligned} m_{\omega_i}^{(q+1)} &= \frac{1}{2} (m_{\omega_i,(1)}^{(q+1)} + m_{\omega_i,(2)}^{(q+1)}); \\ (\sigma_{\omega_i}^2)^{(q+1)} &= \frac{1}{2} (\widehat{\gamma}_{(1)}(0) + \widehat{\gamma}_{(2)}(0)); \\ \alpha_{\omega_i}^{(q+1)} &= \frac{\log\left[\frac{\widehat{\gamma}_{(1)}(0)}{(\sigma_{\omega_i}^2)^{(q+1)}}\right]}{\log(2)} \end{aligned}$$

$$\begin{aligned} \text{avec } m_{\omega_i,(1)}^{(q+1)} &= \frac{1}{r} \sum_{l=1}^r \widetilde{y}_{n_l}, \quad m_{\omega_i,(2)}^{(q+1)} = \frac{1}{r} \sum_{l=1}^r \widetilde{y}_{n_l+1}, \quad \widehat{\gamma}_{(1)}(0) = \frac{1}{r} \sum_{l=1}^r (\widetilde{y}_{n_l} - m_{\omega_i,(1)}^{(q+1)})^2, \\ \widehat{\gamma}_{(2)}(0) &= \frac{1}{r} \sum_{l=1}^r (\widetilde{y}_{n_l+1} - m_{\omega_i,(2)}^{(q+1)})^2 \text{ et } \widehat{\gamma}_{(1)} = \frac{1}{r} \sum_{l=1}^r (\widetilde{y}_{n_l} - m_{\omega_i,(1)}^{(q+1)}) (\widetilde{y}_{n_l+1} - m_{\omega_i,(2)}^{(q+1)}) \end{aligned}$$

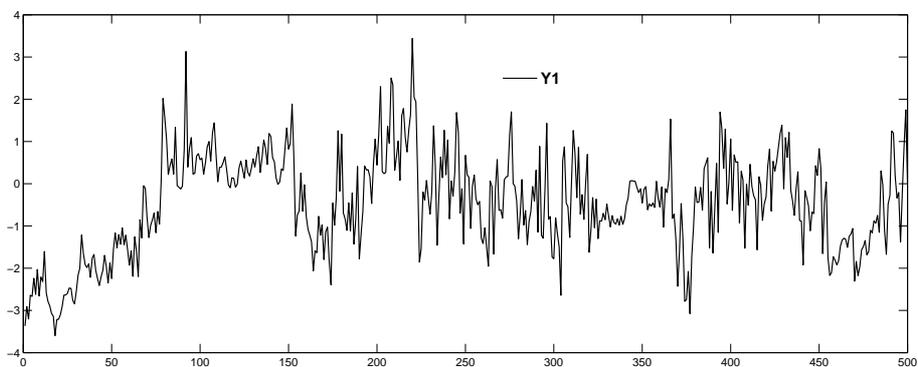
Les résultats des trois premières segmentations, obtenues avec CMC-BI, sont présentés à la figure (V.4), et les taux d'erreur sont donnés dans la première colonne de la table (V.1). Nous constatons que les données simulées selon CMC-BI sont presque parfaitement restaurées; par contre, lorsque les données sont à mémoire longue les résultats sont soit très mauvais, soit relativement médiocres. L'association de CMC-BI avec ICE apparaît ainsi, lorsque les données sont à mémoire longue, comme étant peu robuste.

Les résultats des trois segmentations suivantes, obtenues avec CCLSM-GML, sont présentés à la figure (V.5), et les taux d'erreur sont donnés dans la deuxième colonne de la table (V.1). Nous constatons que les données simulées selon CMC-BI sont restaurées avec la même qualité que celle donnée par la méthode fondée sur CMC-BI. Ce constat est intéressant car CMC-BI n'est pas, à proprement parlé, un cas particulier de CCLSM-GML. Cela montre que CCLSM-GML peut gérer efficacement des bruits non corrélés.

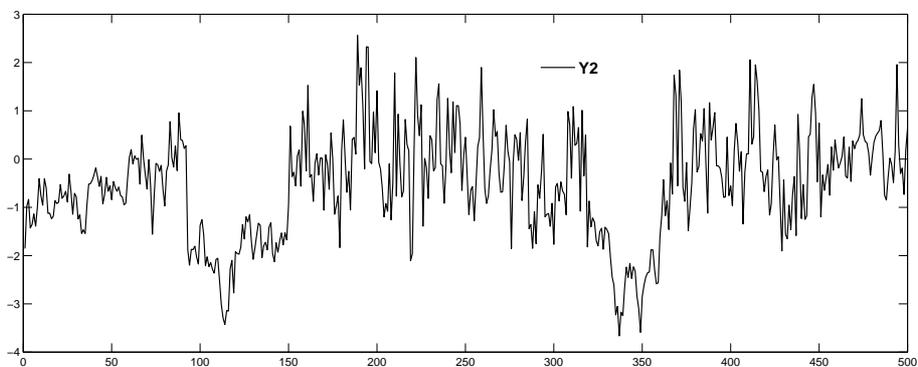
Dans la deuxième série on s'intéresse au filtrage de Y .



(a)



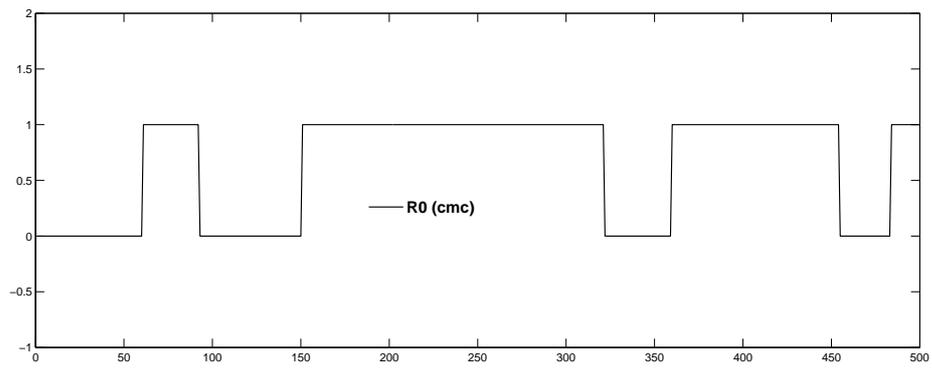
(b)



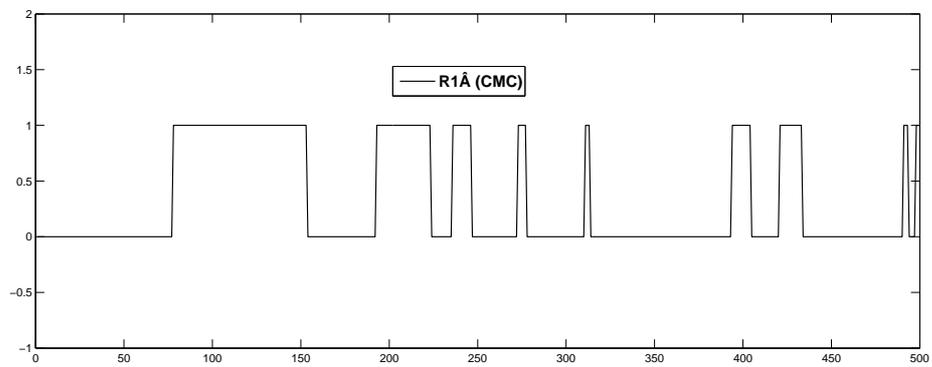
(c)

FIGURE V.3 – Version bruitée du chaîne R avec (a) $y_{1:N}^1$: bruit gaussien indépendant $\mathcal{N}_1(0, 1)$ $\mathcal{N}_2(2, 1)$ (b) $y_{1:N}^2$: bruit à mémoire longue de même moyenne 0 et de même variance 1, $\alpha_{\omega_1} = 0.1$ $\alpha_{\omega_2} = 0.9$ (c) $y_{1:N}^3$: bruit à mémoire longue de même moyenne 0 et de même variance 1, $\alpha_{\omega_1} = 0.1$ $\alpha_{\omega_2} = 1$

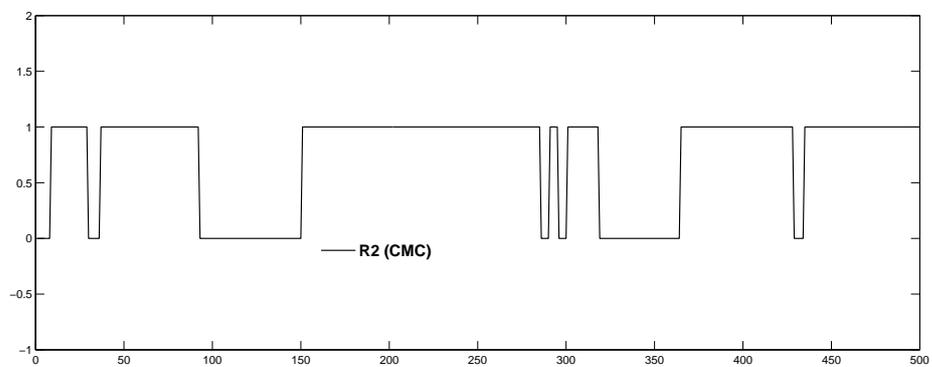
La segmentation des processus $y_{1:N}^1$, $y_{1:N}^2$ et $y_{1:N}^3$ par le modèle de chaîne de Markov cachée est donnée par la figure (V.4) et par le modèle de chaîne cachée partiellement de Markov sur la figure (V.5)



(a')

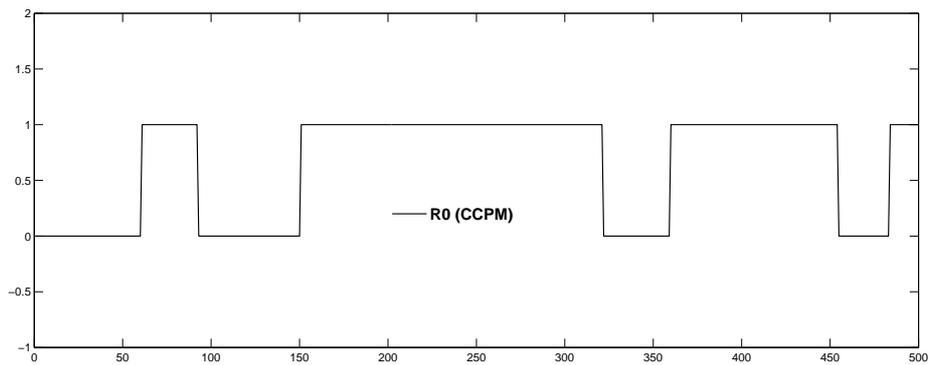


(b')

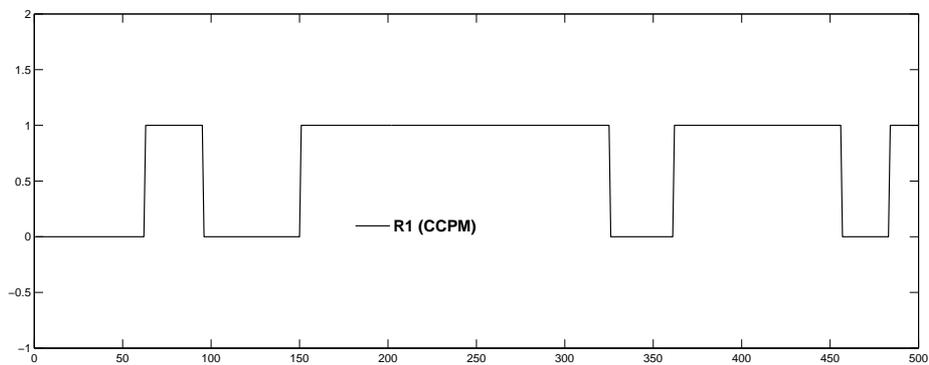


(c')

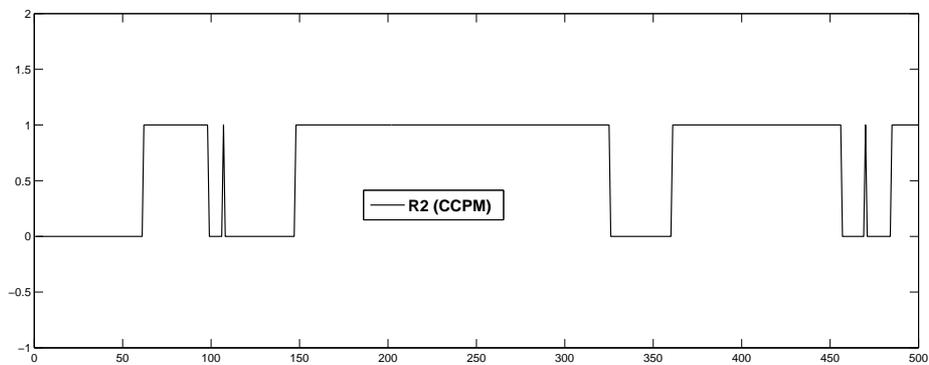
FIGURE V.4 – Segmentation non supervisée par CMC-BI



(a'')



(b'')



(c'')

FIGURE V.5 – Segmentation non supervisée par CCLSM-GML

Les taux d'erreur de ces expériences sont résumés dans la table (V.1). Nous pouvons remarquer que les deux modèles ont la même performance que dans le cas (a), par contre dans les deux autres cas (b) et (c), le modèle CCPM à mémoire longue est plus performant du fait que le modèle CMC ne prend pas en compte les données corrélées.

	CMC-BI	CCLSM-GML
$y_{1:N}^1$	0.2	0.2
$y_{1:N}^2$	61.6	2.0
$y_{1:N}^3$	13.6	5.1

TABLE V.1 – Taux d’erreur de segmentations en %

Finalement, nous appliquons l’algorithme de lissage pour calculer $\mathbb{E}[X_n|y_{1:N}]$ selon le modèle :

$$\forall 2 \leq n \leq N, X_n = F_n(R_n, Y_n)X_{n-1} + G_n(R_n, Y_n)V_n + H_n(R_n, Y_n) \quad (\text{V.24})$$

avec pour tout $2 \leq n \leq N$, $F_n(\omega_1, Y_n) = \frac{3 \times Y_n}{10}$, $F_n(\omega_2, Y_n) = -\frac{3 \times Y_n}{10}$, $H_n(\omega_1, Y_n) = \frac{1}{2}$, $H_n(\omega_2, Y_n) = -\frac{1}{2}$ et $G_n(R_n, Y_n) = \frac{1}{2}$ et $V_n \sim \mathcal{N}(0, 1)$.

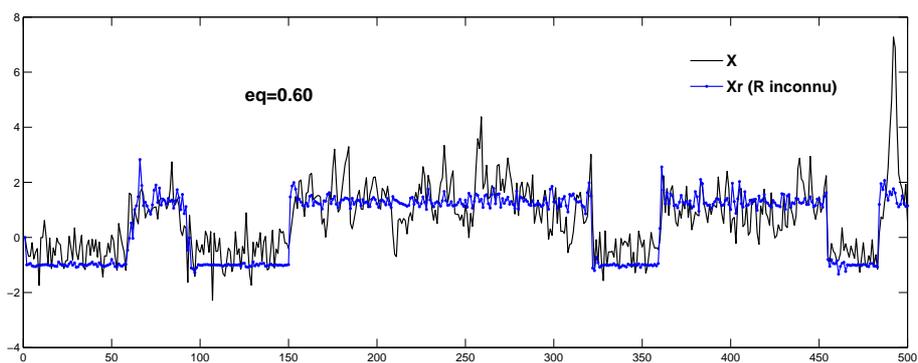
Pour chacun des trois cas (a), (b), et (c) de la figure (V.3), on procède au lissage par la nouvelle méthode générale. Les résultats obtenus sont présentés sur la figure (V.5), respectivement en a”, b”, et c”. Dans chaque cas on présente deux trajectoires : la vraie ($X = x$ simulée selon (V.24)) et l’estimée (à R inconnu). L’écart entre les deux trajectoires est mesuré par l’erreur quadratique moyenne (eq) qui est, respectivement, de 0.60, 0.60, et 0.61. Nous cherchons à comparer notre méthode avec celle utilisant la vraie trajectoire des sauts donné à la figure (V.2) Sur la figure (V.7), nous présentons ainsi les trajectoires du processus X calculées en utilisant le vrai processus R et les observations $y_{1:N}^i$ avec les trajectoires estimées par le lissage (figure (V.6)). Pour tout $i = 1, 2, 3$, la trajectoire estimée connaissant la trajectoire des sauts se calcule simplement par :

$$\forall 2 \leq n \leq N, x_n^i = F_n(r_n, y_n^i)x_{n-1}^i + H_n(r_n, y_n^i) \quad (\text{V.25})$$

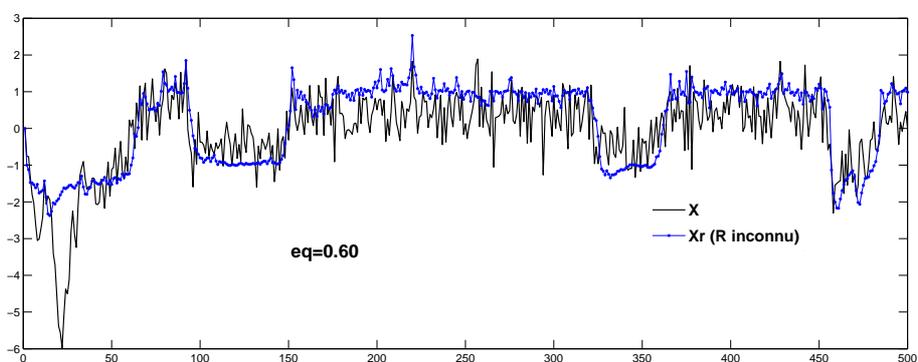
Les résultats obtenus sont présentés sur la figure (V.6), respectivement en a””, b””, et c””. Comme à la figure (V.7), dans chaque cas on présente deux trajectoires : la vraie ($X = x$ simulée selon (V.24)) et l’estimée (à R connu) par (V.25). L’écart entre les deux trajectoires est mesuré par l’erreur quadratique moyenne (eq) qui est, respectivement, de 0.39, 0.27, et 0.29.

Naturellement, les résultats obtenus avec la trajectoire des sauts connue donne de meilleurs résultats ; cependant, la différence n’est pas très grande et visuellement les résultats sont assez proches.

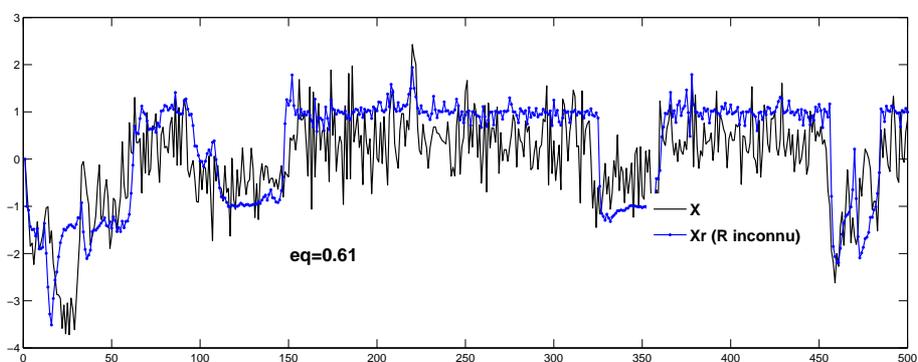
Rappelons que les résultats de la Figure 5.7 sont obtenus de manière "semi-supervisée" : la loi de (R, Y) est estimée, mais les paramètres F_n , B_n et G_n sont connus.



a''



b''



c''

FIGURE V.6 – Les vraies trajectoires $X = x$ et celles obtenues par lissage fondé sur la nouvelle méthode (R inconnu). eq est l'erreur quadratique moyenne

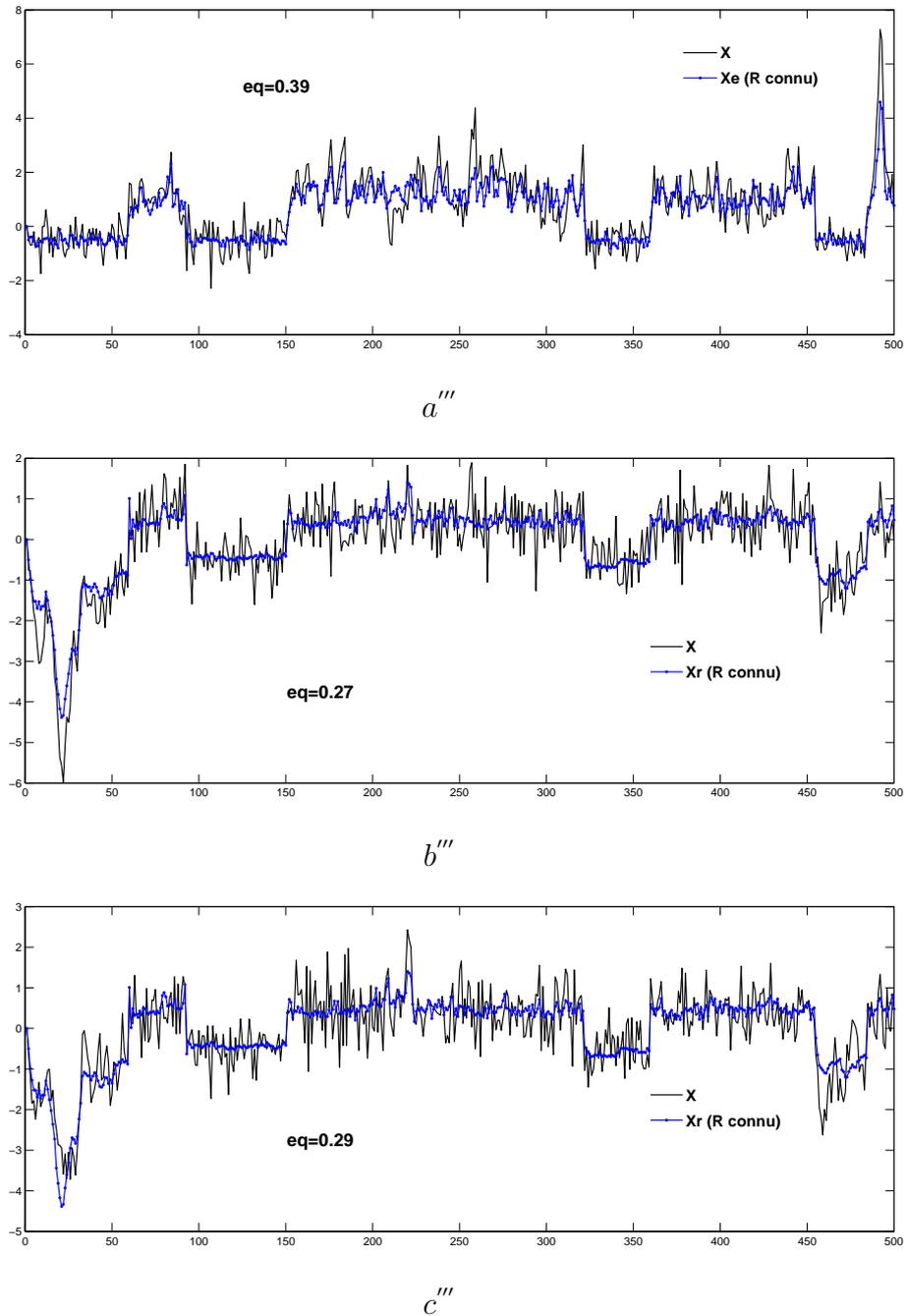


FIGURE V.7 – Les vraies trajectoires $X = x$ et celles obtenues par (V.25), avec R connu. eq est l’erreur quadratique moyenne.

V.5 Conclusion

Ce dernier chapitre a été consacré au nouveau modèle, développé en collaboration avec Noufel Abbassi, permettant le filtrage et le lissage optimaux en présence des sauts aléatoires. Son originalité consiste en introduction des bruits à mémoire longue. On quitte ainsi le cadre des triplets markoviens et l’on se place dans le

cadre des triplets "partiellement" de Markov. Dans un premier temps, nous avons rappelé les pré-requis dans le domaine des processus à dépendance longue et les travaux précédents sur les modèles de chaînes partiellement de Markov. Les modèles partiellement de Markov furent préalablement étudiés par P. Lanchantin et J. Lapyade [64, 70] dans le cas de processus cachés à valeurs discrètes. Nous nous sommes inspirés de ces modèles pour développer des algorithmes permettant de faire du lissage et du filtrage pour les modèles à sauts markoviens. Le lissage et le filtrage sont alors possibles car le nouveau modèle permet de calculer d'une manière exacte et linéaire en N les quantités $\mathbb{E}[X_n|y_{1:N}]$ et $\mathbb{E}[X_n|y_{1:n}]$. Les premiers résultats de simulations et de restauration présentés sont encourageant. Cependant, des expériences plus complètes sont nécessaires pour valider définitivement les nouveaux modèles et les traitements associés. Par ailleurs, le problème de l'estimation des paramètres n'a été traité que partiellement, aboutissant à des méthodes "semi-supervisées". Là encore des études complémentaires sont nécessaires pour étudier les possibilités des traitements entièrement non supervisés.

Les comparaisons de l'efficacité de nos modèles avec les approches classiques par filtrage particulaire font probablement partie des perspectives parmi les plus intéressantes.

Conclusion et perspectives

Le travail de notre thèse se situe dans le domaine de traitement statistique des signaux. L'objectif de cette thèse était de proposer et d'étudier des nouveaux modèles pour la segmentation des signaux et des images. Les modèles de Markov cachés [5, 12, 53] permettent d'estimer la réalisation $x = (x_n)_{1:N}$ d'un processus caché $X = (X_s)_{s \in \mathcal{S}}$ à partir des données observées $y = (y_n)_{1:N}$ qui sont la réalisation d'un processus $Y = (Y_n)_{1:N}$. Les probabilités conditionnelles $(x_n|y)$ sont calculables du fait que X conditionnellement à Y est markovien qui est une conséquence de l'hypothèse X est de Markov. Cette hypothèse n'est pas toujours vérifiée dans les cas réels et elle impose des contraintes restrictives sur la loi $p(y|x)$. Des extensions généralisent les modèles de Markov cachés ont été proposées : modèles de Markov Couples (MMC) [91, 93, 104] et modèles de Markov Triplets (MMT) [36, 92, 94]. Dans le MMC nous supposons que le couple (X, Y) est de Markov ce qui permet d'éviter des contraintes sur X et dans le MMT nous introduisons un processus auxiliaire $U = (U_n)_{1:N}$ qui peut avoir une interprétation physique ou non et nous supposons que le couple $T = (X, U, Y)$ est de Markov. Le processus U peut avoir plusieurs utilisations donnant une très grande variété de modèles particuliers. En effet, il permet de modéliser la semi-markovianité d'une chaîne de Markov caché (CMC) ou plus généralement d'une chaîne de Markov Couple (CMC Couple), une autre utilisation du processus auxiliaire est de modéliser la non-stationnarité. Dans cette thèse, nous avons proposé un nouveau modèle particulier de CMT dont le processus auxiliaire U est continu, et nous avons utilisé l'approximation de Laplace pour estimer la réalisation x du processus X à partir de l'observé y en supposant que X et Y sont indépendants conditionnellement à U . Les résultats de la segmentation des images simulées et artificielles sont meilleurs avec notre nouveau modèle qu'avec le modèle classique du CMC, et cette amélioration se manifeste clairement dans le cas des bruits corrélés. Également, nous avons appliqué le modèle du CMT pour proposer des nouveaux modèles qui traitent le filtrage et le lissage des chaînes cachés à saut markovien [49, 99, 100]. Ces modèles permettent de prendre en considération les bruits à dépendance longue en utilisant le modèle de chaînes partiellement de Markov qui ont été présentés par J. Lapuyade [70].

Comme perspective, on peut envisager de faire de l'estimation des paramètres pour le modèle du champ Gaussien-Markovien triplet par l'algorithme ICE et aussi l'estimation des paramètres de la chaîne X dans le cas des modèles cachés à saut markovien. on peut suggérer aussi de comparer les nouveaux modèles de filtrage et de lissage avec les modèles qui sont basés sur le filtrage particulière et d'appliquer ces nouveaux modèles sur des données réelles.

Annexe A

K-means

L'algorithme K-means est une méthode itérative de classification, que nous l'utilisons pour initialiser différents algorithmes d'estimation des paramètres. Le résultat de la classification en K classes ($|\Omega| = K$) d'un processus y par K-means est noté x^0 , et le vecteur des paramètres initial θ^0 est obtenu en choisissant un estimateur $\widehat{\theta}$ à partir des données complètes (x, y)

$$\theta^0 = \widehat{\theta}(x^0, y)$$

L'algorithme K-means se présente comme suit :

□ Initialisation :

- i) Affecter d'une manière aléatoire à chaque pixel $s \in \mathcal{S}$ une classe $\omega_k \in \Omega$.
- ii) Calculer pour chaque classe ω_k la masse de sa centre m_k

$$m_k = \frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} y_s 1(x_s = \omega_k)$$

□ Répéter jusqu'à stabilisation :

- i) Affecter à chaque pixel $s \in \mathcal{S}$ la classe ω_k qui minimise sa distance par rapport au centres des classes.

$$\omega_k = \underset{1 \leq j \leq K}{\operatorname{argmin}} \|y_s - m_j\|$$

- ii) Recalculer les masses de centres des classes.

Annexe B

Approximation de Laplace

La méthode de Laplace permet d'approximer numériquement un intégral de la forme sous certaines conditions :

$$I = \int_{\mathbb{R}} \exp[Mf(x)] dx \quad (\text{B.1})$$

avec $M \in \mathbb{R}_+^*$ et $f : \mathbb{R} \rightarrow \mathbb{R}_+^*$ deux fois dérivable.

Un développement de Taylor à l'ordre 2 au voisinage de x_0 de f :

$$f(x) = f(x_0) + (x - x_0)f'(x_0) + \frac{(x - x_0)^2}{2}f''(x_0) + O((x - x_0)^3) \quad (\text{B.2})$$

si f admet un maximum global stable en x_0 ($f'(x_0) = 0$ et $f''(x_0) < 0$)

$$f(x) \simeq f(x_0) + \frac{(x - x_0)^2}{2}f''(x_0)$$

l'approximation de I :

$$I \simeq \exp[Mf(x_0)] \int_{\mathbb{R}} \exp\left[M \frac{(x - x_0)^2}{2} f''(x_0)\right] dx \quad (\text{B.3})$$

$$\simeq \exp[Mf(x_0)] \sqrt{\frac{2\pi}{M|f''(x_0)|}} \quad (\text{B.4})$$

Annexe C

Vecteurs gaussiens

Définition C.1 On dit qu'un vecteur réel aléatoire $X = (X_1, \dots, X_n)$ est un vecteur réel gaussien si et seulement si pour toute suite de réels $a = (a_1, \dots, a_n) \in \mathbb{R}^n$, la variable aléatoire $Z = \sum_{i=1}^n a_i X_i$ est une variable réelle gaussienne.

Remarque C.1 Soit $X = (X_1, \dots, X_n)$ est un vecteur réel gaussien :

i) Pour toute $i = 1, \dots, n$ X_i est une variable réelle gaussienne de loi $\mathcal{N}(m_i, \sigma_i^2)$

ii) Si $Y = \sum_{i=1}^n \alpha_i X_i$, on a :

$$\mathbb{E}[Y] = \sum_{i=1}^n \alpha_i m_i$$

$$\mathbb{V}[Y] = \sum_{i,j}^n \alpha_i \alpha_j \text{Cov}(X_i, X_j)$$

Soit $X = (X_1, \dots, X_n)$ est un vecteur réel gaussien, on appelle espérance de X le vecteur $m = \begin{pmatrix} m_1 \\ \vdots \\ m_n \end{pmatrix}$ où $m_i = \mathbb{E}[X_i]$. On appelle matrice de covariance de X la matrice Γ telle que $\Gamma_{i,j} = \text{Cov}(X_i, X_j)$. On note la loi de X :

$$X \sim \mathcal{N}_n(m, \Gamma).$$

Soient X^1 et X^2 deux vecteurs réels gaussiens de dimensions respectives n_1 et n_2 , de moyennes respectives m^1 et m^2 et de matrices de covariances respectives $\Sigma^{1,1}$ et $\Sigma^{2,2}$, tel que le couple $X = (X^1, X^2)$ est un vecteur réel gaussien de moyenne $m = \begin{pmatrix} m^1 \\ m^2 \end{pmatrix}$ et de matrice de covariance $\Sigma = \begin{pmatrix} \Sigma^{1,1} & \Sigma^{1,2} \\ \Sigma^{2,1} & \Sigma^{2,2} \end{pmatrix}$. La loi $p(x^2|x^1)$ est une gaussienne de moyenne :

$$m^{2|1} = m^2 + \Sigma^{2,1}(\Sigma^{1,1})^{-1}(x^1 - m^1),$$

et de matrice de covariance :

$$\Sigma^{2|1} = \Sigma^{2,2} - \Sigma^{2,1}(\Sigma^{1,1})^{-1}\Sigma^{1,2}.$$

Soient X^1 et $(X^2|X^1 = x^1)$ deux vecteurs réels gaussiens de moyenne respectives m^1 et $A.x^1 + B$ et de matrices de covariances respectives $\Sigma^{1,1}$ et $\Sigma^{2,1}$ alors le couple $X = (X^1, X^2)$ est un vecteur gaussien de moyenne

$$m = \begin{pmatrix} m^1 \\ A.m^1 + B \end{pmatrix}$$

et de matrice de covariance :

$$\Sigma = \begin{pmatrix} \Sigma^{1,1} & \Sigma^{1,1}A^T \\ A\Sigma^{1,1} & \Sigma^{2,1} + A\Sigma^{1,1}A^T \end{pmatrix}$$

Bibliographie

- [1] C. Andrieu, M. Davy, and A. Doucet. Efficient particle filtering for jump markov systems. application to time-varying autoregressions. *IEEE Trans. on Signal Process*, 51(7) :1762–1770, 2003.
- [2] B. Annelies. *Reconnaissance et compréhension des gestes, applications à la langue des signes*. PhD thesis, Paris 6, 1996.
- [3] J. K. Baker. The dragon system. *IEEE Trans. on Acoustics Speech and Signal Processing*, 23(1) :24–29, 1975.
- [4] V. S. Barbu and N. Limnios. Semi-markov chains and hidden semi-markov models toward applications. *Their Use in Reliability and DNA Analysis*, 191, 2008.
- [5] L.E. Baum, T. Petrie, G. Soules, and N. Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chain. *Ann. of Math. Statistics*, 41 :164–171, 1970.
- [6] D. Benboudjema. *Champs de Markov Triplets et segmentation Bayésienne non supervisée d’images*. Optimisation et sûreté des systèmes, Université de Technologie de Troyes-Institut National des Télécommunications, 2005.
- [7] B. Benmiloud and W. Pieczynski. Estimation des paramètres dans les chaînes de markov cachées et segmentation d’images. *Traitement du Signal*, 12(5) :433–454, 1995.
- [8] F. Le Ber, M. Benoit, C. Scott, J. F. Mari, and C. Mignolet. Studying crop sequences with carrotage a hmm-based data mixing software. *Ecological Modelling*, 191(1) :170–185, 2006.
- [9] J. Besag. Spatial interaction and the statistical analysis of lattice systems. *J. of the Royal Statistical Society Series B*, 36(2) :192–236, 1974.
- [10] J. Besag. On the statistical analysis of dirty picture. *J. of the Royal Statistical Society Series B*, 48(3) :259–302, 1986.
- [11] J. Besag and C. Kooperberg. On conditional intrinsic autoregressions. *Biometrika*, pages 733–746, 1995.
- [12] J. Besag, J. York, and A. Mollé. Bayesian analysis of agricultural field experiments (with discussion). *J. of the Royal Statistical Society Series B*, 61 :691–746, 1991.
- [13] I. Bloch, Y. Gousseau, H. Maitre, D. Matignon, M. Sigelle, and F. Tupin. *Le traitement des images*, volume Tome 1. 2003-2004.

- [14] A. Bonin, B. Chalmond, and B. Lavayssière. Monte-carlo simulation of industrial radiography images and experimental designs. *NDT et E. International*, 35(8) :503–510, 2002.
- [15] G. E. P. Box and G. M. Jenkins. *Times Series Analysis Forecasting and Control*. Holden Day, 1970.
- [16] M. Broniatowski, G. Celeux, and J. Diebolt. Reconnaissance de mélange de densités par un algorithme d'apprentissage probabiliste. *Data Analysis and Informatics*, 3 :359–373, 1984.
- [17] N. Brunel. *Sur quelques extensions des chaînes de Markov cachées et couples, Applications à la segmentation non-supervisée de signaux radar*. Mathématiques appliquées, Université Paris 6, 2005.
- [18] O. Cappé, E. Moulines, and T. Ryden. *Inference in hidden Markov Models*. Springer, Series in Statistics, 2005.
- [19] G. Celeux, D. Chauveau, and J. Diebolt. A stochastic version of the em algorithm : an experimental study. *Journal of Statistical Computation and Simulation*, 55 :287–314, 1996.
- [20] G. Celeux and J. Diebolt. The sem algorithm : a probabilistic teacher algorithm derived from the em algorithm for the mixture problem. *Computational Statistics Quarterly*, 2(1) :73–82, 1985.
- [21] G. Celeux and J. Diebolt. L'algorithme sem : un algorithme d'apprentissage probabiliste pour la reconnaissance de mélanges de densités. *Revue de Statistique Appliquée*, 34, 1986.
- [22] G. Celeux and J. Diebolt. A stochastic approximation type em algorithm for the mixture problem. *In Stochastics and Stochastics Reports*, 41 :119–134, 1992.
- [23] G. Celeux and J. Diebolt. Une version de type recuit simulé de l'algorithme em. Technical report, INRIA, 2006.
- [24] B. Chalmond. *Éléments de modélisation pour l'analyse d'images*. Springer, 2000.
- [25] M. Chen, A. Kundu, and J. Zhou. Off-line handwritten work recognition using a hidden markov model type stochastic network. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 16(5) :481–496, 1994.
- [26] R. Christensen. *Loglinear models*. Springer-Verlag, 1990.
- [27] O. Chryssaphinou, M. Karaliopoulou, and N. Limnios. On discrete time semi-markov chains and applications in words occurrences. *Communications in Statistics, Theory and Methods*, 37 :1306–1322, 2008.
- [28] O. Chryssaphinou, M. Karaliopoulou, and N. Limnios. On discrete time semi-markov chains and applications in words occurrences. *Communications in statistics. Theory and methods*, 37(8) :1306–1322, 2008.
- [29] G. A. Churchill. Stochastic models for hertergenous dna squences. *Bulletin of Mathematic Biology*, 51 :79–94, 1989.
- [30] J. P. Cocquerez and S. Philipp. *Analyse d'images : filtrage et segmentation*. Masson, 1995.

-
- [31] O. L. V. Costa, M. D. Fragoso, and R. P. Marques. Discrete time markov jump linear systems. Springer-Verlag, New York, 2005.
- [32] Y. Delignon, A. Marzouki, and W. Pieczynski. Estimation of generalized mixture and its application in image segmentation. *IEEE Trans. on Image Processing*, 6(10) :1364–1375, 1997.
- [33] J. P. DELMAS. Relations entre les algorithmes d’estimation itératives em et ice avec exemples d’applications. *GRETSI Groupe d’Etudes du Traitement du Signal et des Images*, pages 185–188, 1995.
- [34] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *J. of the Royal Statistical Society Series B*, 39 :1–38, 1977.
- [35] S. Derrode and W. Pieczynski. Signal and image segmentation using pairwise markov chains. *IEEE Trans. on Signal Processing*, 6(52) :2477–2489, 2004.
- [36] F. Desbouvries and W. Pieczynski. Modèles de markov triplet et filtrage de kalman, triplet markov models and kalman filtering. *Comptes Rendus de l’Académie des Sciences*, 336(8) :667–670, Février 2003.
- [37] P. A. Devijver. Baum’s forward-backward algorithm revisited. *Pattern Recognition*, 3(6) :369–373, 1985.
- [38] A. M. Djafari. Entropie en traitement du signal. cnrs-supélec-ups.
- [39] A. M. Djafari and G. Demoment. Using entropy in image reconstruction and restauration. *Traitement du signal*, 5(4) :235–248, 1988.
- [40] A. Doucet, A. N. Gordon, and V. Krishnamurthy. Particle filters for state estimation of jump markov linear systems. *IEEE Trans. On Signal Processing*, 49 :613–624, 2001.
- [41] Tyrone E. Duncan. *Some applications of fractional brownian motion to linear systems*. Springer, 1999.
- [42] J. B. Durand. *Modèles à structure cachée : inférence, sélection des modèles et applications*. Mathématiques appliquées, Université Grenoble I-Joseph Fourier, 2003.
- [43] B. Ait el Fquih and F. Desbouvries. Kalman filtering in triplet markov chains. *IEEE Trans. on Signal Processing*, 54(8) :2957–2963, 2006.
- [44] B. Ait el Fquih and F. Desbouvries. Bayesian smoothing algorithms in pairwise and triplet markov chains. In *Proceedings of the 2005 IEEE Workshop on Statistical Signal Processing*, Bordeaux, France, July 17-20 2005.
- [45] B. Ait el Fquih and F. Desbouvries. Exact and approximate bayesian smoothing algorithms in partially observed markov chains. In *Proceedings of the IEEE Nonlinear Statistical Signal Processing Workshop (NSSPW’06)*, Cambridge, UK, September 13-15 2006.
- [46] B. Ait el Fquih and F. Desbouvries. Kalman filtering in triplet markov chains. *IEEE Trans. on Signal Processing*, 54(8) :57–63, August 2006.
- [47] B. Ait el Fquih and F. Desbouvries. On bayesian fixed-interval smoothing algorithms. *IEEE Trans. on Automatic Control*, 53(10) :2437–2442, November 2008.

- [48] Y. Ephraim and N. Merhav. Hidden markov process. *IEEE Trans. on Information Theory*, 48(6) :1518–1569, 2002.
- [49] N. Abbassi et W. Pieczynski. Filtrage exact partiellement non supervisé dans les modèles cachés à sauts markoviens. In *GRETSI 2009*, Dijon, France, 8-11 septembre 2009.
- [50] M. A. Ferreira and V. De Oliveira. Bayesian reference analysis for gaussian markov random fields. *Journal of Multivariate Analysis*, 98 :789–812, 2007.
- [51] R. Fjørtoft, Y. Delignon, W. Pieczynski, M. Sigelle, and F. Tupin. Unsupervised segmentation of radar images using hidden markov chains and hidden markov random fields. *IEEE Trans. on Geoscience and Remote Sensing*, 3(41) :675–686, 2003.
- [52] J. Galambos and E. Seneta. Regularly varying sequences. *Proceedings of the American Mathematical Society*, 41(1) :110–116, 1973.
- [53] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions and the bayesian restoration of images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 6(6) :721–741, 1984.
- [54] J. K. Ghosh, M. Delampady, and T. Samanta. An introduction to bayesian analysis theory and methods. *Springer texts in Statistics*, 2006.
- [55] N. Giordana and W. Pieczynski. Estimation of generalized multisensor hidden markov chains and unsupervised image segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(5) :465–475, 1997.
- [56] C. W. J. Granger and R. Joyeux. An introduction to long memory time series and fractional differencing. *Journal of Time Series Analysis*, 1 :15–30, 1980.
- [57] J. Heikkinen and E. Arjas. Non-parametric bayesian estimation of a spatial poisson intensity. *Scand. Journal Statist.*, 25 :435–450, 1998.
- [58] J. R. M. Hosking. Fractional differencing. *Biometrika*, 68(1) :165–176, 1981.
- [59] J. N. Kapur. *Maximum entropy models in science and engineering*. Wiley, 1989.
- [60] J. N. Kapur and H. K. Kesavan. *Generalised maximum entropy principle (with application)*. Sanford Education Press Waterloo, Canada, 1987.
- [61] L. Knorr-Held and J. Besag. Modelling risk from a disease in time and space. *Statist. Med.*, 17 :2045–2060, 1998.
- [62] T. Koski. *Hidden Markov models for Bioinformatics*. Kluwer Academic Publishers, 2001.
- [63] S. Kullback. *Information theory and statistics*. Wiley, 1959.
- [64] P. Lanchantin. *Chaînes de Markov Triplets et Segmentation Non Supervisée de Signaux*. Mathématiques appliquées, Université de Technologie de Troyes-Institut National des Télécommunications, 2006.
- [65] P. Lanchantin, J. Lapuyade-Lahorgue, and W. Pieczynski. Unsupervised segmentation of triplet markov chains with long-memory noise. *Signal Processing*, 5(88) :1134–1151, May 2008.

-
- [66] P. Lanchantin, J. Lapuyade-Lahorgue, and W. Pieczynski. Unsupervised segmentation of randomly switching data hidden with non-gaussian correlated noise. *Signal Processing*, 91(2) :163–175, 2011.
- [67] P. Lanchantin and W. Pieczynski. Unsupervised non stationary image segmentation using triplet markov chains. In *Advanced Concepts for Intelligent Vision Systems (ACIVS 04)*, Brussels, Belgium, Aug. 31-Sept. 2004.
- [68] P. Lanchantin and W. Pieczynski. Chaînes et arbres de markov évidentiels avec applications à la segmentation des processus non stationnaires. *Traitement du Signal*, 22(1) :15–26, 2005.
- [69] P. Lanchantin and W. Pieczynski. Unsupervised restoration of hidden non stationary markov chain using evidential priors. *IEEE Trans. on Signal Processing*, 53(8) :3091–3098, 2005.
- [70] J. Lapuyade-Lahorgue. *Sur divers extensions des chaînes de Markov cachées avec applications au traitement des signaux radar*. Mathématiques appliquées, Thèse de l’Université Paris VI, 2009.
- [71] J. Lapuyade-Lahorgue and W. Pieczynski. Unsupervised segmentation of hidden semi-markov non stationary chains. In *Twenty six International Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering, MaxEnt*, Paris, France, July 2006.
- [72] J. Lapuyade-Lahorgue and W. Pieczynski. Partially markov models and unsupervised segmentation of semi-markov chains hidden with long dependence noise. In *International Symposium on Applied Stochastic Models and Data Analysis, ASMDA*, Crete, Grece, May 2007.
- [73] J. Lapuyade-Lahorgue and W. Pieczynski. Unsupervised segmentation of new semi-markov chains hidden with long depend noise. *Signal Processing*, 90(11) :2899–2910, 2010.
- [74] S. L. Lauritzen. Graphical models. *Clarendon Press Oxford*, 1996.
- [75] S. L. Lauritzen. *Complex Stochastic Systems*, chapter Causal Inference from Graphical Models. CRC Press, 2000.
- [76] B. G. Leroux. Maximum likelihood estimation for hidden markov models. *Stochastic Processes and their applications*, 40 :127–143, 1992.
- [77] N. Limnios. Estimation of the stationary distribution of semi-markov processes with borel state space. *Statistics and Probability Letters*, 76(14) :1536–1542, 2006.
- [78] N. Limnios and G. Oprisan. Semi-markov processes and reliability. Birkhäuser, 2001.
- [79] Tristan Lorino. *Eléments de statistiques - processus stochastiques*, Février 2005.
- [80] M. Ben Mabrouk and W. Pieczynski. Unsupervised segmentation of random discrete data using triplet markov chains. In *International Symposium on Applied Stochastic Models and Data Analysis, ASMDA*, Crete, Grece, May 2007.
- [81] B. B. Mandelbort and J.W. Van Ness. Fractional brownian motions, fractional noises and applicaions. *Revue of the Society of Industriel and applied Mathematics*, 10(4) :422–437, 1968.

- [82] K. V. Mardia. Multidimensional multivariate gaussian markov random fields with application to image processing. *J. Multiv. Anal.*, 24 :265–284, 1988.
- [83] N. Marhic, P. Masson, and W. Pieczynski. Mélange de lois et segmentation non supervisée des données spot. *Statistique et Analyse des Données*, 16(2) :59–81, 1991.
- [84] M. Mignotte, J. Meunier, J.P. Soucy, and C. Janicki. Comparison of deconvolution techniques using a distribution mixture parameter estimation : Application in spect imagery. *Journal of Electronic Imaging*, 11(1) :11–25, 2002.
- [85] A. Mohammed-Djafari. *Problèmes inverses en imagerie et en vision*. Lavoisier, 2009.
- [86] E. Monfrini. *Identifiabilité et Méthode des Moments dans les mélanges généralisés de distributions du système de Pearson*. Mathématiques appliquées, Thèse de l'Université Paris VI, 2002.
- [87] A. Peng and W. Pieczynski. Adaptive mixture estimation and unsupervised local bayesian image segmentation. *Graphical Models and Image Processing*, 57(5) :389–399, 1995.
- [88] P. Perez. *Modèles et algorithmes pour l'analyse probabiliste des images*. PhD thesis, Université de Renne 1, 2003.
- [89] W. Pieczynski. Statistical image segmentation. *Machine Graphics and Vision*, 1(1/2 (Proceedings of GKIP0'92)) :261–268, May 18-23 1992.
- [90] W. Pieczynski. Statistical image segmentation. *Machine Graphics and Vision*, 1(1/2) :261–268, 1992.
- [91] W. Pieczynski. Arbres de markov couple. *Comptes Rendus de l'Académie des Sciences -Mathématique, Série I*, 335(1) :79–82, 2002.
- [92] W. Pieczynski. Chaînes de markov triplet, triplet markov chains. *Comptes Rendus de l'Académie des Sciences-Mathématique*, 335(3) :275–278, 2002.
- [93] W. Pieczynski. Pairwise markov chains. *IEEE Trans. on Pattern Analysis and Machine Intellegence*, 25(5) :634–639, 2003.
- [94] W. Pieczynski. Triplet partially markov chains and trees. In *2nd International Symposium on Image/Video Communications over fixed and mobile networks (ISIVC04)*, Brest, France, 7 - 9 july 2004.
- [95] W. Pieczynski. Convergence of the iterative conditional estimation and application to mixture proportion identification. In *IEEE Statistical Signal Processing Workshop*, Madison, Wisconsin, USA, 26-29 August 2007.
- [96] W. Pieczynski. Multisensor triplet markov chains and theory of evidence. *International Journal of Approximate Reasoning*, 45(1) :1–16, 2007.
- [97] W. Pieczynski. Exact calculation of optimal filter in hidden markov switching long-memory chain. In *3rd International Conference on Mathematics and Statistics*, Athens, Greece, 15-18 June 2009.
- [98] W. Pieczynski. Exact smoothing in hidden conditionally markov switching chains. In *XIII International Conference Applied Stochastic Models and Data Analysis*, Vilnius Lithuania, June 30- July 3 2009.

-
- [99] W. Pieczynski and N. Abbassi. Exact filtering and smoothing in short or long memory stochastic switching systems. In *IEEE International Workshop on Machine Learning for Signal Processing*, Grenoble, France, September 2-4 2009.
- [100] W. Pieczynski, N. Abbassi, and M. Ben Mabrouk. Exact filtering and smoothing in markov switching systems hidden with gaussian long memory noise. In *XIII International Conference Applied Stochastic Models and Data Analysis*, Vilnius Lithuania, June 30- July 3 2009.
- [101] W. Pieczynski and F. Desbouvries. On triplet markov chains. In *International Symposium on Applied Stochastic Models and Data Analysis, (ASMDA 2005)*, Brest, France, May 2005.
- [102] W. Pieczynski and F. Desbouvries. Exact bayesian smoothing in triplet switching markov chains. In S. Co 2009, editor, *Complex data modeling and computationally intensive statistical methods for estimation and prediction*, Milan, Italy, September 14-16 2009.
- [103] W. Pieczynski, C. Hulard, and T. Veit. Triplet markov chains in hidden signal restoration. In *SPIE's International Symposium on Remote Sensing*, Crete, Greece, 22-27 September 2002.
- [104] W. Pieczynski and A. N. Tebbache. Pairwise markov random fields and its application in textured images segmentation. In *Proceedings of IEEE Southwest Symposium on Image Analysis and Interpretation (SSIAI'2000)*, pages 106–110, Austin, Texas, Etats-Unis, 2-4 April 2000.
- [105] H. E. Rauch, F. Tung, and C. T. Striebel. Maximum likelihood estimates of linear dynamic systems. *AIAA Journal*, 3(8) :1445–1450, August 1965.
- [106] H. Rue. Fast sampling of gaussian markov random fields. *J. of the Royal Statistical Society Series B*, 65 :325–338, 2001.
- [107] H. Rue, S. Martino, and N. Chopin. Approximate bayesian inference for latent gaussian models using integrated nested laplace approximations (with discussion). *J. of the Royal Statistical Society Series B*, 71 B :319–392, 2009.
- [108] H. Rue, I. Steinsland, and S. Erland. Approximating hidden gaussian markov random fields. *J. of the Royal Statistical Society Series B*, 66 :877–892, 2004.
- [109] F. Salzenstein, C. Collet, S. LeCam, and M. Hatt. Non stationary fuzzy markov chains. *Pattern Recognition Letters*, 28(16) :2201–2208, 2007.
- [110] F. Salzenstein and W. Pieczynski. Sur le choix de méthode de segmentation statistique d'images. *Traitement du Signal*, 15(2) :120–127, 1998.
- [111] G. Samorodnitsky and M. S. Taqqu. *Stable non-Gaussian Random Process*. 1994.
- [112] C. Shannon and W. Weaver. The mathematical theory of communication. *Bell System Technical Journal*, 27(379-423), 1948.
- [113] P. Smyth, D. Heckerman, and M. Jordan. Probabilistic independence networks for hidden markov probability models. *Neural Computation*, 9 :227–269, 1997.
- [114] C. Tison, J. M. Nicolas, F. Tupin, and H. Maître. A new statistical model of urban areas in high resolution sar images for markovian segmentation. *IEEE Trans. on Geoscience and Remote Sensing*, 42(10) :2046–2057, 2004.

- [115] G. C. Wei and M. A. Tanner. A monte carlo implementation of the em algorithm and the poor man's data augmentation algorithms. *Journal of the American Statistical Association*, 85 :699–704, 1990.
- [116] M. West and J. Harrison. *Bayesian Forecasting and Dynamic Models*. New York, 1997.
- [117] G. Winkler. *Image Analysis, random fields and Markov Chain Monte Carlo Methods : a mathematical introduction*. Springer, 2003.
- [118] C. F. Jeff Wu. On the convergence properties of the em algorithm. *The Annals of statistics*, 11(1), 1983.