



HAL
open science

Chaînes de Markov triplets et segmentation non supervisée d'images

Clément Fernandes

► **To cite this version:**

Clément Fernandes. Chaînes de Markov triplets et segmentation non supervisée d'images. Mathématique discrète [cs.DM]. Institut Polytechnique de Paris, 2022. Français. NNT : 2022IPPAS019 . tel-03952646

HAL Id: tel-03952646

<https://theses.hal.science/tel-03952646>

Submitted on 23 Jan 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



INSTITUT
POLYTECHNIQUE
DE PARIS

NNT : 2022IPPAS019

Thèse de doctorat



Chaînes de Markov triplets et segmentation non supervisée d'images

Thèse de doctorat de l'Institut Polytechnique de Paris
Préparée à Télécom SudParis

École doctorale n°574 mathématiques Hadamard (EDMH)
Spécialité de doctorat : Mathématiques appliquées

Thèse présentée et soutenue à Evry, le 21 Novembre 2022, par

Clément Fernandes

Composition du Jury :

Cédric RICHARD Professeur, Université Côte d'Azur (– UMR 7293)	Président
François SEPTIER Professeur, Université Bretagne Sud (– UMR 6205)	Rapporteur
Jean-Yves TOURNERET Professeur, ENSEEIHT (– UMR 5505)	Rapporteur
Florence TUPIN Professeure, Télécom Paris (– LTCI)	Examineur
Nikolaos LIMNIOS Professeur, Université de Technologie de Compiègne (– LMAC)	Examineur
Wojciech PIECZYNSKI Professeur, Télécom SudParis (– SAMOVAR)	Directeur de thèse
Mokrane ABDICHE Pilote recherche et innovation, Segula Matra Automotive	Invité

Remerciements

Je tiens d'abord à remercier grandement Wojciech PIECZYNSKI, mon directeur de thèse, pour sa grande disponibilité, son encadrement de qualité, ainsi que ses nombreux conseils qui ont été une aide très précieuse tout au long de ces années.

J'adresse ensuite mes remerciements aux membres du jury pour l'intérêt qu'ils ont porté à ce travail, ainsi que pour leurs questions et remarques pertinentes : Cédric RICHARD qui m'a fait l'honneur de présider mon jury, François SEPTIER et Jean-Yves TOURNERET qui ont accepté d'être les rapporteurs de ce travail et Florence TUPIN et Nikolaos LIMNIOS qui ont accepté d'examiner cette thèse.

Je remercie Mokrane ABDICHE, responsable administratif de cette thèse CIFRE à Segula Matra automotive, qui a su faire en sorte que celle-ci se déroule dans de bonnes conditions.

Je remercie aussi Julien FOUTH, le responsable technique de la thèse en entreprise, pour sa participation, ainsi que l'entreprise Segula Matra Automotive et l'Association Nationale de la Recherche et de la Technologie (ANRT) pour leur support financier.

J'aimerais remercier le personnel et les doctorants du laboratoire SAMOVAR, ainsi que mes collègues de Segula Matra Automotive pour leur accueil et les nombreuses discussions fructueuses sur mes sujets de recherches que j'ai pu avoir avec eux. En particulier j'aimerais exprimer ma reconnaissance et ma sympathie à Elie AZERAF.

Enfin mes derniers remerciements vont à ma famille, en particulier à mes parents Pierre et Rosenn FERNANDES qui m'ont énormément soutenu et encouragé durant ces trois années.

Sommaire

Remerciements	1
Sommaire	2
Notations	6
Introduction	10
Chapitre 1. Contexte	13
1.1 Le problème de segmentation bayésienne d'images	14
1.1.1 Position du problème	14
1.1.2 Stratégie bayésienne	14
1.1.3 Estimateurs bayésiens	15
1.2 Estimation des paramètres non supervisée	17
1.2.1 Algorithme EM	17
1.2.2 Algorithme SEM	19
1.2.3 Algorithme ICE	20
1.3 Chaînes de Markov cachées	20
1.3.1 Définition du modèle	21
1.3.2 Inférence	22
1.3.3 Estimation des paramètres non supervisée	23
1.4 Conclusion	25
Chapitre 2. Chaînes de Markov couples et triplets	27
2.1 Chaînes de Markov couples	27
2.1.1 Définition du modèle	28
2.1.2 Inférence	29
2.1.3 Estimation des paramètres non supervisée	29
2.2 Chaînes de Markov triplet	31
2.2.1 Définition du modèle	32
2.2.2 Inférence	33

Sommaire

2.2.3	Estimation des paramètres non supervisée et identifiabilité.....	33
2.3	Chaînes de Markov triplet cachées.....	35
2.3.1	Définition du modèle	35
2.3.2	Inférence	37
2.3.3	Estimation des paramètres non supervisée	37
2.4	Chaînes de Markov évidentielles	39
2.4.1	Non-stationnarité et chaînes de Markov triplets.....	39
2.4.2	Théorie de l'évidence	41
2.4.3	Définition du modèle de chaînes de Markov évidentielles.....	42
2.4.4	Inférence et estimation des paramètres non supervisée.....	44
2.5	Chaînes de semi-Markov cachées à temps discret	46
2.5.1	Chaînes de semi-Markov cachées classiques	46
2.5.2	Chaînes de semi-Markov par Markov triplet.....	47
2.5.3	Comparaison des lois de temps de séjour.....	48
2.5.4	Semi-Markov par Markov triplet et segmentation d'images.....	49
2.6	Conclusion.....	53
Chapitre 3.	Chaînes de semi-Markov évidentielles	54
3.1	Chaînes de semi-Markov évidentielles HESMC-Z.....	55
3.1.1	Définition du modèle	55
3.1.2	Inférence et estimation des paramètres.....	56
3.2	Chaînes de semi-Markov évidentielles HESMC-V	57
3.2.1	Définition du modèle	58
3.2.2	Inférence et Estimation des paramètres	60
3.3	Résultats expérimentaux	61
3.3.1	Segmentation de signaux unidimensionnels synthétiques.....	62
3.3.2	Segmentation d'images synthétiques.....	66
3.4	Comparaison avec les chaînes évidentielles semi-Markov	72
3.5	Conclusion.....	73
Chapitre 4.	Chaînes de Markov conditionnelles et balayage de Peano contextuel	75
4.1	Parcours de Peano contextuel.....	76

Sommaire

4.2	Chaînes de Markov conditionnelles	78
4.2.1	Définition du modèle	78
4.2.2	Inférence	80
4.2.3	Estimation des paramètres non supervisée	81
4.3	Résultats expérimentaux	82
4.3.1	Images synthétiques segmentées de manière supervisée.....	83
4.3.2	Images synthétiques segmentées de manière non supervisée.....	85
4.3.3	Image réelle	89
4.4	Conclusion.....	89
Chapitre 5.	Chaînes de Markov couples et triplets conditionnelles.....	90
5.1	Combiner le Peano contextuel et les chaînes de Markov triplets cachées	90
5.1.1	Définition du modèle	91
5.1.2	Inférence	92
5.1.3	Estimation des paramètres non supervisée	93
5.2	Chaînes de Markov évidentielles conditionnelles	94
5.2.1	Définition du modèle	94
5.2.2	Inférence et estimation des paramètres.....	95
5.2.3	Résultats expérimentaux.....	96
5.3	Chaînes de semi-Markov conditionnelles	100
5.3.1	Définition du modèle	100
5.3.2	Inférence et Estimation des paramètres	101
5.3.3	Résultats expérimentaux.....	102
5.4	Chaînes de Markov couples conditionnelles.....	105
5.4.1	Définition du modèle	105
5.4.2	Inférence	106
5.4.3	Estimation des paramètres non supervisée	107
5.4.4	Résultats expérimentaux.....	108
5.5	Conclusion.....	112
Conclusion et Perspectives.....		113
Annexes.....		115

Sommaire

Bibliographie.....	119
Liste des figures	129
Liste des tableaux	134

Notations

Symboles

X, U, Y	Variables aléatoires (en majuscule)
x, u, y	Réalisations respectives des variables aléatoires (en minuscules)
\mathbb{R}	Ensemble des nombres réels
\mathbb{R}^+	Ensemble des nombres réels positifs
\mathbb{N}	Ensemble des entiers naturels
$ A $	Cardinal de l'ensemble A
$p(\cdot)$	Loi de probabilité
$p(\cdot \cdot)$	Loi de probabilité conditionnelle
A/B	Ensemble A privé du sous ensemble B , $B \subset A$
$E[.]$	Espérance
$E[. \cdot]$	Espérance conditionnelle
$A \times B$	Produit cartésien des ensembles A et B
$\{1, \dots, N\}$	L'ensemble des entiers naturels allant de 1 à N
2^A	Ensemble des sous-ensembles de A
\emptyset	Ensemble vide
1_A	Fonction indicatrice de l'ensemble A
$N(\mu, \sigma^2)$	Loi normale de moyenne μ et de variance σ^2
$U([a, b])$	Loi uniforme sur le segment $[a, b]$

Notations

Acronymes

MAP	Maximum <i>A Posteriori</i>
MPM	Mode des marginales <i>a Posteriori</i> (de l'anglais : Mode of Posterior Marginals)
EMV	Estimateur du Maximum de Vraisemblance
EM	Espérance-Maximisation
SEM	Espérance-Maximisation Stochastique (de l'anglais : Stochastic Expectation-Maximisation)
ICE	Estimation conditionnelle itérative (de l'anglais : Iterative Conditional Estimation)
HMC	Chaîne de Markov Cachée (de l'anglais : Hidden Markov Chain)
PMC	Chaîne de Markov Couple (de l'anglais : Pairwise Markov Chain)
TMC	Chaîne de Markov Triplet (de l'anglais : Triplet Markov Chain)
HTMC	Chaîne de Markov Triplet Cachée (de l'anglais : Hidden Triplet Markov Chain)
HEMC	Chaîne de Markov Evidentielle Cachée (de l'anglais : Hidden Evidential Markov Chain)
HSMC	Chaîne de Semi-Markov Cachée (de l'anglais : Hidden Semi-Markov Chain)
HESMC	Chaîne de Semi-Markov Evidentielle Cachée (de l'anglais : Hidden Evidential Semi-Markov Chain)
HSEMC	Chaîne de Semi-Markov Cachée (de l'anglais : Hidden Evidential Semi-Markov Chain)
HMF	Champs de Markov Caché (de l'anglais : Hidden Markov Field)
CMC	Chaîne de Markov Conditionnelle (de l'anglais : Conditional Markov Chain)
PS	Balayage de Peano (de l'anglais : Peano Scan)
CPS	Balayage de Peano Contextuel (de l'anglais : Contextual Peano Scan)

Notations

CMC-CPS	Chaîne de Markov Conditionnelle avec Balayage de Peano Contextuel (de l'anglais : Conditional Markov Chain with Contextual Peano Scan)
HMC-PS	Chaîne de Markov Cachée avec Balayage de Peano (de l'anglais : Hidden Markov Chain with Peano Scan)
PMF	Champs de Markov Couple (de l'anglais Pairwise Markov Field)
TMF	Champs de Markov Triplet (de l'anglais Triplet Markov Field)
HEMF	Champs de Markov Evidentiel Caché (de l'anglais Hidden Evidential Markov Field)
CTMC	Chaîne de Markov Triplet Conditionnelle (de l'anglais : Conditional Triplet Markov Chain)
CTMC-CPS	Chaîne de Markov Triplet Conditionnelle avec Balayage de Peano Contextuel (de l'anglais : Conditional Triplet Markov Chain with Contextual Peano Scan)
CEMC	Chaîne de Markov Evidentielle Conditionnelle (de l'anglais : Conditional Evidential Markov Chain)
CEMC-CPS	Chaîne de Markov Evidentielle Conditionnelle avec Balayage de Peano Contextuel (de l'anglais : Conditional Evidential Markov Chain with Contextual Peano Scan)
HEMC-PS	Chaîne de Markov Evidentielle Cachée avec Balayage de Peano (de l'anglais : Hidden Evidential Markov Chain with Peano Scan)
CSMC	Chaîne de Semi-Markov Conditionnelle (de l'anglais : Conditional Semi-Markov Chain)
CSMC-CPS	Chaîne de Semi-Markov Conditionnelle avec Balayage de Peano Contextuel (de l'anglais : Conditional Semi-Markov Chain with Contextual Peano Scan)
HSMC-PS	Chaîne de Semi-Markov Cachée avec Balayage de Peano (de l'anglais : Hidden Semi-Markov Chain with Peano Scan)
CPMC	Chaîne de Markov Couple Conditionnelle (de l'anglais : Conditional Pairwise Markov Chain)
CPMC-CPS	Chaîne de Markov Couple Conditionnelle avec Balayage de Peano Contextuel (de l'anglais : Conditional Pairwise Markov Chain with Contextual Peano Scan)

Notations

PMC-PS

Chaîne de Markov Couple avec Balayage de Peano (de
l'anglais : Pairwise Markov Chain with Peano Scan)

Introduction

Les chaînes de Markov cachées (HMC) sont très utilisées pour la segmentation bayésienne non supervisée de données discrètes. On compte, parmi leurs nombreux domaines d'applications, le traitement de la parole ou des images, mais aussi l'analyse des génomes et la prédiction en économie et finances et en météorologie [1]. Leur succès tient du fait qu'elles permettent le traitement de nombreuses données liées stochastiquement de manière rapide et peu coûteuse en ressources informatiques. De plus elles sont particulièrement robustes et, malgré leur simplicité, elles sont suffisamment efficaces dans de nombreuses situations. En particulier pour la segmentation d'image, malgré leur nature unidimensionnelle, elles sont capables, grâce à une transformation des images bidimensionnelles en séquences monodimensionnelles avec le balayage de Peano (PS), de produire des résultats satisfaisants. Cependant, dans certains cas, on peut préférer des modèles plus complexes tels que les champs de Markov cachées (HMF) malgré leur plus grande complexité en temps, pour leurs meilleurs résultats. Par ailleurs les modèles markoviens (aussi bien les chaînes, que les champs) ont été étendus, dans des travaux menés au département CITI, TSP, depuis 2002, à des modèles de Markov couples (PMM) puis triplets (TMM). Ces derniers, obtenus par l'adjonction d'un – ou plusieurs – processus latent au couple (données cachées, données observées) ont donné, en certaines situations, de fortes améliorations des résultats des segmentations non supervisées d'images obtenus avec les HMM. Par exemple, lorsque le temps de séjour n'est pas géométrique, les chaînes de semi-Markov cachées (HSMC) ont tendance à être plus performantes que les HMC, et on peut dire de même pour les chaînes de Markov évidentielles cachées (HEMC) dans le cas de données non-stationnaires. Dans cette thèse, nous allons proposer des modélisations originales, généralisant les HMC, HSMC, et HEMC, et valider l'intérêt de telle modélisations à travers la segmentation d'images non supervisée. Notre travail se découpe en trois grands axes.

Notre premier objectif est de construire une nouvelle chaîne de Markov triplet, qui étend simultanément les HSMC et les HEMC. L'intérêt potentielle d'une telle modélisation est grand sachant que les HSMC et les HEMC sont déjà deux extensions des HMC ayant prouvé leur efficacité, et que les domaines d'applications des HMC sont très nombreux. En particulier, pour la segmentation d'image, la théorie de l'évidence permet une segmentation beaucoup plus fine que les HMC, et la semi-Markovianité permettrait de mieux prendre en compte dans le modèle le fait que les objets ou zones à segmenter sont quasiment toujours composés de plusieurs pixels, et que leur taille peut varier. Ainsi combiner ces deux modélisations dans ce cadre pourrait avoir un intérêt non négligeable.

Introduction

Cependant, les chaînes de Markov ont un problème pour le traitement d'images. Leur structure unidimensionnelle n'est pas adaptée aux images qui sont, elles, bidimensionnelles. On peut utiliser des techniques de balayage pour transformer un ensemble de pixels bidimensionnel en une chaîne unidimensionnelle, mais ces techniques entraînent forcément une perte d'information. Ainsi dans un deuxième temps nous avons comme objectif de proposer un balayage particulier permettant de conserver plus d'information venant du voisinage d'un pixel dans l'image. Ce balayage induit un nouveau modèle de chaîne dont seulement la loi *a posteriori* est markovienne.

Enfin, comme cela a déjà été fait pour les chaînes de Markov cachées, nous avons comme objectif final de proposer l'extension de la nouvelle chaîne évoquée dans le paragraphe précédent aux processus couples et triplets, et en particulier les triplets de semi-Markov et les triplets évidentiels.

L'organisation de cette thèse est la suivante :

Le premier chapitre sera consacré à la segmentation bayésienne d'images. Nous y présenterons dans un premier temps les hypothèses constituant le fondement du problème de segmentation bayésienne d'images, problème qui servira d'illustrations à tous nos travaux présentés dans ce document, ainsi que les estimateurs bayésiens associés. Dans un deuxième temps, nous répertorierons les méthodes générales d'estimation de paramètres en non-supervisé. Enfin nous discuterons des chaînes de Markov cachées, qui constituent la base de tous les modèles étudiés dans cette thèse.

Le deuxième chapitre portera sur les chaînes de Markov couples et triplets. Dans un premier temps, nous présenterons les PMC puis les TMC générales. Ces dernières posant un problème d'identifiabilité pour la segmentation d'images non supervisée, nous nous limiterons par la suite aux chaînes de Markov triplet cachées qui ne présentent pas cet inconvénient. Puis nous présenterons deux chaînes de Markov triplet cachées intéressantes : les chaînes de Markov évidentielles cachées et les chaînes de semi-Markov cachées.

Le troisième chapitre présentera deux nouvelles manières de combiner semi-markovianité et théorie de l'évidence dans de nouvelles chaînes de Markov triplets cachées. Il mettra ensuite en lumière l'intérêt de telles modélisations en segmentation d'images à travers des segmentations non supervisées d'images synthétiques, puis présentera une comparaison entre ces deux nouvelles chaînes et un autre triplet markovien, plus ancien, combinant lui aussi semi-markovianité et théorie de l'évidence d'une autre manière.

Le quatrième chapitre portera sur une nouvelle manière de transformer une image bidimensionnelle en chaîne unidimensionnelle pour permettre aux chaînes de Markov de segmenter une image. Ce balayage, appelé balayage de Peano contextuel, permet d'associer à chaque observation sur les pixels, deux observations voisines du pixel en question dans l'image

Introduction

mais pas dans la chaîne. Pour prendre en compte ces nouvelles observations, nous verrons ensuite que nous devons construire un nouveau modèle de chaînes : les chaînes de Markov cachées conditionnelles. Enfin nous mettrons en lumière l'intérêt de ces dernières, combinées avec notre nouveau balayage de Peano contextuel, pour la segmentation non supervisée d'images en les comparant avec les chaînes de Markov cachées classiques et les champs de Markov cachés.

Pour finir, le cinquième chapitre portera sur diverses extensions possibles des chaînes de Markov cachées conditionnelles, à savoir les chaînes de Markov évidentielles conditionnelles, les chaînes de semi-Markov conditionnelles et enfin les chaînes de Markov couples conditionnelles. Nous présenterons, pour chacune de ces extensions, leur intérêt en segmentation d'images grâce à des études de segmentations non supervisées d'images synthétiques.

Chapitre 1. Contexte

La segmentation de données est un problème clé du traitement de signal. En effet, elle peut intervenir à plusieurs niveaux dans le processus de traitement de données. Elle peut soit constituer une solution à part entière à un problème donné, soit être un prétraitement, dont le résultat servira de base pour d'autres algorithmes. Dans tous les cas la qualité de la segmentation influencera nécessairement le traitement d'un signal.

Le principe de la segmentation est de partitionner un ensemble de données en sous-ensembles selon des critères prédéfinis, ces derniers dépendant directement de la méthode de segmentation choisie. Il existe deux grandes familles de méthodes [2] : la segmentation fondée sur les régions, qui consiste à regrouper des données présentant des propriétés communes et la segmentation fondée sur les contours, où le but est de trouver des frontières délimitant des régions d'intérêt. La segmentation bayésienne s'inscrit dans la première famille [3]. Elle repose sur la modélisation probabiliste des données observées et d'un phénomène caché, ce dernier étant le résultat de la segmentation. En effet, en l'absence de lien déterministe entre le processus caché et les données observés, il convient d'adopter un lien probabiliste entre ceux-ci. Ce lien constitue notre hypothèse sur les interactions entre les deux phénomènes considérés. Cependant, celui-ci n'étant pas connu *a priori*, il faut donc être capable de l'estimer. Pour cela plusieurs options existent : on peut procéder de manière paramétrique (en se donnant une famille de loi dépendant d'un paramètre) ou non paramétrique (en ne faisant aucune hypothèse sur la forme/nature/type de loi). De plus on peut choisir une approche supervisée (en utilisant un échantillon d'apprentissage où les observations ainsi que la partition de celles-ci sont fournies) ou non supervisée (en travaillant avec seulement les observations). Dans le cadre de nos travaux, nous nous intéresserons à une méthode courante dans la segmentation d'images qui consiste à définir une loi de probabilité jointe de manière paramétrique, puis estimer les paramètres de la loi à partir des données observables seulement. Cette approche a l'avantage de ne pas nécessiter d'échantillon d'apprentissage. Une fois la loi estimée, on peut utiliser les estimateurs bayésiens pour estimer notre processus caché, à partir du processus observé. Enfin le choix de la famille de loi de probabilité est une partie importante du problème. Cette famille doit être aussi riche que possible afin de refléter la réalité que l'on souhaite modéliser et en même temps, elle doit rester assez simple pour que les calculs associés aux différentes procédures d'estimation et de segmentation bayésienne demeurent faisables. Un compromis s'impose ainsi entre richesse et simplicité lors de la définition de cette loi jointe. Vu la taille des configurations possibles des données, il est très coûteux voire impossible de calculer les

probabilités *a posteriori* requises pour les différentes procédures d'estimation. De plus l'estimation de paramètres, surtout dans le cas non-supervisé, peut impliquer, pour certaines familles de loi, des calculs extrêmement coûteux. Des hypothèses simplificatrices sont alors émises de façon à alléger les calculs tout en maintenant la loi jointe assez représentative.

Dans ce chapitre, nous présentons tout d'abord les fondements du problème de segmentation bayésienne d'images, problème qui servira d'illustrations à tous nos travaux présentés dans ce document. Dans un deuxième temps, nous présentons les méthodes générales d'estimation de paramètres en non-supervisé. Enfin la troisième partie porte sur un modèle probabiliste en particulier : les chaînes de Markov cachées, qui constitue la base de tous les travaux présentés dans cette thèse.

1.1 Le problème de segmentation bayésienne d'images

Le traitement d'images est un domaine où la segmentation est utilisée abondamment. En effet elle constitue la première étape de nombreuses applications telles que la reconnaissance d'objets, l'analyse et la compréhension d'images, le diagnostic médical [2]. Il existe deux possibilités quant à l'hypothèse du lien entre les observations et le résultat de la segmentation : on peut supposer un lien déterministe entre eux, comme dans les méthodes de seuillage par exemple, ou un lien probabiliste. Nous présentons ici une approche bayésienne, fondée sur ce deuxième choix.

1.1.1 Position du problème

On considère la situation suivante : nous observons une image (en niveaux de gris où en couleurs) et nous voulons classer les pixels de celle-ci en plusieurs groupes. Pour modéliser le problème on considère donc deux processus aléatoires $X = (X_s)_{s \in S}$, les groupes considérés et $Y = (Y_s)_{s \in S}$, les pixels observés. Pour tout $s \in S$, X_s prend ses valeurs dans l'espace fini des classes $\Omega = \{\omega_1, \dots, \omega_k\}$ et Y_s dans \mathbb{R} . Le problème de la segmentation est alors celui de l'estimation de $X = x = (x_s)_{s \in S}$ à partir de l'observation $Y = y = (y_s)_{s \in S}$. On considère alors une loi de probabilité sur $\Omega^{|S|} \times \mathbb{R}^{|S|}$, définie par la probabilité $p(x)$ sur $\Omega^{|S|}$, dite loi *a priori*, et une densité conditionnelle $p(y|x)$ sur $\mathbb{R}^{|S|}$. Soit $p(x, y) = p(x)p(y|x)$. Pour calculer la loi *a posteriori* $p(x|y)$ qui va nous intéresser dans ce cas, il suffit d'appliquer la règle de Bayes :

$$p(x|y) = \frac{p(x, y)}{\sum_{x' \in \Omega^{|S|}} p(x', y)} = \frac{p(x)p(y|x)}{\sum_{x' \in \Omega^{|S|}} p(x')p(y|x')} \quad (1.1)$$

1.1.2 Stratégie bayésienne

Reprenons notre loi de probabilité du couple (X, Y) à valeur dans $\Omega^{|S|} \times \mathbb{R}^{|S|}$. Pour $(x, y) \in \Omega^{|S|} \times \mathbb{R}^{|S|}$, réalisation de (X, Y) , le problème de la segmentation bayésienne est celui de l'estimation de la réalisation inobservable x , à partir de la réalisation observable y . Une

Contexte

stratégie de classification \hat{s} associée à toute observation y une estimation $\hat{s}(y) = \hat{x}$ de la réalisation x . Enfin il faut choisir une fonction de perte $L: \Omega^{|\mathcal{S}|} \times \Omega^{|\mathcal{S}|} \rightarrow \mathbb{R}^+$. Celle-ci mesure l'impact d'une erreur de prédiction et dépend donc du problème rencontré, cependant L doit vérifier :

$$L(x', x) = \begin{cases} 0 & \text{si } x' = x \\ \lambda_{x', x} & \text{sinon} \end{cases} \quad (1.2)$$

La perte moyenne associée à une stratégie est $E[L(\hat{s}(Y), X)]$. Par définition la stratégie bayésienne notée \hat{s}_B est la stratégie qui minimise la perte moyenne. Soit :

$$E[L(\hat{s}_B(Y), X)] = \min_{\hat{s}} E[L(\hat{s}(Y), X)] \quad (1.3)$$

Pour notre observation y , l'estimateur bayésien associé à la fonction de perte L est donc :

$$\hat{x} = \operatorname{argmin}_{x' \in \Omega^{|\mathcal{S}|}} \sum_{x \in \Omega^{|\mathcal{S}|}} L(x', x) p(x|y) \quad (1.4)$$

1.1.3 Estimateurs bayésiens

Un estimateur bayésien est, comme rappelé dans la partie précédente, le résultat d'une stratégie bayésienne. Il dépend donc de la fonction de perte choisie. Nous allons présenter ici les estimateurs associés aux fonctions de perte classiques dans le cadre de la segmentation bayésienne. Nous partirons de la fonction de perte courante $L_{0/1}$ définie telle que :

$$\forall (\omega_i, \omega_j) \in \Omega^2, L_{0/1}(\omega_i, \omega_j) = \begin{cases} 0 & \text{si } \omega_i = \omega_j \\ 1 & \text{sinon} \end{cases} \quad (1.5)$$

1.1.3.1 Estimateur du Maximum a Posteriori

Le premier estimateur est l'estimateur du maximum *a posteriori* (MAP) qui est associé à la fonction de perte :

$$L_{MAP}(x', x) = L_{0/1}(x', x) \quad (1.6)$$

C'est une fonction de perte « tout ou rien » qui va pénaliser indifféremment une erreur de prédiction sur un seul site ou plusieurs sites. Autrement dit, la perte obtenue par cette fonction est la même que l'on se trompe sur un seul site ou tous les sites. L'estimateur du MAP associé à cette fonction de perte est donc :

$$\begin{aligned} \hat{x}_{MAP} &= \operatorname{argmin}_{x' \in \Omega^{|\mathcal{S}|}} \sum_{x \in \Omega^{|\mathcal{S}|}} L_{MAP}(x', x) p(x|y) \\ &= \operatorname{argmin}_{x' \in \Omega^{|\mathcal{S}|}} \sum_{x \in \Omega^{|\mathcal{S}|} \setminus \{x'\}} p(x|y) \\ &= \operatorname{argmin}_{x' \in \Omega^{|\mathcal{S}|}} 1 - p(x'|y) \end{aligned} \quad (1.7)$$

Contexte

$$= \operatorname{argmax}_{x' \in \Omega^{|S|}} p(x'|y)$$

L'estimateur MAP maximise donc directement la probabilité a posteriori $p(x|y)$ et permet donc de retrouver la séquence x qui a le plus de chance d'avoir généré l'observation y .

On peut remarquer que la fonction de perte associé au MAP est assez sévère, ce qui fait qu'elle n'est pas adaptée à tous les problèmes. En revanche elle est efficace quand le caractère séquentiel du résultat est important, par exemple dans les problèmes liés au traitement du langage naturel.

1.1.3.2 Estimateur du Mode des marginales a Posteriori

Le deuxième estimateur est celui du mode des marginales *a posteriori* (MPM) qui est associé à la fonction de perte :

$$L_{MPM}(x', x) = \sum_s L_{0/1}(x'_s, x_s) \quad (1.8)$$

Contrairement au MAP, la fonction de perte pour le MPM est la somme des erreurs sur chaque site, elle augmente donc en fonction du nombre de sites où l'on s'est trompé. L'estimateur du MPM associé à cette fonction de perte est donc :

$$\begin{aligned} \forall s \in S, \hat{x}_{MPM} &= \operatorname{argmin}_{x' \in \Omega^{|S|}} \sum_{x \in \Omega^{|S|}} L_{MPM}(x', x) p(x|y) \\ &= \operatorname{argmin}_{x' \in \Omega^{|S|}} \sum_{x \in \Omega^{|S|}} \sum_s L_{0/1}(x'_s, x_s) p(x|y) \\ &= \operatorname{argmin}_{x' \in \Omega^{|S|}} \sum_s \sum_{x_s \in \Omega / \{x'_s\}} p(x_s|y) \\ &= \operatorname{argmin}_{x' \in \Omega^{|S|}} \sum_s 1 - p(x'_s|y) \\ &= \operatorname{argmax}_{x' \in \Omega^{|S|}} \sum_s p(x'_s|y) \end{aligned} \quad (1.9)$$

L'estimateur MPM maximise donc les probabilités marginales *a posteriori* $p(x_s|y)$ et permet donc de minimiser le nombre moyen de sites erronés.

Le MPM est bien plus souple que le MAP. En particulier, en segmentation d'images, la performance d'un modèle est souvent mesurée grâce à un taux d'erreur qui se calcule en faisant la somme (ou la moyenne) des erreurs de prédictions pour chaque pixel [2]. Il se trouve que c'est exactement ce que minimise le MPM, c'est pourquoi dans la suite nous utiliserons l'estimateur du MPM.

1.2 Estimation des paramètres non supervisée

La solution apportée par les estimateurs bayésien au problème de segmentation d'images est efficace à condition que l'on connaisse la loi jointe reliant les données observées et les groupes dans lesquels on veut classer celles-ci. En pratique c'est rarement le cas, il nous faut donc être capable d'estimer cette loi. Cette estimation peut se faire de manière supervisée, sur un jeu de données d'apprentissage contenant les données observées et les données cachées associées, ou de manière non-supervisée, c'est-à-dire seulement à partir des observations seules. Dans la suite, nous discutons surtout de l'estimateur du maximum de vraisemblance, qui revient à maximiser, par rapport aux paramètres, la probabilité d'apparition des données dont nous disposons (dans le cas supervisé, on maximisera donc la loi jointe du couple données observées et données cachées, alors que dans le cas non supervisé, on maximisera la loi marginale des observations). Nous avons choisi de nous intéresser à la segmentation d'images à partir des données observées seulement, dans une approche paramétrique (donc avec une famille de lois dépendant d'un vecteur de paramètres). Il faut remarquer que l'on ne peut pas toujours atteindre le maximum de vraisemblance, cependant il existe des méthodes itératives d'approximation. Ainsi nous présentons dans cette partie trois algorithmes populaires dans la littérature, permettant d'estimer les paramètres de la loi de manière non supervisée. Le premier est l'algorithme Expectation-Maximisation (EM) [4], puis sa version stochastique : Stochastic Expectation-Maximisation (SEM) [5], [6], [7] et enfin l'estimation conditionnelle itérative (ICE) [8] [9] [10].

1.2.1 Algorithme EM

L'algorithme EM (Expectation Maximisation) repose sur l'estimateur du maximum de vraisemblance (EMV). Soit une famille de lois pour le couple (X, Y) à valeur dans $\Omega^{|\mathcal{S}|} \times \mathbb{R}^{|\mathcal{S}|}$, indexée par un vecteur de paramètres $\theta \in \Theta$. L'EMV est l'estimateur $\hat{\theta}$ de θ vérifiant :

$$\hat{\theta} = \operatorname{argmax}_{\theta} p(x, y|\theta) \quad (1.10)$$

En général, on travaille plutôt avec la log-vraisemblance, et (1.10) devient :

$$\hat{\theta} = \operatorname{argmax}_{\theta} \log(p(x, y|\theta)) \quad (1.11)$$

Dans le cas de données partiellement observées, l'EMV n'est généralement pas calculable analytiquement. En effet supposons que l'on observe que $Y = y$, X étant à présent un processus latent, (1.11) devient :

$$\hat{\theta} = \operatorname{argmax}_{\theta} \log(p(y|\theta)) = \operatorname{argmax}_{\theta} \log\left(\sum_x p(x, y|\theta)\right) \quad (1.12)$$

La somme sur x dans le logarithme de pose des problèmes pour le calcul analytique de $\hat{\theta}$.

Contexte

L'algorithme EM propose une solution pour approcher de manière itérative l'EMV dans le cas de données incomplètes en partant d'un jeu de paramètres initiaux θ^0 et en construisant, à chaque itération $q + 1$, θ^{q+1} à partir de θ^q [4]. Il repose sur deux idées. La première est qu'il est généralement plus facile de calculer l'EMV sur la loi du couple que sur les y seuls. On peut ensuite exprimer la vraisemblance sur les y en fonction de cette vraisemblance jointe de la manière suivante :

$$\log(p(y|\theta)) = \log(p(x, y|\theta) - \log(p(x|y, \theta)) \quad (1.13)$$

La deuxième idée est que la maximisation de l'espérance de la vraisemblance jointe conditionnée par les observations y et $\theta^q \in \Theta$ un paramètre quelconque obtenu à l'itération q et *a priori* différent de θ , à savoir $E[\log(p(x, y|\theta)|Y, \theta^q)]$, par rapport à θ , entraîne une augmentation de $\log(p(y|\theta))$. En effet :

$$\log(p(y|\theta)) - \log(p(y|\theta^q)) = \log(p(x, y|\theta) - \log(p(x, y|\theta^q) + \log \frac{p(x|y, \theta^q)}{p(x|y, \theta)} \quad (1.14)$$

Prenons à présent l'espérance de (1.14) conditionnée par les observations y et θ^q . On obtient alors :

$$\begin{aligned} & \log(p(y|\theta)) - \log(p(y|\theta^q)) \\ &= E[\log(p(x, y|\theta)|Y, \theta^q) - E[\log(p(x, y|\theta^q)|Y, \theta^q)] + E\left[\log \frac{p(x|y, \theta^q)}{p(x|y, \theta)} \middle| Y, \theta^q\right] \end{aligned} \quad (1.15)$$

Enfin $E\left[\log \frac{p(x|y, \theta^q)}{p(x|y, \theta)} \middle| Y, \theta^q\right] > 0$ car c'est la divergence de Kullback-Leibler entre $p(x|y, \theta^q)$ et $p(x|y, \theta)$. Ainsi choisir θ qui maximise $E[\log(p(x, y|\theta)|Y, \theta^q)]$ a pour effet d'augmenter $\log(p(y|\theta))$.

L'algorithme EM est finalement un algorithme itératif, où pour chaque itération q , on va effectuer deux étapes [4]:

- La première étape (l'étape Expectation) est le calcul de l'espérance $E[\log(p(x, y|\theta)|Y, \theta^q)]$
- La deuxième étape (l'étape Maximisation) est la maximisation de $E[\log(p(x, y|\theta)|Y, \theta^q)]$ par rapport à θ :

$$\theta^{q+1} = \underset{\theta}{\operatorname{argmax}} E[\log(p(x, y|\theta)|Y, \theta^q)]$$

En pratique on arrête l'algorithme EM après un nombre d'itérations jugé suffisant, ou lorsque $\log(p(y|\theta^{q+1})) - \log(p(y|\theta^q)) < \tau$, τ étant fixé au préalable.

Cet algorithme assure donc la croissance de $\log(p(y|\theta))$ tout au long de son déroulement. Cependant il ne garantit pas de trouver le maximum de $\log(p(y|\theta))$ dans le cas général. En particulier, l'algorithme est susceptible de tomber dans un maximum local. Cela rend aussi

l'algorithme particulièrement sensible à son initialisation θ^0 . En effet, le calcul de la suite des θ^q étant déterministe, une fois θ^0 fixé, le fait de tomber ou non dans un maximum local dépend très largement du θ^0 choisi. Enfin dans certain cas, il est compliqué d'évaluer où de maximiser le terme $E[\log(p(x, y|\theta)|Y, \theta^q)]$. C'est pourquoi d'autres alternatives ont été développées pour pallier ces défauts.

1.2.2 Algorithme SEM

L'algorithme SEM (Stochastic Expectation Maximisation) est une alternative stochastique à l'algorithme EM [5], [6], [7]. Il repose sur la même idée principale que ce dernier, à savoir qu'il est généralement plus facile de calculer l'EMV sur la loi du couple que sur les y seuls. Cependant, au lieu de maximiser $E[\log(p(x, y|\theta)|Y, \theta^q)]$ par rapport à θ à chaque itération, le SEM va proposer une approche itérative stochastique pour maximiser $\log(p(y|\theta))$ qui ne nécessite pas le calcul de $E[\log(p(x, y|\theta)|Y, \theta^q)]$.

Pour effectuer le SEM, les conditions préalables sont les suivantes : avoir un moyen de générer un x selon la loi *a posteriori* $p(x|y, \theta)$ et être capable de calculer la quantité $\operatorname{argmax}_{\theta} \log(p(x, y|\theta))$, EMV sur les données complètes. Une fois ces conditions remplies, l'algorithme se déroule de la manière suivante :

- On se donne une initialisation θ^0
- Pour chaque itération q on réalise les deux étapes suivantes :
 - On génère x^q selon la loi $p(x|y, \theta^q)$
 - On calcule $\theta^{q+1} = \operatorname{argmax}_{\theta} (\log(p(x^q, y|\theta)))$

En pratique on arrête l'algorithme SEM après un nombre d'itérations jugé suffisant, ou lorsque $|\log(p(y|\theta^{q+1})) - \log(p(y|\theta^q))| < \tau$, τ étant fixé au préalable.

Cet algorithme ne garantit pas de trouver le maximum de $\log(p(y|\theta))$ dans le cas général. De plus, il est possible que $\log(p(y|\theta^{q+1})) < \log(p(y|\theta^q))$ pour un q donné. Cependant son principal avantage par rapport à l'algorithme EM est que grâce au caractère aléatoire de la mise à jour du paramètre θ à chaque itération, celui-ci peut éviter de rester bloqué dans un maximum local de la fonction de vraisemblance. De plus, le SEM présente comme avantage de ne pas avoir à faire le calcul de la quantité $\operatorname{argmax}_{\theta} E[\log(p(x, y|\theta)|Y, \theta^q)]$ qui peut poser problème dans certain cas, et ne nécessite que la possibilité de simuler selon la loi $p(x|y, \theta^q)$ et le calcul de $\theta^{q+1} = \operatorname{argmax}_{\theta} (\log(p(x^q, y, \theta)))$, ce qui peut être plus simple et plus rapide pour certaines familles de lois. Cependant la nature stochastique de SEM peut entraîner une convergence plus lente de l'algorithme par rapport à EM dans certains cas.

1.2.3 Algorithme ICE

L'algorithme ICE repose sur une idée différente des deux précédents algorithmes. En effet il ne fait pas intervenir la notion d'EMV [8] [9] [10]. Soit une famille de lois pour le couple (X, Y) à valeur dans $\Omega^{|S|} \times R^{|S|}$, indexée par un vecteur de paramètres $\theta \in \Theta$. Nous recherchons $\hat{\theta} = \hat{\theta}(X, Y)$. Nous savons que la meilleure approximation, au sens de l'erreur quadratique moyenne, est l'espérance conditionnelle. Ainsi si l'on observe Y seulement, l'idée est de chercher à calculer $E[\hat{\theta}|Y]$. Le problème est que $E[\hat{\theta}|Y]$ dépend du paramètre θ , qui nous est inconnu. On va alors, pour approcher $E[\hat{\theta}|Y]$, construire une méthode itérative où pour chaque itération q , nous allons calculer $\theta^{q+1} = E[\hat{\theta}|Y, \theta = \theta^q]$. Enfin si la quantité $E[\hat{\theta}|Y, \theta = \theta^q]$ n'est pas directement calculable, mais que l'on possède un moyen de simuler X selon la loi $p(x|y, \theta^q)$, on peut utiliser la loi des grands nombres pour approcher $E[\hat{\theta}|Y, \theta = \theta^q]$ de la manière suivante : $\theta^{q+1} = \frac{1}{N}(\hat{\theta}(x_1^{(q)}, y) + \dots + \hat{\theta}(x_N^{(q)}, y))$ avec $x_1^{(q)}, \dots, x_N^{(q)}$ N simulations de X par $p(x|y, \theta^q)$.

Finalement, en prenant un paramètre θ à plusieurs valeurs, se décomposant de la manière suivante : $\theta = (\theta_1, \theta_2)$, de telle sorte que $E[\hat{\theta}_1|Y, \theta = \theta^q]$ soit calculable analytiquement, mais $E[\hat{\theta}_2|Y, \theta = \theta^q]$ ne le soit pas, on arrive à la démarche itérative suivante :

- On se donne une initialisation θ^0
- Pour chaque itération q on réalise les quatre étapes suivantes :
 - On calcule $\theta_1^{q+1} = E[\hat{\theta}_1|Y, \theta = \theta^q]$
 - On génère N simulations $x_1^{(q)}, \dots, x_N^{(q)}$, selon la loi $p(x|y, \theta^q)$
 - On calcule $\theta_2^{q+1} = \frac{1}{N}(\hat{\theta}_2(x_1^{(q)}, y) + \dots + \hat{\theta}_2(x_N^{(q)}, y))$
 - $\theta^{q+1} = (\theta_1^{q+1}, \theta_2^{q+1})$

En pratique on arrête l'algorithme ICE après un nombre d'itérations jugé suffisant, ou lorsque $|\log(p(y|\theta^{q+1})) - \log(p(y|\theta^q))| < \tau$, τ étant fixé au préalable.

On obtient finalement un algorithme partiellement stochastique (s'il existe des composantes θ_2 de θ telles que $E[\hat{\theta}_2|Y, \theta = \theta^q]$ ne soit pas calculable), dont on connaît certaines garanties sur sa convergence [9] [11].

1.3 Chaînes de Markov cachées

Une fois le problème de segmentation bayésienne posé, et les méthodes d'estimation des paramètres mentionnées, il faut s'intéresser à la famille de lois que l'on va considérer. Celle-ci doit permettre d'effectuer les calculs présentés dans les sections précédentes de manière efficace et rapide. En particulier, le modèle de chaînes de Markov cachées est compatible avec la segmentation bayésienne évoquée plus tôt car l'estimateur du MPM (et MAP) est simple à

calculer et l'estimation des paramètres de manière non-supervisée est faisable via les algorithmes EM, SEM et ICE mentionnés dans la section précédente. Malgré leur simplicité, les modélisations par chaînes de Markov cachées permettent d'aborder de nombreux problèmes et se montrent particulièrement robustes et suffisamment efficaces dans un grand nombre de situations. Par exemple, elles ont été utilisées avec succès dans les domaines de traitement de la parole [12], d'analyse d'images [13] [14] [15] [16], de reconnaissance d'écriture manuscrite [17], d'analyse de la structure du génome [18], ou encore pour la modélisation de points imprimés microscopiques pour l'identification de la source d'impression d'un document [19].

1.3.1 Définition du modèle

Soit $X = (X_1, \dots, X_N)$ un processus caché qui prend ses valeurs à partir d'un ensemble fini de classes $\Omega = \{\omega_1, \dots, \omega_K\}$ et qui doit être estimé à partir d'un processus observable $Y = (Y_1, \dots, Y_N)$ qui prend ses valeurs dans \mathbb{R} . Le processus couple (X, Y) est dit chaîne de Markov cachée (HMC) si X est une chaîne de Markov et si les Y_1, \dots, Y_N sont indépendants conditionnellement à X .

La loi de la chaîne de Markov X , dite loi a priori, s'écrit alors de la manière suivante :

$$p(x) = p(x_1) \prod_{n=1}^{N-1} p(x_{n+1}|x_n) \quad (1.15)$$

Le processus Y observable est considéré comme une version bruitée du processus X . Dans une chaîne de Markov cachée, les Y sont supposés indépendants entre eux conditionnellement à X , autrement dit :

$$p(y|x) = p(y_1|x_1) \prod_{n=1}^{N-1} p(y_{n+1}|x_{n+1}) \quad (1.16)$$

La loi $p(y|x)$ est généralement désignée comme la distribution du bruit. Si pour tous n les lois $p(y_n|x_n)$ sont gaussiennes, on parle de HMC à bruit gaussien.

Finalement, la loi du couple (X, Y) s'écrit de la manière suivante :

$$p(x, y) = p(x_1)p(y_1|x_1) \prod_{n=1}^{N-1} p(x_{n+1}|x_n)p(y_{n+1}|x_{n+1}) \quad (1.17)$$

Son graphe des dépendances de probabilité est présenté sur la **Figure 1**.

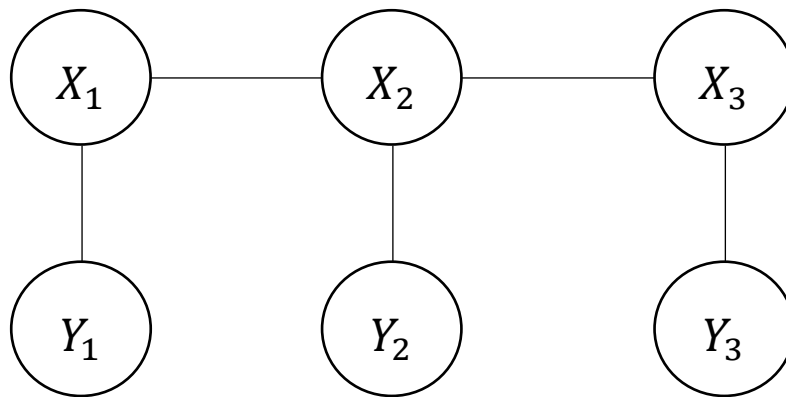


Figure 1. Graphe de probabilité d'une chaîne de Markov cachée pour $N=3$

Une HMC est dite homogène si les transitions $p(x_{n+1}|x_n)$ ne dépendent pas de n . On peut alors définir toute la chaîne à l'aide d'une matrice de transition de taille $K \times K$ pour $p(x_{n+1}|x_n)$, K moyennes et K variances pour $p(y_{n+1}|x_{n+1})$ et enfin K paramètres pour $p(x_1)$ et K moyennes et K variances pour $p(y_1|x_1)$. Il faut noter que dans une chaîne de Markov homogène, les distributions marginales $p(x_n, y_n)$ peuvent varier en fonction de n . Si, en plus, $p(x_n, y_n)$ ne dépend pas de n , on parle de HMC stationnaires. Dans ces dernières, la loi est complètement définie par $p(x_1, x_2, y_1, y_2)$, égal à $p(x_n, x_{n+1}, y_n, y_{n+1})$ pour tout $n = 2 \dots N - 1$. Il suffit alors de définir une matrice de taille $K \times K$ pour $p(x_1, x_2)$, en s'assurant que les marginales $p(x_1)$ et $p(x_2)$ soient identiques, K moyennes et K variances pour $p(y_n|x_n)$. Ces propriétés sont importantes pour la suite. En effet elles permettent la définition d'une chaîne de Markov cachées à partir d'un petit nombre de paramètres, et elles sont essentielles à l'estimation des paramètres si l'on ne possède qu'une seule séquence.

1.3.2 Inférence

Les chaînes de Markov cachées gaussiennes sont compatibles avec l'estimation bayésienne car les calculs liés aux estimateurs bayésiens sont faciles et rapides. En particulier, l'estimateur du MPM, donné par :

$$[\hat{s}_{MPM}(y) = \hat{x}] \Leftrightarrow \left[\forall n \in \{1, \dots, N\}, \quad p(\hat{x}_n|y) = \max_{x_n \in \Omega} p(x_n|y) \right] \quad (1.18)$$

est calculable dès qu'on sait évaluer les lois $p(x_n|y)$. Dans le cas d'une chaîne de Markov cachée ce calcul est faisable avec une complexité en temps de $O(K^2N)$ si on suppose une complexité en espace de $O(KN)$. Ce calcul s'effectue de la manière suivante :

- i. Calculer les probabilités « forward » $\alpha_n(x_n, u_n) = p(x_n, y_1, \dots, y_n)$ avec :
$$\alpha_1(x_1) = p(x_1, y_1)$$

$$\alpha_{n+1}(x_{n+1}) = \sum_{x_n \in \Omega} \alpha_n(x_n) p(x_{n+1}|x_n) p(y_{n+1}|x_{n+1}) \quad (1.19)$$

- ii. Calculer les probabilités « backward » $\beta_n(x_n) = p(y_{n+1}, \dots, y_N | x_n)$ avec :

$$\beta_N(x_N) = 1$$

$$\beta_n(x_n) = \sum_{x_{n+1} \in \Omega} \beta_{n+1}(x_{n+1}) p(x_{n+1}|x_n) p(y_{n+1}|x_{n+1}) \quad (1.20)$$

- iii. Calculer $p(x_n|y)$ avec :

$$p(x_n|y) = \frac{\alpha_n(x_n) \beta_n(x_n)}{\sum_{x_n \in \Omega} \alpha_n(x_n) \beta_n(x_n)} \quad (1.21)$$

Il est aussi possible de calculer l'estimateur MAP dans une HMC, avec l'algorithme de Viterbi [20] [21] avec la même complexité en temps de $O(K^2N)$, toujours si on suppose une complexité en espace de $O(KN)$. Cependant, comme nous nous intéressons au MPM, nous ne détaillerons pas la mise en œuvre de cet algorithme.

1.3.3 Estimation des paramètres non supervisée

Dans le cadre de la segmentation d'images non supervisée à laquelle nous nous intéressons, il faut s'assurer que les paramètres de la famille de modèles choisie puissent bien être estimés à partir des observations $Y = y$ seulement. C'est le cas pour les HMC gaussiennes, en effet il existe des versions de l'algorithme EM [22], SEM [15] et ICE [8] que nous allons présenter ci-dessous. Si on s'intéresse à la complexité informatique de ces méthodes, il est impossible de la donner pour les algorithmes complets car la convergence de ceux-ci n'est pas garantie. Cependant la complexité en temps d'une itération de EM, SEM ou ICE est la même. Elle est de $O(K^2N)$ en temps si on suppose une complexité en espace de $O(KN)$.

1.3.3.1 Algorithme EM pour les chaînes de Markov

Soit une famille de HMC (X, Y) homogène à bruit gaussien, indexée par un paramètre θ . Ce dernier est défini par $\pi_k = p(x_1 = \omega_k)$, $p_{kl} = p(x_2 = \omega_l | x_1 = \omega_k)$ et les moyennes et variances μ_k, σ_k^2 associées aux probabilités $p(y_n | x_n = \omega_k)$. Notons :

$$p_{k,n} = p(x_n = \omega_k, | y) \quad (1.22)$$

$$p_{kl,n,n+1} = p(x_{n+1} = \omega_l, x_n = \omega_k | y) \quad (1.23)$$

$$p_{kl,n} = p(x_{n+1} = \omega_l | x_n = \omega_k, y) \quad (1.24)$$

On peut montrer que les $p_{kl,n,n+1}$ sont données par :

$$p_{kl,n,n+1} = \frac{\alpha_n(x_n) p(x_{n+1}|x_n) p(y_{n+1}|x_{n+1}) \beta_{n+1}(x_{n+1})}{\sum_{x_n \in \Omega} \alpha_n(x_n) p(x_{n+1}|x_n) p(y_{n+1}|x_{n+1}) \beta_{n+1}(x_{n+1})} \quad (1.25)$$

L'algorithme EM pour les HMC se déroule comme suit [22] :

Contexte

- On initialise l'algorithme avec $\theta = \theta^0$
- Pour chaque itération q , on utilise θ^q pour calculer $p_{k,n}^{(q)}$ avec (1.21) et $p_{kl,n,n+1}^{(q)}$ avec (1.25). Ensuite on calcule θ^{q+1} de la manière suivante :

$$\pi_k^{(q+1)} = \frac{\sum_{n=1}^N p_{k,n}^{(q)}}{N} \quad (1.26)$$

$$p_{kl}^{(q+1)} = \frac{\sum_{n=1}^{N-1} p_{kl,n,n+1}^{(q)}}{\sum_{n=1}^{N-1} p_{k,n}^{(q)}} \quad (1.27)$$

$$\mu_k^{(q+1)} = \frac{\sum_{n=1}^N p_{k,n}^{(q)} y_n}{\sum_{n=1}^N p_{k,n}^{(q)}} \quad (1.28)$$

$$\sigma_k^{2,(q+1)} = \frac{\sum_{n=1}^N p_{k,n}^{(q)} (y_n - \mu_k^{(q+1)})^2}{\sum_{n=1}^N p_{k,n}^{(q)}} \quad (1.29)$$

1.3.3.2 Algorithme SEM pour les chaînes de Markov

Reprenons la famille de HMC définie dans la partie précédente. La loi a posteriori d'une chaîne de Markov cachée est markovienne :

$$p(x|y) = p(x_1|y) \prod_{n=1}^{N-1} p(x_{n+1}|x_n, y), \quad (1.30)$$

où $p(x_1|y)$ est calculé avec (1.21), et les $p(x_{n+1}|x_n, y)$ s'expriment en fonction des probabilités « backward » :

$$p(x_{n+1}|x_n, y) = \frac{p(x_{n+1}|x_n) p(y_{n+1}|x_{n+1}) \beta_{n+1}(x_{n+1})}{\beta_n(x_n)} \quad (1.31)$$

L'algorithme SEM pour les HMC se déroule comme suit [15] :

- On initialise l'algorithme avec $\theta = \theta^0$
- Pour chaque itération q , on utilise θ^q pour simuler x^q selon (1.30). Ensuite on calcule θ^{q+1} de la manière suivante :

$$\pi_k^{(q+1)} = \frac{\sum_{n=1}^N 1_{\{x_n^{(q)} = \omega_k\}}}{N} \quad (1.32)$$

$$p_{kl}^{(q+1)} = \frac{\sum_{n=1}^{N-1} 1_{\{x_n^{(q)} = \omega_k, x_{n+1}^{(q)} = \omega_l\}}}{\sum_{n=1}^{N-1} 1_{\{x_n^{(q)} = \omega_k\}}} \quad (1.33)$$

$$\mu_k^{(q+1)} = \frac{\sum_{n=1}^N \mathbf{1}_{\{x_n^{(q)} = \omega_k\}} y_n}{\sum_{n=1}^N \mathbf{1}_{\{x_n^{(q)} = \omega_k\}}} \quad (1.34)$$

$$\sigma_k^{2,(q+1)} = \frac{\sum_{n=1}^N \mathbf{1}_{\{x_n^{(q)} = \omega_k\}} (y_n - \mu_k^{(q+1)})^2}{\sum_{n=1}^N \mathbf{1}_{\{x_n^{(q)} = \omega_k\}}} \quad (1.35)$$

1.3.3.3 Algorithme ICE pour les chaînes de Markov

L'algorithme ICE peut être appliqué dans le cadre des HMC. En effet, en posant $\theta \in \Theta$, $\theta = (\theta_1, \theta_2)$, avec $\theta_1 = (\{\pi_k\}_{1 \leq k \leq K}, \{p_{kl}\}_{1 \leq k \leq K, 1 \leq l \leq K})$ et $\theta_2 = (\{\mu_k\}_{1 \leq k \leq K}, \{\sigma_k^2\}_{1 \leq k \leq K})$, $\forall \theta^q \in \Theta$, $E[\widehat{\theta}_1 | Y, \theta = \theta^q]$ est calculable à l'aide de (1.21) et (1.25), et on peut simuler $x^{1,q}, \dots, x^{R,q}$ selon (1.30) pour approcher $E[\widehat{\theta}_2 | Y, \theta = \theta^q]$

Cet algorithme se déroule comme ceci [8]:

- On initialise l'algorithme avec $\theta = \theta^0$
- Pour chaque itération q , on utilise θ^q pour calculer $p_{k,n}^{(q)}$ avec (1.21) et $p_{kl,n,n+1}^{(q)}$ avec (1.25) et simuler $x^{1,q}, \dots, x^{R,q}$ selon (1.30). Ensuite on calcule θ^{q+1} de la manière suivante :

$$\pi_k^{(q+1)} = \frac{p_{k,1}^{(q)} + \dots + p_{k,N}^{(q)}}{N} \quad (1.36)$$

$$p_{kl}^{(q+1)} = \frac{p_{kl,1,2}^{(q)} + \dots + p_{kl,N-1,N}^{(q)}}{p_{k,1}^{(q)} + \dots + p_{k,N}^{(q)}} \quad (1.37)$$

$$\mu_k^{(q+1)} = \frac{\sum_{r=1}^R \sum_{n=1}^N \mathbf{1}_{\{x_n^{r,(q)} = \omega_k\}} y_n}{\sum_{r=1}^R \sum_{n=1}^N \mathbf{1}_{\{x_n^{r,(q)} = \omega_k\}}} \quad (1.38)$$

$$\sigma_k^{2,(q+1)} = \frac{\sum_{r=1}^R \sum_{n=1}^N \mathbf{1}_{\{x_n^{r,(q)} = \omega_k\}} (y_n - \mu_k^{(q+1)})^2}{\sum_{r=1}^R \sum_{n=1}^N \mathbf{1}_{\{x_n^{r,(q)} = \omega_k\}}} \quad (1.39)$$

1.4 Conclusion

Dans ce chapitre, nous avons présenté tous les fondements sur lesquels se basent nos travaux. Tout d'abord, nous avons montré comment le problème de segmentation d'images peut être résolu par une approche bayésienne. Nous avons ensuite exposé les stratégies bayésiennes associées à deux fonctions de perte courantes (MAP et MPM) en montrant comment celles-ci pouvaient proposer une réponse optimale au problème de segmentation. Puis nous avons passé en revue différents algorithmes d'estimation des paramètres en non-supervisé. Enfin nous avons présenté un modèle très classique en segmentation bayésienne, les chaînes de Markov cachées,

Contexte

qui de par leur simplicité et leur robustesse, sont très efficaces dans de nombreux cas d'applications. Cependant la simplicité des HMC fait aussi leur faiblesse. En effet, leur construction est fondée sur des hypothèses fortes, souvent non vérifiées en situations réelles. C'est pourquoi différentes extensions des HMC ont été proposées, dans l'idée de garder cette simplicité de traitement, tout en allégeant les hypothèses de construction des modèles. En particulier nous allons nous intéresser aux chaînes de Markov couples et triplets, décrites dans le chapitre suivant.

Chapitre 2. Chaînes de Markov couples et triplets

Les chaînes de Markov cachées ont été étendues aux chaînes de Markov couples [23], [24], qui permettent de considérer directement la markovianité du couple (X, Y) sans que la loi *a priori* (loi de X) soit nécessairement markovienne. Puis les chaînes de Markov couples ont été étendues aux chaînes de Markov triplets [25], qui considèrent l'ajout d'un troisième processus stochastique U tel que le triplet (X, U, Y) soit de Markov, sans que la loi *a priori* de X , ni que la loi du couple (X, Y) soient de Markov. Le processus U en question, appelé processus auxiliaire est quelconque tant que (X, U, Y) est de Markov. Son interprétation va dépendre de l'application considérée (il se peut qu'il n'y ait pas d'interprétation). Il est la clé de l'intérêt des chaînes de Markov triplet, car il permet à celles-ci de modéliser des situations que les HMC sont incapables de bien gérer, telle que la non-stationnarité de la chaîne de Markov [26], [27], [28], la non-stationnarité du bruit [29], la modélisation du bruit par mélange de gaussiennes [30]. Les chaînes de Markov couples ont été appliquées avec succès au traitement du signal [31], [32], à la segmentation d'images [33] [34] [35], et à la segmentation de textes [36], tandis que les chaînes de Markov triplet ont amélioré les traitements par chaînes de Markov cachées dans de nombreux domaines tels que la segmentation d'image [37] [38] [39], la modélisation de l'indice de végétation par différence normalisée [40], la modélisation du génome [41], la classification d'activités physiques [42], la modélisation de remboursement de crédit à la consommation [43], ou encore la segmentation de données non-stationnaires floues [44], [45].

Ainsi dans ce chapitre, nous présentons d'abord les modèles de PMC puis les TMC généraux. Ces derniers posant un problème d'identifiabilité pour la segmentation d'images non supervisée, nous nous limitons par la suite aux chaînes de Markov triplet cachées qui ne présentent pas cet inconvénient. Puis nous présentons deux chaînes de Markov triplet cachées intéressantes : les chaînes de Markov évidentielles cachées et les chaînes de semi-Markov cachées.

2.1 Chaînes de Markov couples

La généralisation des chaînes de Markov cachées aux chaînes de Markov couples (PMC) permet de modéliser des situations plus complexes. En particulier dans une chaîne de Markov couple, le processus caché n'est pas nécessairement de Markov [24]. Cependant nous allons voir que les PMC restent compatibles avec la segmentation bayésienne. En effet l'estimateur du MPM (et MAP) n'est au final pas plus coûteux à calculer que dans les HMC et l'estimation des paramètres de manière non-supervisée reste faisable. De plus, l'extension des HMC aux PMC permet, dans certaines situations, de diviser l'erreur de segmentation par deux [46].

2.1.1 Définition du modèle

Soit $X = (X_1, \dots, X_N)$ un processus caché qui prend ses valeurs à partir d'un ensemble fini de classes $\Omega = \{\omega_1, \dots, \omega_K\}$ et qui doit être estimé à partir d'un processus observable $Y = (Y_1, \dots, Y_N)$ qui prend ses valeurs dans \mathbb{R} . Le processus couple (X, Y) est dit chaîne de Markov couple si (X, Y) est de Markov. Finalement, posons $Z = (X, Y)$, la loi de Z s'écrit de la manière suivante :

$$p(z) = p(z_1) \prod_{n=1}^{N-1} p(z_{n+1}|z_n) \quad (2.1)$$

Son graphe de probabilité est présenté sur la **Figure 2**.

La loi $p(y|x)$ est généralement désignée comme la distribution du bruit et peut être très complexe. En particulier, même si $p(y_n, y_{n+1}|x_n, x_{n+1})$ est une loi gaussienne pour tout n , $p(y|x)$ n'est pas gaussien dans le cas général. On peut directement remarquer que les PMC généralisent les HMC. En effet, posant $p(z_{n+1}|z_n) = p(x_{n+1}|x_n)p(y_{n+1}|y_n)$ pour tout n , (2.1) nous donne immédiatement la loi d'une HMC.

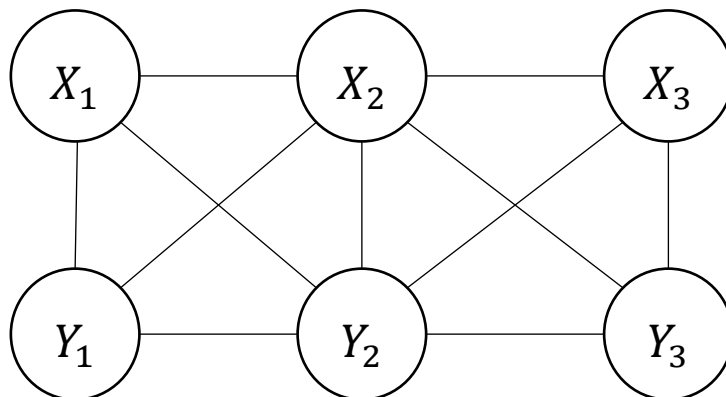


Figure 2. Graphe de probabilité d'une chaîne de Markov couple pour $N=3$

Un point important des PMC est que même si les marginales $p(x)$ et $p(y)$ ne sont pas de Markov, les lois conditionnelles $p(x|y)$ et $p(y|x)$ le sont. Pour montrer cela, il suffit de remarquer que $p(x|y) = \frac{p(z)}{p(y)} = \frac{p(z_1) \prod_{n=1}^{N-1} p(z_{n+1}|z_n)}{p(y)}$, y étant fixe ici. Cette décomposition assure que pour tout n , $p(x_n|x_{n-1}, \dots, x_1, y) = p(x_n|x_{n-1}, y)$. On peut faire exactement le même raisonnement pour $p(y|x)$, x et y jouant un rôle symétrique ici. Cette propriété est particulièrement intéressante car elle implique que la loi *a posteriori* $p(x|y)$ est Markovienne. Cela va servir dans le calcul de l'estimateur MPM et dans l'estimation des paramètres.

Une PMC est dite stationnaire si $p(z_n, z_{n+1})$ ne dépend pas de n . Dans ce cas, la loi est complètement définie une fois que l'on a explicité $p(z_1, z_2)$ qui est égal à $p(z_n, z_{n+1})$ pour

tous $n = 2 \dots N - 1$. Sachant que $p(z_1, z_2) = p(x_1, x_2)p(y_1, y_2|x_1, x_2)$, il suffit alors, pour définir toute la chaîne, de construire une matrice de taille $K \times K$ pour $p(x_1, x_2)$, en s'assurant que les marginales $p(x_1)$ et $p(x_2)$ soient identiques, et K^2 vecteurs moyennes et K^2 matrices de covariances pour $p(y_1, y_2|x_1, x_2)$ correspondant aux K^2 valeurs possibles du couple (X_1, X_2) .

2.1.2 Inférence

Les chaînes de Markov couples sont compatibles avec l'estimation bayésiennes car les calculs liés aux estimateurs bayésiens sont faciles et rapides. En particulier si nous regardons l'estimateur du MPM, donné par (1.18), on remarque qu'il est nécessaire de pouvoir évaluer les marginales a posteriori $p(x_n|y)$. Dans le cas d'une chaîne de Markov couples ce calcul est faisable avec une complexité en temps de $O(K^2N)$ si on suppose une complexité en espace de $O(KN)$. Ce calcul s'effectue de la manière suivante :

- i. Calculer les probabilités « forward » $\alpha_n(x_n)$ avec :

$$\begin{aligned} \alpha_1(x_1) &= p(z_1), \\ \alpha_{n+1}(x_{n+1}) &= \sum_{x_n \in \Omega} \alpha_n(x_n)p(z_{n+1}|z_n); \end{aligned} \quad (2.2)$$

- ii. Calculer les probabilités « backward » $\beta_n(x_n)$ avec :

$$\begin{aligned} \beta_N(x_N) &= 1, \\ \beta_n(x_n) &= \sum_{x_{n+1} \in \Omega} \beta_{n+1}(x_{n+1})p(z_{n+1}|z_n); \end{aligned} \quad (2.3)$$

- iii. Calculer $p(x_n|y)$ avec :

$$p(x_n|y) = \frac{\alpha_n(x_n)\beta_n(x_n)}{\sum_{x_n \in \Omega} \alpha_n(x_n)\beta_n(x_n)}. \quad (2.4)$$

Sachant que, dans le cas stationnaire, on a :

$$p(z_{n+1}|z_n) = \frac{p(z_n, z_{n+1})}{p(z_n)} = \frac{p(x_n, x_{n+1})p(y_n, y_{n+1}|x_n, x_{n+1})}{\sum_{x_{n+1} \in \Omega} p(x_n, x_{n+1})p(y_n, y_{n+1}|x_n, x_{n+1})} \quad (2.5)$$

Il est aussi possible de calculer l'estimateur MAP dans une PMC, avec l'algorithme de Viterbi [32] avec la même complexité en temps de $O(K^2N)$, toujours si on suppose une complexité en espace de $O(KN)$, cependant comme nous nous intéressons au MPM ici, nous ne détaillerons pas la mise en œuvre de cet algorithme.

2.1.3 Estimation des paramètres non supervisée

L'estimation des paramètres pour les PMC est un peu plus compliquée que dans les HMC. En particulier, pour l'algorithme EM, on ne peut plus maximiser analytiquement la quantité $E[\log(p(x, y|\theta)|Y, \theta^q)]$ à chaque itération. Cependant des variantes des algorithmes SEM [47]

et ICE [23], [24] fonctionnent encore, et permettent l'estimation des paramètres des PMC dans le cadre non-supervisé. La complexité en temps d'une itération de ICE ou SEM est la même : elle est de $O(K^2N)$ en temps si on suppose une complexité en espace de $O(KN)$, comme pour les HMC.

2.1.3.1 Algorithme SEM pour les chaînes de Markov couples

Soit une famille de PMC $Z = (X, Y)$ stationnaires avec $p(y_n, y_{n+1}|x_n, x_{n+1})$ gaussien, et indexée par un vecteur de paramètres θ . Cette famille est définie par $p_{kl} = p(x_2 = \omega_l, x_1 = \omega_k)$, les vecteurs moyennes $\mu_{kl} = \begin{pmatrix} \mu_{kl}^1 \\ \mu_{kl}^2 \end{pmatrix}$, et les matrices de covariances Γ_{kl} associées aux probabilités $p(y_n, y_{n+1}|x_n = \omega_k, x_{n+1} = \omega_l)$.

La loi a posteriori d'une chaîne de Markov couple est de Markov, et est calculée de la manière suivante :

$$p(x|y) = p(x_1|y) \prod_{n=1}^{N-1} p(x_{n+1}|x_n, y) \quad (2.6)$$

$p(x_1|y)$ est calculé avec (2.4), et $p(x_{n+1}|x_n, y)$ s'exprime en fonction des probabilité « backward » :

$$p(x_{n+1}|x_n, y) = \frac{p(z_{n+1}|z_n)\beta_{n+1}(x_{n+1})}{\beta_n(x_n)} \quad (2.7)$$

L'algorithme SEM pour les PMC se déroule comme il suit [47]:

- On initialise l'algorithme avec $\theta = \theta^0$
- Pour chaque itération q , on utilise θ^q pour simuler x^q selon (2.6). Ensuite on calcule θ^{q+1} de la manière suivante :

$$p_{kl}^{(q+1)} = \frac{\sum_{n=1}^{N-1} 1_{\{x_n^{(q)} = \omega_k, x_{n+1}^{(q)} = \omega_l\}}}{N - 1} \quad (2.8)$$

$$\mu_{kl}^{(q+1)} = \frac{\sum_{n=1}^{N-1} 1_{\{x_n^{(q)} = \omega_k, x_{n+1}^{(q)} = \omega_l\}} \begin{pmatrix} y_n \\ y_{n+1} \end{pmatrix}}{\sum_{n=1}^{N-1} 1_{\{x_n^{(q)} = \omega_k, x_{n+1}^{(q)} = \omega_l\}}} \quad (2.9)$$

$$\Gamma_{kl}^{(q+1)} = \frac{\sum_{n=1}^{N-1} 1_{\{x_n^{(q)} = \omega_k, x_{n+1}^{(q)} = \omega_l\}} \begin{pmatrix} y_n - \mu_{kl}^{1,(q+1)} \\ y_{n+1} - \mu_{kl}^{2,(q+1)} \end{pmatrix} \begin{pmatrix} y_n - \mu_{kl}^{1,(q+1)} \\ y_{n+1} - \mu_{kl}^{2,(q+1)} \end{pmatrix}^t}{\sum_{n=1}^{N-1} 1_{\{x_n^{(q)} = \omega_k, x_{n+1}^{(q)} = \omega_l\}}} \quad (2.10)$$

2.1.3.2 Algorithme ICE pour les chaînes de Markov couples

L'algorithme ICE peut être appliqué dans le cadre des PMC. En effet, reprenons notre famille de PMC $Z = (X, Y)$ stationnaires avec $p(y_n, y_{n+1}|x_n, x_{n+1})$ gaussien, et indexée par un paramètre θ . Notons :

$$p_{kl,n,n+1} = p(x_{n+1} = \omega_l, x_n = \omega_k | y) \quad (2.11)$$

On peut calculer $p_{kl,n,n+1}$ de la manière suivante :

$$p_{kl,n,n+1} = \frac{\alpha_n(x_n)p(z_{n+1}|z_n)\beta_{n+1}(x_{n+1})}{\sum_{x_n \in \Omega} \alpha_n(x_n)p(z_{n+1}|z_n)\beta_{n+1}(x_{n+1})} \quad (2.12)$$

$p(z_{n+1}|z_n)$ se calculant avec (2.5). Ainsi l'algorithme se déroule de la manière suivante [23], [24] :

- On initialise l'algorithme avec $\theta = \theta^0$
- Pour chaque itération q , on utilise θ^q pour calculer $p_{kl,n,n+1}^{(q)}$ avec (2.12) et simuler $x^{1,q}, \dots, x^{R,q}$ selon (2.6). Ensuite on calcule θ^{q+1} de la manière suivante :

$$p_{kl}^{(q+1)} = \frac{\sum_{n=1}^{N-1} p_{kl,n,n+1}^{(q)}}{N-1} \quad (2.13)$$

$$\mu_{kl}^{(q+1)} = \frac{\sum_{r=1}^R \sum_{n=1}^{N-1} \mathbf{1}_{\{x_n^{r,q} = \omega_k, x_{n+1}^{r,q} = \omega_l\}} \begin{pmatrix} y_n \\ y_{n+1} \end{pmatrix}}{\sum_{r=1}^R \sum_{n=1}^{N-1} \mathbf{1}_{\{x_n^{r,q} = \omega_k, x_{n+1}^{r,q} = \omega_l\}}} \quad (2.14)$$

$$\Gamma_{kl}^{(q+1)} = \frac{\sum_{r=1}^R \sum_{n=1}^{N-1} \mathbf{1}_{\{x_n^{r,q} = \omega_k, x_{n+1}^{r,q} = \omega_l\}} \begin{pmatrix} y_n - \mu_{kl}^{1,(q+1)} \\ y_{n+1} - \mu_{kl}^{2,(q+1)} \end{pmatrix}^t \begin{pmatrix} y_n - \mu_{kl}^{1,(q+1)} \\ y_{n+1} - \mu_{kl}^{2,(q+1)} \end{pmatrix}}{\sum_{r=1}^R \sum_{n=1}^{N-1} \mathbf{1}_{\{x_n^{r,q} = \omega_k, x_{n+1}^{r,q} = \omega_l\}}} \quad (2.15)$$

2.2 Chaînes de Markov triplet

Les chaînes de Markov triplets (TMC) sont une extension des chaînes de Markov couples qui consiste à rajouter un troisième processus latent, qui a ou non une signification concrète. Ce troisième processus est ajouté de façon à ce que les observations et les données cachées à prédire soient considérées comme la loi marginale d'un triplet Markovien [25]. L'ajout de ce troisième processus démultiplie grandement la capacité du modèle à gérer des situations complexes. Cependant nous allons voir que comme pour les PMC, dans le cas où le troisième processus est à valeurs discrètes en nombre fini, les TMC restent compatibles avec certaines segmentations bayésiennes. En effet il est possible de calculer efficacement l'estimateur bayésien du MPM (pour le MAP en revanche ce n'est pas le cas) et l'estimation des paramètres reste possible dans le cas général [48].

2.2.1 Définition du modèle

Soit $X = (X_1, \dots, X_N)$ un processus caché qui prend ses valeurs à partir d'un ensemble fini de classes $\Omega = \{\omega_1, \dots, \omega_K\}$ et qui doit être estimé à partir d'un processus observable $Y = (Y_1, \dots, Y_N)$ qui prend ses valeurs dans \mathbb{R} . Soit un troisième processus $U = (U_1, \dots, U_N)$ à valeurs dans $\Lambda = \{\lambda_1, \dots, \lambda_L\}$. Le processus triplet (X, U, Y) est dit chaîne de Markov triplet si (X, U, Y) est de Markov. Finalement, posons $T = (X, U, Y)$, la loi de T s'écrit de la manière suivante :

$$p(t) = p(t_1) \prod_{n=1}^{N-1} p(t_{n+1}|t_n) \quad (2.16)$$

Son graphe de probabilité est présenté sur la **Figure 3**.

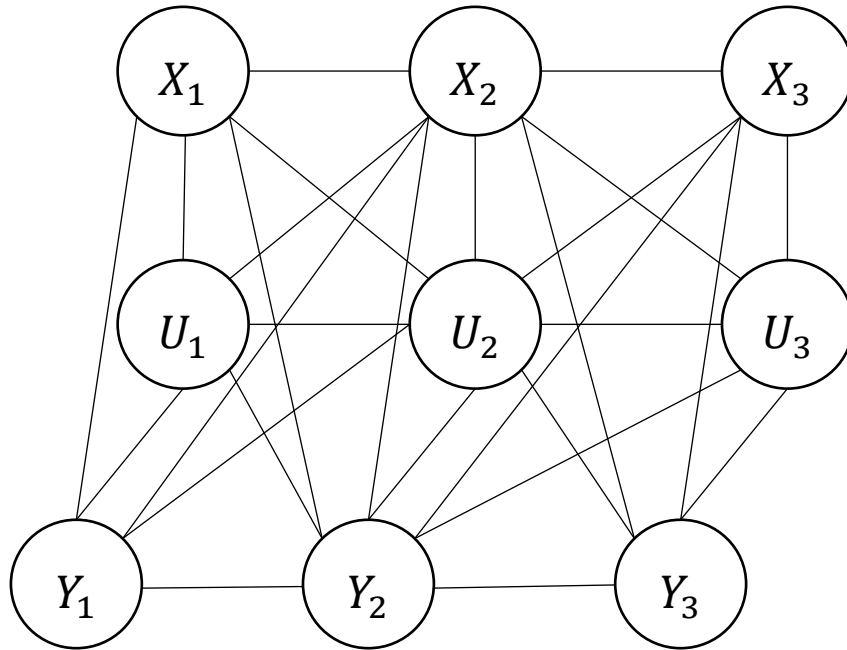


Figure 3. Graphe de probabilité d'une chaîne de Markov triplet pour $N=3$

Notons que la loi de « bruit » $p(y|x) = \sum_u p(y|x, u)$ est très complexe, ce qui montre aussi qu'elle est très complète, en particulier, si pour tout n , $p(y_n, y_{n+1}|u_n, x_n, u_{n+1}, x_{n+1})$ est une loi gaussienne, ni $p(y|x, u)$, ni $p(y|x)$ n'est gaussien en général. Les TMC généralisent les PMC, en effet, pour $X = U$ on retrouve une PMC.

Une TMC est dite stationnaire si les lois $p(t_n, t_{n+1})$ ne dépendent pas de n . Dans ce cas, la loi est complètement définie une fois que l'on a explicité $p(t_1, t_2)$ qui est égal à $p(t_n, t_{n+1})$ pour tout $n = 2 \dots N - 1$. Sachant que $p(t_1, t_2) = p(x_1, u_1, u_2, x_2)p(y_1, y_2|x_1, u_1, u_2, x_2)$, il suffit alors, pour définir toute la chaîne, de construire une matrice de taille $KL \times KL$ pour

$p(x_1, u_1, u_2, x_2)$, en s'assurant que les marginales $p(x_1, u_1)$ et $p(u_2, x_2)$ soient identiques, et K^2L^2 vecteurs moyennes et K^2L^2 matrices de covariances pour $p(y_1, y_2|x_1, u_1, u_2, x_2)$.

2.2.2 Inférence

Les chaînes de Markov triplets sont compatibles avec certaines estimations bayésiennes car les calculs liés aux estimateurs correspondants sont faciles et rapides. En particulier, si nous regardons l'estimateur du MPM, donnée par (1.18), on remarque qu'il est nécessaire de pouvoir évaluer $p(x_n|y)$ pour tout $n \in \{1, \dots, N\}$. Dans le cas d'une chaîne de Markov triplet ce calcul est faisable avec une complexité en temps de $O(K^2L^2N)$ si on suppose une complexité en espace de $O(KLN)$. En effet, posons $V = (X, U)$, alors $T = (V, Y)$ est une chaîne de Markov couple. Ainsi nous pouvons calculer $p(v_n|y)$ de la manière suivante :

- i. Calculer les probabilités « forward » $\alpha_n(v_n)$ avec :

$$\begin{aligned} \alpha_1(v_1) &= p(t_1) \\ \alpha_{n+1}(v_{n+1}) &= \sum_{v_n \in \Omega \times \Lambda} \alpha_n(v_n) p(t_{n+1}|t_n) \end{aligned} \quad (2.17)$$

- ii. Calculer les probabilités « backward » $\beta_n(v_n)$ avec :

$$\begin{aligned} \beta_N(v_N) &= 1 \\ \beta_n(v_n) &= \sum_{v_{n+1} \in \Omega \times \Lambda} \beta_{n+1}(v_{n+1}) p(t_{n+1}|t_n) \end{aligned} \quad (2.18)$$

- iii. Calculer $p(v_n|y)$ avec :

$$p(v_n|y) = \frac{\alpha_n(v_n)\beta_n(v_n)}{\sum_{v_n \in \Omega \times \Lambda} \alpha_n(v_n)\beta_n(v_n)} \quad (2.19)$$

- iv. Il suffit enfin de remarquer que $p(v_n|y) = p(x_n, u_n|y)$ et donc :

$$p(x_n|y) = \sum_{u_n \in \Lambda} p(x_n, u_n|y) \quad (2.20)$$

Par contre, il n'est plus possible d'appliquer l'algorithme de Viterbi. Ainsi l'estimateur du MAP n'est pas calculable de façon efficace dans une chaîne de Markov triplet.

2.2.3 Estimation des paramètres non supervisée et identifiabilité

L'estimation non supervisée des paramètres des TMC est identique à celle des paramètres des PMC. En particulier comme on l'a vu, en posant $V = (X, U)$, alors la TMC $T = (X, U, Y)$ peut être vu comme une PMC (V, Y) . Ainsi on ne peut pas utiliser l'algorithme EM. Cependant, les algorithmes SEM et ICE permettent l'estimation des paramètres en non supervisée. En effet soit une famille de TMC $T = (X, U, Y)$ stationnaires à bruit gaussien, indexée par un paramètre θ . En posant $V = (X, U)$ on se ramène strictement à l'estimation de paramètres pour la chaîne de Markov couple avec V à valeur dans $\Omega \times \Lambda$ à la place de X . On peut donc utiliser respectivement les équations (2.8) -(2.10) ou (2.13) -(2.15) à chaque itération pour estimer les

paramètres d'un triplet dans le cadre du SEM ou d'ICE. On voit immédiatement qu'une itération de ces algorithmes a une complexité en temps de $O(K^2L^2N)$ si on suppose une complexité en espace de $O(KLN)$.

Cependant contrairement aux PMC, un problème d'identifiabilité apparaît dans le cadre de la segmentation non supervisée avec une TMC. La non identifiabilité dans la segmentation non supervisée est courante. En effet on la retrouve avec des mélanges gaussiens : la distribution d'un mélange de K gaussiennes (et donc la fonction de vraisemblance) est invariante à la permutation des labels indexés $k = 1, \dots, K$ [49]. C'est aussi le cas pour les chaînes de Markov cachées, avec $X = (X_1, \dots, X_N)$ à valeurs dans $\Omega^N = (\omega_1, \dots, \omega_K)^N$. Cependant, cette non identifiabilité ne pose pas problème lorsque l'on veut classer les observations $Y = (Y_1, \dots, Y_N)$ en groupes. En effet quel que soit le label du groupe, c'est la séparation qui est importante. Ainsi dans ce cas-ci, les maxima de vraisemblance associés à chaque permutation des labels donneront les mêmes résultats du MPM, à la permutation des labels près. Mais ce n'est plus le cas dans le triplet Markovien, à cause du calcul (2.20) qui intervient dans le calcul du MPM et qui impose une somme sur U , et donc une identifiabilité partielle, à savoir qu'il faut être capable de différentier X de U . Or de par la nature du triplet, il est impossible de différentier X de U dans le cadre non supervisé [50]. Cela pose problème lorsque l'on veut utiliser l'estimateur MPM. Ce problème ne peut pas être résolu pour la TMC générale. Cependant, il existe des TMC où cet inconvénient disparaît. On peut mettre en lumière deux types de TMC où le problème disparaît. Le premier type est une TMC où la loi de U (ou X) serait connue et donc fixe (elle ne ferait pas partie des paramètres à estimer). Le deuxième type est une TMC particulière où certaines dépendances auraient été supprimées de telle sorte que U (ou X) ne soient plus directement dépendant de Y , c'est-à-dire que $p(u|x, y) = p(u|x)$ ou $(p(x|u, y) = p(x|u))$, ce simple fait rend les permutations entre les labels de X et les labels de U non équivalentes, et la vraisemblance changera en fonction de ces permutations. Un exemple de graphe de probabilité d'un tel modèle est présenté dans la **figure 4**. Ainsi dans la suite nous nous intéresserons à ce dernier type de triplets qui ne comporte pas l'inconvénient mentionné (car notre domaine d'application choisi est la segmentation non-supervisée), mais qui permet quand même de considérer la loi de U comme paramètre et de l'estimer.

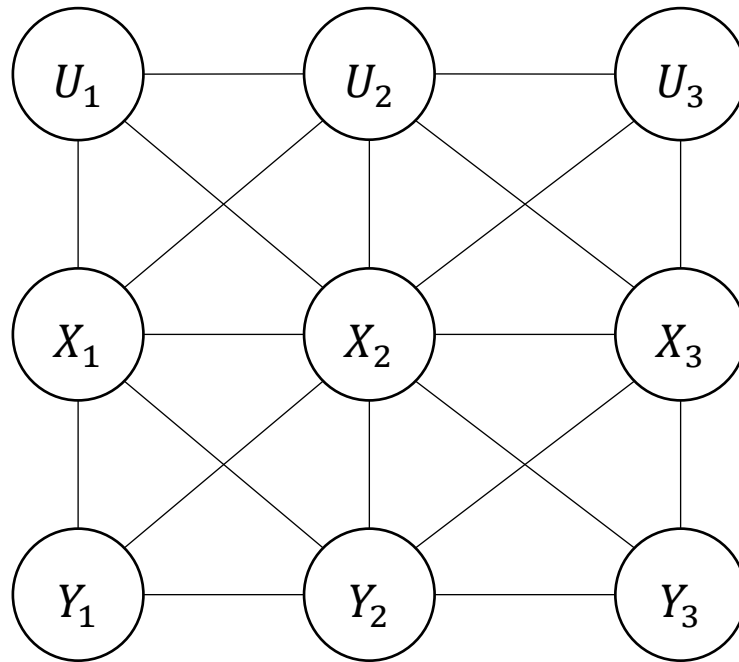


Figure 4. Exemple de TMC, $N = 3$, où l'identifiabilité de (X, U) ne pose pas de problème dans le cas non supervisé

On peut remarquer par ailleurs que ce problème n'existe pas lorsque l'on apprend les paramètres de manière supervisée : en effet le fait que l'on observe X et Y mais pas U dans ce cas implique que X et U apparaissent distinctes. On peut alors utiliser des algorithmes SEM ou ICE pour estimer les paramètres (sur ce point on rappelle que même en supervisé, on ne dispose pas de réalisation de U dans le cas général, donc on est obligé d'utiliser les techniques d'estimation avec données incomplètes comme SEM ou ICE).

2.3 Chaînes de Markov triplet cachées

Le problème des modèles de TMC générales, évoqué dans la section 2.2.3, à savoir l'impossibilité de différencier X et U dans le cas d'une segmentation non supervisée, a pour conséquence que ceux-ci ne sont pas très pratiques à utiliser dans ce cas. Cependant, comme mentionné précédemment il existe des triplets particuliers qui ne présentent pas cet inconvénient. C'est le cas du modèle que nous présentons ici, les chaînes de Markov triplet cachées (HTMC).

2.3.1 Définition du modèle

Soit $X = (X_1, \dots, X_N)$ un processus caché qui prend ses valeurs dans un ensemble fini de classes $\Omega = \{\omega_1, \dots, \omega_K\}$ et qui doit être estimé à partir d'un processus observable $Y = (Y_1, \dots, Y_N)$ qui prend ses valeurs dans \mathbb{R} . Soit un troisième processus $U = (U_1, \dots, U_N)$ à valeur dans $\Lambda = \{\lambda_1, \dots, \lambda_L\}$. Nous appelons chaînes de Markov triplet cachées le triplet $T = (X, U, Y)$ vérifiant :

$$p(t_1) = p(x_1, u_1)p(y_1|x_1)$$

$$p(t_{n+1}|t_n) = p(x_{n+1}, u_{n+1}|x_n, u_n)p(y_{n+1}|x_{n+1}) \quad (2.21)$$

On peut remarquer que (2.21) est une TMC simplifiée, obtenue en supposant : $p(x_{n+1}, u_{n+1}|x_n, u_n, y_n) = p(x_{n+1}, u_{n+1}|x_n, u_n)$, et $p(y_{n+1}|x_n, u_n, y_n, x_{n+1}, u_{n+1}) = p(y_{n+1}|x_{n+1})$. Son graphe de probabilité est présenté sur la **Figure 5**. Par ailleurs, si on pose $V = (X, U)$ alors le couple (V, Y) est une chaîne de Markov cachée. Notons que dans une chaîne de Markov triplet cachée X n'est pas, en général, une chaîne de Markov, et (X, Y) n'est pas non plus, en général, une chaîne de Markov couple.

Dans une HTMC stationnaire, la loi est complètement définie une fois que l'on a explicité $p(t_1, t_2)$ qui est égal à $p(t_n, t_{n+1})$ pour tous $n = 2 \dots N - 1$. Il suffit alors, pour définir toute la chaîne, de construire une matrice de taille $KL \times KL$ pour $p(x_1, u_1, u_2, x_2)$, en s'assurant que les marginales $p(x_1, u_1)$ et $p(u_2, x_2)$ soient identiques, et K moyennes et K variances pour $p(y_1|x_1)$. On peut aussi considérer une HTMC homogène, définie par une matrice de transition de taille $KL \times KL$ pour $p(u_2, x_2|x_1, u_1)$, KL paramètres pour $p(x_1, u_1)$ et enfin K moyennes et K variances pour $p(y_1|x_1)$.

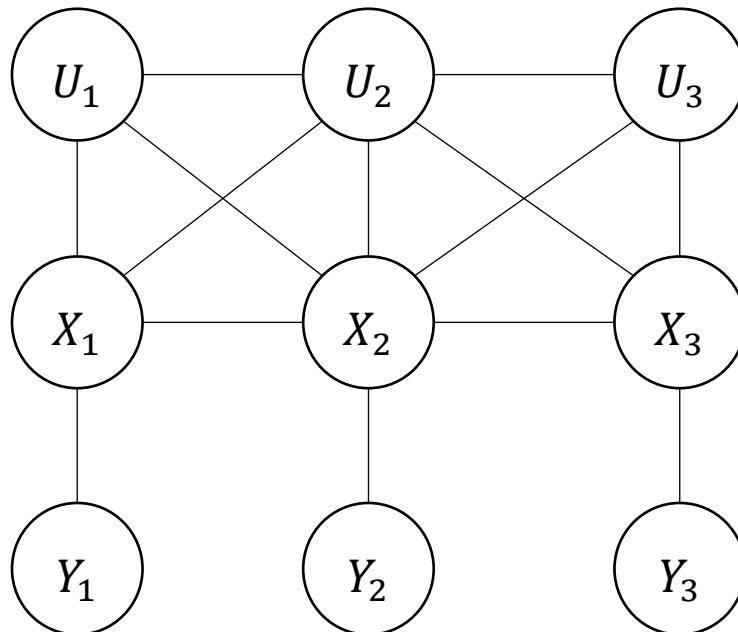


Figure 5. Graphe de probabilité d'une chaîne de Markov triplet cachée pour $N=3$

2.3.2 Inférence

Les HTMC étant un cas particulier de TMC, l'inférence se déduit directement de celle utilisée pour les TMC. Cependant, compte tenu des simplifications de (2.17) -(2.20) introduites par (2.21), on peut expliciter le calcul de l'estimateur du MPM, donnée par (1.18) pour le HTMC en particulier :

- i. Calculer les probabilités « forward » $\alpha_n(x_n, u_n)$ avec :

$$\alpha_1(x_1, u_1) = p(x_1, u_1)p(y_1|x_1)$$

$$\alpha_{n+1}(x_{n+1}, u_{n+1}) = \sum_{(x_n, u_n) \in \Omega \times \Lambda} \alpha_n(x_n, u_n)p(x_{n+1}, u_{n+1}|x_n, u_n)p(y_{n+1}|x_{n+1}) \quad (2.22)$$

- ii. Calculer les probabilités « backward » $\beta_n(x_n, u_n)$ avec :

$$\beta_N(x_N, u_N) = 1$$

$$\beta_n(x_n, u_n) = \sum_{(x_{n+1}, u_{n+1}) \in \Omega \times \Lambda} \beta_{n+1}(x_{n+1}, u_{n+1})p(x_{n+1}, u_{n+1}|x_n, u_n)p(y_{n+1}|x_{n+1}) \quad (2.23)$$

- iii. Calculer $p(x_n, u_n|y)$ avec :

$$p(x_n, u_n|y) = \frac{\alpha_n(x_n, u_n)\beta_n(x_n, u_n)}{\sum_{(x_n, u_n) \in \Omega \times \Lambda} \alpha_n(x_n, u_n)\beta_n(x_n, u_n)} \quad (2.24)$$

- iv. Calculer $p(x_n|y)$ avec :

$$p(x_n|y) = \sum_{u_n \in \Lambda} p(x_n, u_n|y) \quad (2.25)$$

Le calcul de l'estimateur du MPM a une complexité en temps de $O(K^2L^2N)$ si on suppose une complexité en espace de $O(KLN)$. C'est la même complexité que pour les TMC générales.

Comme pour les TMC générales cependant, il est impossible d'utiliser l'algorithme de Viterbi pour les HTMC. On ne peut donc pas calculer l'estimateur du MAP de manière efficace.

2.3.3 Estimation des paramètres non supervisée

Les HTMC étant des TMC particulières, on pourrait appliquer directement les algorithmes d'estimations de paramètres des TMC aux HTMC. Cependant, contrairement aux TMC généraux, on peut remarquer que pour la HTMC $T = (X, U, Y)$, si on pose $V = (X, U)$, alors le couple (V, Y) est une chaîne de Markov cachée. Cela simplifie quelque peu l'estimation des paramètres. En particulier, l'algorithme EM est utilisable dans les HTMC alors qu'il ne l'était pas, sous sa forme exacte, dans les TMC. Dans cette section nous allons détailler seulement le déroulement des algorithmes EM et SEM car c'est ceux que nous avons utilisés dans la suite de nos travaux, cependant l'algorithme ICE est aussi applicable dans le cas des HTMC. On peut remarquer qu'une itération de chacun de ces trois algorithmes a une complexité en temps de $O(K^2L^2N)$ si on suppose une complexité en espace de $O(KLN)$, ce qui ne diffère pas des TMC généraux.

2.3.3.1 Algorithme EM pour les chaînes de Markov triplet cachées

Soit une famille de HTMC (X, U, Y) homogènes à bruit gaussien, indexée par un vecteur de paramètres θ . Ce dernier est composé de $\pi_{kl} = p(x_1 = \omega_k, u_1 = \lambda_l)$, $p_{ijkl} = p(x_2 = \omega_k, u_2 = \lambda_l | x_1 = \omega_i, u_1 = \lambda_j)$ et les moyennes et variances μ_k, σ_k^2 associées aux probabilités $p(y_n | x_n = \omega_k)$. Notons :

$$p_{kl,n} = p(x_n = \omega_k, u_n = \lambda_l | y) \quad (2.26)$$

$$p_{ijkl,n,n+1} = p(x_{n+1} = \omega_k, u_{n+1} = \lambda_l, x_n = \omega_i, u_n = \lambda_j | y) \quad (2.27)$$

$$p_{ijkl,n} = p(x_{n+1} = \omega_k, u_{n+1} = \lambda_l | x_n = \omega_i, u_n = \lambda_j, y) \quad (2.28)$$

On peut calculer $p_{kl,n,n+1}$ de la manière suivante :

$$p_{kl,n,n+1} = \frac{\alpha_n(x_n, u_n) p(x_{n+1}, u_{n+1} | x_n, u_n) p(y_{n+1} | x_{n+1}) \beta_{n+1}(x_{n+1}, u_{n+1})}{\sum_{(x_n, u_n) \in \Omega \times \Lambda} \alpha_n(x_n, u_n) p(x_{n+1}, u_{n+1} | x_n, u_n) p(y_{n+1} | x_{n+1}) \beta_{n+1}(x_{n+1}, u_{n+1})} \quad (2.29)$$

L'algorithme EM pour les HTMC se déroule comme suit :

- On initialise l'algorithme avec $\theta = \theta^0$
- Pour chaque itération q , on utilise θ^q pour calculer $p_{kl,n}^{(q)}$ avec (2.24) et $p_{ijkl,n,n+1}^{(q)}$ avec (2.29). Ensuite on calcule θ^{q+1} de la manière suivante :

$$\pi_{kl}^{(q+1)} = \frac{\sum_{n=1}^N p_{kl,n}^{(q)}}{N} \quad (2.30)$$

$$p_{ijkl}^{(q+1)} = \frac{\sum_{n=1}^{N-1} p_{ijkl,n,n+1}^{(q)}}{\sum_{n=1}^{N-1} p_{kl,n}^{(q)}} \quad (2.31)$$

$$\mu_k^{(q+1)} = \frac{\sum_{n=1}^N \sum_{l=1}^m p_{kl,n}^{(q)} y_n}{\sum_{n=1}^N \sum_{l=1}^m p_{kl,n}^{(q)}} \quad (2.32)$$

$$\sigma_k^{2,(q+1)} = \frac{\sum_{n=1}^N \sum_{l=1}^m p_{kl,n}^{(q)} (y_n - \mu_k^{(q+1)})^2}{\sum_{n=1}^N \sum_{l=1}^m p_{kl,n}^{(q)}} \quad (2.33)$$

2.3.3.2 Algorithme SEM pour les chaînes de Markov triplet cachées

On rappelle que $V = (X, U)$, (V, Y) est une chaîne de Markov cachée. Ainsi la loi a posteriori d'une HTMC peut se calculer de la manière suivante :

$$p(x, u | y) = p(x_1, u_1 | y) \prod_{n=1}^{N-1} p(x_{n+1}, u_{n+1} | x_n, u_n, y) \quad (2.34)$$

$p(x_1, u_1 | y)$ est calculé avec (2.24), et $p(x_{n+1}, u_{n+1} | x_n, u_n, y)$ s'exprime en fonction des probabilité « backward » :

$$p(x_{n+1}, u_{n+1} | x_n, u_n, y) = \frac{p(x_{n+1}, u_{n+1} | x_n, u_n) p(y_{n+1} | x_{n+1}) \beta_{n+1}(x_{n+1}, u_{n+1})}{\beta_n(x_n, u_n)} \quad (2.35)$$

L'algorithme SEM pour les HTMC se déroule comme il suit :

- On initialise l'algorithme avec $\theta = \theta^0$
- Pour chaque itération q , on utilise θ^q pour simuler (x^q, u^q) selon (2.34). Ensuite on calcule θ^{q+1} de la manière suivante :

$$\pi_{kl}^{(q+1)} = \frac{\sum_{n=1}^N \mathbf{1}_{\{x_n^{(q)} = \omega_k, u_n^{(q)} = \lambda_l\}}}{N} \quad (2.36)$$

$$p_{ijkl}^{(q+1)} = \frac{\sum_{n=1}^{N-1} \mathbf{1}_{\{x_n^{(q)} = \omega_i, u_n^{(q)} = \lambda_j, x_{n+1}^{(q)} = \omega_k, u_{n+1}^{(q)} = \lambda_l\}}}{\sum_{n=1}^{N-1} \mathbf{1}_{\{x_n^{(q)} = \omega_k, u_n^{(q)} = \lambda_l\}}} \quad (2.37)$$

$$\mu_k^{(q+1)} = \frac{\sum_{n=1}^N \mathbf{1}_{\{x_n^{(q)} = \omega_k\}} y_n}{\sum_{n=1}^N \mathbf{1}_{\{x_n^{(q)} = \omega_k\}}} \quad (2.38)$$

$$\sigma_k^{2,(q+1)} = \frac{\sum_{n=1}^N \mathbf{1}_{\{x_n^{(q)} = \omega_k\}} (y_n - \mu_k^{(q+1)})^2}{\sum_{n=1}^N \mathbf{1}_{\{x_n^{(q)} = \omega_k\}}} \quad (2.39)$$

2.4 Chaînes de Markov évidentielles

Les HMC classiques perdent en efficacité dans le cas de la segmentation de données non stationnaires de manière non supervisée. Cependant les TMC peuvent être utilisées pour traiter ce problème. En particulier les modèles de Markov à saut [27] [29] et les modèles de Markov évidentiels [28] [26] permettent tous deux d'améliorer nettement les segmentations des HMC dans le cas de données non stationnaires. Nous nous intéressons plus particulièrement aux chaînes de Markov évidentielles. En effet la théorie de Dempster-Shafer peut être utile dans de nombreuses situations [51] [52] [53] [54] [55], et peut être utilisée dans un contexte markovien [56] [57] [58] [59]. Les chaînes de Markov évidentielles sont une des façons de le faire, en se basant sur les modèles de TMC.

2.4.1 Non-stationnarité et chaînes de Markov triplets

Il nous semble important de préciser la notion de non stationnarité discutée tout au long de ce document, ainsi que sa signification concrète dans la segmentation d'images. Pour cela, nous allons introduire deux types de non stationnarité. Dans le premier, que l'on retrouve dans l'article [27], les paramètres du modèle changent aléatoirement, avec un nombre fini de possibilités. Comme ces paramètres sont stationnaires entre deux changements, nous parlerons

de « stationnarité par morceaux ». Le deuxième type de non stationnarité est plus général, et concerne des paramètres pouvant évoluer aléatoirement de manière continue, ou même de manière hybride (mélange entre discret et continu).

Considérons à présent le HMC classique (X, Y) , où chaque X_n prend ses valeurs dans $\Omega = \{\omega_1, \dots, \omega_K\}$ et Y_n prend ses valeurs dans \mathbb{R} . Le couple (X, Y) est markovien et sa distribution est déterminée par la probabilité initiale $p(x_1, y_1)$ et les transitions de la forme :

$$p(x_{n+1}, y_{n+1} | x_n, y_n) = p(x_{n+1} | x_n) p(y_{n+1} | x_{n+1}) \quad (2.40)$$

Supposons en plus que les distributions $p(y_n | x_n)$ sont gaussiennes et que la HMC est homogène. Nous rappelons alors que, comme vu en section 1.31, $p(x, y)$ est totalement définie par une matrice de transition de taille $K \times K$ pour $p(x_{n+1} | x_n)$, K paramètres pour $p(x_1)$ et K moyennes et K variances associées à la loi gaussienne $p(y_n | x_n)$. Dans la suite nous appellerons ces paramètres θ . Imaginons par la suite qu'il y ait deux HMC homogènes de paramètres θ_1, θ_2 respectivement. Enfin mettons nous dans la situation où la loi du couple (X, Y) est un HMC tel que le modèle ait pour paramètre θ_1 pour la séquence $(X_1, Y_1), \dots, (X_m, Y_m)$, et θ_2 pour la séquence $(X_{m+1}, Y_{m+1}), \dots, (X_N, Y_N)$. Une telle HMC est appelée HMC à sauts. Une extension de ce modèle à un nombre fini de saut est facilement réalisable. Nous allons considérer ce type de modèle dans la suite. Ce genre de situation est courant en segmentation d'images. En effet considérons deux exemples :

- Soit y une ligne dans une image aérienne optique avec deux classes « eau » et « forêt » et soit x les classes cachées. A cause des nuages, le soleil éclaire une partie de l'image, alors que l'autre partie est dans l'ombre. Du coup les $p(y_n | x_n)$ sont différents dans les zones éclairées et les zones d'ombres, et ainsi nous avons deux vecteurs différents θ_1, θ_2 . Une telle situation admet un « bruit à saut » [29].
- Soit y encore une ligne dans une image aérienne optique avec deux classes, mais cette fois les classes sont « zone urbaine » et « forêt ». Supposons que $N = 500$ et que les pixels de l'image représentent une zone d'aire 25 m^2 . Il y a une ville dans l'image, dont le centre contient des parcs pour $(x_1, y_1), \dots, (x_{40}, y_{40})$, on trouve la banlieue pour $(x_{41}, y_{41}), \dots, (x_{150}, y_{150})$, la forêt pour $(x_{151}, y_{151}), \dots, (x_{490}, y_{490})$, et enfin un village pour $(x_{491}, y_{491}), \dots, (x_{500}, y_{500})$. Il est évident que les $p(x_{n+1} | x_n)$ sont différents pour chacune de ces zones. Cette situation admet donc « des transitions de classes à saut ».

On peut noter qu'il est possible de gérer les deux situations simultanément. Lorsque nous connaissons l'emplacement des sauts, nous avons simplement une HMC non stationnaire classique, qui est « stationnaire par morceaux ». Le cas intéressant est celui où ne connaît pas l'emplacement des sauts. Pour les estimer, nous devons définir une séquence aléatoire U , prenant ses valeurs dans l'ensemble des paramètres $\Theta = \{\theta_1, \dots, \theta_M\}$ modélisant les différentes

stationnarités. On peut alors définir la distribution du triplet (X, U, Y) par $p(x, u, y) = p(u)p(x, y|u)$, avec $p(u)$ markovienne et $p(x, y|u)$ loi d'un HMC (1.15). La loi d'un tel modèle s'écrit :

$$p(x, y) = p(u_1)p(x_1, y_1|u_1), \prod_{n=1}^{N-1} p(u_{n+1}|u_n)p(x_{n+1}|x_n, u_{n+1})p(y_{n+1}|x_{n+1}) \quad (2.41)$$

Le point intéressant est que l'équation (2.41) peut être utilisée pour estimer (x, u) à partir de y . En effet, c'est une loi particulière de HTMC, et cette HTMC peut être supposé stationnaire sans que cela n'implique la stationnarité des lois $p(x, y)$ et $p(x, y|u)$. Ainsi l'estimation des paramètres en non supervisée est faisable et l'estimateur du MPM est calculable rapidement. On a donc pu convertir un HMC à saut, qui est « stationnaire par morceaux » en une HTMC stationnaire, permettant l'estimation non supervisée de X et U . On peut noter que dans le triplet (X, U, Y) considéré ici, le couple (X, Y) n'est pas de Markov en général, alors qu'il est de Markov conditionnellement à U .

Le second type de non-stationnarité est plus général. Son principe est que le vecteur de paramètres θ peut évoluer de manière continue, et ne se limite pas à un nombre fini de valeurs. Par exemple si on reprend notre premier exemple (i), et que l'on considère que la nature des nuages fait que ceux-ci produisent une ombre sur l'image qui varie en intensité, les situations « sombre » et « éclairé » ne sont plus réellement distinctes et le niveau de luminosité évolue continuellement, avec un éclairage plus ou moins important. Dans ce cas, les paramètres de $p(y_{n+1}|x_{n+1})$ évoluent de manière continue. De manière similaire, dans le second exemple (ii), on peut imaginer que la transition entre le centre-ville et la banlieue implique une modification continue de $p(x_{n+1}|x_n)$. Dans cette situation les modèles de Markov triplet à saut peuvent toujours être utilisés, mais ils correspondent moins à la réalité terrain. Par ailleurs, il existe une autre modélisation par un Markov triplet qui, en pratique, semble mieux adaptée à ce genre de cas. En effet, expérimentalement, les chaînes de Markov évidentielles que nous allons présenter dans la suite semblent mieux gérer ce type de non-stationnarité [26]. Cependant l'interprétation de U dans ce dernier modèle, ainsi que les justifications théoriques de son efficacité dans le cas d'une non-stationnarité continue ne sont pas clairement établies. On peut cependant trouver un début d'explication de cela dans [28], dans le cas simple d'une segmentation à deux classes de couples indépendants $(X_1, Y_1), \dots, (X_N, Y_N)$.

2.4.2 Théorie de l'évidence

La théorie de Dempster-Shafer ou théorie de l'évidence [53] constitue la base du modèle de chaînes de Markov cachées évidentielles (HEMC). Cette théorie permet de modéliser des informations que l'on peut avoir sur les éléments d'un ensemble $\Omega = \{\omega_1, \dots, \omega_K\}$ par des « mesures floues », qui peuvent être vues comme des généralisations de la mesure de probabilité

classique. En notant $2^\Omega = \{A_1, \dots, A_q\}$, $q = 2^K$, l'ensemble des sous-ensembles de Ω , définissons une probabilité classique P sur 2^Ω . De par l'axiome de l'additivité, P est entièrement déterminée par ses valeurs sur les singletons $\{\omega_1\}, \dots, \{\omega_K\}$. On dit ainsi que P est définie sur Ω , ce qui est un abus de langage car en toute rigueur P est définie sur 2^Ω . Une mesure floue est une généralisation des probabilités, qui retire l'axiome de l'additivité. Ainsi, cette mesure est définie sur 2^Ω par des fonctions de masses : soit 2^{2^Ω} l'ensemble des sous-ensembles de 2^Ω , une fonction de masse m est une probabilité sur 2^{2^Ω} telle que $m(\{\emptyset\}) = 0$.

Un élément central de cette théorie est la règle de fusion (ou règle de Dempster-Shafer) qui généralise la formule de Bayes des probabilités, en l'appliquant aux masses. Soit m_0, m_1, \dots, m_r , $r + 1$ fonctions de masses définies sur $2^\Omega = \{A_1, \dots, A_q\}$, $q = 2^K$. La règle de fusion est la suivante :

$$m(B) = \frac{1}{1 - H} \sum_{B_0 \cap B_1 \cap \dots \cap B_r = B \neq \emptyset} \left[\prod_{j=0}^r m_j(B_j) \right] \quad (2.42)$$

avec $H = \sum_{B_0 \cap B_1 \cap \dots \cap B_r \neq \emptyset} \left[\prod_{j=0}^r m_j(B_j) \right]$. On notera cette fusion $m = m_0 \otimes m_1 \otimes \dots \otimes m_r$. Enfin, un résultat important est le suivant : si une au moins parmi les fonctions de masses m_0, m_1, \dots, m_r est assimilable à une probabilité, c'est-à-dire qu'elle est nulle en dehors des singletons $\{\omega_1\}, \dots, \{\omega_K\}$, alors $m = m_0 \otimes m_1 \otimes \dots \otimes m_r$ est assimilable à une probabilité. Ce résultat va nous permettre de construire un triplet markovien (qui est donc une loi de probabilité) à partir de masses de Dempster-Shafer.

2.4.3 Définition du modèle de chaînes de Markov évidentielles

Soit $X = (X_1, \dots, X_N)$ un processus caché qui prend ses valeurs à partir d'un ensemble fini de classes $\Omega = (\omega_1, \dots, \omega_K)$ et qui doit être estimé à partir d'un processus observable $Y = (Y_1, \dots, Y_N)$ qui prend ses valeurs dans \mathbb{R} . Notons 2^Ω l'ensemble puissance de Ω et $2^{(\Omega^N)}$ l'ensemble puissance de (Ω^N) . Introduisons un troisième processus aléatoire $U = (U_1, \dots, U_N)$, ou chaque U_n prend ses valeurs dans 2^Ω . Définissons par ailleurs m comme une masse de Dempster-Shafer sur $(2^\Omega)^N$ nulle sur $2^{(\Omega^N)} / (2^\Omega)^N$ et telle que :

$$m(u_1, \dots, u_N) = m(u_1) \prod_{i=1}^{N-1} m(u_{i+1} | u_i) \quad (2.43)$$

Cela signifie que pour tout $n = 1, \dots, N - 1$ et $u_n \in 2^\Omega$, $u_{n+1} \rightarrow m(u_{n+1} | u_n)$ est une masse de Dempster-Shafer sur 2^Ω . La distribution d'une chaîne de Markov évidentielle (HEMC) est une distribution de Markov obtenue par la normalisation de la fonction φ définie sur $\Omega^N \times (2^\Omega)^N$ par :

$$\varphi(x, u, y) = \prod_{n=1}^{N-1} \varphi_n(x_n, u_n, y_n, x_{n+1}, u_{n+1}, y_{n+1}) \quad (2.44)$$

où

$$\varphi_1(x_1, u_1, y_1, x_2, u_2, y_2) = 1_{[x_1 \in u_1, x_2 \in u_2]} m(u_1) m(u_2 | u_1) p(y_1 | x_1) p(y_2 | x_1) \quad (2.45)$$

$$\begin{aligned} \forall n \in [2, N-1], \varphi_n(x_n, u_n, y_n, x_{n+1}, u_{n+1}, y_{n+1}) \\ = 1_{[x_{n+1} \in u_{n+1}]} m(u_{n+1} | u_n) p(y_{n+1} | x_{n+1}) \end{aligned} \quad (2.46)$$

Après normalisation de (2.44) (donc division par $\sum_{(x,u,y)} \varphi(x, u, y)$), on obtient qu'une HEMC est un triplet Markovien vérifiant :

$$\begin{aligned} p(t_1) &= p(u_1) p(x_1 | u_1) p(y_1 | x_1) \\ p(t_{n+1} | t_n) &= p(u_{n+1} | u_n) p(x_{n+1} | u_{n+1}) p(y_{n+1} | x_{n+1}) \end{aligned} \quad (2.47)$$

avec

$$\forall n \in [1, \dots, N], p(x_n | u_n) = \frac{1_{\{x_n \in u_n\}}}{|u_n|} \quad (2.48)$$

$$\forall n \in [1, N-1], p(u_{n+1} | u_n) = \frac{|u_{n+1}| m(u_{n+1} | u_n) \beta_{n+1}^*(x_{n+1}, u_{n+1})}{\sum_{u_{n+1}} |u_{n+1}| m(u_{n+1} | u_n) \beta_{n+1}^*(x_{n+1}, u_{n+1})}, \quad (2.49)$$

l'opérateur $|\cdot|$ dénotant le cardinal d'un ensemble, et :

$$\begin{aligned} \beta_N^*(x_N, u_N) &= 1 \\ \forall n \in [1, N-1], \beta_n^*((x_n, u_n, x_{n+1}, u_{n+1})) \\ &= \sum_{(x_{n+1}, u_{n+1})} \varphi_n(x_n, u_n, x_{n+1}, u_{n+1}) \beta_{n+1}^*(x_{n+1}, u_{n+1}) \end{aligned}$$

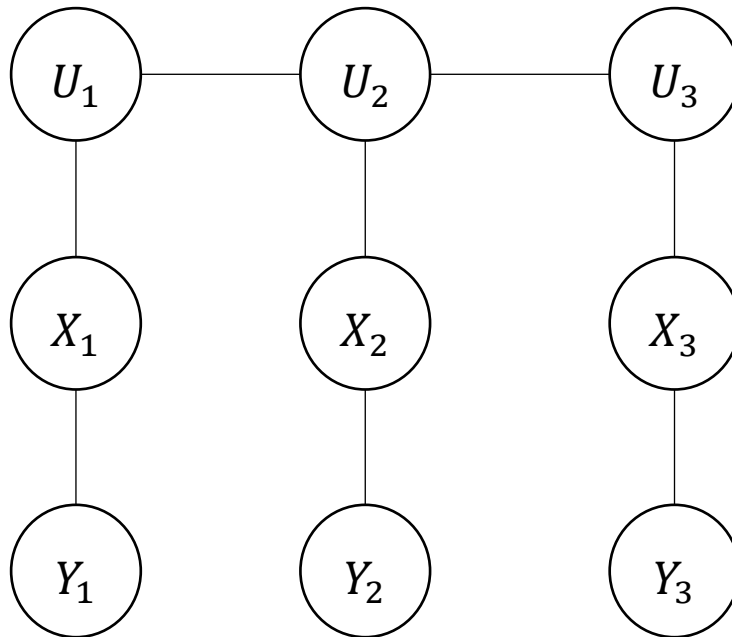


Figure 6. Graphe de probabilité d'une chaîne de Markov évidentielle cachée pour $N=3$

Le calcul de normalisation pour obtenir (2.47) -(2.49) est présenté dans la section A des annexes. Le graphe de probabilité d'une HEMC est présenté en **figure 6**. On peut remarquer que les HEMC sont une HTMC particulière, donc soit $V = (X, U)$, (V, Y) est une chaîne de Markov cachée. Mais on peut aussi s'apercevoir qu'en posant $Z = (X, Y)$, (U, Z) est une chaîne de Markov cachée.

Dans une HEMC stationnaire, la loi est complètement définie une fois que l'on a explicité $p(t_1, t_2)$ qui est égal à $p(t_n, t_{n+1})$ pour tous $n = 2 \dots N - 1$. Il suffit alors, pour définir toute la chaîne, de construire une matrice de taille $(|2^\Omega| - 1) \times (|2^\Omega| - 1)$ pour $p(u_1, u_2)$, en s'assurant que les marginales $p(u_1)$ et $p(x_2)$ soient identiques, et K moyennes et K variances pour $p(y_1|x_1)$. On peut aussi considérer une HEMC homogène, qui sera alors défini par une matrice de transition de taille $(|2^\Omega| - 1) \times (|2^\Omega| - 1)$ pour $p(u_2|u_1)$, $|2^\Omega| - 1$ paramètres pour $p(u_1)$ et enfin K moyennes et K variances pour $p(y_1|x_1)$. La notation $|\cdot|$ représente ici le cardinal d'un ensemble.

2.4.4 Inférence et estimation des paramètres non supervisée

L'estimateur du MPM dans une chaîne de Markov évidentielle peut être calculé si l'on remarque que les HEMC sont une HTMC particulière. Ainsi il suffit d'utiliser (2.22) -(2.25) pour calculer les $p(x_n|y)$. Pour l'estimation des paramètres non supervisée, on pourrait dire la même chose. Cependant les HEMC ont la particularité que les $p(x_n|u_n)$ sont fixés par (2.48) et ne font donc pas partie des paramètres à estimer. Nous allons présenter les versions des algorithmes EM et SEM pour les HEMC ci-dessous.

2.4.4.1 Algorithme EM pour les chaînes de Markov évidentielles

Soit une famille de HEMC (X, U, Y) homogène à bruit gaussien, indexée par un vecteur de paramètres θ . Ce dernier est défini par $\pi_l = p(u_1 = \lambda_l)$, $p_{il} = p(u_2 = \lambda_l | u_1 = \lambda_i)$ et les moyennes et variances μ_k, σ_k^2 associées aux probabilités $p(y_n | x_n = \omega_k)$. Notons :

$$p_{kl,n} = p(x_n = \omega_k, u_n = \lambda_l | y) \quad (2.50)$$

$$p_{ijkl,n,n+1} = p(x_{n+1} = \omega_k, u_{n+1} = \lambda_l, x_n = \omega_i, u_n = \lambda_j | y) \quad (2.51)$$

$$p_{ijkl,n} = p(x_{n+1} = \omega_k, u_{n+1} = \lambda_l | x_n = \omega_i, u_n = \lambda_j, y) \quad (2.52)$$

et posons $C = |2^\Omega| - 1$.

L'algorithme EM pour les HEMC se déroule comme il suit :

- On initialise l'algorithme avec $\theta = \theta^0$
- Pour chaque itération q , on utilise θ^q pour calculer $p_{kl,n}^{(q)}$ avec (2.24) et $p_{ijkl,n,n+1}^{(q)}$ avec (2.29). Ensuite, on calcule θ^{q+1} de la manière suivante :

$$\pi_l^{(q+1)} = \frac{\sum_{n=1}^N \sum_{k=1}^K p_{kl,n}^{(q)}}{N} \quad (2.53)$$

$$p_{il}^{(q+1)} = \frac{\sum_{n=1}^{N-1} \sum_{j=1}^K \sum_{k=1}^K p_{ijkl,n,n+1}^{(q)}}{\sum_{n=1}^{N-1} \sum_{k=1}^K p_{kl,n}^{(q)}} \quad (2.54)$$

$$\mu_k^{(q+1)} = \frac{\sum_{n=1}^N \sum_{l=1}^C p_{kl,n}^{(q)} y_n}{\sum_{n=1}^N \sum_{l=1}^C p_{kl,n}^{(q)}} \quad (2.55)$$

$$\sigma_k^{2,(q+1)} = \frac{\sum_{n=1}^N \sum_{l=1}^C p_{kl,n}^{(q)} (y_n - \mu_k^{(q+1)})^2}{\sum_{n=1}^N \sum_{l=1}^C p_{kl,n}^{(q)}} \quad (2.56)$$

On s'assure par ailleurs, tout au long de l'algorithme, que les $p(x_n | u_n)$ restent fixés par (2.48).

2.4.4.2 Algorithme SEM pour les chaînes de Markov évidentielles

Comme les HEMC sont des HTMC particulières, leur loi *a posteriori* est donnée par (2.34) et (2.35).

L'algorithme SEM pour les HEMC se déroule comme il suit :

- On initialise l'algorithme avec $\theta = \theta^0$
- Pour chaque itération q , on utilise θ^q pour simuler (x^q, u^q) selon (2.34). Ensuite on calcule θ^{q+1} de la manière suivante :

$$\pi_l^{(q+1)} = \frac{\sum_{n=1}^N \sum_{k=1}^K \mathbf{1}_{\{x_n^{(q)} = \omega_k, u_n^{(q)} = \lambda_l\}}}{N} \quad (2.57)$$

$$p_{ijkl}^{(q+1)} = \frac{\sum_{n=1}^{N-1} \sum_{j=1}^K \sum_{k=1}^K \mathbf{1}_{\{x_n^{(q)}=\omega_i, u_n^{(q)}=\lambda_j, x_{n+1}^{(q)}=\omega_k, u_{n+1}^{(q)}=\lambda_l\}}}{\sum_{n=1}^{N-1} \sum_{k=1}^K \mathbf{1}_{\{x_n^{(q)}=\omega_k, u_n^{(q)}=\lambda_l\}}} \quad (2.58)$$

$$\mu_k^{(q+1)} = \frac{\sum_{n=1}^N \mathbf{1}_{\{x_n^{(q)}=\omega_k\}} y_n}{\sum_{n=1}^N \mathbf{1}_{\{x_n^{(q)}=\omega_k\}}} \quad (2.59)$$

$$\sigma_k^{2,(q+1)} = \frac{\sum_{n=1}^N \mathbf{1}_{\{x_n^{(q)}=\omega_k\}} (y_n - \mu_k^{(q+1)})^2}{\sum_{n=1}^N \mathbf{1}_{\{x_n^{(q)}=\omega_k\}}} \quad (2.60)$$

On s'assure par ailleurs, tout au long de l'algorithme, que les $p(x_n|u_n)$ restent fixés par (2.48).

2.5 Chaînes de semi-Markov cachées à temps discret

Sous le terme de chaînes de semi-Markov à temps discret se cachent en fait deux modèles légèrement différents. Le premier modèle, que nous nommons chaînes de semi-Markov cachées classiques, a été développé pour modéliser un temps de séjour quelconque dans une classe donnée, qui est forcément géométrique dans une chaîne de Markov cachée [60] [61] [62]. Ce modèle propose un temps de séjour modélisé explicitement par une famille de lois choisie. Le deuxième modèle désigne un triplet Markovien particulier, construit par l'ajout d'un troisième processus latent à la HMC, qui modélise le temps de séjour *minimal* restant dans une classe [30] [27] avant la prochaine transition. Ces modélisations ne sont pas équivalentes. Les chaînes de semi-Markov cachées (HSMC), que ce soit par l'une des modélisations, ou l'autre, ont montré leur efficacité dans de nombreux domaines [63] tels que la télédétection [64] [65], la reconnaissance d'écriture manuscrite [66] [67], la segmentation d'images [27] [68] [69], la modélisation d'activités physiques [30] [70] [71] et l'analyse de l'ADN. Ainsi dans cette partie nous présentons la chaîne de semi-Markov par Markov triplet et ses différences avec l'approche classique du semi-Markov.

2.5.1 Chaînes de semi-Markov cachées classiques

Soit $X = (X_1, \dots, X_N)$ un processus caché qui prend ses valeurs à partir d'un ensemble fini de classes $\Omega = \{\omega_1, \dots, \omega_K\}$ et qui doit être estimé à partir d'un processus observable $Y = (Y_1, \dots, Y_N)$ qui prend ses valeurs dans \mathbb{R} . Soit un troisième processus $D = (D_1, \dots, D_T)$ à valeur dans $\Lambda = \{1, \dots, M\}$, où M est le temps maximum passé dans un état donné et peut être infini dans certaines applications. On note $X_{[n_1:n_2]} = \omega_k$ pour signifier que l'état ω_k commence au n_1 -ième élément de la chaîne et fini au n_2 -ième, avec une durée de $n_2 - n_1 + 1$. On va définir une nouvelle séquence aléatoire $I = (I_1, \dots, I_T)$ à valeur dans $\Omega = \{\omega_1, \dots, \omega_K\}$ telle que $I_1 = X_{[1:d_1]}$, $I_2 = X_{[d_1+1:d_1+d_2]}$, ..., $I_T = X_{[\sum_{t=1}^{T-1} d_t+1:\sum_{t=1}^T d_t]}$, et $\sum_{t=1}^T d_t = N$. Une chaîne de semi-Markov classique est telle que (I, D) soit de Markov [72] :

$$p(i, d) = p(i_1, d_1) p(Y_{[1:d_1]} | i_1) \prod_{t=1}^{T-1} p(i_{t+1}, d_{t+1} | i_t, d_t) p\left(Y_{[\sum_{k=1}^t d_{k+1} : \sum_{k=1}^{t+1} d_k]} | i_{t+1}\right) \quad (2.61)$$

De façon équivalente, on peut aussi définir la séquence aléatoire suivante : $S = (S_1, \dots, S_N)$ telle que $S_n = n - \sum_{l=1}^{t-1} d_l$, $t - 1$ étant le nombre de changement d'états (de transition d'un état vers un autre différent) ayant eu lieu jusqu'à t . Une chaîne de semi-Markov classique est alors telle que (X, S) soit de Markov [72].

Une chaîne de semi-Markov classique homogène est totalement définie par : la probabilité initiale $\pi_{k,g} = p(i_1 = \omega_k, d_1 = g) = p(X_{[1:g]} = \omega_k)$, les transitions $p_{(k,h),(l,g)} = p(i_{t+1} = \omega_l, d_{t+1} = h | i_t = \omega_k, d_t = g) = p(X_{[n+1:n+h]} = \omega_l | X_{[n-g+1:n]} = \omega_k)$ et la loi des observations $b_{lh}(Y_{[n+1:n+h]}) = p(Y_{[n+1:n+h]} | X_{[n+1:n+h]} = \omega_l)$. Dans un semi-Markov classique, il faut que $p_{(k,h),(k,d)} = 0$. La loi de d représente ainsi la loi du temps exact passé dans chaque état ω_k . Les algorithmes de calcul du MPM et l'estimation des paramètres avec EM sont possibles dans les chaînes de semi-Markov classiques [72].

On peut remarquer que cette définition se base sur des processus stochastiques à temps discret. Il existe aussi une définition plus générale des chaînes de semi-Markov avec des processus stochastiques à temps continu [73].

2.5.2 Chaînes de semi-Markov par Markov triplet

Soit $X = (X_1, \dots, X_N)$ un processus caché qui prend ses valeurs à partir d'un ensemble fini de classes $\Omega = \{\omega_1, \dots, \omega_K\}$ et qui doit être estimé à partir d'un processus observable $Y = (Y_1, \dots, Y_N)$ qui prend ses valeurs dans \mathbb{R} . Soit un troisième processus $U = (U_1, \dots, U_N)$ à valeur dans $\Lambda = \{0, \dots, M - 1\}$, tel que $U_n = e \neq 0$ implique que X_n va rester dans le même état au moins pour un temps e . Ainsi $(X_n, U_n) = (x_n, e)$ implique que $x_n = x_{n+1} = x_{n+2} = \dots = x_{n+e}$. Pour $e = 0$, on peut avoir $x_{n+1} = x_n$ ou non. Pour a, b, c, e posons $\delta_{(a,b)}(c, e) = 1$ si $(a, b) = (c, e)$, et $\delta_{(a,b)}(c, e) = 0$ si $(a, b) \neq (c, e)$. Nous appelons chaînes de semi-Markov triplet cachées (HSMC) les HTMC $T = (X, U, Y)$ vérifiant :

$$p(x_{n+1}, u_{n+1} | x_n, u_n) = \begin{cases} \delta_{(x_{n+1}, u_{n+1})}(x_n, u_n - 1) & \text{si } u_n > 0 \\ p^*(x_{n+1}, u_{n+1} | x_n, u_n = 0) & \text{si } u_n = 0 \end{cases} \quad (2.62)$$

où p^* est une loi quelconque sur $\Omega \times \Lambda$

Les transitions (2.62) s'interprètent de la manière suivante : pour $u_n = 0$, le prochain couple (x_{n+1}, u_{n+1}) est généré selon $p^*(x_{n+1}, u_{n+1} | x_n, u_n = 0)$, alors que pour $u_n \neq 0$, (x_{n+1}, u_{n+1}) est déterministe et $x_{n+1} = x_n$, $u_{n+1} = u_n - 1$. A la différence des chaînes de semi-Markov classiques, u_n est le temps minimum restant dans l'état x_n . Le graphe de probabilité de la HSMC est le même que celui de la HTMC, présenté à la **figure 5**.

Les HSMC étant bien des HTMC particulières, l'estimateur du MPM est obtenu par (2.22)-(2.25) qui permettent de calculer les $p(x_n|y)$. Pour l'estimation des paramètres non supervisée, les algorithmes EM et SEM pour les HTMC, présentés en section 2.3.3 fonctionnent également pour les HSMC. En particulier, si on initialise $p(x_{n+1}, u_{n+1}|x_n, u_n)$ de sorte que cette probabilité respecte (2.62), alors les équations (2.31) et (2.37) nous garantissent qu'à chaque itération, le nouveau $p(x_{n+1}, u_{n+1}|x_n, u_n)$ calculé respecte (2.62).

2.5.3 Comparaison des lois de temps de séjour

Dans les HSMC classiques, on a accès directement à la loi du temps de séjour dans chaque état ω_l , c'est en effet la loi $p(X_{[n+1:n+d]} = \omega_l | X_{[n-h+1:n]} = \omega_k)$, définie à partir du processus aléatoire D et supposée invariante selon n , dans le cas d'une HSMC homogène. Elle dépend de l'état précédent ω_k et, en général, de la durée h passée dans ce dernier. Chaque D est à valeur dans $\Lambda = \{1, \dots, M\}$, où M est le temps maximum passé dans un état donné et peut être infini dans certaines applications. On a donc totale liberté sur la loi et le support de D , ce qui est un avantage quand on veut modéliser un temps de séjour quelconque. Cependant la complexité algorithmique du calcul du MPM et d'une itération de l'algorithme d'estimation des paramètres dépendent du support de D , ainsi considérer des temps de séjour maximum très grand peut provoquer des coûts de calcul énormes [72]. Dans les HSMC par TMC, on définit $U = (U_1, \dots, U_N)$, sur $\Lambda = \{0, \dots, M-1\}$, où U_n est le temps minimum restant dans l'état x_n . Comme pour les HSMC classiques, la complexité algorithmique du calcul du MPM et d'une itération de l'estimation des paramètres dépend du support des U_n . Cependant, comme U_n est le temps minimum restant dans un état, un support fini pour les U_n engendre une loi du temps de séjour exact avec un support infini, ce qui permet de prendre en compte des temps de séjour exact très long, avec une complexité algorithmique raisonnable. Cependant le défaut de cette modélisation est que l'on ne spécifie pas la loi du temps de séjour exact, celle-ci est induite par notre définition de la loi de $U = (U_1, \dots, U_N)$ et, contrairement à la HSMC classique, elle a une forme particulière.

Considérons une HSMC par TMC homogène, et introduisons les notations suivantes : $p_{k,l} = p(x_{n+1} = \omega_l | x_n = \omega_k, u_n = 0)$, $q_{k,l,m} = p(u_{n+1} = m | x_{n+1} = \omega_l, x_n = \omega_k, u_n = 0)$, et $B^M(j, k)$ l'ensemble des compositions rétreintes à M de k à j parts, c'est-à-dire que [74]:

$$b(j, k) \in B^M(j, k) \Leftrightarrow \{b(j, k) = (b_1, \dots, b_j), \sum_{r=1}^j b_r = k, \forall r \in [1, \dots, j] b_r \in \mathbb{N} \text{ et } b_r \leq M\} \quad (2.63)$$

On remarque en premier que dans le cas de la HSMC triplet présenté en section 2.5.2, on a :

$$p(X_{[n+1:n+g]} = \omega_l | X_{[n-h+1:n]} = \omega_k) = p(X_{[n+1:n+g]} = \omega_l | X_n = \omega_k) \quad (2.64)$$

Soit le temps de séjour dans l'état courant ne dépend pas du temps de séjour dans l'état précédent. On peut maintenant expliciter $a_{k,l}(g) = p(X_{[n+1:n+g]} = \omega_l | X_n = \omega_k)$ [75]:

$$a_{k,l}(g) = p_{k,l}(1 - p_{k,k}) \sum_{j=1}^g p_{k,k}^{j-1} \sum_{b(j,d) \in B^M(j,g)} q_{k,l,(b_1-1)} \prod_{m=2}^j q_{k,k,(b_m-1)} \quad (2.65)$$

En remarquant qu'il ne peut exister de composition restreintes $B^M(1, g)$ pour $g > M$, $B^M(2, g)$ pour $g > 2M, \dots, B^M(n, g)$ pour $d > nM$, on peut ainsi montrer que :

Pour $(n-1)M < g \leq nM$, $n \in \mathbb{N}$:

$$a_{kl}(g) = p_{kk}^{(n-1)} p_{kl}(1 - p_{kk}) \sum_{j=n}^g p_{k,k}^{j-n} \sum_{b(j,d) \in B^M(j,g)} q_{k,l,(b_1-1)} \prod_{m=2}^j q_{k,k,(b_m-1)} \quad (2.66)$$

Ainsi la loi du temps de séjour dans une HSMC par TMC a une forme « pseudo-géométrique », elle n'est pas quelconque comme dans les HSMC classiques et s'affaiblit par paliers au fur et à mesure que l'on s'éloigne du maximum de temps de séjour minimum M . De plus elle ne dépend pas de la durée de l'état précédent, ce qui fait que les HSMC par TMC sont moins générales que les HSMC classique. Cependant le dernier point, à savoir l'indépendance du temps de séjour dans un état par rapport à la durée de l'état précédent, est assez facile à surmonter en introduisant un second U représentant le temps passé dans une classe dans les HSMC par TMC.

2.5.4 Semi-Markov par Markov triplet et segmentation d'images

Nous présentons ici une étude rapide pour déterminer l'intérêt des HSMC par TMC dans la segmentation d'images non supervisée. Celles-ci ont déjà été utilisés dans ce cadre [27] [68] [69], mais souvent d'autres modélisations ont été ajoutées, si bien qu'il est difficile de voir, expérimentalement, leur apport dans ce domaine. Nous proposons ainsi une simple étude de segmentation d'une image synthétique à deux classes, bruitée par trois jeux de bruits gaussiens de plus en plus fort. Ces bruits sont construits en bruitant les pixels des images avec, à chaque fois, deux gaussiennes indépendantes de même variance, suivant la classe du pixel (noir ou blanc). Pour augmenter la force du bruit entre chaque couple de gaussiennes, nous conservons la même variance et nous réduisons l'écart entre les deux moyennes. Nous avons choisi pour cette étude une image contenant de larges zones homogènes de même classe. L'idée derrière cela est que les HSMC permettent de mieux représenter des lois de séjour « longs » dans un état car ils n'imposent pas une loi géométrique comme les HMC. Ainsi de très grandes zones de même classe devraient favoriser un modèle de HSMC. Par ailleurs la HSMC étant plus général que la HMC, on peut être assuré, sous conditions que l'estimation des paramètres est correcte, que celui-ci fera aussi bien que la HMC. Nous pensons donc qu'il est plus intéressant de regarder des cas où le HSMC a de grandes chances d'améliorer la HMC. Les bruits considérés dans cette étude sont très forts ; en effet, les HMC sont déjà très performants dans le cas d'images assez homogènes [15]. L'idée pour se placer dans un cas où la HSMC serait plus

intéressant que la HMC est donc de renforcer le bruit pour augmenter l'influence de la loi *a priori* que nous pensons être plus adaptée à l'image choisie dans la HSMC que dans la HMC. Enfin nous avons aussi voulu regarder l'influence du choix de M , le maximum du temps minimum de séjour dans un état. Pour cela nous segmentons l'image avec une HMC et quatre HSMC avec $M = 2, 5, 10, 20$. Elles seront notées respectivement HSMC-2, HSMC-5, HSMC-10, HSMC-20. Les trois jeux de bruits choisis sont $\{N(0,1) - N(0.4,1)\}$, $\{N(0,1) - N(0.3,1)\}$, et $\{N(0,1) - N(0.2,1)\}$. La segmentation est réalisée avec l'estimateur MPM. Les paramètres sont estimés avec l'algorithme EM. Nous estimons les valeurs initiales de $p(x_1 = \omega_i)$ et $p(x_2 = \omega_k | x_1 = \omega_i)$ à partir d'une segmentation à deux classes, réalisée avec l'algorithme du K-means utilisé sur les données bruitées. Celui-ci est initialisé avec la méthode « K-moyennes++ » [76], dont le principe est de répartir les k points des moyennes initiales de la manière suivante : le point de moyenne initial du premier cluster est choisi aléatoirement parmi les données, puis chaque point de moyenne initiale est choisi parmi les points restants, avec une probabilité proportionnelle au carré de la distance entre le point et le cluster le plus proche. Pour les HSMC nous initialisons $p(x_2 = \omega_k | x_1 = \omega_i, u_1 = 0) = p(x_2 = \omega_k | x_1 = \omega_i)$, puis $p(u_1 | x_1 = \omega_i)$ et $p(u_2 | x_2 = \omega_k, x_1 = \omega_i, u_1 = 0)$ comme des lois uniformes sur $\{0, 1, \dots, M - 1\}$. Enfin pour transformer les images bidimensionnelles en signaux unidimensionnels, on utilise le parcours de Peano [77] [78], qui sera présenté avec plus de détail en section 4.1. Les résultats des différentes segmentations sont présentés dans les **figures 7, 8, 9**, avec les taux d'erreur pour chaque bruit et chaque modèle dans le **tableau 1**. Cette segmentation rapide permet d'observer deux choses : d'une part la HSMC a bien un intérêt en segmentation d'images, elle permet d'améliorer la HMC dans des cas d'images avec de large zones homogènes de même classe et très bruitées. En effet comme nous l'avions supposé, dans le cas d'un bruit trop faible comme $\{N(0,1) - N(0.4,1)\}$, la HSMC et la HMC sont équivalents, car la loi *a priori* compte moins dans ce cas. Cependant quand le bruit devient très fort, $\{N(0,1) - N(0.3,1)\}$ et $\{N(0,1) - N(0.2,1)\}$, la HSMC commence à avoir un intérêt par rapport au HMC. On peut par ailleurs remarquer que prendre un temps de séjour M plus long, a tendance à améliorer les résultats de la HSMC. C'est attendu par la théorie car une HSMC avec un M plus grand est plus général qu'une HSMC avec un M plus petit. Par ailleurs, ces résultats montrent un très bon comportement de EM, même avec un M relativement élevé. Enfin, notons que les bruits utilisés dans cette étude sont très forts, en particulier le dernier $\{N(0,1) - N(0.2,1)\}$. Au vu du peu de différences entre les performances de la HSMC-10 et la HSMC-20 (la HSMC-20 n'améliore la HSMC-10 que pour le dernier bruit), nous avons choisi de nous limiter à prendre $M = 10$ dans la suite de nos travaux. En effet, nous comptons ajouter aux HSMC d'autres modélisation par TMC dans des triplets plus généraux, et il convient alors de limiter le nombre d'état de U pour que la complexité en temps et en espace n'augmente pas trop.

Tableau 1. Taux d'erreur (en %) de la segmentation d'image synthétique à deux classes pour chaque modèle et chaque bruit considérés

Bruit	HMC	HSMC-2	HSMC-5	HSMC-10	HSMC-20
$N(0,1) - N(0.4,1)$	4.5%	4.5%	4.5%	4.5%	4.5%
$N(0,1) - N(0.3,1)$	30.9%	34.9%	28.0%	6.2%	6.2%
$N(0,1) - N(0.2,1)$	46.0%	45.6%	44.7%	34.5%	22.3%

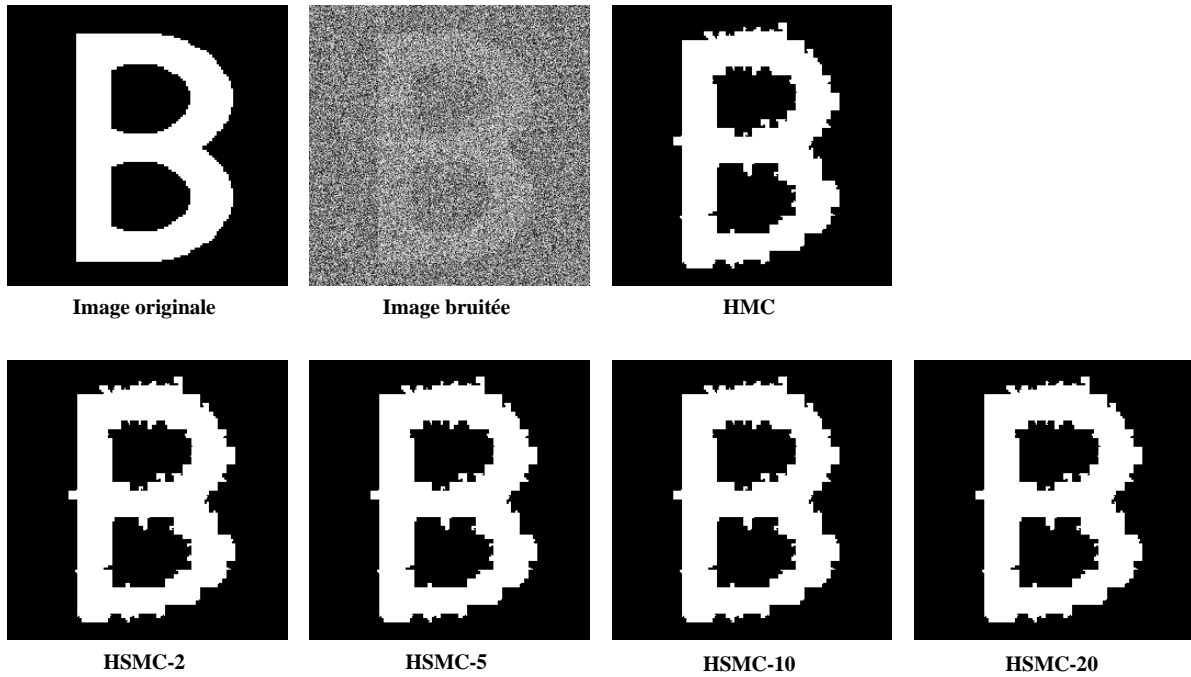


Figure 7. Segmentation d'une image synthétique à deux classes (Image originale) bruitée par un bruit gaussien $N(0,1)$, $N(0.4,1)$ (Image bruitée). HMC, HSMC-2, HSMC-5, HSMC-10 et HSMC-20 : segmentation non supervisée avec les modèles en question.

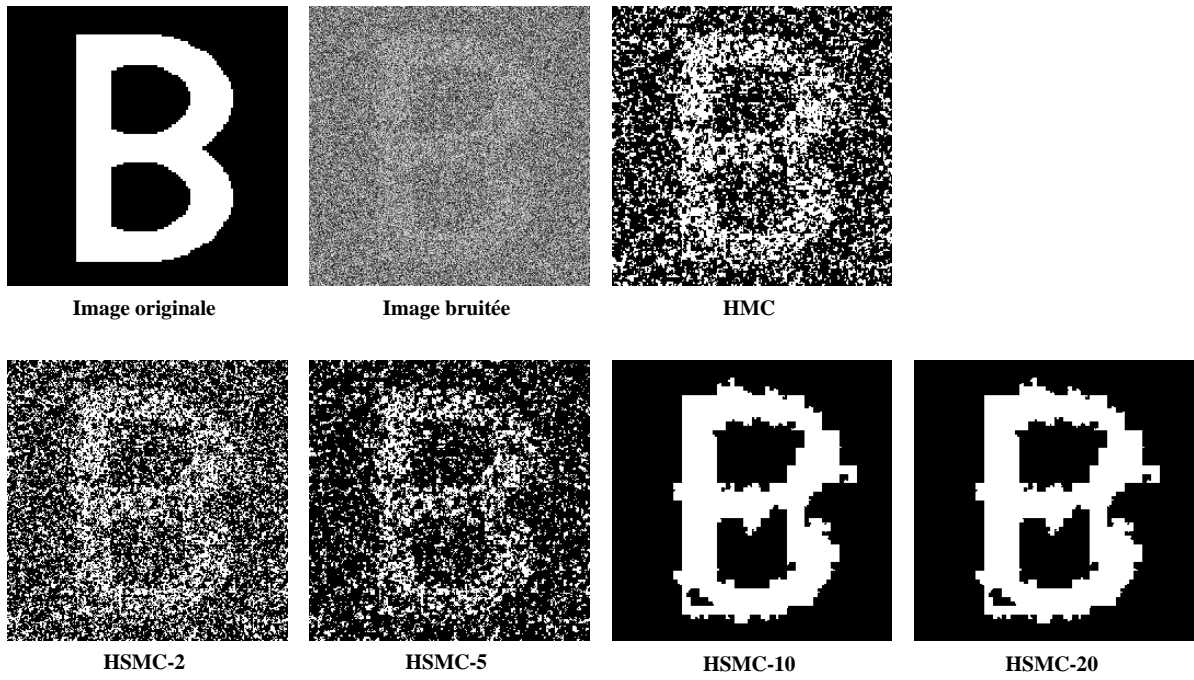


Figure 8. Segmentation d'une image synthétique à deux classes (Image originale) bruitée par un bruit gaussien $N(0, 1)$, $N(0, 3, 1)$ (Image bruitée). HMC, HSMC-2, HSMC-5, HSMC-10 et HSMC-20 : segmentation non supervisée avec les modèles en question.

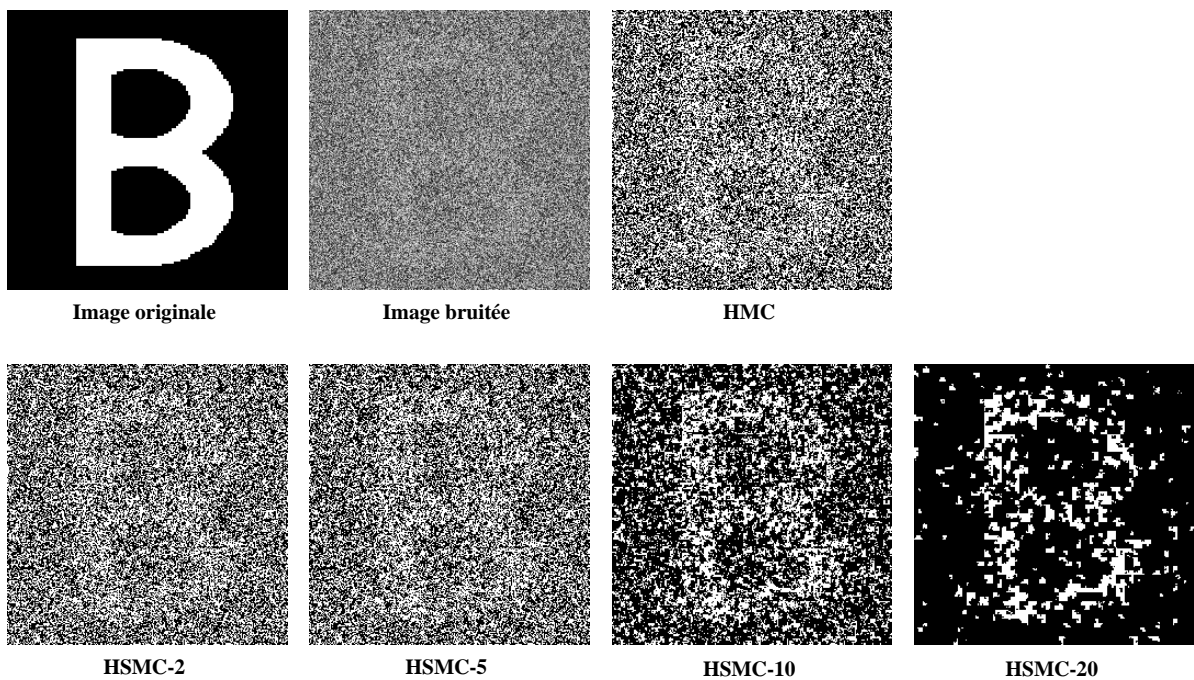


Figure 9 Segmentation d'une image synthétique à deux classes (Image originale) bruitée par un bruit gaussien $N(0, 1)$, $N(0, 2, 1)$ (Image bruitée). HMC, HSMC-2, HSMC-5, HSMC-10 et HSMC-20 : segmentation non supervisée avec les modèles en question.

2.6 Conclusion

Dans cette partie nous avons exposé les modèles de PMC qui généralisent les HMC puis les modèles de TMC qui généralisent les PMC. Mais nous avons aussi exposé un problème d'identifiabilité dans les TMC générales qui diminuait leur intérêt dans le cas de la segmentation non-supervisée. Nous avons ensuite discuté de plusieurs TMC particulières qui ne présentaient pas cet inconvénient, et qui avaient déjà fait leurs preuves en segmentation d'image : les HEMC permettant de segmenter des données non stationnaires et les HSMC permettant de considérer une loi de temps de séjour dans un état différente d'une loi géométrique. Les HSMC par TMC diffèrent des HSMC classiques car elles modélisent le temps de séjour minimal dans un état, plutôt que le temps exact. Ainsi elles permettent de considérer des temps de séjour exact plus long sans augmenter la complexité algorithmique des calculs, mais contrairement aux HSMC classiques, la loi du temps de séjour exact dans une HSMC par TMC n'est pas quelconque. On ne peut donc pas modéliser n'importe quelle loi de séjour exact avec une HSMC par TMC. Notre contribution à ce stade se trouve dans la définition du modèle de HEMC par une loi initiale et des transitions classiques, contrairement à ce qui avait été fait avant. Cette nouvelle définition du HEMC facilite grandement l'ajout de nouveaux processus au triplet évidentiel. Enfin nous avons mis en lumière un intérêt des HSMC par TMC pour la segmentation d'images homogènes très bruitées. On peut alors se demander si combiner les HEMC et les HSMC aurait un intérêt en segmentation d'images. C'est la question à laquelle nous allons tenter de répondre dans le prochain chapitre.

Chapitre 3. Chaînes de semi-Markov évidentielles

Les chaînes de Markov triplets offrent un contexte théorique très large. En effet dans la définition d'une telle chaîne, aucune spécificité sur le processus U n'est mentionné. Ainsi on peut prendre des processus U à nombre arbitraire de classes et avec une loi arbitraire elle aussi. En particulier, il est possible de considérer un $U = (U^1, \dots, U^r)$, avec chaque $U^i = (U_1^i, \dots, U_N^i)$ modélisant une propriété particulière de (X, Y) . On peut ainsi aisément combiner les modèles vus précédemment dans un triplet plus général. Jusqu'à présent seuls deux TMC de la sorte ont été étudiés. Dans [27] les auteurs considèrent $U = (U^1, U^2)$, avec U^1 modélisant les sauts de la chaîne X et U^2 modélisant la semi-Markovianité de X . Appliqué à la segmentation d'images, le modèle proposé améliore simultanément les modèles de Markov à sauts, (X, U^1, Y) et les chaînes de semi-Markov (X, U^2, Y) . Dans [42] les auteurs considèrent $U = (U^1, U^2)$, où U^1 modélise le fait que le bruit soit assimilé à un mélange gaussien, alors que U^2 modélise la semi-markovianité de X . Ils appliquent le modèle à la reconnaissance d'activités physiques. Nous nous intéressons ici à la combinaison entre le semi-Markov et la théorie de l'évidence dans un triplet markovien original $U = (U^1, U^2)$, où la première chaîne U^1 modélise la nature évidentielle du couple (X, Y) , alors que la deuxième chaîne U^2 modélise la nature semi-markovienne de X . On va voir qu'il est possible de créer plusieurs triplets fusionnant la théorie de Demspster-Shafer et la semi-markovianité. En effet on pourrait partir d'une chaîne de semi-Markov et rajouter un U modélisant la théorie de l'évidence. Cette approche a déjà été théorisée [79] sans que le modèle ne soit testé en pratique. On peut aussi partir d'une chaîne évidentielle et rajouter un U modélisant la semi-markovianité. C'est ce que nous avons choisi de faire ici. Nous allons ainsi voir qu'il y a deux possibilités distinctes de faire cela.

Dans un premier temps nous présentons la première manière de construire le modèle de ce nouveau triplet, appelé chaînes de semi-Markov évidentielle cachée (HESMC-Z), qui généralise simultanément les HEMC [26], et les HSMC [60] [62] [72]. Nous montrons que le calcul du MPM est possible et nous proposons une méthode d'estimation des paramètres en non supervisé de ce modèle, basée sur l'algorithme EM. Nous proposons ensuite une variante de la HESMC avec une loi différente (HESMC-V) et nous montrons que le calcul du MPM et l'estimation des paramètres par EM sont aussi possibles dans ce cas. Dans un troisième temps, nous validons nos modèles sur des segmentations non supervisées d'images synthétiques. Enfin nous présentons l'approche de chaînes évidentielles de semi-Markov cachées (HSEMC) [79], et les différences de ces dernières avec les HESMC.

3.1 Chaînes de semi-Markov évidentielles HESMC-Z

La première manière de fusionner les HEMC et les HSMC dans un triplet markovien plus général est de considérer une HEMC (U^1, X, Y) , et ajouter un processus U^2 tel que la loi de U^1 soit semi-markovienne. On va ainsi présenter la construction d'un tel triplet, et montrer que le calcul du MPM et l'estimation des paramètres par EM restent possibles.

3.1.1 Définition du modèle

Soit $X = (X_1, \dots, X_N)$ un processus caché qui prend ses valeurs à partir d'un ensemble fini de classes $\Omega = \{\omega_1, \dots, \omega_K\}$: et qui doit être estimé à partir d'un processus observable $Y = (Y_1, \dots, Y_N)$ qui prend ses valeurs dans \mathbb{R} . Notons 2^Ω l'ensemble puissance de Ω . Introduisons un troisième processus aléatoire, la séquence $U^1 = (U_1^1, \dots, U_N^1)$ avec chaque U_n^1 à valeur dans 2^Ω de telle sorte que (X, U^1, Y) forme une chaîne de Markov évidentielle, dont la construction est présentée dans la section 2.4.3. On sait alors qu'en posant $Z = (X, Y)$, (U^1, Z) est une HMC. On peut ainsi appliquer les mêmes transformations que dans la section 2.5.2 pour introduire la semi-markovianité du couple (U^1, Z) . Afin d'obtenir une chaîne de semi-Markov évidentielle, nous construisons une séquence aléatoire $U^2 = (U_1^2, \dots, U_N^2)$ avec chaque U_n^2 à valeur dans $\Lambda = \{0, 1, \dots, M-1\}$, tel que (2.62) soit vérifié avec (U^1, U^2) à la place de (X, U) . La distribution du HESMC-Z $(U^2, U^1, Z) = (X, U^1, U^2, Y)$ est celle d'un HTMC définie par :

$$\begin{aligned} p(t_1) &= p(u_1^1, u_1^2) p(x_1 | u_1^1) p(y_1 | x_1) \\ p(t_{n+1} | t_n) &= p(u_{n+1}^1, u_{n+1}^2 | u_n^1, u_n^2) p(x_{n+1} | u_{n+1}^1) p(y_{n+1} | x_{n+1}) \end{aligned} \quad (3.1)$$

avec :

$$p(u_{n+1}^1, u_{n+1}^2 | u_n^1, u_n^2) = \begin{cases} \delta_{(u_{n+1}^1, u_{n+1}^2)}(u_n^1, u_n^2 - 1) & \text{si } u_n^2 > 0 \\ p^*(u_{n+1}^1, u_{n+1}^2 | u_n^1, u_n^2 = 0) & \text{si } u_n^2 = 0 \end{cases} \quad (3.2)$$

et :

$$\forall n \in \{1, \dots, N\}, p(x_n | u_n^1) = \frac{1_{\{x_n \in u_n^1\}}}{|u_n^1|} \quad (3.3)$$

Le graphe de probabilité d'une HESMC-Z est présenté à la **figure 10**.

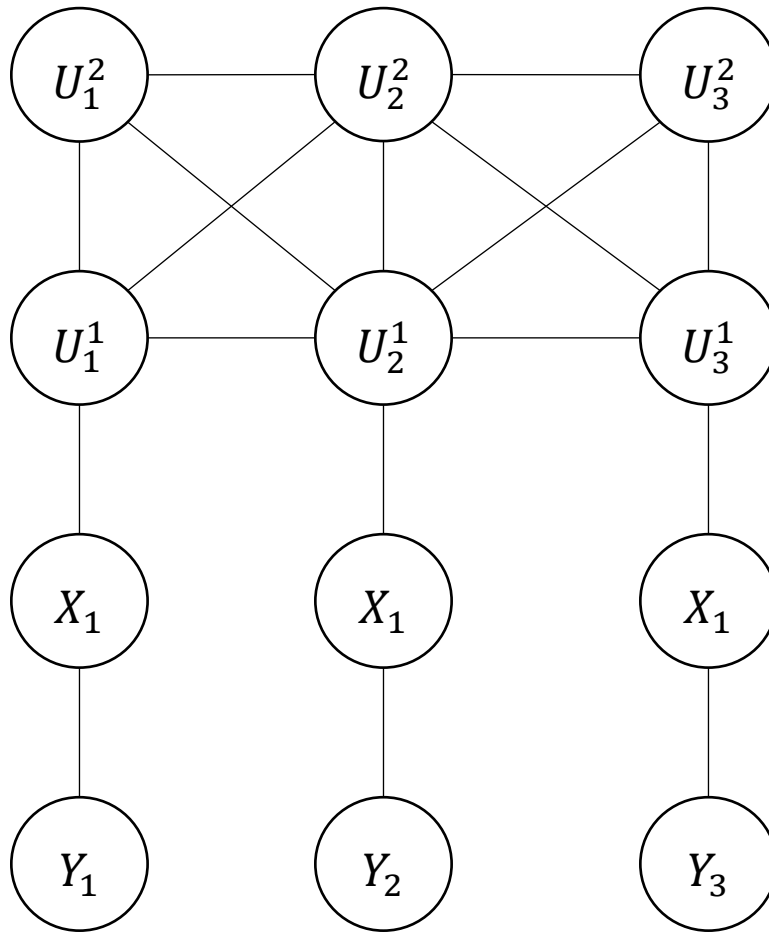


Figure 10. Graphe de probabilité d'un HESMC-Z pour $N=3$

La HESMC-Z proposée étend bien simultanément les HSMC et les HEMC. En effet si on prend, pour construire la partie évidentielle d'un HESMC-Z, une masse de Dempster-Shafer m nulle en dehors des singletons $\{\omega_1\}, \dots, \{\omega_N\}$ dans (2.43), alors la HESMC-Z se réduit à un HSMC. De même si on prend $p(u_n^2 \neq 0) = 0$ pour tout $n = 1, \dots, N$, alors la HESMC-Z se réduit à un HEMC.

3.1.2 Inférence et estimation des paramètres

L'estimateur du MPM dans une HESMC-Z peut être calculé si l'on remarque que les HESMC sont une HTMC particulière. Ainsi il suffit d'utiliser (2.22) -(2.25) et calculer les $p(x_n|y)$ en posant $u_n = (u_n^1, u_n^2)$. Pour l'estimation des paramètres non supervisée on pourrait dire la même chose. Cependant les HESMC-Z ont, comme les HEMC, la particularité que les $p(x_n|u_n)$ sont fixés par (3.3) et ne font donc pas partie des paramètres à estimer. Nous allons présenter une version de l'algorithme EM adapté au HESMC-Z, que nous avons utilisé pour nos expériences dans la suite. Soit une famille de HESMC-Z (X, U^1, U^2, Y) homogènes à bruit

gaussien, indexée par un vecteur de paramètres θ . Ce dernier est défini par $\pi_{lp} = p(u_1^1 = l, u_1^2 = p)$, $p_{jlp} = p(u_{n+1}^1 = l, u_{n+1}^2 = p | u_n^1 = j, u_n^2 = r)$ et les moyennes et variances μ_k, σ_k^2 associées aux distributions $p(y_n | x_n = \omega_k)$. Notons :

$$p_{klp,n} = p(x_n = \omega_k, u_n^1 = l, u_n^2 = p | y) \quad (3.4)$$

$$p_{ijklpr,n,n+1} = p(x_{n+1} = \omega_k, u_{n+1}^1 = l, u_{n+1}^2 = p, x_n = \omega_i, u_n^1 = j, u_n^2 = r | y) \quad (3.5)$$

$$p_{ijklpr,n} = p(x_{n+1} = \omega_k, u_{n+1}^1 = l, u_{n+1}^2 = p | x_n = \omega_i, u_n^1 = j, u_n^2 = r, y) \quad (3.6)$$

Et posons $C = |2^\Omega| - 1$

L'algorithme EM pour les HESMC-Z peut être déduit directement de celui pour les HTMC. Il se déroule comme suit :

- On initialise l'algorithme avec $\theta = \theta^0$
- Pour chaque itération q on utilise θ^q pour calculer $p_{klp,n}^{(q)}$ avec (2.24) et $p_{ijklpr,n,n+1}^{(q)}$ avec (2.29), en posant $u_n = (u_n^1, u_n^2)$. Ensuite on calcule θ^{q+1} de la manière suivante :

$$\pi_{lp}^{(q+1)} = \frac{\sum_{n=1}^N \sum_{k=1}^K p_{klp,n}^{(q)}}{N} \quad (3.7)$$

$$p_{jlp}^{(q+1)} = \frac{\sum_{n=1}^{N-1} \sum_{i=1}^K \sum_{k=1}^K p_{ijklpr,n,n+1}^{(q)}}{\sum_{n=1}^{N-1} \sum_{k=1}^K p_{klp,n}^{(q)}} \quad (3.8)$$

$$\mu_k^{(q+1)} = \frac{\sum_{n=1}^N \sum_{l=1}^C \sum_{p=0}^{D-1} p_{klp,n}^{(q)} y_n}{\sum_{n=1}^N \sum_{l=1}^C \sum_{p=0}^{D-1} p_{klp,n}^{(q)}} \quad (3.9)$$

$$\sigma_k^{2,(q+1)} = \frac{\sum_{n=1}^N \sum_{l=1}^C \sum_{p=0}^{D-1} p_{klp,n}^{(q)} (y_n - \mu_k^{(q+1)})^2}{\sum_{n=1}^N \sum_{l=1}^C \sum_{p=0}^{D-1} p_{klp,n}^{(q)}} \quad (3.10)$$

On s'assure par ailleurs, tout au long de l'algorithme, que pour tout $n \in [1, \dots, N]$, $p(x_n | u_n)$ reste fixé par (3.3). De plus on peut remarquer que si on initialise p_{jlpq} de sorte que cette probabilité respecte (3.2), alors l'équation (3.8) nous garantit qu'à chaque itération le nouveau p_{jlpq} calculé respecte (3.2).

Le calcul du MPM, ainsi qu'une itération de l'algorithme EM ont tous deux une complexité de $O(K^2 C^2 M^2 N)$ en temps, si on suppose une complexité en espace de $O(KCMN)$.

3.2 Chaînes de semi-Markov évidentielles HESMC-V

La deuxième manière de fusionner les HEMC et les HSMC dans un triplet markovien plus général est de considérer une HEMC (U^1, X, Y) , et ajouter un processus U^2 tel que la loi couple

(U^1, X) soit semi-Markovienne. Nous présentons dans cette partie la construction d'un tel triplet, et montrons que comme dans la HESMC-Z, le calcul du MPM et l'estimation des paramètres par EM restent possibles, avec la même complexité en temps que le HESMC-Z.

3.2.1 Définition du modèle

Soit $X = (X_1, \dots, X_N)$ un processus caché qui prend ses valeurs à partir d'un ensemble fini de classes $\Omega = \{\omega_1, \dots, \omega_K\}$: et qui doit être estimé à partir d'un processus observable $Y = (Y_1, \dots, Y_N)$ qui prend ses valeurs dans \mathbb{R} . Notons 2^Ω l'ensemble puissance de Ω . Comme dans la section précédente, introduisons un troisième processus aléatoire, la séquence $U^1 = (U_1^1, \dots, U_N^1)$ avec chaque U_n^1 à valeur dans 2^Ω de telle sorte que (X, U^1, Y) forme une chaîne de Markov évidentielle, dont la construction est présentée dans la section 2.4.3. Cette fois ci on va s'appuyer sur le fait qu'en posant $V = (X, U^1)$, (V, Y) est une HMC. On peut alors appliquer les mêmes transformations que dans la section 2.5.2 pour introduire la semi-markovianité au couple (V, Y) . Afin d'obtenir notre chaîne de semi-Markov évidentielles, nous construisons une séquence aléatoire $U^2 = (U_1^2, \dots, U_N^2)$ avec chaque U_n^2 à valeur dans $\Lambda = \{0, 1, \dots, M-1\}$, tel que (2.62) soit vérifié avec (V, U^2) à la place de (X, U) . La distribution du HESMC-V $(U^2, V, Y) = (U^2, U^1, X, Y)$ est celle d'une HTMC définie par :

$$\begin{aligned} p(t_1) &= p(u_1^1, u_1^2, x_1)p(y_1|x_1) \\ p(t_{n+1}|t_n) &= p(u_{n+1}^1, u_{n+1}^2, x_{n+1}|u_n^1, u_n^2, x_n)p(y_{n+1}|x_{n+1}), \end{aligned} \quad (3.11)$$

avec

$$p(u_{n+1}^1, u_{n+1}^2, x_{n+1}|u_n^1, u_n^2, x_n) = \begin{cases} \delta_{(u_{n+1}^1, x_{n+1}, u_{n+1}^2)}(u_n^1, x_n, u_n^2 - 1) & \text{si } u_n^2 > 0 \\ p^*(u_{n+1}^1, u_{n+1}^2, x_{n+1}|u_n^1, x_n, u_n^2 = 0) & \text{si } u_n^2 = 0 \end{cases} \quad (3.12)$$

et

$$p^*(u_{n+1}^1, u_{n+1}^2, x_{n+1}|u_n^1, x_n, u_n^2 = 0) = p^*(u_{n+1}^1, u_{n+1}^2|u_n^1, x_n, u_n^2 = 0)p(x_{n+1}|u_{n+1}^1) \quad (3.13)$$

où

$$p(x_n|u_n^1) = \frac{1_{\{x_n \in u_n^1\}}}{|u_n^1|} \quad (3.14)$$

Le graphe de probabilité d'une HESMC-V est présenté à la **figure 11**.

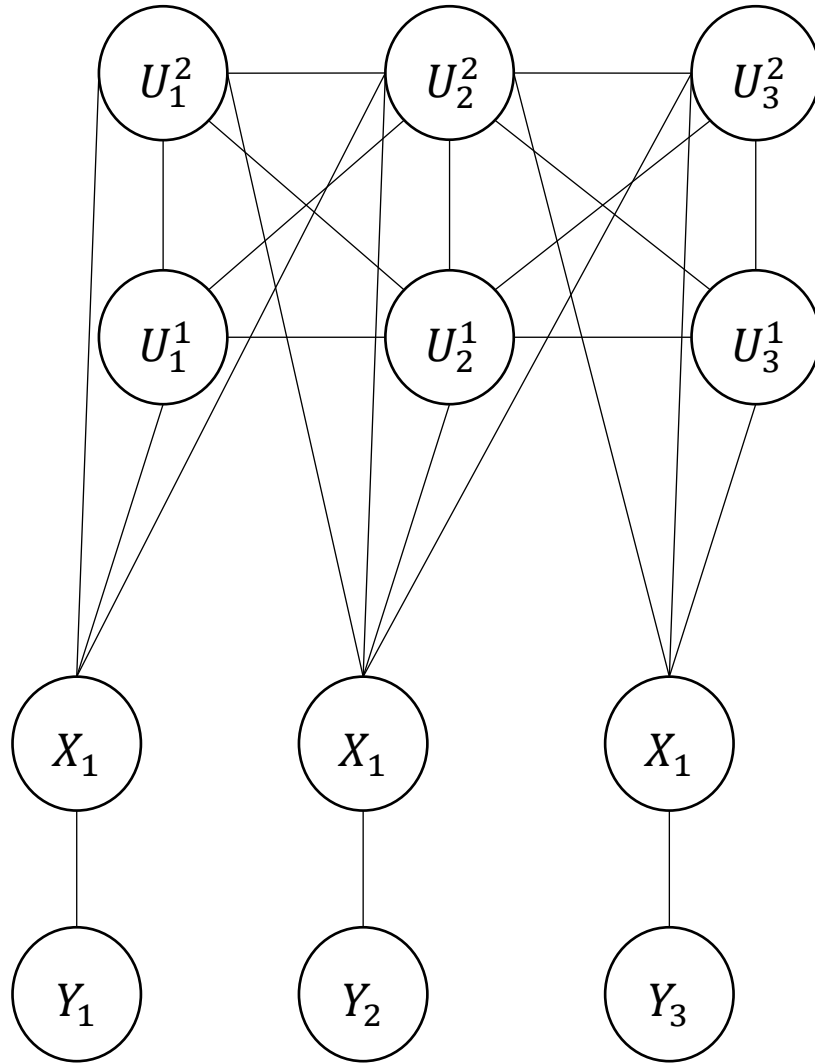


Figure 11 Graphe de probabilité d'une HESMC-V pour $N=3$

On peut remarquer que la HESMC-Z et la HESMC-V sont deux modèles différents, et aucun n'est plus général que l'autre. Leur différence principale est la suivante : dans un HESMC-Z, si $u_n^2 > 0$, alors u_{n+1}^1 est déterministe sachant u_n^1 et fixé par (3.2), soit $u_{n+1}^1 = u_n^1$, mais x_{n+1} est toujours aléatoire, à valeurs dans u_{n+1}^1 , avec la loi (3.3). Dans une HESMC-V en revanche, si $u_n^2 > 0$, alors u_{n+1}^1 et x_{n+1} sont déterministes sachant (u_n^1, x_n) et fixés par (3.12), soit $(u_{n+1}^1, x_{n+1}) = (u_n^1, x_n)$. La même remarque sur les HESMC-Z peut être faite quant à la généralité du modèle par rapport au HEMC et HSMC : si on prend, pour construire la partie évidentielle d'un HESMC-V, une masse de Dempster-Shafer m nulle en dehors des singletons $\{\omega_1\}, \dots, \{\omega_N\}$ dans (2.43), alors le HESMC-V se réduit à un HSMC. De même si on prend $p(u_n^2 \neq 0) = 0$ pour tout n , alors la HESMC-V se réduit à une HEMC.

3.2.2 Inférence et Estimation des paramètres

Comme dans les HESMC-Z, l'estimateur du MPM dans une HESMC-V peut être calculé car c'est une HTMC particulière. Ainsi il suffit d'utiliser (2.22) -(2.25) pour calculer les $p(x_n|y)$, en posant $\forall n, u_n = (u_n^1, u_n^2)$. Pour l'estimation des paramètres non supervisée, on peut utiliser une version de l'algorithme EM adapté au HESMC-V. Nous allons détailler cet EM, que nous avons utilisé pour nos expériences dans la suite. Soit une famille de HESMC-V $(X, U^1, U^2 | Y)$ homogènes à bruit gaussien, indexée par un vecteur de paramètres θ . Ce dernier est composé de $\pi_{klp} = p(x_1 = \omega_k, u_1^1 = l, u_1^2 = p)$, $p_{ijklpr} = p(x_{n+1} = \omega_k, u_{n+1}^1 = l, u_{n+1}^2 = p, x_n = \omega_i, u_n^1 = j, u_n^2 = r)$ et les moyennes et variances μ_k, σ_k^2 associées aux distributions $p(y_n|x_n = \omega_k)$. Notons :

$$p_{klp,n} = p(x_n = \omega_k, u_n^1 = l, u_n^2 = p|y) \quad (3.15)$$

$$p_{ijklpr,n,n+1} = p(x_{n+1} = \omega_k, u_{n+1}^1 = l, u_{n+1}^2 = p, x_n = \omega_i, u_n^1 = j, u_n^2 = r|y) \quad (3.16)$$

$$p_{ijklpr,n} = p(x_{n+1} = \omega_k, u_{n+1}^1 = l, u_{n+1}^2 = p | x_n = \omega_i, u_n^1 = j, u_n^2 = r, y) \quad (3.17)$$

et posons $C = |2^\Omega| - 1$

L'algorithme EM pour les HESMC-V peut être déduit directement de celui pour les HTMC. Il se déroule comme suit :

- On initialise l'algorithme avec $\theta = \theta^0$
- Pour chaque itération q on utilise θ^q pour calculer $p_{klp,n}^{(q)}$ avec (2.24) et $p_{ijklpr,n,n+1}^{(q)}$ avec (2.29), en posant $u_n = (u_n^1, u_n^2)$. Ensuite on calcule θ^{q+1} de la manière suivante :

$$\pi_{klp}^{(q+1)} = \frac{\sum_{n=1}^N p_{klp,n}^{(q)}}{N} \quad (3.18)$$

$$p_{ijklpr}^{(q+1)} = \frac{\sum_{n=1}^{N-1} p_{ijklpr,n,n+1}^{(q)}}{\sum_{n=1}^{N-1} p_{klp,n}^{(q)}} \quad (3.19)$$

$$\mu_k^{(q+1)} = \frac{\sum_{n=1}^N \sum_{l=1}^C \sum_{p=0}^{D-1} p_{klp,n}^{(q)} y_n}{\sum_{n=1}^N \sum_{l=1}^C \sum_{p=0}^{D-1} p_{klp,n}^{(q)}} \quad (3.20)$$

$$\sigma_k^{2,(q+1)} = \frac{\sum_{n=1}^N \sum_{l=1}^C \sum_{p=0}^{D-1} p_{klp,n}^{(q)} (y_n - \mu_k^{(q+1)})^2}{\sum_{n=1}^N \sum_{l=1}^C \sum_{p=0}^{D-1} p_{klp,n}^{(q)}} \quad (3.21)$$

On peut remarquer que si on initialise p_{ijklpq} de sorte que cette probabilité respecte (3.12), alors l'équation (3.19) nous garantit qu'à chaque itération le nouveau p_{ijklpq} calculé respecte (3.12).

Le calcul du MPM ainsi qu'une itération de l'algorithme EM ont tous deux la même complexité en temps que pour le HESMC-Z, soit $O(K^2 C^2 M^2 N)$ en temps si on suppose une complexité en espace de $O(KCMN)$

3.3 Résultats expérimentaux

Dans cette partie, nous présentons des études expérimentales comparant la HESMC-Z et la HESMC-V aux HSMC, HEMC et aux chaînes de semi-Markov non-stationnaires (NSHSMC) [68] ainsi qu'aux HMC pour avoir une référence. L'idée est de répondre à plusieurs questions que nous nous posons sur les HESMC-Z et les HESMC-V. La première question est de savoir si ces deux modèles peuvent améliorer simultanément la HEMC et la HSMC. En effet celles-ci sont comme on l'a vu, plus générales que ces dernières. Cependant cela ne veut pas dire qu'en pratique, les résultats de ces modèles seront forcément meilleurs. En effet la généralité entraîne aussi l'augmentation du nombre de paramètres pour le modèle et nous n'avons pas la garantie que leur estimation se passe bien. La deuxième question est de voir si les HESMC-Z et HESMC-V ont un intérêt par rapport au NSHSMC, en particulier pour la segmentation d'images. Enfin la dernière question est de voir comment HESMC-Z et HESMC-V se comparent. Dans toutes les expériences présentées, on fait les hypothèses suivantes. Pour les masses m de Dempster-Shafer intervenant dans les HEMC, HESMC-Z, HESMC-V, nous avons choisi m comme nulle en dehors des singletons $\{\omega_k\}, k = 1, \dots, K$ et de Ω . Le maximum du temps de séjour minimal M dans les HSMC, HESMC-Z, HESMC-V et NSHSMC est fixé à $M = 10$, ce qui nous semble un bon compromis entre généralité et nombre de paramètres au vu des expériences de la section 2.5.4. Pour la NSHSMC, nous avons choisi le nombre de sauts égal à deux. Les paramètres des différents modèles ont été estimés par l'algorithme EM avec une initialisation que nous allons détailler pour les modèles HSMC, HEMC, HESMC-Z et HESMC-V. Pour initialiser, nous segmentons d'abord les données observées en deux ou trois classes (suivant le nombre de classe du signal ou de l'image d'origine) par un algorithme k-means pour calculer les valeurs initiales de $p(x_1 = \omega_i)$ et $p(x_2 = \omega_k | x_1 = \omega_i)$. Celui-ci est initialisé avec la méthode « K-moyennes++ » [76]. Puis nous procédons de manière différente pour chaque modèle :

- Pour le HSMC, nous fixons $p(x_2 = \omega_k | x_1 = \omega_i, u_1 = 0) = p(x_2 = \omega_k | x_1 = \omega_i)$. Puis nous définissons $p(u_1 | x_1 = \omega_i)$ and $p(u_2 | x_2 = k, x_1 = i, u_1 = 0)$ comme des lois uniformes sur $\{0, 1, \dots, M - 1\}$.
- Pour le HEMC, nous fixons $p(u_2 = \{\omega_k\}, u_1 = \{\omega_i\}) = (1 - \varepsilon) \times p(x_2 = \omega_k, x_1 = \omega_i)$ et $p(u_2 = \{\omega_k\}, u_1 = \Omega) = p(u_2 = \Omega, u_1 = \{\omega_i\}) = p(u_2 = \Omega, u_1 = \Omega) = \varepsilon / (2 \times \text{card}(\Omega) + 1)$, avec $\varepsilon = 0.25$.
- Pour le HESMC-Z, nous construisons d'abord $p(u_1^1, u_1^2)$ de la même manière que nous avons construit $p(u_2, u_1)$ dans le HEMC. Ensuite nous construisons $p(u_1^1, u_1^2)$ et

$p(u_2^1, u_2^2 | u_1^1, u_1^2 = 0)$ de la même manière que nous avons construit $p(x_1, u_1)$ et $p(x_2, u_2 | x_1, u_1 = 0)$ dans le HSMC.

- Pour le HESMC-V, nous construisons d'abord $p(u_1^1, u_1^2)$ de la même manière que nous avons construit $p(u_2, u_1)$ dans le HEMC. Ensuite nous construisons $p(x_1, u_1^1, u_1^2)$ et $p(x_2, u_2^1, u_2^2 | x_1, u_1^1, u_1^2 = 0)$ de la même manière que nous avons construit $p(x_1, u_1)$ et $p(x_2, u_2 | x_1, u_1 = 0)$ dans le HSMC.

3.3.1 Segmentation de signaux unidimensionnels synthétiques

Nous comparons ici les cinq modèles sur cinq signaux synthétiques unidimensionnels. Les signaux cachés sont générés en utilisant différentes chaînes de Markov non stationnaires et ils sont bruités par un bruit gaussien. Chacun des signaux est segmenté de manière non supervisée par HSMC, HEMC, HESMC-Z, HESMC-V et NSHSMC, plus une HMC pour référence. Pour la génération des signaux, nous avons tiré aléatoirement chacune des valeurs d'un signal selon une chaîne de Markov cachée à deux états, avec une probabilité initiale $\pi = \begin{pmatrix} \delta_0 \\ 1-\delta_0 \end{pmatrix}$ et une matrice de transition $A_n = \begin{pmatrix} \delta_n & 1-\delta_n \\ 1-\delta_n & \delta_n \end{pmatrix}$, en utilisant cinq séquences (δ_n) différentes :

$$\text{Expérience 1 : } \delta_n = \frac{3}{4} + \frac{1}{4} \sin\left(\frac{n}{5}\right)$$

$$\text{Expérience 2 : } \delta_n = \frac{1}{2} + \frac{1}{2} \sin\left(\frac{n}{5}\right)$$

$$\text{Expérience 3 : } \delta_n = \frac{1}{2} + \left(\frac{n}{10\pi} - \left\lfloor \frac{1}{2} + \frac{n}{10\pi} \right\rfloor\right)$$

$$\text{Expérience 4 : } \delta_n = \sqrt{1 - 4 \left(\frac{n}{10\pi} - \left\lfloor \frac{1}{2} + \frac{n}{10\pi} \right\rfloor\right)^2}$$

$$\text{Expérience 5 : } \delta_n = \sqrt{1 - 4 \left(\frac{n}{10} - \left\lfloor \frac{1}{2} + \frac{n}{10} \right\rfloor\right)^2}$$

Les différentes variations de δ_n dans les expériences 1, 2, 3, 4, 5 peuvent être observées à la **figure 12**. On remarquera que les signaux 1, 2, 3, 4, sont non-stationnaires de manière irrégulière, alors que le signal 5 est non stationnaire mais avec des non-stationnarités périodiques. Nous bruitons chacun des signaux ainsi obtenus par un bruit gaussien indépendant $N(0,1)$ et $N(2,1)$. Pour la taille des signaux, nous avons pris $N = 10\,000$. L'erreur de segmentation ainsi que les paramètres du bruit estimés par chacun des modèles sont présentés dans le **tableau 2**. Nous donnons aussi le pourcentage de réduction de l'erreur de chacun des modèles par rapport au HMC. Par exemple pour l'expérience 1, comme $\frac{(0.127-0.111)}{0.127} = 0.126$, le HESMC-Z améliore le HMC de 13%.

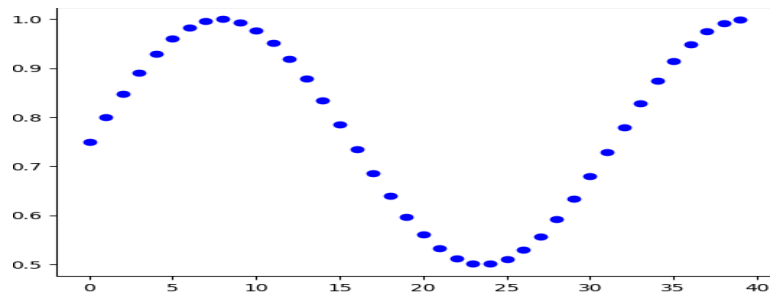
A partir de ces résultats, on note les points suivants :

- La conclusion principale est que les HESMC-Z et les HESMC-V améliorent toujours simultanément les HSMC et les HEMC, et l'amélioration peut être significative.
- Dans le cas de non stationnarité irrégulière des signaux 1, 3, 4, HESMC-Z semble avoir l'avantage sur HESMC-V et NSHSMC. Pour NSHSMC, c'est consistant avec le fait que celui-ci combine la semi-markovianité avec un modèle de Markov à sauts, plus adapté pour des sauts de stationnarités finis, alors que les HESMC combinent la semi-markovianité avec une chaîne de Markov évidentielle plus à même de segmenter des cas de sauts de stationnarités infinis, comme expliqué à la section 2.4.1. Cependant pour les différences de performance entre HESMC-Z et HESMC-V, nous n'avons pas d'éléments d'explication précis. De plus pour le signal 2, NSHSMC est meilleur que HESMC-Z et HESMC-V, alors que celui-ci est aussi non stationnaire de manière irrégulière. Enfin les différences de performances ne sont pas très grandes.
- Dans le cas de stationnarité périodique, signal 5, NSHSMC est bien meilleur que HESMC-Z et HESMC-V. C'est encore une fois consistant avec la section 2.4.1, mais ce qui est surprenant c'est l'importance de la différence de performance.
- L'algorithme EM est d'une remarquable efficacité, si l'on se réfère aux paramètres du bruit qui sont estimés avec une excellente précision. Ce n'était pas sûr à l'avance au vu du nombre de paramètres à estimer pour certains modèles et de la non-stationnarité des signaux cachés.
- HSMC a de bons résultats dans le cadre de la segmentation de signaux non stationnaires. Ce dernier point est surprenant car celui-ci a été construit pour modéliser un temps de séjour dans un état non géométrique, et non pas pour gérer la non-stationnarité. Cela mériterait une étude plus approfondie avec des situations plus complexes.

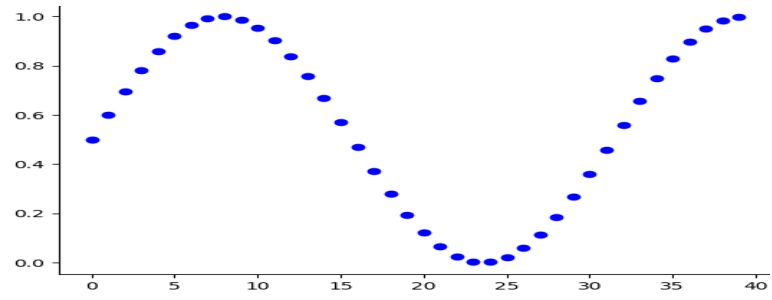
Tableau 2. Paramètres estimés et ratio d’erreur. M : moyenne, V : Variance, ER : ratio d’erreur, pour la segmentation des cinq signaux (Exp. 1, Exp. 2, Exp. 3, Exp. 4, Exp. 5) par les six modèles (HMC, HEMC, HESMC-Z, HESMC-V, NSHSMC)

		HMC		HSMC		HEMC		HESMC-Z		HESMC-V		NSHSMC	
		M	V	M	V	M	V	M	V	M	V	M	V
Exp.1	ω_1	0.09	1.17	-0.01	0.99	-0.01	0.98	0.01	1.00	-0.00	0.98	0.03	0.98
	ω_2	1.88	1.21	2.02	0.97	1.99	0.98	2.02	0.99	1.99	0.99	2.03	1.00
	ER	0.127		0.117 (8%)		0.117 (8%)		0.111 (13%)		0.114 (10%)		0.114 (10%)	
Exp.2		M	V	M	V	M	V	M	V	M	V	M	V
	ω_1	0.10	1.06	0.02	1.02	-0.12	0.92	-0.03	0.97	-0.01	0.99	-0.07	0.93
	ω_2	2.09	0.94	2.05	1.00	1.88	1.11	1.95	1.03	1.98	1.07	1.97	0.99
ER	0.159		0.122 (23%)		0.145 (9%)		0.115 (28%)		0.113 (29%)		0.098 (38%)		
Exp.3		M	V	M	V	M	V	M	V	M	V	M	V
	ω_1	-0.05	0.98	-0.01	1.03	0.03	1.05	-0.01	0.98	-0.04	0.96	0.04	0.96
	ω_2	1.93	1.11	1.99	1.00	2.06	0.92	1.99	1.01	1.99	0.99	2.03	0.95
ER	0.161		0.150 (7%)		0.152 (6%)		0.142 (12%)		0.145 (10%)		0.148 (8%)		
Exp.4		M	V	M	V	M	V	M	V	M	V	M	V
	ω_1	0.10	1.15	0.03	0.92	0.04	1.06	0.01	0.96	-0.01	0.99	-0.03	0.93
	ω_2	1.89	1.19	2.04	1.04	2.00	1.04	2.00	1.01	1.99	0.98	2.01	0.94
ER	0.121		0.102 (16%)		0.099(18%)		0.092 (24%)		0.096 (20%)		0.093 (23%)		
Exp.5		M	V	M	V	M	V	M	V	M	V	M	V
	ω_1	0.10	1.15	0.02	1.04	0.04	1.06	0.01	1.05	0.02	1.04	0.04	1.01
	ω_2	1.89	1.19	1.99	1.05	2.00	1.04	2.03	1.03	1.90	1.15	2.00	1.00
ER	0.122		0.097 (20%)		0.113 (7%)		0.088 (28%)		0.091 (25%)		0.063 (48%)		

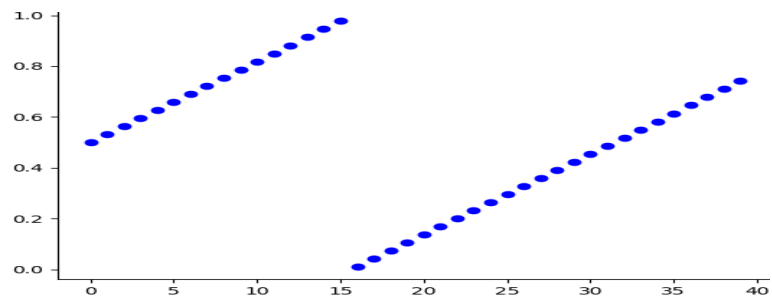
Chaînes de semi-Markov évidentielles



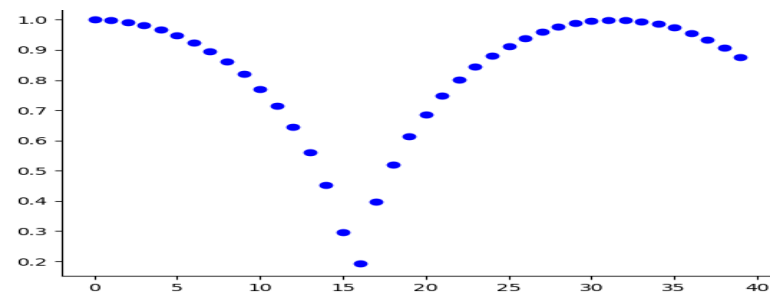
Exp. 1



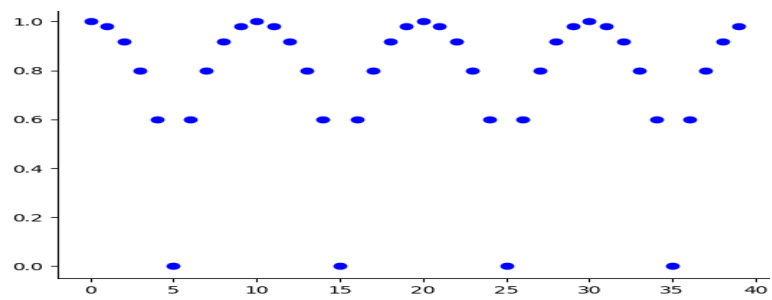
Exp. 2



Exp. 3



Exp. 4



Exp. 5

Figure 12. Variations de δ_n dans les expériences 1 (Exp. 1), 2 (Exp. 2), 3 (Exp. 3), 4 (Exp.4), 5 (Exp. 5).

3.3.2 Segmentation d'images synthétiques

Dans ce paragraphe, nous exposons l'intérêt des HESMC-Z et HESMC-V dans le cadre de la segmentation d'images. Pour cela nous segmentons quatre images synthétiques par les cinq modèles HSMC, HEMC, HESMC-Z, HESMC-V, et NSHSMC, toujours avec une HMC pour référence. Les images bidimensionnelles seront transformées en signaux unidimensionnels par un balayage de Peano [77] [78] pour pouvoir utiliser les différents modèles. Les images présentées ici ne sont pas choisies au hasard. La première image, dont les segmentations sont présentées en **figure 13**, contient de grandes zones homogènes qui devraient favoriser la segmentation par les HSMC, comme montré en section 2.5.4. Cependant comme les HMC sont déjà très performantes sur ce genre d'images, nous avons décidé, pour augmenter l'influence de la loi *a priori* dans la segmentation, de la bruiteur avec un bruit gaussien indépendant très élevé : $N(0,1)$ et $N(0.3,1)$. La seconde image, dont les segmentations sont présentées en **figure 14**, contient des détails très fins, qui sont difficiles à segmenter avec des modèles tels que le HMC. Mais nous savons que les HEMC peut améliorer les HMC de manière significative dans ce cas [28], à condition que le bruit ne soit pas trop fort. C'est pourquoi nous avons bruité cette image avec un bruit gaussien indépendant relativement faible : $N(0,1)$ et $N(2,1)$. La troisième image, dont les segmentations sont présentées en **figure 15**, est une image à trois classes qui a été conçue pour contenir à la fois des détails fins et de larges zones homogènes. Elle est bruitée par un bruit gaussien indépendant $N(0,1)$, $N(0.3,1)$ et $N(3,1)$, de telle sorte que les détails fins soient peu bruités alors que les grandes zones homogènes le soient de façon importante. Comme la troisième, la quatrième image, dont les segmentations sont présentées en **figure 16**, est une image à trois classes, conçue pour contenir à la fois des détails fins et de larges zones homogènes. Nous l'avons bruitée avec un bruit original non-stationnaire obtenu de la façon suivante. Pour chaque $n = 1, \dots, N$, nous tirons en premier aléatoirement selon une loi uniforme sur $[a, b]$ de longueur 2, centrée sur une valeur différente pour chaque classe. Nous utilisons les valeurs ainsi construites comme les moyennes de lois gaussiennes, avec une variance de 1, pour bruiteur le pixel en question. Nous allons noter ce type de bruit, qui n'est pas gaussien, avec $N(U[a, b], 1)$. Ce bruit a été conçu pour tester la robustesse de chacun des six modèles utilisés pour la segmentation, car il n'est pas gaussien alors que ceux-ci supposent une loi du bruit gaussienne. Les paramètres du bruit ainsi que le ratio d'erreur des segmentations dans les **figures 13, 14, 15 et 16**, sont présentés dans les **tableaux 3, 4, 5 et 6**.

Les résultats obtenus permettent de présenter les conclusions suivantes :

- Les HSMC ont un intérêt dans la segmentation d'images lorsqu'on a de grandes zones homogènes et un bruit très fort, comme dans la **figure 13**, où ils améliorent significativement les HMC. On peut l'expliquer car la loi *a priori* d'un HSMC est mieux adaptée aux grandes zones homogènes que celle du HMC qui suppose un temps de

séjour dans un état géométrique. On peut remarquer aussi que les HESMC-Z, et les NSHSMC sont très proches du HSMC en performance, alors que le HESMC-V l'améliore.

- Les HEMC ont un intérêt lorsque l'image n'est pas très homogène et contient simultanément des zones larges de même classe et des détails très fin, comme l'image de la **figure 14**. C'est un fait connu [28], mais le point intéressant ici est que la HESMC-Z fait aussi bien que la HEMC dans ce cas. On voit par ailleurs que la HESMC-V a plus de mal à retrouver le résultat de la HEMC et le dégrade un peu, tandis que la NSHSMC, qui est elle aussi conçu pour gérer la non-stationnarité, n'est pas capable de restaurer les détails fins. On peut alors supposer que les détails fins sont mieux gérés par un modèle qui est construit pour le second type de non-stationnarité (à variations infinies, continue) présentée à la section 2.4.1, que les modèles conçus pour le premier type (sauts finis de stationnarité).
- Dans nos expériences, la HESMC-Z peut améliorer simultanément les HSMC et les HEMC jusqu'à 40%. La HESMC-V peut les améliorer simultanément jusqu'à 50%. On le voit sur les segmentations des figures **15 et 16**. Cela semble être le cas sur des images présentant à la fois de grandes zones homogènes de même classe et des détails fins. De plus le bruit continu non-stationnaire semble renforcer cet effet.
- La HESMC-Z peut améliorer la NSHSMC de 10% au maximum et la HESMC-V peut les améliorer de 20% au maximum. Cela semble être le cas sur des images présentant à la fois de grandes zones homogènes de même classe et des détails fins, bruitées avec un bruit continu non-stationnaire comme à la **figure 16**. Si le bruit est stationnaire en revanche (**figure 15**), alors l'écart entre la NSHSMC et les deux HESMC est réduit et la NSHSMC peut même être plus performant que ceux-ci. C'est consistant avec la théorie car les NSHSMC combine la semi-markovianité avec un modèle de Markov à sauts, plus adapté pour des sauts de stationnarités finis, alors que les HESMC combinent la semi-markovianité avec une chaîne de Markov évidentielle plus à même de segmenter des cas de non-stationnarité continue comme le bruit de la **figure 16**.
- Si on compare la HESMC-V et la HESMC-Z, on peut remarquer que la HESMC-V est souvent meilleure en segmentation d'images (**tableaux 3, 5, 6**), cependant cette dernière a plus de mal à segmenter les détails fins que la HESMC-Z (**tableau 4**).
- L'efficacité de l'estimation des paramètres par EM est remarquable. En effet pour différencier les modèles nous utilisons des bruits forts et complexes (**figures 13, 15 et 16**) et si on regarde les estimées du bruit (**tableaux 3, 5, 6**), on remarque que cette estimation reste correcte. C'est encore plus surprenant si on considère que les données ne suivent vraisemblablement aucun des six modèles. En effet la chaîne obtenue après le balayage de Peano d'une image a une structure très complexe. De plus dans la

segmentation de la **figure 16**, tous les modèles supposent un bruit gaussien, alors que celui-ci ne l'est pas. Si on regarde l'estimation de la HESMC-Z spécifiquement, on voit que dans les situations où seule la HSMC a un intérêt, l'estimation des paramètres réduit la HESMC-Z à une HSMC, et de même lorsque seule la HEMC a un intérêt. Enfin quand les deux ont un intérêt, la HESMC-Z améliorent les deux modèles simultanément. Ces résultats sont compatibles avec le fait que la HESMC-Z est plus générale que la HSMC et la HEMC, ce qui confirme que l'algorithme EM est suffisamment efficace pour que la plus grande généralité de la HESMC-Z joue son rôle.

Tableau 3 Paramètres estimés et ratio d'erreur. M : moyenne, V : Variance, pour la segmentation de l'image présentée à la **figure 13** par les six modèles (HMC, HEMC, HESMC-Z, HESMC-V, NSHSMC)

	Paramètres estimés du bruit			Taux d'erreur (gain en % par rapport au HMC)
		ω_1	ω_2	
Vérité	M, V	0, 1	0.3, 1	
HMC	M, V	0.17, 0.95	0.35, 0.98	0.42
HSMC	M, V	0.03, 0.99	0.28, 1.01	0.20 (52%)
HEMC	M, V	0.17, 0.96	0.27, 1.00	0.32 (24%)
HESMC-Z	M, V	0.06, 0.98	0.26, 1.00	0.21 (50%)
HESMC-V	M, V	0.01, 1.04	0.33, 1.01	0.17 (60%)
NSHSMC	M, V	0.03, 1.01	0.28, 1.01	0.21 (50%)

Tableau 4. Paramètres estimés et ratio d’erreur. M : moyenne, V : Variance, pour la segmentation de l’image présentée à la **figure 14** par les six modèles (HMC, HEMC, HESMC-Z, HESMC-V, NSHSMC)

	Paramètres estimés du bruit			Taux d’erreur (gain en % par rapport au HMC)
		ω_1	ω_2	
Vérité	M, V	0, 1	2, 1	
HMC	M, V	0.99, 2.02	1.99, 1.01	0.23
HSMC	M, V	0.99, 2.02	1.99, 1.02	0.23 (0%)
HEMC	M, V	-0.01, 1.01	2.00, 1.00	0.08 (65%)
HESMC-Z	M, V	0.09, 1.14	2.00, 1.02	0.08 (65%)
HESMC-V	M, V	0.67, 1.88	2.02, 1.03	0.14 (39%)
NSHSMC	M, V	0.99, 2.02	1.99, 1.02	0.23 (0%)

Tableau 5 Paramètres estimés et ratio d’erreur. M : moyenne, V : Variance, pour la segmentation de l’image présentée à la **figure 15** par les six modèles (HMC, HEMC, HESMC-Z, HESMC-V, NSHSMC)

	Paramètres estimés du bruit				Taux d’erreur (gain en % par rapport au HMC)
		ω_1	ω_2	ω_3	
Vérité	M, V	0, 1	0.3, 1	3, 1	
HMC	M, V	0.12, 0.94	0.33, 1.00	1.50, 3.03	0.42
HSMC	M, V	0.02, 1.00	0.27, 1.01	1.50, 3.08	0.19 (54%)
HEMC	M, V	0.10, 0.95	0.37, 0.97	3.01, 1.02	0.35 (16%)
HESMC-Z	M, V	0.03, 1.00	0.28, 1.00	3.02, 1.01	0.16 (62%)
HESMC-V	M, V	0.02, 1.00	0.27, 1.00	2.91, 1.21	0.16 (62%)
NSHSMC	M, V	0.02, 1.00	0.31, 0.99	2.99, 1.00	0.15 (64%)

Tableau 6. Paramètres estimés et ratio d'erreur. M : moyenne, V : Variance, pour la segmentation de l'image présentée à la **figure 16** par les six modèles (HMC, HEMC, HESMC-Z, HESMC-V, NSHSMC)

	Paramètres estimés du bruit				Taux d'erreur (gain en % par rapport au HMC)
		ω_1	ω_2	ω_3	
Vérité	M, V	0, 1.33	2, 1.33	4, 1.33	
HMC	M, V	-0.19, 1.26	0.05, 1.38	2.55, 2.65	0.29
HSMC	M, V	0.01, 1.38	1.92, 1.48	2.73, 3.62	0.16 (45%)
HEMC	M, V	-0.52, 1.11	0.20, 1.30	2.59, 2.54	0.17 (41%)
HESMC-Z	M, V	-0.02, 1.34	2.02, 1.58	4.12, 1.21	0.09 (69%)
HESMC-V	M, V	0.03, 1.37	1.97, 1.40	4.00, 1.42	0.08 (72%)
NSHSMC	M, V	-0.01, 1.33	1.96, 1.40	3.74, 1.69	0.10 (66%)

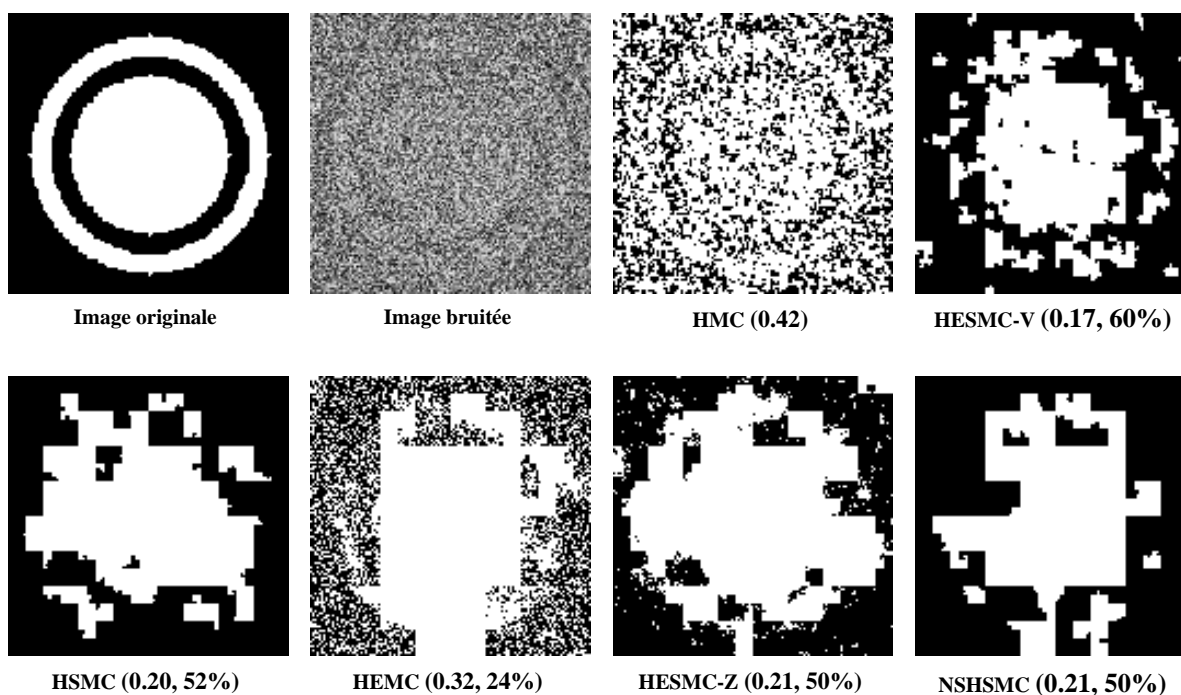


Figure 13. Segmentation d'une image a deux classes, bruitée par un bruit gaussien $N(0, 1)$, $N(0.3, 1)$. HMC, HSMC, HEMC, HESMC-Z, HESMC-V et NSHSMC : résultat, avec taux d'erreur et amélioration par rapport à HMC entre parenthèses, de la segmentation non supervisée avec le modèle correspondant.

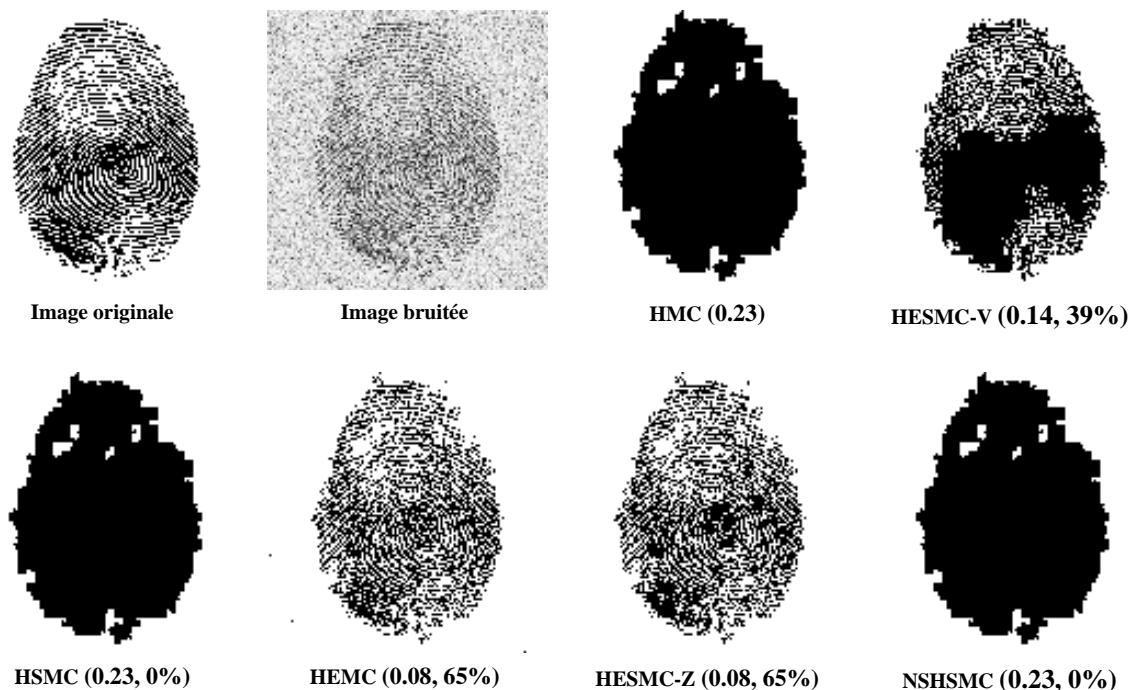


Figure 14. Segmentation d'une image a deux classes (Image originale), bruitée par un bruit gaussien $N(0, 1)$, $N(2, 1)$ (Image bruitée). HMC, HSMC, HEMC, HESMC-Z, HESMC-V et NSHSMC : résultat, avec taux d'erreur et amélioration par rapport à HMC entre parenthèses, de la segmentation non supervisée avec le modèle en question

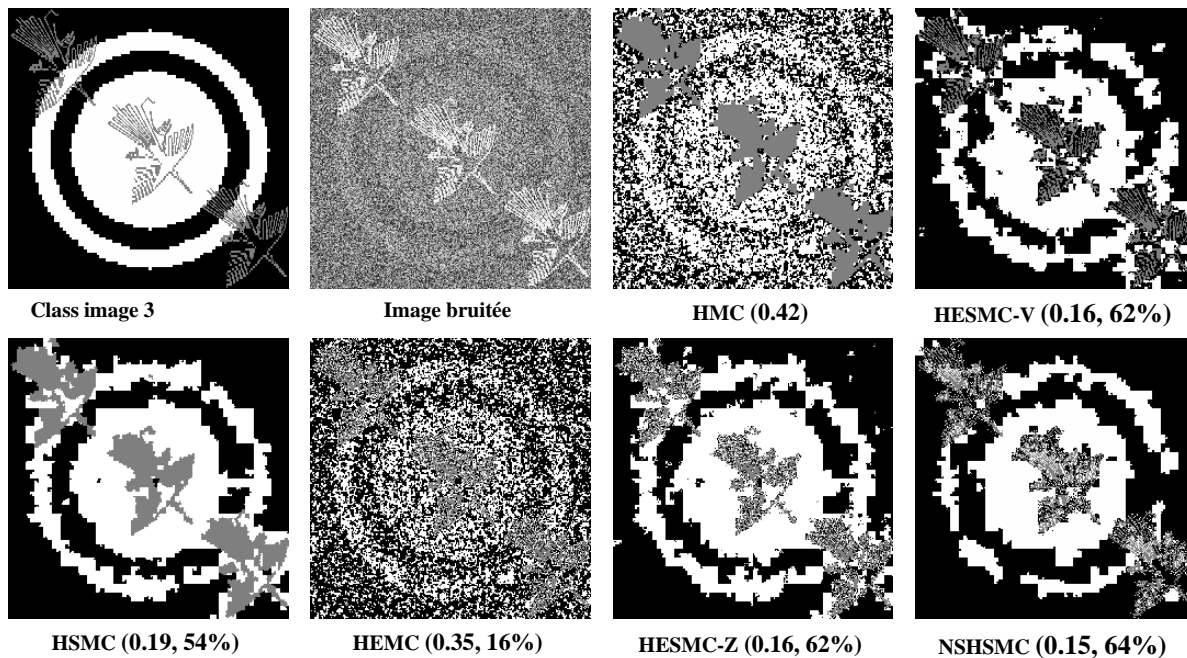


Figure 15. Segmentation d'une image a trois classes (Image originale), bruitée par un bruit gaussien $N(0, 1)$, $N(0.3, 1)$, $N(3, 1)$ (Image bruitée). HMC, HSMC, HEMC, HESMC-Z, HESMC-V et NSHSMC : résultat, avec taux d'erreur et amélioration par rapport à HMC entre parenthèses, de la segmentation non supervisée avec le modèle en question

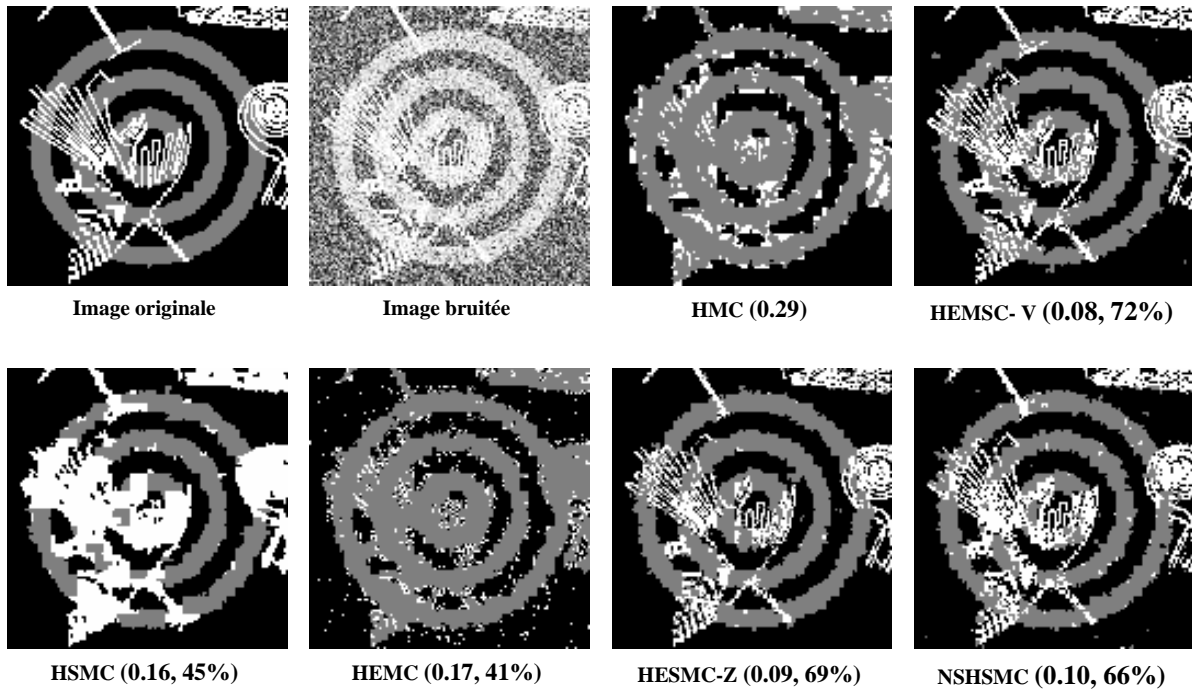


Figure 16. Segmentation d'une image a trois classes (Image originale), bruitée par un bruit non stationnaire $N(U([-1, 1]), 1)$, $N(U([1, 3]), 1)$, $N(U([3, 5]), 1)$ (Image bruitée). HMC, HSMC, HEMC, HEMSC-Z, HEMSC-V et NSHSMC : résultat, avec taux d'erreur et amélioration par rapport à HMC entre parenthèses, de la segmentation non supervisée avec le modèle en question

3.4 Comparaison avec les chaînes évidentielles semi-Markov

Dans nos modèles de chaînes de semi-Markov évidentielles, nous sommes partis d'une HEMC, que nous avons ensuite étendue en rajoutant la semi-markovianité, que ce soit sur U^1 ou sur (X, U^1) . L'auteur de [79] propose de faire l'inverse, c'est-à-dire de partir d'une HSMC, puis de rajouter le processus U^2 pour modéliser la notion d'évidentialité sur (X, U^1) . Soit $X = (X_1, \dots, X_N)$ un processus caché qui prend ses valeurs à partir d'un ensemble fini de classes $\Omega = \{\omega_1, \dots, \omega_K\}$ et qui doit être estimé à partir d'un processus observable $Y = (Y_1, \dots, Y_N)$ qui prend ses valeurs dans \mathbb{R} . Introduisons un troisième processus aléatoire, la séquence $U^1 = (U_1^1, \dots, U_N^1)$ avec chaque U_n^1 à valeur dans $\Lambda = (0, 1, \dots, M - 1)$ de tel sorte que (X, U^1, Y) soit une HSMC, respectant donc (2.62). Notons $2^{\Omega \times \Lambda}$ l'ensemble puissance de $\Omega \times \Lambda$. Posons $V = (X, U^1)$ et construisons une séquence aléatoire $U^2 = (U_1^2, \dots, U_N^2)$ avec chaque U_n^2 à valeur dans $2^{\Omega \times \Lambda}$, tel que (2.47) soit respecté, avec V à la place de X . Ainsi La distribution de la chaîne évidentielle de semi-Markov (HSEMC) $(U^2, V, Y) = (U^2, U^1, X, Y)$ est celle d'une HTMC définie par :

$$\begin{aligned}
 p(t_1) &= p(u_1^2)p(x_1, u_1^1|u_1^2)p(y_1|x_1) \\
 p(t_{n+1}|t_n) &= p(u_{n+1}^2|u_n^2)p(x_{n+1}, u_{n+1}^1|u_{n+1}^2)p(y_{n+1}|x_{n+1})
 \end{aligned} \tag{3.22}$$

avec

$$p(x_{n+1}, u_{n+1}^1 | u_{n+1}^2) = \frac{1_{\{(x_{n+1}, u_{n+1}^1) \in u_{n+1}^2\}}}{|u_{n+1}^2|} \quad (3.23)$$

L'application du MPM ainsi que l'estimation des paramètres par EM sont alors possibles [79] puisque la HSEMC est une HTMC particulière. Cependant la complexité en temps de celle-ci dépend du nombre d'états cachés, comme pour toute HTMC, et dans le cas de la HSEMC, le nombre d'états cachés est de $(2^{KM} - 1) \times K \times M$, contrairement aux HESMC-Z et HESMC-V où il est de $(2^K - 1) \times K \times M$. Ainsi la HSEMC, bien qu'intéressante théoriquement, est moins pratique expérimentalement car sa complexité croît extrêmement vite lorsqu'on considère un nombre assez conséquent de classes pour X et un maximum de temps de séjour minimum M un peu long.

3.5 Conclusion

Dans ce chapitre, nous avons proposé deux nouveaux triplets markoviens, les HESMC-Z et les HESMC-V. Ils ont été obtenus par l'introduction de deux chaînes auxiliaires : la première est basée sur la théorie de l'évidence pour gérer la non-stationnarité, alors que la seconde utilise la semi-markovianité pour modéliser un temps de séjour dans un état plus flexible que dans les HMC [60]. En ce qui concerne la segmentation d'images, les HESMC considérés peuvent avoir de l'intérêt dans des images contenant à la fois des grandes zones homogènes et des détails fins, ainsi que du bruit non-stationnaire, où elles peuvent améliorer significativement les HEMC, HSMC et NSHSMC. Le HESMC-V semble un peu plus performant que le HESMC-Z dans la segmentation d'images en général, mais il rencontre des difficultés à segmenter les détails fins. En particulier il est moins performant que la HEMC sur ce point, alors que la HSEMC-Z y arrive bien et fait aussi bien que la HEMC. En plus, comme vu dans les segmentations présentées, HESMC-Z permet être aussi performant que la HSMC quand celle-ci est intéressante, et la même chose peut être dite pour les HEMC. En considérant que le temps d'exécution du MPM et celui de l'estimation des paramètres pour le HESMC-Z sont compétitifs avec ceux de HSMC et HEMC, si les opérations pour chaque état caché sont exécutées en parallèles, ce modèle peut permettre de ne pas choisir entre la modélisation HEMC et la modélisation HSMC en utilisant un modèle qui inclut les deux. Enfin cette approche du triplet fusionnant l'évidentiel et le semi-Markov est plus prometteuse dans la pratique que celle proposée dans [79], car cette dernière implique un grand nombre d'état cachés, ce qui augmente rapidement la complexité informatique des calculs du MPM et de l'estimation des paramètres. Cependant bien que l'efficacité de ces modèles en segmentation d'images soit avérée, leur approche de la structure de l'image est insatisfaisante. En effet, les HESMC restent des chaînes de Markov et modélisent donc des processus unidimensionnels. Or une image est par définition bidimensionnelle. Ainsi on pourrait se demander comment faire en sorte d'améliorer ce point, tout en n'augmentant pas trop la complexité informatique du MPM et de l'estimation des

Chaînes de semi-Markov évidentielles

paramètres, qui peut déjà être relativement élevée dans un triplet avec de nombreux état caché. Ce point est l'objet de l'étude présentée dans le chapitre suivant.

Chapitre 4. Chaînes de Markov conditionnelles et balayage de Peano contextuel

Les chaînes de Markov présentent un désavantage certain lorsque l'on traite des données avec des dépendances en deux dimensions comme une image. En effet la chaîne suppose toujours des données avec une dépendance en une dimension. On peut cependant transformer une image bidimensionnelle en chaîne unidimensionnelle par un balayage des pixels dans un certain ordre, tel que le balayage ligne par ligne, colonne par colonne ou encore le parcours de Peano [77] [78]. Cela permet la segmentation d'images par des chaînes de Markov, mais ces techniques de balayage ne sont pas idéales. En effet après application d'une de celles-ci, nous perdons une partie des dépendances entre les données, résultat du passage d'un voisinage bidimensionnel à un voisinage unidimensionnel. Par ailleurs d'autres modélisations markoviennes n'ont pas cet inconvénient. C'est le cas des champs de Markov cachés (HMF) qui prennent en compte un voisinage bi-dimensionnel, et sont plus adaptés à la segmentation d'image [80] [81] [82]. Ceux-ci ont été par exemple appliqués avec succès à la segmentation d'IRM de cerveaux [83], à la reconstruction d'images [84] [85] et à la séparation spectrale dans les images hyperspectrales [86] [87]. Cependant, les calculs directs de loi *a posteriori* et d'estimation des paramètres ne sont pas faisables, et on doit utiliser des méthodes itératives telles que l'échantillonneur de Gibbs ou l'algorithme de Metropolis [88]. Cela augmente significativement la complexité informatique en temps de ces modèles et les rend inadaptés dans certaines situations. L'utilisation de HMC combinés au parcours de Peano peut être une alternative peu coûteuse en temps de calcul aux HMF [89] et donne parfois des résultats de qualité assez proche de ceux-ci [15] [90]. La combinaison HMC et parcours de Peano a été utilisée pour segmenter des images de types très variés [37] [91] [92] [93] [94] [95] [90]. Malgré le fait que les données obtenues avec un balayage de Peano sur une image ont une structure complexe et ne sont pas, *a priori*, markoviennes, les différentes méthodes mentionnées peuvent donner des résultats intéressants, montrant ainsi la robustesse des HMC.

Nous présentons dans un premier temps une extension du parcours de Peano, pour prendre en compte une partie du voisinage bi-dimensionnel dans une image. Elle consiste à associer à chaque pixel de la HMC obtenue avec le balayage de Peano (PS), deux observations des pixels qui sont voisines du pixel en question dans l'image, mais pas dans la HMC. Ce parcours, que nous appellerons balayage de Peano contextuel (CPS), induit un nouveau modèle, que nous appellons chaînes de Markov conditionnelles avec un balayage de Peano contextuel (CMC-CPS), avec une structure plus complexe que les HMC mais dont la loi *a posteriori* est toujours

markovienne [96]. Nous présentons ainsi comment calculer l'estimateur du MPM et estimer les paramètres en non supervisé avec une variante de l'algorithme SEM, dans ce modèle. Enfin nous réalisons une étude de segmentation d'images synthétiques pour comparer le CMC-CPS aux HMC et aux HMF.

4.1 Parcours de Peano contextuel

Le parcours de Peano classique repose sur la courbe de Hilbert, qui est une courbe continue remplissant un carré [77] [78]. Un exemple de construction de cette courbe est donné en **figure 17**, et le **tableau 7** donne un exemple de l'ordre dans lequel sont choisis les pixels dans un balayage de Peano, pour une image 4×4 . On remarque directement que deux des quatre voisins d'un pixel dans l'image ne le sont plus dans le signal obtenu par le PS (**figure 18**). C'est pour pallier ce problème que nous introduisons le balayage de Peano contextuel, qui se construit comme suit. Soit $X = (X_1, \dots, X_N)$ un processus caché qui prend ses valeurs à partir d'un ensemble fini de classes $\Omega = \{\omega_1, \dots, \omega_K\}$ et qui doit être estimé à partir d'un processus observable $Y = (Y_1, \dots, Y_N)$ qui prend ses valeurs dans \mathbb{R} . On considère que la séquence $1, \dots, N$ a été obtenue après le PS. Soit S l'ensemble des pixels de l'image en deux dimensions, soit un ensemble de pixels carré de dimension $N = 2^R \times 2^R$. Pour tout $n = 2, \dots, N - 1$, on note s le pixel correspondant à n dans S . Le point précédent $n - 1$ dans la séquence sera noté r , et le suivant $n + 1$, sera appelé t . Pour construire le CPS, nous associons à chaque $n = 2, \dots, N - 1$, un v_n et un w_n , tels que v_n et w_n soient des voisins de s dans l'image, mais différent de r et t (donc pas voisins dans la chaîne). Ainsi chaque pixel (n, s) a quatre voisins dans S , les deux $(r, n - 1)$ et $(t, n + 1)$, qui sont aussi voisins de (n, s) dans le balayage de Peano, et les deux (v, v_n) et (w, w_n) qui ne le sont pas. Nous construisons ainsi une nouvelle séquence d'observation $Y^* = \left((Y_{v_1}, Y_1, Y_{w_1}), \dots, (Y_{v_N}, Y_N, Y_{w_N}) \right)$ associée à (X_1, \dots, X_N) . Nous proposons dans le **tableau 8** un exemple de construction de la séquence Y^* pour une image 4×4 dont la numérotation des pixels par le balayage de Peano est donnée dans le **tableau 7**. Si on énumère les points de ce parcours comme $(1, 2, \dots, N)$, le CPS associé est une séquence de triplets (ce sont des couples pour les points du bord de l'image et les coins de l'image marquant respectivement le début et la fin du balayage, et des singletons sur les deux autres coins de l'image).

$$([1, v_1, w_1], [2, v_2, w_2], \dots, [N, v_N, w_N]) \quad (4.1)$$

On peut remarquer que la construction du CPS dépend de la définition du voisinage d'un pixel. Nous avons en fait présenté ici un CPS correspondant au voisinage des quatre plus proches voisins (4NN-CPS). Il est tout à fait possible d'étendre cette notion de voisinage aux huit plus proches voisins (8NN-CPS). Cela entraînerait un CPS qui associe à chaque pixel

Chaînes de Markov conditionnelles et balayage de Peano contextuel

l'observation sur six des huit plus proches voisins de ce dernier (les deux voisins restants étant les voisins dans la chaîne et l'image).

Tableau 7. Ordre de parcours des pixels d'une image 4×4 d'un balayage de Peano commençant dans le coin haut gauche

1	2	15	16
4	3	14	13
5	8	9	12
6	7	10	11

Tableau 8. Les observations associées aux pixels 1,2, 3, ..., 16 du tableau 7 par le balayage de Peano contextuel, qui sont (y_1, y_4) , (y_2, y_3) , (y_{14}, y_3, y_8) , (y_1, y_4) , (y_5, y_8) , y_6 , (y_7, y_{10}) , (y_8, y_3, y_5) , ..., (y_{16}, y_{13}) .

1	2	3	4	5	6	7	8
y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8
-	-	y_{14}	-	-	-	-	y_3
y_4	y_{15}	y_8	y_1	y_8	-	y_{10}	y

9	10	11	12	13	14	15	16
y_9	y_{10}	y_{11}	y_{12}	y_{13}	y_{14}	y_{15}	y_{16}
y_{12}	-	-	-	-	y_9	y_2	y_{13}
y_{14}	y_7	-	y_9	y_{16}	y_3	-	-

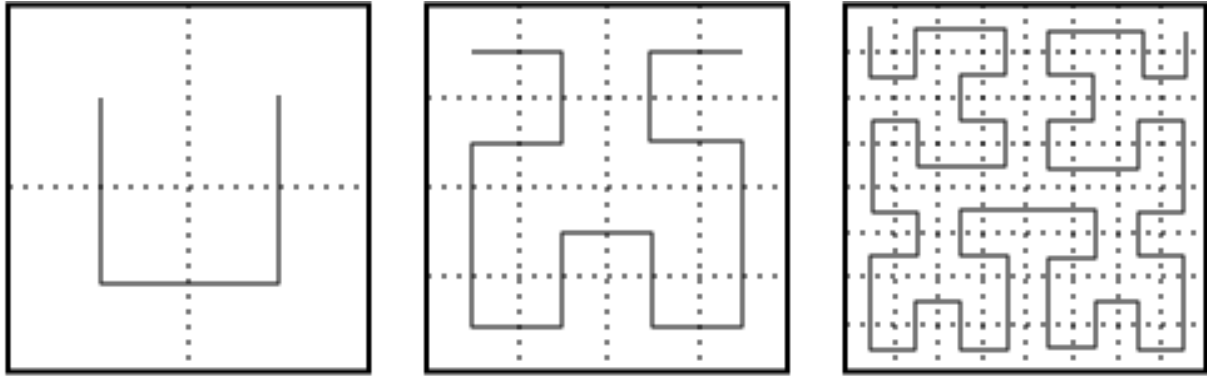


Figure 17. Construction du balayage de Peano.

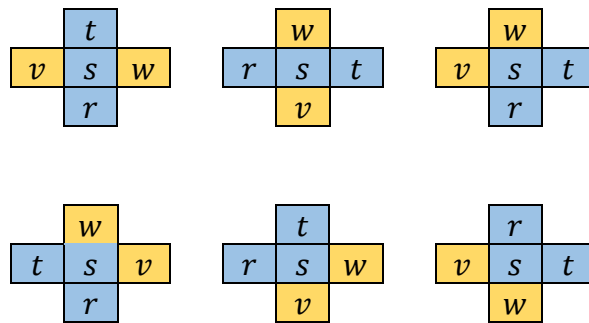


Figure 18. Les six configurations spatiales possibles pour les voisins de s (pour s hors bords et coins de l'image) ajoutés par le Peano contextuel. En bleu les voisins de s , aussi voisins dans le balayage de Peano. En vert les voisins de s qui ne le sont plus dans le balayage de Peano, et dont les observations sont associées à l'observation y_s sur s .

4.2 Chaînes de Markov conditionnelles

Le balayage de Peano contextuel permet d'ajouter, à une observation sur un pixel, les observations associées aux pixels voisins dans l'image, mais qui ne sont pas voisins dans le parcours de Peano classique. Il faut maintenant voir comment prendre en compte ces nouvelles observations d'un point de vue du modèle probabiliste. Nous proposons ici une modification du modèle classique de la HMC pour que celui-ci puisse prendre en compte ces observations. Nous allons voir que dans le modèle proposé, la loi du couple (X, Y) n'est plus markovienne et seule $p(x|y)$ l'est, cependant nous allons montrer que l'estimateur du MPM est toujours calculable, et l'estimation des paramètres par l'algorithme SEM est toujours possible.

4.2.1 Définition du modèle

Soit S un ensemble de pixels carré de dimension $N = 2^R \times 2^R$. Soit $(1, 2, \dots, N)$ un PS de S , et soit $([1, v_1, w_1], [2, v_2, w_2], \dots, [N, v_N, w_N])$ le CPS associé au PS. Soit $X = (X_1, \dots, X_N)$ un processus caché qui prend ses valeurs dans d'un ensemble fini de classes $\Omega = \{\omega_1, \dots, \omega_K\}$ et

qui doit être estimé à partir d'un processus observable $Y = (Y_1, \dots, Y_N)$ qui prend ses valeurs dans \mathbb{R} .

Soient s, t voisins dans S , sans être voisins dans le PS. S'ils sont voisins horizontaux, posons :

$$p^{hor}(y_t|x_s) = \sum_{x_t} p^{hor}(x_t|x_s)p(y_t|x_t) \quad (4.2)$$

De la même manière, s'ils sont voisins verticaux, on pose :

$$p^{ver}(y_t|x_s) = \sum_{x_t} p^{ver}(x_t|x_s)p(y_t|x_t) \quad (4.3)$$

Ainsi pour $s \in S$ et v, w voisins de s dans S l'ensemble des pixels, mais pas voisins dans le PS, on a :

$$p(y_v, y_s, y_w|x_s) = p(y_s|x_s)p^{a(v,s)}(y_v|x_s)p^{a(w,s)}(y_w|x_s) \quad (4.4)$$

où $a(v, s) = hor$ si v, s sont des voisins horizontaux, et $a(v, s) = ver$ si v, s sont des voisins verticaux. On définit $a(w, s)$ de la même manière. Par exemple, en considérant (y_{14}, y_3, y_8) dans le **tableau 8**, on voit que 3, 14 sont des voisins horizontaux, alors que 3, 8 sont des voisins verticaux. Ainsi nous obtenons :

$$p(y_{14}, y_3, y_8|x_3) = p(y_3|x_3)p^{hor}(y_{14}|x_3)p^{ver}(y_8|x_3) \quad (4.5)$$

Par analogie avec la distribution d'une HMC classique (1.15), nous définissons un nouveau modèle, appelé « chaînes de Markov conditionnelles pour un balayage de Peano contextuel » (CMC-CPS), de la manière suivante :

$$q(x, y) = p(x_1)p((y_1, y_{v_1}, y_{w_1}|x_1) \prod_{n=1}^{N-1} p(x_{n+1}|x_n)p(y_{n+1}, y_{v_{n+1}}, y_{w_{n+1}}|x_{n+1})) \quad (4.6)$$

Il faut bien noter ici que $q(x, y)$ n'est pas la densité de probabilité du couple (X, Y) par rapport à la mesure usuelle $(c \times \lambda)^N$ sur $(\Omega \times \mathbb{R})^N$, avec c la mesure de comptage et λ la mesure de Lebesgue. Cependant en considérant la constante de normalisation :

$$\kappa = \frac{1}{\sum_x \int q(x, y) dy} \quad (4.7)$$

Nous avons :

$$p(x, y) = \kappa q(x, y) \quad (4.8)$$

Le nom « chaînes de Markov conditionnelles pour un balayage de Peano contextuel » spécifie que X est markovien conditionnellement à Y , soit $p(x|y)$ est une loi markovienne. Cela va permettre de calculer l'estimateur MPM et donc faire de la segmentation bayésienne, comme nous allons le montrer dans la prochaine section. Cependant ni $p(x)$, ni $p(x, y)$, ni $p(y|x)$ ne sont markoviennes, ce qui rend κ non immédiatement calculable dans la pratique.

La définition du modèle de CMC-CPS ne dépend en fait pas du type de balayage choisi. Nous avons ici pris un parcours de Peano, mais l'idée principale de la section (4.1), à savoir qu'un balayage pour transformer une image bidimensionnelle en chaîne unidimensionnelle va toujours entraîner la perte de deux voisins sur quatre, est aussi vrai pour un parcours ligne par ligne ou colonne par colonne. Ainsi on peut toujours faire la même démarche que celle de la section (4.1) dans d'autres parcours que celui de Peano, et ajouter à l'observation sur un pixel, les observations sur les pixels voisins dans l'image mais pas dans la chaîne. On pourrait même se passer de la notion d'image et de bi dimensionnalité, et, dans un signal unidimensionnel, décider d'associer à l'observation sur un site n , des observations sur des sites $(k)_{\substack{k \neq n-1 \\ k \neq n+1}}$ qui ne sont pas voisins de n .

4.2.2 Inférence

La loi du processus couple (X, Y) obtenue par (4.6) et (4.7) n'est pas une HMC classique et $p(y|x)$ a une structure très complexe. Cependant, comme $p(x|y)$ est markovien, la segmentation bayésienne est faisable. En effet pour calculer l'estimateur MPM donné par (1.18), il nous faut être capable de calculer $p(x_n|y)$. Nous présentons tout d'abord un résultat intermédiaire important pour ce calcul.

Lemme 4.1

Soit $Z = (Z_1, \dots, Z_N)$ une chaîne stochastique prenant ses valeurs dans un ensemble fini Ω . Alors :

- Z est markovien si et seulement s'il existe $N - 1$ fonctions $\varphi_1, \dots, \varphi_{N-1}$ de Ω^2 vers \mathbb{R}^+ telles que :

$$p(z) \propto \prod_{n=1}^{N-1} \varphi_n(z_n, z_{n+1}) \quad (4.9)$$

avec \propto exprimant la relation de proportionnalité.

- Si (4.9) est vérifiée, la probabilité initiale $p(z_1)$ et les transitions $p(z_{n+1}|z_n)$ s'expriment en fonction des $\varphi_1, \dots, \varphi_{N-1}$ de la manière suivante :

$$p(z_1) = \frac{\beta_1(z_1)}{\sum_{z_1} \beta_1(z_1)}$$

$$\text{Pour } n = 1, \dots, N - 1, \quad p(z_{n+1}|z_n) = \frac{\varphi_n(z_n, z_{n+1})\beta_{n+1}(z_{n+1})}{\beta_n(z_n)} \quad (4.10)$$

où $\beta_1(z_2), \dots, \beta_N(z_N)$ sont calculées à partir des récursions rétrogrades (« backward »):

$$\begin{aligned} \beta_N(z_N) &= 1 \\ \beta_n(z_n) &= \sum_{z_{n+1}} \varphi_{n+1}(z_n, z_{n+1})\beta_{n+1}(z_{n+1}) \end{aligned} \quad (4.11)$$

- Une fois $p(z_1)$ et $p(z_{n+1}|z_n)$ données, chaque $p(z_n)$ peut être calculé à partir des récursion progressive (« forward ») :

$$\forall n \in [2, \dots, N], p(z_n) = \sum_{z_n} p(z_n|z_{n-1})p(z_{n-1}) \quad (4.12)$$

Pour résumer, une fois que les fonctions $\varphi_1, \dots, \varphi_{N-1}$ vérifiant (4.9) sont données, il est possible de calculer, avec une complexité proportionnelle à N , les transitions $p(z_{n+1}|z_n)$ et les distributions marginales $p(z_n)$ de la chaîne de Markov $Z = (Z_1, \dots, Z_N)$. Il est donc suffisant, pour une chaîne de Markov, de définir les fonction $\varphi_1, \dots, \varphi_{N-1}$ vérifiant (4.9), ce qui simplifie l'introduction des différents modèles. La démonstration de ce lemme est présentée en section B des annexes.

Si on retourne au calcul de $p(x_n|y)$ dans la CMC-CPS, on peut remarquer que $p(x|y) \propto q(x, y)$, avec $q(x, y)$ défini par (4.6). On a alors :

$$p(x|y) \propto \prod_{n=1}^{N-1} \varphi_n(x_n, x_{n+1}, y) \quad (4.13)$$

avec :

$$\varphi_1(x_1, x_2, y) = p(x_1, x_2)p(y_1, y_{v_1}, y_{w_1}|x_1)p(y_2, y_{v_2}, y_{w_2}|x_2)$$

$$\text{Pour } n = 2, \dots, N-1, \varphi_n(x_{n+1}, x_n, y) = p(x_{n+1}|x_n)p(y_{n+1}, y_{v_{n+1}}, y_{w_{n+1}}|x_{n+1}) \quad (4.14)$$

Finalement, les fonctions $\varphi_1, \dots, \varphi_{N-1}$ vérifient (4.9), et sont de la forme :

$$\varphi_1(x_1, x_2, y) = \varphi_1(x_1, x_2, y_1, y_{v_1}, y_{w_1}, y_2, y_{v_2}, y_{w_2})$$

$$\text{Pour } n = 2, \dots, N-1, \varphi_n(x_{n+1}, x_n, y) = \varphi_n(x_{n+1}, x_n, y_{n+1}, y_{v_{n+1}}, y_{w_{n+1}}) \quad (4.15)$$

De plus elles sont faciles à calculer par (4.14). On peut donc appliquer le **lemme 4.1** pour calculer $p(z_1|y)$ et $p(z_{n+1}|z_n, y)$, et finalement on obtient $p(x_n|y)$, toujours par le **lemme 4.1**.

4.2.3 Estimation des paramètres non supervisée

Supposons que $p(x, y)$ est la distribution d'une HMC à un bruit gaussien $p(y|x)$, avec deux transitions différentes $p^{ver}(x_{n+1}|x_n)$ et $p^{hor}(x_{n+1}|x_n)$ suivant que les pixels $n+1$ et n sont respectivement des voisins verticaux ou horizontaux dans l'image originale. Pour K classes $\Omega = \{\omega_1, \dots, \omega_K\}$, les paramètres de la HMC sont : K^2 probabilités $p^{ver} = (p_{ij}^{ver})_{1 \leq i, j \leq K}$, avec $p_{ij}^{ver} = p(x_{n+1} = \omega_j, x_n = \omega_i)$ pour $n, n+1$ voisins verticaux dans l'image, K^2 probabilités $p^{hor} = (p_{ij}^{hor})_{1 \leq i, j \leq K}$, avec $p_{ij}^{hor} = p(x_{n+1} = \omega_j, x_n = \omega_i)$ pour $n, n+1$ voisins horizontaux dans l'image, K moyennes $\mu = (\mu_i)_{1 \leq i \leq K}$ et K variances $\sigma^2 = (\sigma_i^2)_{1 \leq i \leq K}$ associées aux K distributions gaussiennes $(p(y_n|x_n = \omega_i))_{1 \leq i \leq K}$. Si on s'intéresse maintenant au CMC-CPS, en choisissant $p^{hor}(x_{v_n}|x_n)$ et $p^{ver}(x_{w_n}|x_n)$ intervenant dans $p(y_{v_n}, y_n, y_{w_n}|x_n)$ comme

égales à p_{ij}^{hor} et p_{ij}^{ver} respectivement, celui-ci possède exactement les mêmes paramètres que le HMC. Ainsi le problème est d'estimer $\theta = (p^{hor}, p^{ver}, \mu, \sigma^2)$ à partir de l'image observée $Y = y$ seulement. Cependant l'algorithme EM n'est pas applicable ici. En effet, d'après (4.6)-(4.8), on ne peut calculer la vraisemblance jointe qu'à une constante près, constante qui n'est pas calculable. On ne peut donc pas calculer $\underset{\theta}{\operatorname{argmax}} E[\log(p(x, y|\theta)|Y, \theta^q)]$ à chaque itération q .

Cependant on peut remarquer que l'on a accès à la loi *a posteriori* $p(x|y)$ dont on sait qu'elle est markovienne grâce au **Lemme 4.1**, avec la loi initiale et les transitions calculables avec (4.10). Ainsi on peut utiliser une variante de l'algorithme SEM, où l'on simule x^q à partir de (4.10), puis on utilise les estimateurs empiriques classiques de la loi du couple (x_{n+1}, x_n) et des moyennes et variances pour mettre à jour nos paramètres.

On pose H l'ensemble des couples $(n, n + 1)$, avec n et $n + 1$ voisins horizontaux dans l'image et V l'ensemble des couples $(n, n + 1)$, avec n et $n + 1$ voisins verticaux dans l'image. L'algorithme SEM pour les CMC-CPS se déroule alors de la manière suivante :

- On initialise l'algorithme avec $\theta = \theta^0$
- Pour chaque itération q on utilise θ^q pour simuler (x^q) selon (4.10). Ensuite on calcule θ^{q+1} de la manière suivante :

$$p_{ij}^{hor,(q+1)} = \frac{\sum_{(n,n+1) \in H} \mathbf{1}_{\{x_n^{(q)} = \omega_i, x_{n+1}^{(q)} = \omega_j\}}}{|H|} \quad (4.16)$$

$$p_{ij}^{ver,(q+1)} = \frac{\sum_{(n,n+1) \in V} \mathbf{1}_{\{x_n^{(q)} = \omega_i, x_{n+1}^{(q)} = \omega_j\}}}{|V|} \quad (4.17)$$

$$\mu_i^{(q+1)} = \frac{\sum_{n=1}^N \mathbf{1}_{\{x_n^{(q)} = \omega_i\}} y_n}{\sum_{n=1}^N \mathbf{1}_{\{x_n^{(q)} = \omega_i\}}} \quad (4.18)$$

$$\sigma_i^{2,(q+1)} = \frac{\sum_{n=1}^N \mathbf{1}_{\{x_n^{(q)} = \omega_i\}} (y_n - \mu_i^{(q+1)})^2}{\sum_{n=1}^N \mathbf{1}_{\{x_n^{(q)} = \omega_i\}}} \quad (4.19)$$

4.3 Résultats expérimentaux

Dans cette section, nous comparons le nouveau modèle de CMC-CPS avec les HMC et le parcours de Peano classique (HMC-PS), ainsi que les HMF, dans le cadre de la segmentation d'images non-supervisée. L'idée est de répondre à trois questions :

- (i) l'algorithme SEM proposé est-il efficace ?
- (ii) Est-ce que la CMC-CPS améliore la HMC-PS en segmentation d'images ?
- (iii) Comment la CMC-CPS se compare aux HMF ?

Pour répondre à ces questions, nous avons fait trois études séparées. La première étude est une comparaison entre une segmentation supervisée et non supervisée d'images synthétiques bruitées par un bruit gaussien indépendant, pour vérifier la validité du SEM dans la CMC-CPS. La deuxième est une segmentation non supervisée d'images synthétiques bruitées par un bruit gaussien indépendant, avec pour but de comparer la CMC-CPS avec les HMC-PS et les HMF. Enfin la troisième étude est une segmentation d'une image réelle en niveau de gris. L'estimation des paramètres en non supervisé a été faite par l'algorithme SEM pour la HMC-PS et la CMC-CPS. Par ailleurs, spécifiquement pour les CMC-CPS, nous n'avons pas différencié les transitions verticales et horizontales, soit $p^h = p^v$. Pour les HMF nous avons estimé les paramètres par l'algorithme EM gibbsien [85]. L'initialisation pour tous les modèles a été faite grâce à une segmentation à deux classes, réalisée par un algorithme du k-means, celui-ci étant initialisé avec la méthode « K-moyennes++ » [76].

4.3.1 Images synthétiques segmentées de manière supervisée.

Nous présentons ici une étude de segmentations d'images synthétiques, toutes bruitées par un bruit gaussien indépendant $N(0,1)$ et $N(1,1)$. Les paramètres du modèle pour la segmentation sont obtenus de manière supervisée par maximum de vraisemblance pour la HMC-PS, et à la fois de manière supervisée par maximum de vraisemblance, et non supervisée par SEM, pour la CMC-CPS. Cela nous permet de vérifier si les CMC-CPS apportent quelque chose à la segmentation d'images par rapport aux HMC-PS dans un cadre où l'estimateur du maximum de vraisemblance des paramètres se calcule de manière exacte, en comparant les deux segmentations supervisées. Ainsi que de vérifier que le SEM proposé est efficace en comparant pour la CMC-CPS, le résultat de la segmentation supervisée et non supervisée. De ce premier jeu de segmentations présenté à la **figure 19**, nous pouvons conclure deux choses. La première est que les CMC-CPS améliorent toujours les HMC classiques. Le gain est modeste, voir négligeable dans le cas d'image homogène (l'image 1 de la **figure 19**), mais il est plus conséquent dans les autres cas (les images 2, 3 et 4 de la **figure 19**). La deuxième conclusion est que l'estimation des paramètres non supervisée par SEM pour le CMC-CPS est efficace. En effet si on compare les résultats de la segmentation supervisée (ligne d de la **figure 19**) et de la segmentation non supervisée (ligne e de la **figure 19**) pour ce modèle, on remarque que les résultats sont proches voir parfois meilleurs en non supervisé qu'en supervisé.

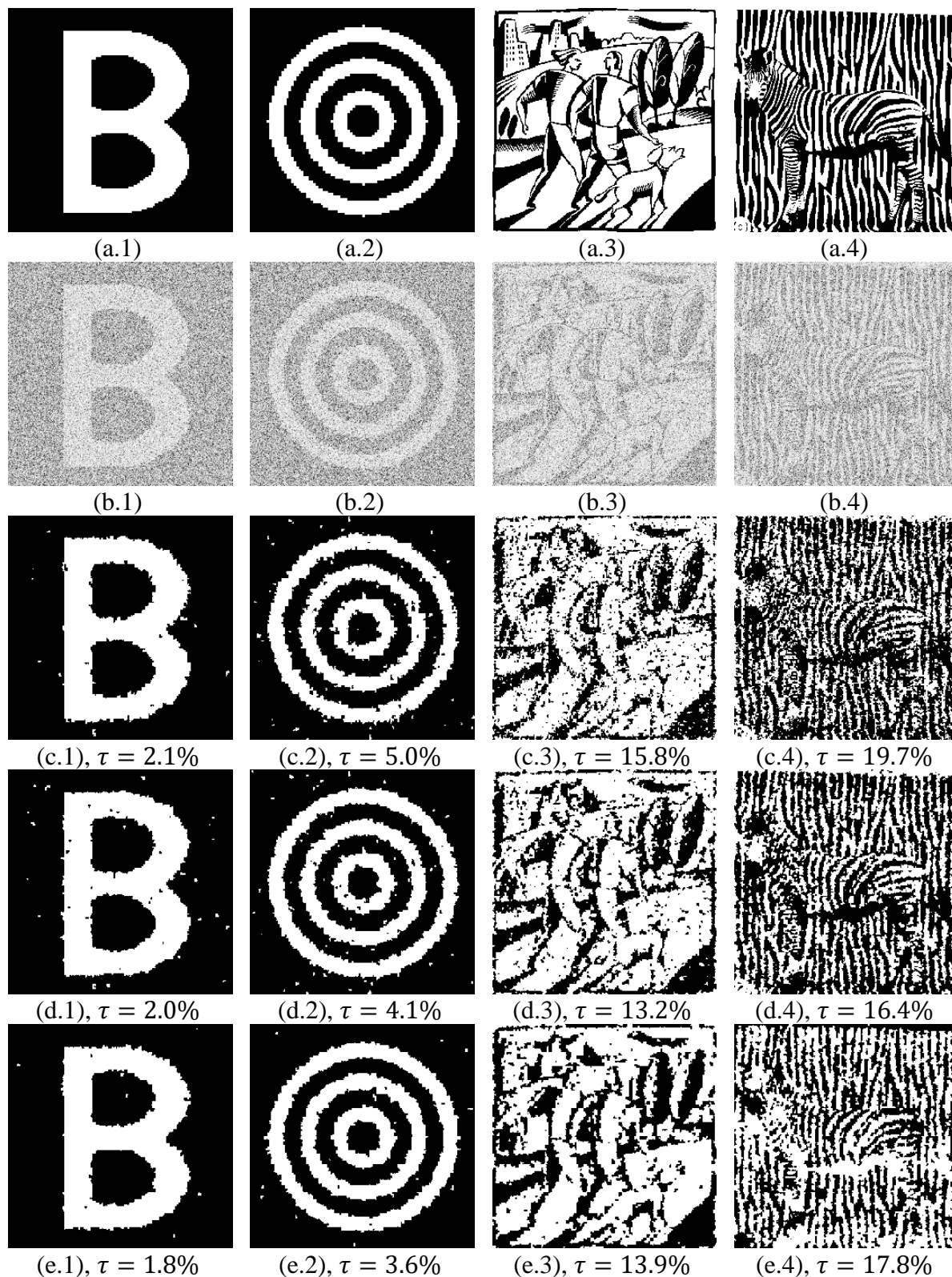


Figure 19. Segmentations de plusieurs images (a.) bruitées par un bruit gaussien indépendant $N(0, 1)$, $N(1, 1)$ (b.), avec HMC-PS de manière supervisée (c.), CMC-CPS de manière supervisée (d.), CMC-CPS de manière non supervisée (e.). τ est le taux d'erreur.

4.3.2 Images synthétiques segmentées de manière non supervisée

La deuxième étude de segmentations concerne à nouveau des images synthétiques, toutes bruitées par un bruit gaussien indépendant $N(0,1)$ et $N(1,1)$. Cette fois cependant tous les paramètres des modèles sont estimés de manière non supervisée, de plus un HMF a été inclus dans la comparaison. Sur le choix des images, nous avons essayé de balayer un éventail d'images différentes mais avec la particularité de ne pas être trop homogène, car nous savons que les CMC-CPS apportent peu aux HMC-PS dans ce cas, d'après les expériences de la section précédente. Selon ce deuxième jeu de segmentations, présenté dans les **figures 20 -25**, nous pouvons conclure deux choses. La conclusion principale est que dans le cadre de la segmentation non supervisée d'images, la CMC-CPS est systématiquement et significativement plus efficace que la HMC-PS avec le parcours de Peano classique. Le gain moyen relatif est d'environ 16%. Une autre conclusion importante est que dans certain cas, mais pas tous, la CMC-CPS donne des résultats comparables au HMF. C'est le cas dans les **figures 20, 23, 25**, où les deux modèles sont très proches, le HMF restant quand même un peu meilleur sauf dans la **figure 23** où c'est le contraire.

Tableau 9. Récapitulatifs des taux d'erreurs de segmentations des modèles de HMC-PS, CMC-CPS et HMF pour les **figures 20-25**.

Image	Bruit	HMC-PS	CMC-CPS	HMF
figure 20	$N(0,1)$, $N(1,1)$	12.2%	9.1%	8.9%
figure 21	$N(0,1)$, $N(1,1)$	15.3%	13.9%	11.6%
figure 22	$N(0,1)$, $N(1,1)$	9.5%	7.4%	6.1%
figure 23	$N(0,1)$, $N(1,1)$	20.1%	17.8%	18.0%
figure 24	$N(0,1)$, $N(1,1)$	16.1%	14.1%	12.0%
figure 25	$N(0,1)$, $N(1,1)$	13.8%	11.7%	10.5%

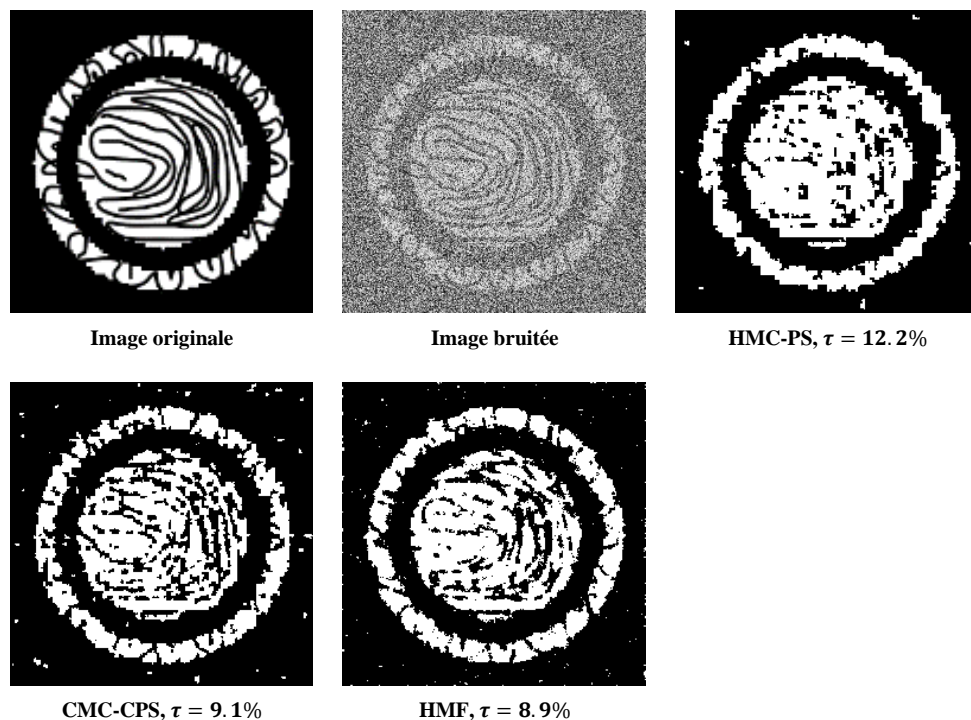


Figure 20. Segmentation d'une image a deux classes (Image originale), bruitée par un bruit gaussien $N(0, 1)$, $N(1, 1)$ (Image bruitée). HMC-PS : résultat de la segmentation non supervisée avec la HMC-PS (taux d'erreur $\tau = 12.2\%$), CMC-CPS : résultat de la segmentation non supervisée avec la CMC-CPS (taux d'erreur $\tau = 9.1\%$), HMF : résultat de la segmentation non supervisée avec le HMF (taux d'erreur $\tau = 8.9\%$).

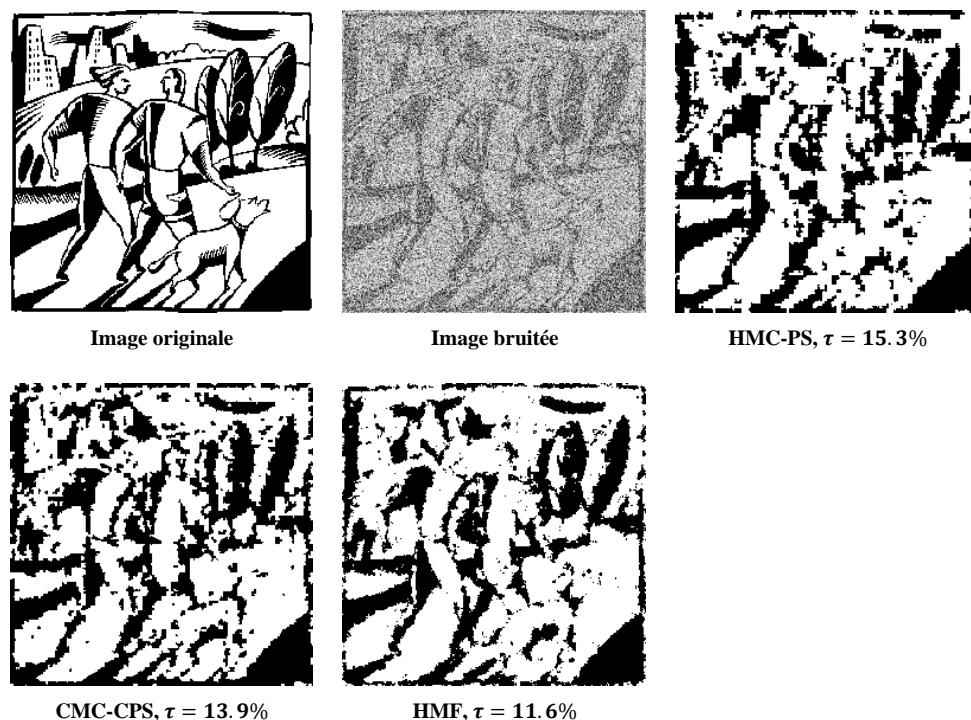


Figure 21. Segmentation d'une image a deux classes (Image originale), bruitée par un bruit gaussien $N(0, 1)$, $N(1, 1)$ (Image bruitée). HMC-PS : résultat de la segmentation non supervisée avec la HMC-PS (taux d'erreur $\tau = 15.3\%$), CMC-CPS : résultat de la segmentation non supervisée avec la CMC-CPS (taux d'erreur $\tau = 13.9\%$), HMF : résultat de la segmentation non supervisée avec le HMF (taux d'erreur $\tau = 11.6\%$).

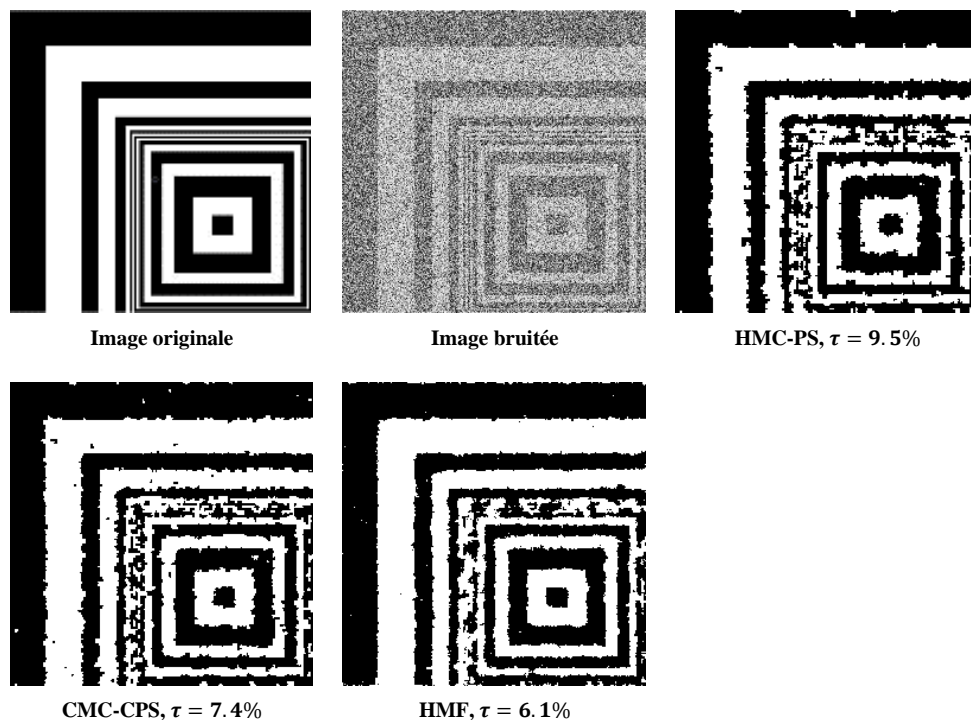


Figure 22. Segmentation d'une image a deux classes (Image originale), bruitée par un bruit gaussien $N(0, 1)$, $N(1, 1)$ (Image bruitée). HMC-PS : résultat de la segmentation non supervisée avec la HMC-PS (taux d'erreur $\tau = 9.5\%$), CMC-CPS : résultat de la segmentation non supervisée avec la CMC-CPS (taux d'erreur $\tau = 7.4\%$), HMF : résultat de la segmentation non supervisée avec le HMF (taux d'erreur $\tau = 6.1\%$).

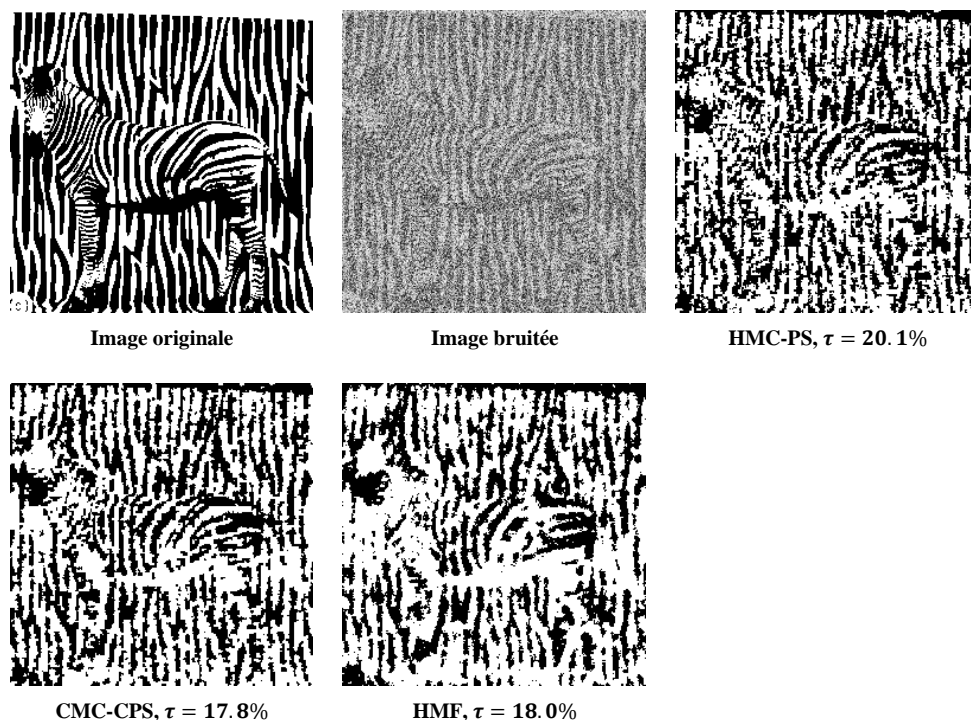


Figure 23. Segmentation d'une image a deux classes (Image originale), bruitée par un bruit gaussien $N(0, 1)$, $N(1, 1)$ (Image bruitée). HMC-PS : résultat de la segmentation non supervisée avec la HMC-PS (taux d'erreur $\tau = 20.1\%$), CMC-CPS : résultat de la segmentation non supervisée avec la CMC-CPS (taux d'erreur $\tau = 17.8\%$), HMF : résultat de la segmentation non supervisée avec le HMF (taux d'erreur $\tau = 18.0\%$).

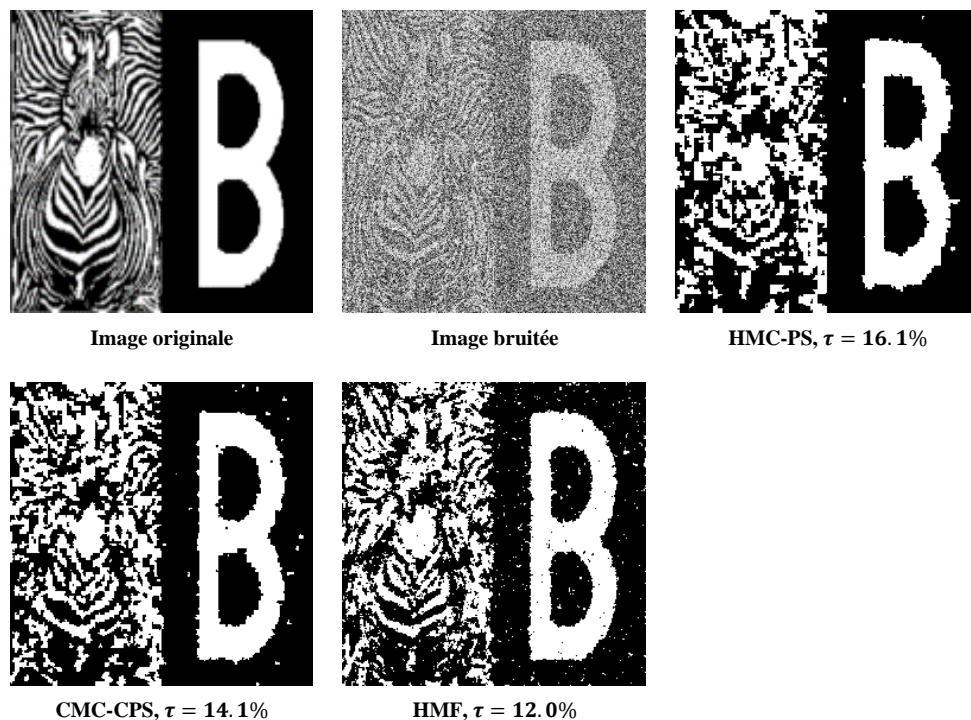


Figure 24. Segmentation d'une image a deux classes (Image originale), bruitée par un bruit gaussien $N(0, 1)$, $N(1, 1)$ (Image bruitée). HMC-PS : résultat de la segmentation non supervisée avec la HMC-PS (taux d'erreur $\tau = 16.1\%$), CMC-CPS : résultat de la segmentation non supervisée avec la CMC-CPS (taux d'erreur $\tau = 14.1\%$), HMF : résultat de la segmentation non supervisée avec le HMF (taux d'erreur $\tau = 12.0\%$).

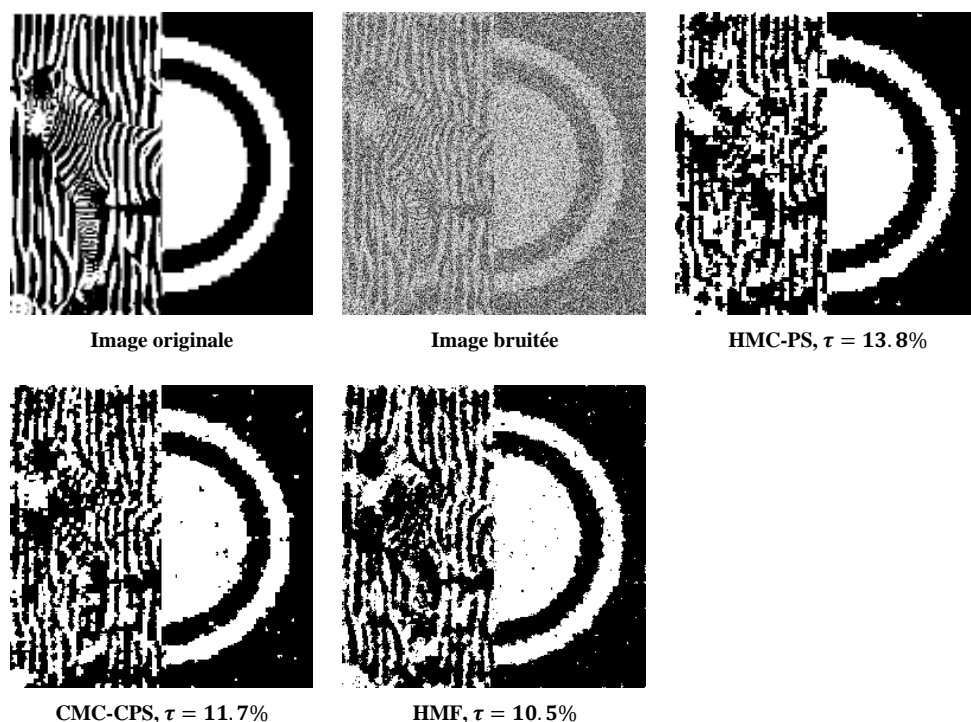


Figure 25. Segmentation d'une image a deux classes (Image originale), bruitée par un bruit gaussien $N(0, 1)$, $N(1, 1)$ (Image bruitée). HMC-PS : résultat de la segmentation non supervisée avec la HMC-PS (taux d'erreur $\tau = 13.8\%$), CMC-CPS : résultat de la segmentation non supervisée avec la CMC-CPS (taux d'erreur $\tau = 11.7\%$), HMF : résultat de la segmentation non supervisée avec le HMF (taux d'erreur $\tau = 10.5\%$).

4.3.3 Image réelle

Enfin dans cette dernière étude de segmentation, nous présentons la segmentation d'une image réelle en niveaux de gris : une radiographie comportant un os fracturé (**figure 26**). On ne peut pas calculer de taux d'erreur dans ce cas, car nous n'avons pas la vérité terrain. Cependant on peut remarquer une chose : bien que la segmentation des os soit plus satisfaisante dans le cas du HMF, ce dernier ne détecte pas la zone fracturée, alors que celle-ci apparaît dans la segmentation par la CMC-CPS. Le modèle de HMC-PS quant à lui est de loin le modèle qui donne le plus mauvais résultat, la segmentation des os par ce dernier n'est pas satisfaisante et la zone de fracture n'apparaît pas.

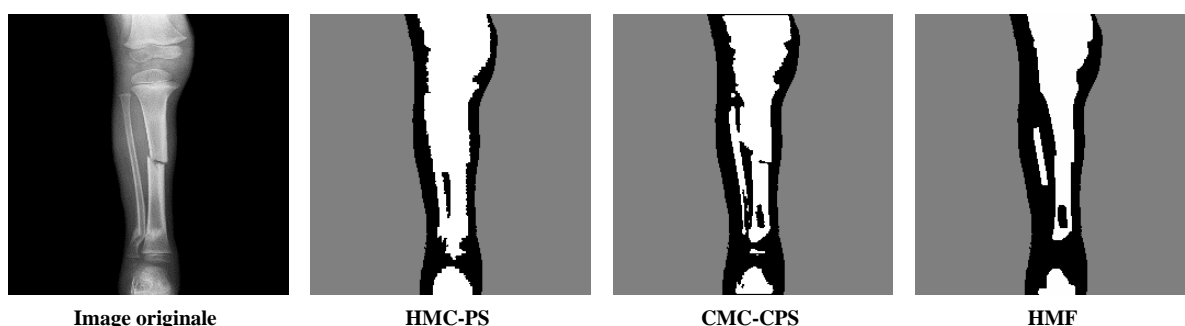


Figure 26. Segmentation d'une image de fracture en niveau de gris (Image originale). HMC, CMC-CPS, HMF : résultat de la segmentation non supervisée avec le modèle en question.

4.4 Conclusion

Nous avons construit, dans ce chapitre, une extension du balayage de Peano permettant de prendre en compte les observations associées aux pixels voisins dans l'image, mais qui ne sont pas voisins dans le parcours de Peano classique. Nous avons ensuite présenté le modèle de chaînes de Markov conditionnelle, permettant de prendre en compte les nouvelles observations obtenues par ce balayage. Ces deux apports combinés améliorent la segmentation d'images non supervisée par la HMC-PS avec un gain relatif de 16% en moyenne sur le taux d'erreur. De plus, dans certaines situations, la qualité des segmentations obtenues par la CMC-CPS s'approche de celle des HMF, tout en étant bien plus rapides. Par ailleurs les HMC étant généralisable aux PMC et TMC, et le parcours de Peano pouvant également être utilisé avec ces modèles, il semble naturel de tenter de généraliser les CMC-CPS aux chaînes de Markov couples et triplets conditionnelles, avec un balayage de Peano contextuel. C'est l'objet de l'étude présentée dans le chapitre suivant.

Chapitre 5. Chaînes de Markov couples et triplets conditionnelles

De même que les chaînes, les HMF ont été étendus aux modèles couples (PMF) [97] [84] et triplets (TMF) [98] , [99]. Dans les TMF, on trouve en particulier les champs de Markov évidentiels cachés (HEMF) [100], les champs de Markov non stationnaires [101] [102], ou encore les champs de Markov à bruit en mélange de gaussiennes [103], les champs de Markov flous [104] et les champs de Markov multi-résolutions [105]. Ces différents TMF ont été appliqués avec succès dans des cas de segmentation d'images [106] [107] [108] [109]. Les TMF sont des modèles très puissants mais ils souffrent du même problème que les HMF, à savoir que les calculs directs de loi *a posteriori* et des estimateurs des paramètres ne sont pas faisables, et on doit utiliser des méthodes itératives telles que l'échantillonneur de Gibbs ou l'algorithme de Metropolis. Le passage du HMF au TMF amplifie ce problème car les complexités de l'échantillonneur de Gibbs et de l'algorithme de Metropolis dépendent du nombre d'états cachés, qui est démultiplié. Par ailleurs, le balayage de Peano a aussi été utilisé dans des modèles plus sophistiqués que les HMC, comme les chaînes de Markov cachées floues [44] [31] [90], ou les chaînes de Markov couples [31], [24]. Ainsi notre idée est d'étendre les CMC-CPS aux triplets conditionnels avec balayage de Peano contextuel (CTMC-CPS) en nous inspirant de l'extension des HMC aux TMC. Cette extension pourrait permettre de proposer une alternative aux TMF, en combinant une modélisation plus adaptée à l'image que les chaînes, et des calculs de loi *a posteriori* et d'estimation des paramètres relativement simples.

Nous présentons ici deux CTMC-CPS. En premier lieu, nous avons étendu les CMC-CPS aux chaînes de Markov évidentielles conditionnelles contextuelles (CEMC-CPS), proposant ainsi une alternative aux HEMF. Puis nous avons construit les chaînes de semi-Markov conditionnelles contextuelles (CSMC-CPS). Enfin nous montrons qu'il est possible de construire des chaînes de Markov couples conditionnelles contextuelles (CPMC-CPS). Pour chacun des modèles nous montrons que le calcul de l'estimateur MPM est possible, ainsi que l'estimation des paramètres par SEM, puis nous présentons des résultats de segmentations sur des images synthétiques.

5.1 Combiner le Peano contextuel et les chaînes de Markov triplets cachés

Les HMC ayant été étendu aux CMC-CPS, on peut se demander comment transposer cette extension aux chaînes de Markov triplet. Nous allons nous intéresser plus particulièrement aux HTMC, présentés à la section 2.3. En effet dans une HTMC (X, U, Y) , si on pose $V = (X, U)$, alors (V, Y) est un HMC. Ainsi la transformation d'une HTMC en chaîne de Markov triplet

conditionnelle avec balayage de Peano contextuelle (CTMC-CPS) se déduit facilement de celle des HMC en CMC-CPS. En remarquant que dans un CTMC-CPS, malgré le fait que le triplet (X, U, Y) n'est plus markovien, $p(x, u|y)$ l'est toujours, on peut utiliser l'estimateur du MPM, et l'estimation des paramètres par l'algorithme SEM.

5.1.1 Définition du modèle

Soit S un ensemble de pixels carré de dimension $N = 2^R \times 2^R$. Soit $(1, 2, \dots, N)$ un PS de S , et soit $([1, v_1, w_1], [2, v_2, w_2], \dots, [N, v_N, w_N])$ le CPS associé au PS. Soit $X = (X_1, \dots, X_N)$ un processus caché qui prend ses valeurs à partir d'un ensemble fini de classes $\Omega = \{\omega_1, \dots, \omega_K\}$ et qui doit être estimé à partir d'un processus observable $Y = (Y_1, \dots, Y_N)$ qui prend ses valeurs dans \mathbb{R} . Soit un troisième processus $U = (U_1, \dots, U_N)$ à valeurs dans $\Lambda = \{\lambda_1, \dots, \lambda_m\}$. Nous limiterons ici notre présentation aux TMC vérifiant (2.21), donc les HTMC. Soit s, t voisins dans S , mais pas dans le PS. S'ils sont voisins horizontaux, posons :

$$p^{hor}(y_t|x_s, u_s) = \sum_{x_t, u_t} p^{hor}(x_t, u_t|x_s, u_s)p(y_t|x_t) \quad (5.1)$$

De la même manière, s'ils sont voisins verticaux, on pose :

$$p^{ver}(y_t|x_s, u_s) = \sum_{x_t, u_t} p^{ver}(x_t, u_t|x_s, u_s)p(y_t|x_t) \quad (5.2)$$

Ainsi pour $s \in S$ et v, w voisins de s dans S l'ensemble des pixels, mais pas voisins dans le PS, on a :

$$p(y_v, y_s, y_w|x_s, u_s) = p(y_s|x_s)p^{a(v,s)}(y_v|x_s, u_s)p^{a(w,s)}(y_w|x_s, u_s) \quad (5.3)$$

Où $a(v, s) = hor$ si v, s sont des voisins horizontaux, et $a(v, s) = ver$ si v, s sont des voisins verticaux.

Par analogie avec la distribution d'une HTMC classique (2.21), nous définissons un nouveau modèle, appelé « chaînes de Markov triplet cachées conditionnelles avec un balayage de Peano contextuel » (CTMC-CPS), de la manière suivante. En notant $Z = (X, U)$, nous posons :

$$q(z, y) = p(z_1)p((y_1, y_{v_1}, y_{w_1}|z_1) \prod_{n=1}^{N-1} p(z_{n+1}|z_n)p(y_{n+1}, y_{v_{n+1}}, y_{w_{n+1}}|z_{n+1})) \quad (5.4)$$

Comme pour les CMC-CPS, $q(z, y)$ n'est pas la densité de probabilité du triplet (X, U, Y) par rapport à la mesure usuelle $(c \times \lambda)^N$ sur $(\Omega \times \mathbb{R})^N$, avec c la mesure de comptage et λ la mesure de Lebesgue. Cependant en considérant la constante de normalisation :

$$\kappa = \frac{1}{\sum_{x,u} \int q(z,y) dy}, \quad (5.5)$$

nous avons :

$$p(z,y) = \kappa q(z,y) \quad (5.6)$$

De la même façon que pour les CMC-CPS, $Z = (X, U)$ est markovien conditionnellement à Y , soit $p(z|y)$ est une loi markovienne. Cela va permettre de calculer l'estimateur MPM et donc faire de la segmentation bayésienne, comme nous allons le montrer dans la prochaine section. Cependant, $p(x, u, y)$ n'est pas markovienne, ce qui rend κ incalculable dans la pratique.

5.1.2 Inférence

Le processus couple (X, U, Y) obtenu par (5.4) n'est pas une HTMC classique, en particulier $p(y|x, u) \neq p(y|x)$. Cependant, comme $p(x, u|y)$ est markovien, la segmentation bayésienne est faisable. En effet pour calculer l'estimateur MPM donné par (1.18), il nous suffit être capable de calculer $p(x_n|y)$. On peut remarquer que $p(x, u|y) \propto q(x, u, y)$, avec $q(x, u, y)$ défini par (5.4). En posant $Z = (X, U)$ on a alors, en vertu du **lemme 4.1** :

$$p(x|y) \propto \prod_{n=1}^{N-1} \varphi_n(z_n, z_{n+1}, y) \quad (5.7)$$

avec :

$$\begin{aligned} \varphi_1(z_1, z_2, y) &= p(z_1, z_2) p(y_1, y_{v_1}, y_{w_1} | z_1) p(y_2, y_{v_2}, y_{w_2} | z_2) \\ \forall n \in \{2, \dots, N\}, \varphi_n(z_{n+1}, z_n, y) &= p(z_{n+1} | z_n) p(y_{n+1}, y_{v_{n+1}}, y_{w_{n+1}} | z_{n+1}) \end{aligned} \quad (5.8)$$

Finalement, les fonctions $\varphi_1, \dots, \varphi_{N-1}$ vérifient (4.9), et sont de la forme :

$$\begin{aligned} \varphi_1(z_1, z_2, y) &= \varphi_1(z_1, z_2, y_1, y_{v_1}, y_{w_1}, y_2, y_{v_2}, y_{w_2}) \\ \forall n \in \{2, \dots, N\}, \varphi_n(z_{n+1}, z_n, y) &= \varphi_n(z_{n+1}, z_n, y_{n+1}, y_{v_{n+1}}, y_{w_{n+1}}) \end{aligned} \quad (5.9)$$

De plus on peut les calculer facilement avec (5.8). On peut donc appliquer le **lemme 4.1** pour obtenir $p(z_1|y)$ et $p(z_{n+1}|z_n, y)$, et finalement on obtient $p(z_n|y)$ toujours par le **lemme 4.1**. Enfin, on termine le calcul de $p(x_n|y)$ avec

$$p(x_n|y) = \sum_{u_n \in \Lambda} p(z_n|y) = \sum_{u_n \in \Lambda} p(x_n, u_n|y) \quad (5.10)$$

ce qui permet d'utiliser l'estimateur MPM (1.18).

5.1.3 Estimation des paramètres non supervisée

Supposons que $p(x, u, y)$ est la distribution d'une HTMC à un bruit gaussien $p(y|x)$, avec deux transitions différentes $p^{ver}(x_{n+1}, u_{n+1}|x_n, u_n)$ et $p^{hor}(x_{n+1}, u_{n+1}|x_n, u_n)$ suivant que les pixels $n + 1$ et n sont respectivement des voisins verticaux ou horizontaux dans l'image originale. Pour K classes $\Omega = \{\omega_1, \dots, \omega_K\}$ pour X_n et m classes $\Lambda = \{\lambda_1, \dots, \lambda_m\}$ pour U_n , les paramètres de la HTMC sont : $K^2\Lambda^2$ probabilités $p^{ver} = (p_{ijkl}^{ver})_{\substack{1 \leq i, j \leq K \\ 1 \leq k, l \leq m}}$, avec $p_{ijkl}^{ver} = p(x_{n+1} = \omega_j, u_{n+1} = \lambda_l, x_n = \omega_i, u_n = \lambda_k)$ pour $n, n + 1$ voisins verticaux dans l'image, $K^2\Lambda^2$ probabilités $p^{hor} = (p_{ijkl}^{hor})_{\substack{1 \leq i, j \leq K \\ 1 \leq k, l \leq m}}$, avec $p_{ijkl}^{hor} = p(x_{n+1} = \omega_j, u_{n+1} = \lambda_l, x_n = \omega_i, u_n = \lambda_k)$ pour $n, n + 1$ voisins horizontaux dans l'image, K moyennes $\mu = (\mu_i)_{1 \leq i \leq K}$ et K variances $\sigma^2 = (\sigma_i^2)_{1 \leq i \leq K}$ associées aux K distributions gaussiennes $(p(y_n|x_n = \omega_i))_{1 \leq i \leq K}$. Si on s'intéresse à la CTMC-CPS, en choisissant $p^{hor}(x_{v_n}, u_{v_n}|x_n, u_n)$ et $p^{ver}(x_{w_n}, u_{w_n}|x_n, u_n)$ intervenant dans $p(y_{v_n}, y_n, y_{w_n}|x_n, u_n)$ comme égales à p_{ijkl}^{hor} et p_{ijkl}^{ver} respectivement, celle-ci possède exactement les mêmes paramètres que la HTMC. Ainsi le problème est d'estimer $\theta = (p^{hor}, p^{ver}, \mu, \sigma^2)$ à partir de l'image observée $Y = y$ seulement. Comme dans les CMC-CPS, on ne peut pas utiliser l'algorithme EM. Cependant on peut remarquer que l'on a accès à la loi *a posteriori* $p(x, u|y)$ dont on sait qu'elle est markovienne grâce au **Lemme 4.1**, avec la loi initiale et les transitions calculables avec (4.10). Ainsi on peut utiliser une variante de l'algorithme SEM, où l'on simule $x^{(q)}$ et $u^{(q)}$ à partir de (4.10), puis on utilise les estimateurs empiriques classiques de la loi du quadruplet $(x_n, u_n, x_{n+1}, u_{n+1})$ et des moyennes et variances pour mettre à jour les paramètres.

On pose H l'ensemble des couples $(n, n + 1)$, avec n et $n + 1$ voisins horizontaux dans l'image et V l'ensemble des couples $(n, n + 1)$, avec n et $n + 1$ voisins verticaux dans l'image. L'algorithme SEM pour les CTMC-CPS se déroule alors de la manière suivante :

- On initialise l'algorithme avec $\theta = \theta^0$
- Pour chaque itération q on utilise θ^q pour simuler $(x^{(q)}, u^{(q)})$ selon (4.10).

Ensuite on calcule θ^{q+1} de la manière suivante :

$$p_{ijkl}^{hor, (q+1)} = \frac{\sum_{(n, n+1) \in H} \mathbf{1}_{\{x_n^{(q)} = \omega_i, u_n^{(q)} = \lambda_k, x_{n+1}^{(q)} = \omega_j, u_{n+1}^{(q)} = \lambda_l\}}}{|H|} \quad (5.11)$$

$$p_{ijkl}^{ver, (q+1)} = \frac{\sum_{(n, n+1) \in V} \mathbf{1}_{\{x_n^{(q)} = \omega_i, u_n^{(q)} = \lambda_k, x_{n+1}^{(q)} = \omega_j, u_{n+1}^{(q)} = \lambda_l\}}}{|V|} \quad (5.12)$$

$$\mu_i^{(q+1)} = \frac{\sum_{n=1}^N \mathbf{1}_{\{x_n^{(q)} = \omega_i\}} y_n}{\sum_{n=1}^N \mathbf{1}_{\{x_n^{(q)} = \omega_i\}}} \quad (5.13)$$

$$\sigma_i^{2,(q+1)} = \frac{\sum_{n=1}^N \mathbf{1}_{\{x_n^{(q)} = \omega_i\}} (y_n - \mu_i^{(q+1)})^2}{\sum_{n=1}^N \mathbf{1}_{\{x_n^{(q)} = \omega_i\}}} \quad (5.14)$$

5.2 Chaînes de Markov évidentielles conditionnelles

Comme dans le cas des chaînes de Markov, on peut aussi introduire la théorie de Dempster-Shafer dans un contexte de champs Markovien [110]. Les HEMF sont des modèles proposant une des manières de le faire. Ils ont l'avantage, par rapport aux HEMC, de proposer un modèle probabiliste plus adapté à la structure d'une image, tout en gardant la capacité à segmenter des données non stationnaires [111], [106]. Cependant, comme dans tous les modèles de champs markoviens, les calculs associés à la loi *a posteriori* et à l'estimation des paramètres sont coûteux. Nous proposons ici une alternative aux HEMF, en nous appuyant sur la théorie des CTMC-CPS introduite précédemment et la construction des HEMC par TMC pour obtenir un modèle de chaînes de Markov évidentielles conditionnelles avec balayage de Peano contextuel (CEMC-CPS). Nous présentons rapidement comment calculer l'estimateur du MPM et comment estimer les paramètres en non-supervisé dans ce modèle, puis nous réalisons une étude de segmentations d'images synthétiques pour évaluer les performances de celui-ci.

5.2.1 Définition du modèle

Soit S un ensemble de pixel carré de dimension $N = 2^R \times 2^R$. Soit $(1, 2, \dots, N)$ un PS de S , et soit $([1, v_1, w_1], [2, v_2, w_2], \dots, [N, v_N, w_N])$ le CPS associé au PS. Soit $X = (X_1, \dots, X_N)$ un processus caché qui prend ses valeurs à partir d'un ensemble fini de classes $\Omega = \{\omega_1, \dots, \omega_K\}$ et qui doit être estimé à partir d'un processus observable $Y = (Y_1, \dots, Y_N)$ qui prend ses valeurs dans \mathbb{R} . Soit un troisième processus $U = (U_1, \dots, U_N)$ ou chaque U_n prend ses valeurs dans 2^Ω . Une CEMC-CPS est une CTMC-CPS vérifiant (5.4) avec :

$$p(z_{n+1}|z_n) = p(u_{n+1}|u_n)p(x_{n+1}|u_{n+1}) \quad (5.15)$$

$$p(y_v, y_s, y_w | x_s, u_s) = p(y_s | x_s) p^{a(v,s)}(y_v | u_s) p^{a(w,s)}(y_w | u_s) \quad (5.16)$$

$a(v, s) = hor$ si v, s sont des voisins horizontaux, et $a(v, s) = ver$ si v, s sont des voisins verticaux.

Avec pour s, t voisins horizontaux dans S , mais pas dans le PS:

$$p^{hor}(y_t | u_s) = \sum_{x_t, u_t} p^{hor}(u_t | u_s) p(x_t | u_t) p(y_t | x_t) \quad (5.17)$$

et pour s, t voisins verticaux dans S , mais pas dans le PS :

$$p^{ver}(y_t | u_s) = \sum_{x_t, u_t} p^{ver}(u_t | u_s) p(x_t | u_t) p(y_t | x_t) \quad (5.18)$$

Enfin :

$$\forall n \in \{1, \dots, N\}, \quad p(x_n | u_n) = \frac{\mathbf{1}_{\{x_n \in u_n\}}}{|u_n|} \quad (5.19)$$

5.2.2 Inférence et estimation des paramètres

Par analogie avec les HEMC, l'estimateur du MPM dans une CEMC-CPS peut être calculé si l'on remarque que les celles-ci sont des CTMC-CPS particulières. Ainsi il suffit d'utiliser le **lemme 4.1** et (5.10) pour calculer les $p(x_n | y)$. Pour l'estimation des paramètres non supervisée, on pourrait aussi se référer aux CTMC-CPS, cependant, comme les HEMC, les CEMC-CPS ont la particularité que $\forall n \in [1, \dots, N], p(x_n | u_n)$ est fixé par (5.19) et ne fait donc pas partie des paramètres à estimer. Nous allons présenter la version de l'algorithme SEM pour les CEMC-CPS ci-dessous.

Supposons que $p(x, u, y)$ est la distribution d'une HEMC avec un bruit gaussien $p(y|x)$, et deux transitions différentes $p^{ver}(u_{n+1} | u_n)$ et $p^{hor}(u_{n+1} | u_n)$ suivant que les pixels $n + 1$ et n sont respectivement des voisins verticaux ou horizontaux dans l'image originale. Pour K classes $\Omega = \{\omega_1, \dots, \omega_K\}$ pour X_n , les U_n sont des processus à $C = |2^\Omega| - 1$ classes. Les paramètres de la HEMC sont donc: C^2 probabilités $p^{ver} = (p_{kl}^{ver})_{1 \leq k, l \leq C}$, avec $p_{kl}^{ver} = p(u_{n+1} = \lambda_l, u_n = \lambda_k)$ pour $n, n + 1$ voisins verticaux dans l'image, C^2 probabilités $p^{hor} = (p_{kl}^{hor})_{1 \leq k, l \leq C}$, avec $p_{kl}^{hor} = p(u_{n+1} = \lambda_l, u_n = \lambda_k)$ pour $n, n + 1$ voisins horizontaux dans l'image, K moyennes $\mu = (\mu_i)_{1 \leq i \leq K}$ et K variances $\sigma^2 = (\sigma_i^2)_{1 \leq i \leq K}$ associés aux K distributions gaussiennes $(p(y_n | x_n = \omega_i))_{1 \leq i \leq K}$. Si on s'intéresse à la CEMC-CPS, en choisissant $p^{hor}(u_{v_n} | u_n)$ et $p^{ver}(u_{w_n} | u_n)$ intervenant dans $p(y_{v_n}, y_n, y_{w_n} | x_n, u_n)$ comme égales à p_{ijkl}^{hor} et p_{ijkl}^{ver} respectivement, celle-ci possède exactement les mêmes paramètres que la HEMC.

On pose H l'ensemble des couples $(n, n + 1)$, avec n et $n + 1$ voisins horizontaux dans l'image et V l'ensemble des couples $(n, n + 1)$, avec n et $n + 1$ voisins verticaux dans l'image. L'algorithme SEM pour les CEMC-CPS se déroule alors de la manière suivante :

- On initialise l'algorithme avec $\theta = \theta^0$
- Pour chaque itération q on utilise θ^q pour simuler $(x^{(q)}, u^{(q)})$ selon (4.10).

Ensuite on calcule θ^{q+1} de la manière suivante :

$$p_{kl}^{hor, (q+1)} = \frac{\sum_{(n, n+1) \in H} \sum_{i=1}^K \sum_{j=1}^K \mathbf{1}_{\{x_n^{(q)} = \omega_i, u_n^{(q)} = \lambda_k, x_{n+1}^{(q)} = \omega_j, u_{n+1}^{(q)} = \lambda_l\}}}{|H|} \quad (5.20)$$

$$p_{kl}^{ver, (q+1)} = \frac{\sum_{(n, n+1) \in V} \sum_{i=1}^K \sum_{j=1}^K \mathbf{1}_{\{x_n^{(q)} = \omega_i, u_n^{(q)} = \lambda_k, x_{n+1}^{(q)} = \omega_j, u_{n+1}^{(q)} = \lambda_l\}}}{|V|} \quad (5.21)$$

$$\mu_i^{(q+1)} = \frac{\sum_{n=1}^N 1_{\{x_n^{(q)}=\omega_i\}} y_n}{\sum_{n=1}^N 1_{\{x_n^{(q)}=\omega_i\}}} \quad (5.22)$$

$$\sigma_i^{2,(q+1)} = \frac{\sum_{n=1}^N 1_{\{x_n^{(q)}=\omega_i\}} (y_n - \mu_i^{(q+1)})^2}{\sum_{n=1}^N 1_{\{x_n^{(q)}=\omega_i\}}} \quad (5.23)$$

5.2.3 Résultats expérimentaux

Dans cette section, nous comparons la CEMC-CPS, avec les HEMC et le parcours de Peano classique (HEMC-PS), les CMC-CPS, les HMF et les HEMF dans le cadre de la segmentation d'images non-supervisée. L'idée est de répondre à deux questions :

(i) est-ce que la CEMC-CPS améliore simultanément la HEMC-PS et la CMC-CPS en segmentation d'images ?

(ii) comment la CEMC-CPS se compare aux HMF et HEMF ?

Pour y répondre, nous avons fait une étude de segmentations non supervisées d'images synthétiques bruitées par un bruit gaussien indépendant. L'estimation des paramètres en non supervisé a été faite par l'algorithme SEM pour HEMC-PS, CMC-CPS et CEMC-CPS. Par ailleurs, pour les CEMC-CPS et les CMC-CPS, nous n'avons pas différencié les transitions verticales et horizontales, soit $p^{hor} = p^{ver}$. Pour les HMF nous avons estimé les paramètres par un algorithme EM gibbsien [85]. L'initialisation pour tous les modèles a été faite grâce à une segmentation à deux classes, réalisée par un algorithme k-means, celui-ci étant initialisé avec la méthode « K-moyennes++ » [76].

Le choix des images dans cette étude est lié avec ce que nous avons déjà observé sur les HEMC-PS et les CMC-CPS. Les trois premières images ont des détails très fins, difficiles à segmenter pour les HMC (**figures 27, 28, 29**), alors que les deux dernières sont des images (**figures 30, 31**) peu homogènes mais sans détails fins, où la CMC-CPS améliore la HMC-PS classique de manière significative. Ainsi on remarque que dans le cas de présence de détails fins (**figures 27, 28, 29**), la CEMC-CPS améliore les CMC-CPS et les HEMC-PS et peut même, dans certaines situations, améliorer les HMF (**figures 28, 29**) et les HEMF (**figure 29**). Par contre, dans le cadre d'images peu homogènes mais sans détails fins (**figures 30, 31**), on remarque que la CEMC-CPS a tendance à dégrader la segmentation par rapport aux CMC-CPS, mais améliore quand même celle par les HEMC-PS.

On peut aussi remarquer que suivant ce que l'on recherche dans la segmentation, le taux d'erreur n'est pas toujours le critère le plus pertinent. En effet si on prend par exemple la **figure 28**, on peut argumenter que la segmentation par CEMC-CPS est plus intéressante que celle par HMF. En effet si on compare les deux segmentations visuellement, les détails de l'empreinte

digitale sont bien mieux restaurés et il y a moins de « points blancs parasites » sur le fond noir dans le cas du modèle de CEMC-CPS, cependant les taux d’erreurs des deux modèles sont très proches.

En conclusion, on peut observer que les CEMC-CPS sont au moins équivalentes aux HEMC et les améliorent souvent, avec un gain moyen relatif sur le taux d’erreur d’environ 16%. Leur intérêt par rapport aux HMF et CMC-CPS apparaît lorsqu’une image contient des détails fins que le modèle segmente mieux ; cependant, quand il n’y a pas de détails fins, les CEMC-CPS peuvent légèrement dégrader les CMC-CPS. Enfin les CEMC-CPS semblent mieux segmenter les détails fins que les HEMF, mais il conviendrait de faire une étude plus approfondie pour clarifier ce point.

Tableau 10. Récapitulatifs des taux d’erreurs de segmentations des modèles de HEMC-PS, CMC-CPS, CEMC-CPS et HMF pour les **figures 27-31** ainsi que HEMF pour la **figure 29**.

Image	Bruit	HEMC-PS	CMC-CPS	CEMC-CPS	HMF	HEMF
figure 27	$N(0,1), N(1,1)$	17.4%	14.9%	11.5%	8.8%	-
figure 28	$N(0,1), N(1,1)$	16.8%	15.5%	12.0%	12.5%	-
figure 29	$N(0,1), N(2,1)$	5.5%	15.8%	5.4%	12.8%	6.7%
figure 30	$N(0,1), N(1,1)$	9.3%	7.4%	8.7%	6.1%	-
figure 31	$N(0,1), N(1,1)$	20.3%	17.8%	18.7%	18.0%	-

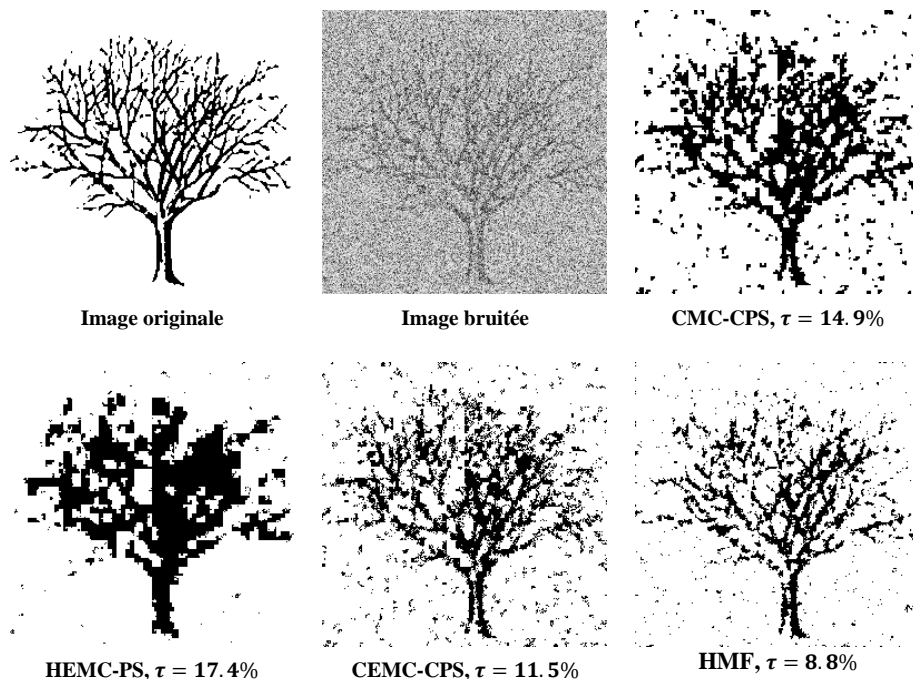


Figure 27. Segmentation d'une image a deux classes (Image originale), bruitée par un bruit gaussien $N(0, 1)$, $N(1, 1)$ (Image bruitée). CMC-CPS : résultat de la segmentation non supervisée avec la CMC-CPS (taux d'erreur $\tau = 14.9\%$), HEMC-PS : résultat de la segmentation non supervisée avec la HEMC-PS (taux d'erreur $\tau = 17.4\%$), CEMC-CPS : résultat de la segmentation non supervisée avec la CEMC-CPS (taux d'erreur $\tau = 11.5\%$), HMF : résultat de la segmentation non supervisée avec le HMF (taux d'erreur $\tau = 8.8\%$).

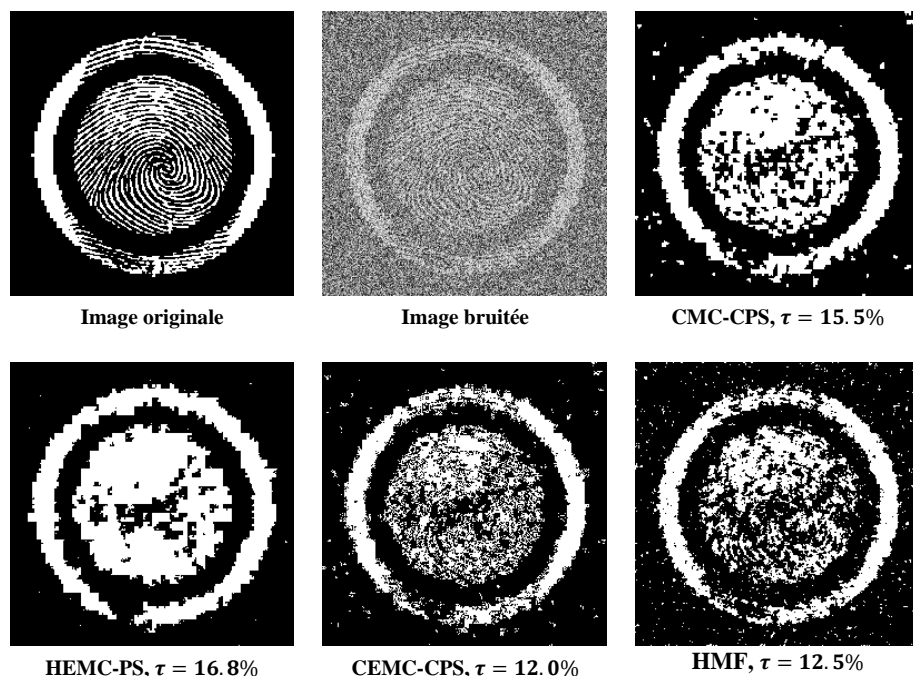


Figure 28. Segmentation d'une image a deux classes (Image originale), bruitée par un bruit gaussien $N(0, 1)$, $N(1, 1)$ (Image bruitée). CMC-CPS : résultat de la segmentation non supervisée avec la CMC-CPS (taux d'erreur $\tau = 15.5\%$), HEMC-PS : résultat de la segmentation non supervisée avec la HEMC-PS (taux d'erreur $\tau = 16.8\%$), CEMC-CPS : résultat de la segmentation non supervisée avec la CEMC-CPS (taux d'erreur $\tau = 12.0\%$), HMF : résultat de la segmentation non supervisée avec le HMF (taux d'erreur $\tau = 12.5\%$).

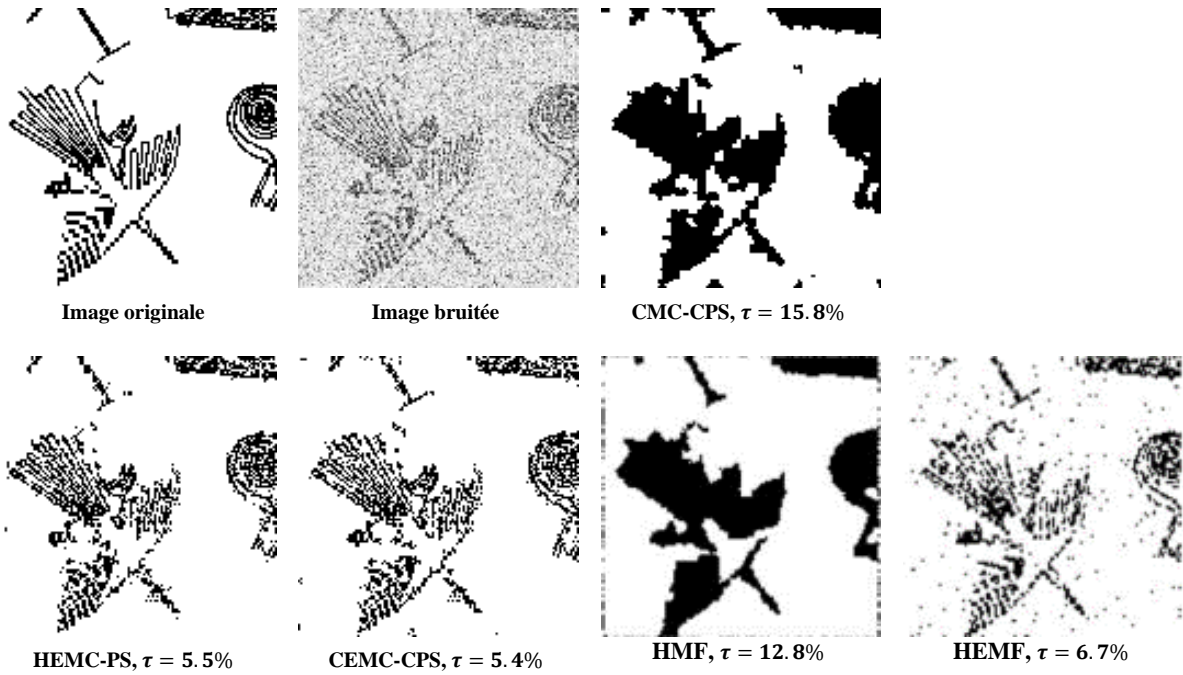


Figure 29. Segmentation d'une image a deux classes (Image originale), bruitée par un bruit gaussien $N(0, 1)$, $N(2, 1)$ (Image bruitée). CMC-CPS : résultat de la segmentation non supervisée avec la CMC-CPS (taux d'erreur $\tau = 15.8\%$), HEMC-PS : résultat de la segmentation non supervisée avec la HEMC-PS (taux d'erreur $\tau = 5.5\%$), CEMC-CPS : résultat de la segmentation non supervisée avec la CEMC-CPS (taux d'erreur $\tau = 5.4\%$), HMF : résultat de la segmentation non supervisée avec le HMF (taux d'erreur $\tau = 12.8\%$), HEMF : résultat de la segmentation non supervisée avec le HEMF (taux d'erreur $\tau = 6.7\%$).

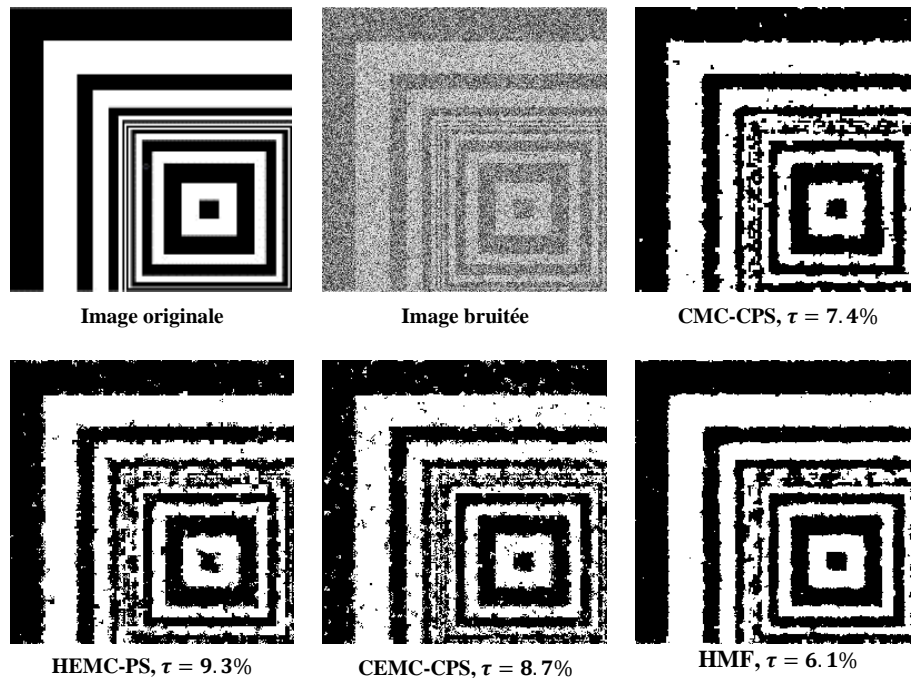


Figure 30. Segmentation d'une image a deux classes (Image originale), bruitée par un bruit gaussien $N(0, 1)$, $N(1, 1)$ (Image bruitée). CMC-CPS : résultat de la segmentation non supervisée avec la CMC-CPS (taux d'erreur $\tau = 7.4\%$), HEMC-PS : résultat de la segmentation non supervisée avec la HEMC-PS (taux d'erreur $\tau = 9.3\%$), CEMC-CPS : résultat de la segmentation non supervisée avec la CEMC-CPS (taux d'erreur $\tau = 8.7\%$), HMF : résultat de la segmentation non supervisée avec le HMF (taux d'erreur $\tau = 6.1\%$).

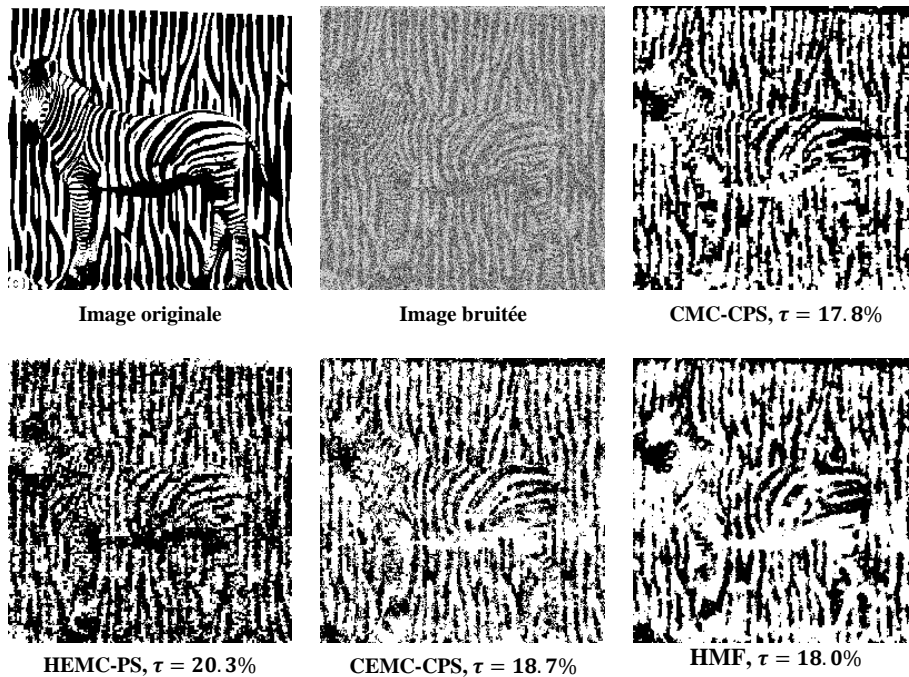


Figure 31. Segmentation d'une image à deux classes (Image originale), bruitée par un bruit gaussien $N(0, 1)$, $N(1, 1)$ (Image bruitée). CMC-CPS : résultat de la segmentation non supervisée avec la CMC-CPS (taux d'erreur $\tau = 17.8\%$), HEMC-PS : résultat de la segmentation non supervisée avec la HEMC-PS (taux d'erreur $\tau = 20.3\%$), CEMC-CPS : résultat de la segmentation non supervisée avec la CEMC-CPS (taux d'erreur $\tau = 18.7\%$), HMF : résultat de la segmentation non supervisée avec le HMF (taux d'erreur $\tau = 18.0\%$).

5.3 Chaînes de semi-Markov conditionnelles

Les HSMC considérées dans cette thèse sont des TMC particuliers permettant de modéliser une loi de séjour dans un état qui n'est pas géométrique [27], or nous avons construit les CTMC-CPS, équivalent des TMC dans le contexte des chaînes de Markov conditionnelles. Ainsi en choisissant un CTMC-CPS particulier, il devient possible de considérer la semi-markovianité avec le balayage de Peano contextuel. Nous proposons ainsi une alternative plus simple que la modélisation de la semi-markovianité à deux dimensions, mais qui peut se révéler suffisamment efficace pour de nombreuses applications. Nous présentons comment calculer l'estimateur du MPM et comment estimer les paramètres en non-supervisé dans ce modèle, puis nous réaliserons une étude de segmentations d'images synthétiques pour évaluer les performances de celui-ci.

5.3.1 Définition du modèle

Soit S un ensemble de pixel carré de dimension $N = 2^R \times 2^R$. Soit $(1, 2, \dots, N)$ un PS de S , et soit $([1, v_1, w_1], [2, v_2, w_2], \dots, [N, v_N, w_N])$ le CPS associé au PS. Soit $X = (X_1, \dots, X_N)$ un processus caché qui prend ses valeurs à partir d'un ensemble fini de classes $\Omega = \{\omega_1, \dots, \omega_K\}$ et qui doit être estimé à partir d'un processus observable $Y = (Y_1, \dots, Y_N)$ qui prend ses valeurs dans \mathbb{R} . Soit un troisième processus $U = (U_1, \dots, U_N)$ à valeur dans $\Lambda = \{0, 1, \dots, M - 1\}$ tel

que $U_n = e \neq 0$ implique que X_n va rester dans le même état au moins pour un temps e . Ainsi $(X_n, U_n) = (x_n, e)$ implique que $x_n = x_{n+1} = x_{n+2} = \dots = x_{n+d}$. Pour $e = 0$, on peut avoir $x_{n+1} = x_n$ ou non. Pour a, b, c, e posons $\delta_{(a,b)}(c, e) = 1$ si $(a, b) = (c, e)$, et $\delta_{(a,b)}(c, e) = 0$ si $(a, b) \neq (c, e)$. Nous appelons chaînes de semi-Markov triplet conditionnelles avec balayage de Peano contextuel (CSMC-CPS) les CTMC-CPS vérifiant (5.4) avec :

$$p(x_{n+1}, u_{n+1} | x_n, u_n) = \begin{cases} \delta_{(x_{n+1}, u_{n+1})}(x_n, u_n - 1) & \text{si } u_n > 0 \\ p^*(x_{n+1}, u_{n+1} | x_n, u_n = 0) & \text{si } u_n = 0 \end{cases} \quad (5.24)$$

et :

$$p(y_v, y_s, y_w | x_s, u_s) = p(y_s | x_s) p^{a(v,s)}(y_v | x_s, u_s) p^{a(w,s)}(y_w | x_s, u_s) \quad (5.25)$$

où $a(v, s) = hor$ si v, s sont des voisins horizontaux, et $a(v, s) = ver$ si v, s sont des voisins verticaux.

Avec pour s, t voisins horizontaux dans S , mais pas dans le PS:

$$p^{hor}(y_t | x_s, u_s) = \sum_{x_t, u_t} p^{hor}(x_t, u_t | x_s, u_s) p(y_t | x_t) \quad (5.26)$$

et pour s, t voisins verticaux dans S , mais pas dans le PS :

$$p^{ver}(y_t | x_s, u_s) = \sum_{x_t, u_t} p^{ver}(x_t, u_t | x_s, u_s) p(y_t | x_t) \quad (5.27)$$

Pour finir :

$$p^{hor}(x_t, u_t | x_s, u_s) = \begin{cases} \delta_{(x_t, u_t)}(x_s, u_s - 1) & \text{si } u_s > 0 \\ p^{*,hor}(x_t, u_t | x_s, u_s = 0) & \text{si } u_s = 0 \end{cases} \quad (5.28)$$

$$p^{ver}(x_t, u_t | x_s, u_s) = \begin{cases} \delta_{(x_t, u_t)}(x_s, u_s - 1) & \text{si } u_s > 0 \\ p^{*,ver}(x_t, u_t | x_s, u_s = 0) & \text{si } u_s = 0 \end{cases} \quad (5.29)$$

5.3.2 Inférence et Estimation des paramètres

Par analogie avec les HSMC, on peut remarquer que les CSMC-CPS sont des CTMC-CPS particuliers. Ainsi l'estimateur du MPM peut être obtenu en utilisant le **lemme 4.1** et (5.10) pour calculer les $p(x_n | y)$. Pour l'estimation des paramètres non supervisée, l'algorithme SEM pour les CTMC-CPS, présenté en section 5.1.3 fonctionne pour les CSMC-CPS. En particulier, si on initialise $p(x_{n+1}, u_{n+1}, x_n, u_n)$ de sorte que cette probabilité respecte (5.24), alors les équations (5.11) et (5.14) nous garantissent, à chaque itération, que le nouveau $p(x_{n+1}, u_{n+1}, x_n, u_n)$ calculé respecte (5.24).

5.3.3 Résultats expérimentaux

Dans cette section, nous comparons la CSMC-CPS, avec les HSMC et le parcours de Peano classique (HSMC-PS), les CMC-CPS et les HMF dans le cadre de la segmentation non-supervisée d'images. L'idée est de répondre à deux questions :

(i) Est-ce que la CSMC-CPS améliore simultanément la HSMC-PS et la CMC-CPS en segmentation d'image ?

(ii) Comment la CSMC-CPS se compare aux HMF ?

Pour y répondre, nous allons présenter une étude de segmentations non supervisées d'images synthétiques bruitées par un bruit gaussien indépendant. L'estimation des paramètres en non supervisé a été faite par l'algorithme SEM pour les HSMC-PS, CMC-CPS et CSMC-CPS. Par ailleurs, pour les CSMC-CPS et les CMC-CPS, nous n'avons pas différencié les transitions verticales et horizontales, soit $p^{hor} = p^{ver}$. Pour les HMF nous avons estimé les paramètres par un algorithme EM gibbsien [85]. L'initialisation pour tous les modèles a été faite grâce à une segmentation à deux classes, réalisée par un algorithme k-means, celui-ci étant initialisé avec la méthode « K-moyennes++ » [76].

Nous avons choisi, comme pour l'étude de segmentations sur la CMC-CPS, des images peu homogènes mais sans détails fins cette fois-ci, car nous savons que la semi-markovianité n'a que peu d'intérêt pour ces derniers. Les résultats des segmentations de chaque image par chacun des modèles sont présentés en **figures 32, 33, 34** et **35**. Ils nous permettent les conclusions suivantes :

- Les CSMC-CPS améliorent systématiquement les HSMC-PS pour la segmentation d'images, avec un gain relatif moyen de 23% sur le taux d'erreur.
- Les CSMC-CPS sont au moins aussi performantes que les CMC-CPS (**figures 32, 33**) et peuvent améliorer les CMC-CPS avec un gain allant jusqu'à 8% sur le taux d'erreur (**figures 34, 35**). Enfin dans certains cas (**figure 34**), les CSMC-CPS peuvent améliorer les résultats de la segmentation par HMF.

En conclusion on peut observer que la combinaison entre semi-markovianité et balayage de Peano contextuel peut améliorer la segmentation d'images par HSMC-PS de manière significative et à un intérêt par rapport aux CMC-CPS : elle fait au moins aussi bien que celle-ci, et peut améliorer légèrement les segmentations dans certains cas. Par rapport aux HMF, la CSMC-CPS se rapproche encore plus du résultat de la segmentation par HMF que la CMC-CPS, pouvant être meilleure dans certains cas particuliers.

Tableau 11. Récapitulatifs des taux d'erreurs de segmentations des modèles de HSMC-PS, CMC-CPS, CSMC-CPS et HMF pour les **figures 32-35**.

Image	Bruit		HSMC-PS	CMC-CPS	CSMC-CPS	HMF
figure 32	$N(0,1)$,	$N(1,1)$	12.3%	9.1%	9.2%	8.9%
figure 33	$N(0,1)$,	$N(1,1)$	9.4%	7.4%	7.2%	6.1%
figure 34	$N(0,1)$,	$N(1,1)$	19.7%	17.8%	16.4%	18.0%
figure 35	$N(0,1)$,	$N(1,1)$	17.3%	13.3%	12.5%	11.9%

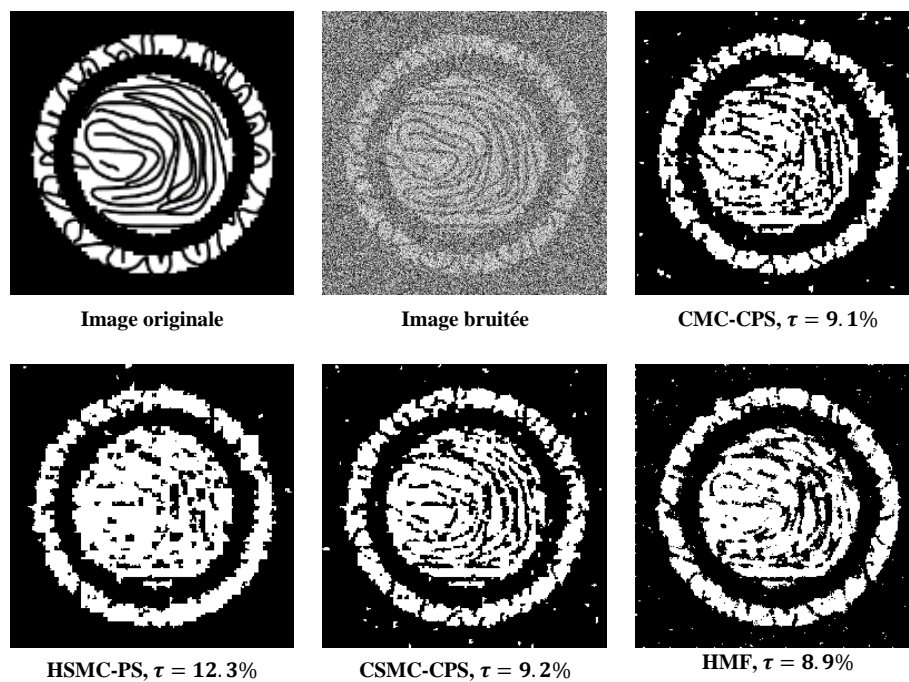


Figure 32. Segmentation d'une image a deux classes (Image originale), bruitée par un bruit gaussien $N(0, 1)$, $N(1, 1)$ (Image bruitée). CMC-CPS : résultat de la segmentation non supervisée avec la CMC-CPS (taux d'erreur $\tau = 9.1\%$), HSMC-PS : résultat de la segmentation non supervisée avec la HSMC-PS (taux d'erreur $\tau = 12.3\%$), CSMC-CPS : résultat de la segmentation non supervisée avec la CSMC-CPS (taux d'erreur $\tau = 9.2\%$), HMF : résultat de la segmentation non supervisée avec le HMF (taux d'erreur $\tau = 8.9\%$).

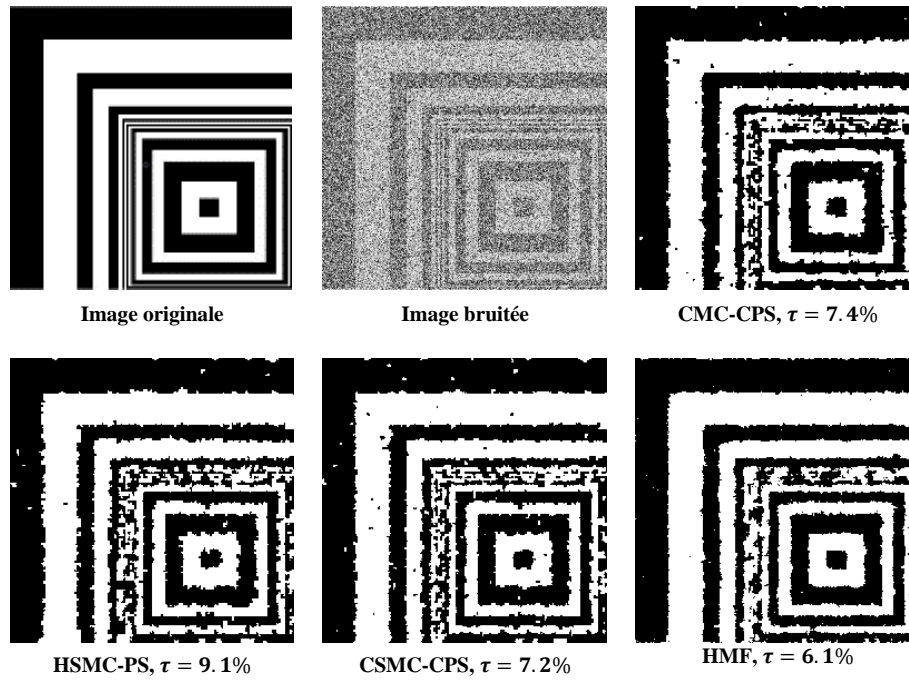


Figure 33. Segmentation d'une image a deux classes (Image originale), bruitée par un bruit gaussien $N(0, 1)$, $N(1, 1)$ (Image bruitée). CMC-CPS : résultat de la segmentation non supervisée avec la CMC-CPS (taux d'erreur $\tau = 7.4\%$), HSMC-PS : résultat de la segmentation non supervisée avec la HSMC-PS (taux d'erreur $\tau = 9.1\%$), CSMC-CPS : résultat de la segmentation non supervisée avec la CSMC-CPS (taux d'erreur $\tau = 7.2\%$), HMF : résultat de la segmentation non supervisée avec le HMF (taux d'erreur $\tau = 6.1\%$).

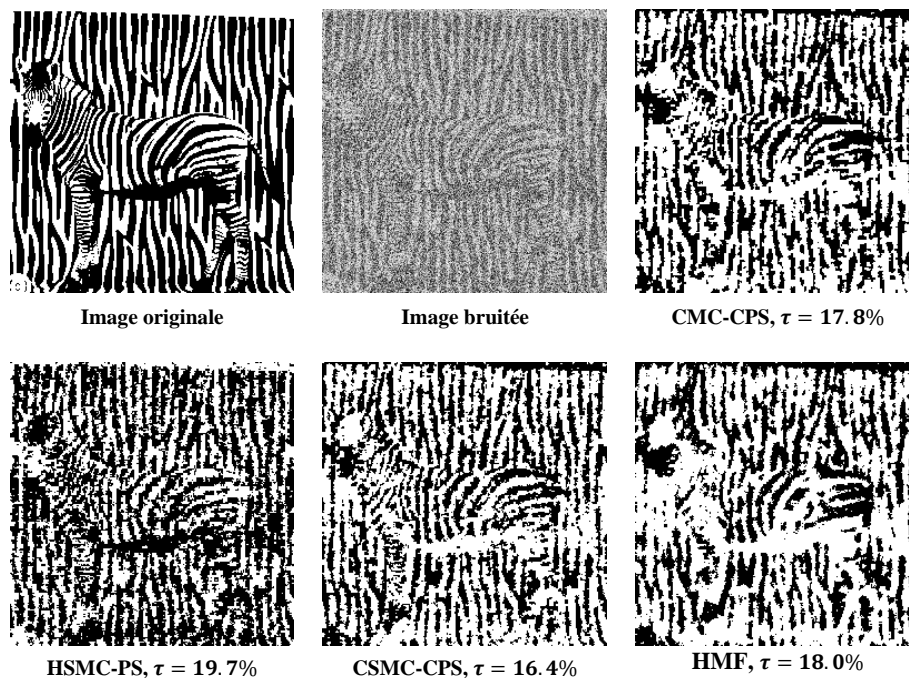


Figure 34. Segmentation d'une image a deux classes (Image originale), bruitée par un bruit gaussien $N(0, 1)$, $N(1, 1)$ (Image bruitée). CMC-CPS : résultat de la segmentation non supervisée avec la CMC-CPS (taux d'erreur $\tau = 17.8\%$), HSMC-PS : résultat de la segmentation non supervisée avec la HSMC-PS (taux d'erreur $\tau = 19.7\%$), CSMC-CPS : résultat de la segmentation non supervisée avec la CSMC-CPS (taux d'erreur $\tau = 16.4\%$), HMF : résultat de la segmentation non supervisée avec le HMF (taux d'erreur $\tau = 18.0\%$).

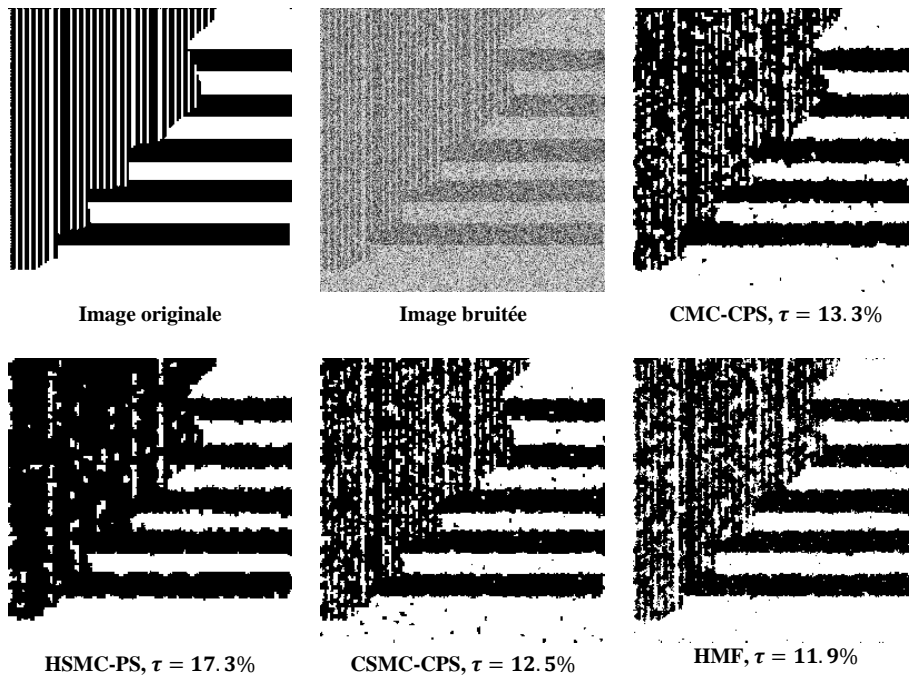


Figure 35. Segmentation d'une image a deux classes (Image originale), bruitée par un bruit gaussien $N(0, 1)$, $N(1, 1)$ (Image bruitée). CMC-CPS : résultat de la segmentation non supervisée avec la CMC-CPS (taux d'erreur $\tau = 13.3\%$), HSMC-PS : résultat de la segmentation non supervisée avec la HSMC-PS (taux d'erreur $\tau = 17.3\%$), CSMC-CPS : résultat de la segmentation non supervisée avec la CSMC-CPS (taux d'erreur $\tau = 12.5\%$), HMF : résultat de la segmentation non supervisée avec le HMF (taux d'erreur $\tau = 11.9\%$).

5.4 Chaînes de Markov couples conditionnelles

De la même façon que les HMC, les HMF ont été étendus aux champs de Markov couples. Cette extension est par ailleurs la base de la construction des champs de Markov triplet généraux. On peut alors, pour proposer une alternative aux PMF ainsi que pour poser les bases d'extensions futures, chercher à étendre les CMC-CPS aux chaînes de Markov couples conditionnelles avec balayage de Peano contextuel (CPMC-CPS). C'est ce que nous présentons ici avec, comme dans les sections précédentes, le calcul de l'estimateur MPM ainsi que l'estimation des paramètres. Nous réalisons finalement une étude de segmentations d'images synthétiques pour évaluer les performances du nouveau modèle.

5.4.1 Définition du modèle

Soit S un ensemble de pixel carré de dimension $N = 2^R \times 2^R$. Soit $(1, 2, \dots, N)$ un PS de S , et soit $([1, v_1, w_1], [2, v_2, w_2], \dots, [N, v_N, w_N])$ le CPS associé au PS. Soit $X = (X_1, \dots, X_N)$ un processus caché qui prend ses valeurs à partir d'un ensemble fini de classes $\Omega = \{\omega_1, \dots, \omega_K\}$ et qui doit être estimé à partir d'un processus observable $Y = (Y_1, \dots, Y_N)$ qui prend ses valeurs dans \mathbb{R} .

Soit s, t voisins dans S , mais pas dans le PS. S'ils sont voisins horizontaux, on pose :

$$p^{hor}(y_t|x_s, y_s) = \sum_{x_t} p^{hor}(y_t, x_t|x_s, y_s) \quad (5.30)$$

De la même manière, s'ils sont voisins verticaux, on pose :

$$p^{ver}(y_t|x_s, y_s) = \sum_{x_t} p^{ver}(y_t, x_t|x_s, y_s) \quad (5.31)$$

Par analogie avec la distribution d'une PMC classique (2.1), nous définissons un nouveau modèle, appelé « chaînes de Markov couples conditionnelles avec un balayage de Peano contextuel » (CPMC-CPS), de la manière suivante :

$$q(x, y) = p(x_1, y_1, y_{v_1}, y_{w_1}) \prod_{n=1}^{N-1} p(y_{v_{n+1}}, y_{w_{n+1}}, x_{n+1}, y_{n+1}, |x_n, y_n) \quad (5.32)$$

avec :

$$\begin{aligned} p(x_1, y_1, y_{v_1}, y_{w_1}) &= p(x_1, y_1) p^{a(v_1,1)}(y_{v_1}|x_1, y_1) p^{a(w_1,1)}(y_{w_1}|x_1, y_1) \\ p(y_{v_{n+1}}, y_{w_{n+1}}, x_{n+1}, y_{n+1}, |x_n, y_n) &= \\ p(y_{n+1}, x_{n+1}|x_n, y_n) p^{a(v_{n+1},n+1)}(y_{v_{n+1}}|x_{n+1}, y_{n+1}) p^{a(w_{n+1},n+1)}(y_{w_{n+1}}|x_{n+1}, y_{n+1}) \end{aligned} \quad (5.33)$$

où $a(v, s) = hor$ si v, s sont des voisins horizontaux, et $a(v, s) = ver$ si v, s sont des voisins verticaux.

Comme précédemment, $q(x, y)$ n'est pas la densité de probabilité du couple (X, Y) selon la mesure usuelle, cependant en considérant la constante de normalisation :

$$\kappa = \frac{1}{\sum_x \int q(x, y) dy}, \quad (5.34)$$

nous avons :

$$p(x, y) = \kappa q(x, y), \quad (5.35)$$

qui définit la loi de la CPMC-CPS.

5.4.2 Inférence

L'estimateur du MPM suppose d'être capable de calculer les marginales *a posteriori* $p(x_n|y)$ (1.18). Dans le cas de la CPMC-CPS, on peut remarquer que $p(x|y) \propto q(x, y)$, avec $q(x, y)$ défini par (5.32). On a alors :

$$p(x|y) \propto \prod_{n=1}^{N-1} \varphi_n(x_n, x_{n+1}, y), \quad (5.36)$$

avec :

$$\varphi_1(x_1, x_2, y) = p(x_1, y_1, y_{v_1}, y_{w_1}, x_2, y_2, y_{v_2}, y_{w_2})$$

$$\forall n \in \{2, \dots, N-1\} \varphi_n(x_{n+1}, x_n, y) = p(y_{v_{n+1}}, y_{w_{n+1}}, x_{n+1}, y_{n+1}, |x_n, y_n) \quad (5.37)$$

Finalement, les fonctions $\varphi_1, \dots, \varphi_{N-1}$ vérifient (4.9), et sont de la forme :

$$\begin{aligned} \varphi_1(x_1, x_2, y) &= \varphi_1(x_1, x_2, y_1, y_{v_1}, y_{w_1}, y_2, y_{v_2}, y_{w_2}) \\ \forall n \in \{2, \dots, N-1\}, \varphi_n(x_{n+1}, x_n, y) &= \varphi_n(x_n, x_{n+1}, y_n, y_{n+1}, y_{v_{n+1}}, y_{w_{n+1}}) \end{aligned} \quad (5.38)$$

et sont calculables par (4.14). On peut alors appliquer le **lemme 4.1** pour obtenir $p(z_1|y)$ et $p(z_{n+1}|z_n, y)$, qui mènent aux $p(x_n|y)$, toujours par le **lemme 4.1**. On peut donc utiliser l'estimateur MPM.

5.4.3 Estimation des paramètres non supervisée

Supposons que $p(x, y)$ est la distribution d'une PMC avec $p(y_n, y_{n+1}|x_n, x_{n+1})$ gaussien pour tout n , et deux lois différentes $p^{ver}(x_{n+1}, x_n)$ et $p^{hor}(x_{n+1}, x_n)$ suivant que les pixels $n+1$ et n sont respectivement des voisins verticaux ou horizontaux dans l'image originale. Pour K classes $\Omega = \{\omega_1, \dots, \omega_K\}$, les paramètres de la PMC sont : K^2 probabilités $p^{ver} = (p_{ij}^{ver})_{1 \leq i, j \leq K}$, avec $p_{ij}^{ver} = p(x_{n+1} = \omega_j, x_n = \omega_i)$ pour $n, n+1$ voisins verticaux dans l'image, K^2 probabilités $p^{hor} = (p_{ij}^{hor})_{1 \leq i, j \leq K}$, avec $p_{ij}^{hor} = p(x_{n+1} = \omega_j, x_n = \omega_i)$ pour $n, n+1$ voisins horizontaux dans l'image, K^2 vecteurs moyennes $\mu = \left\{ \begin{pmatrix} \mu_{i,j}^1 \\ \mu_{i,j}^2 \end{pmatrix} \right\}_{\substack{1 \leq i \leq K \\ 1 \leq j \leq K}}$ et K matrices de covariances $\Gamma = (\Gamma_{ij})_{\substack{1 \leq i \leq K \\ 1 \leq j \leq K}}$ associés aux K^2 distributions gaussiennes $(p(y_{n+1}, y_n | x_n = \omega_i, x_{n+1} = \omega_j))_{\substack{1 \leq i \leq K \\ 1 \leq j \leq K}}$. Si on s'intéresse à la CPMC-CPS, en choisissant $p^{hor}(y_{v_n}, x_{v_n} | x_n, y_n)$ et $p^{ver}(y_{w_n}, x_{w_n} | x_n, y_n)$ intervenant dans $p(y_{v_n}, y_{w_n}, x_n, y_n, |x_{n-1}, y_{n-1})$ comme égales à $p^{hor}(y_{n+1}, x_{n+1} | x_n, y_n)$ et $p^{ver}(y_{n+1}, x_{n+1} | x_n, y_n)$ respectivement, celle-ci possède exactement les mêmes paramètres que la PMC. Ainsi le problème est d'estimer $\theta = (p^{hor}, p^{ver}, \mu, \sigma^2)$ à partir de l'image observée $Y = y$ seulement. Comme dans les CMC-CPS, on ne peut pas utiliser l'algorithme EM. Cependant, on a accès à la loi *a posteriori* $p(x|y)$ dont on sait qu'elle est markovienne grâce au **Lemme 4.1**, avec la loi initiale et les transitions calculables avec (4.10). Ainsi on peut utiliser une variante de l'algorithme SEM, où l'on simule x^q à partir de (4.10), puis on utilise les estimateurs empiriques classiques de la loi du couple (x_{n+1}, x_n) et des moyennes et variances pour mettre les paramètres à jour.

On pose H l'ensemble des couples $(n, n+1)$, avec n et $n+1$ voisins horizontaux dans l'image et V l'ensemble des couples $(n, n+1)$, avec n et $n+1$ voisins verticaux dans l'image. L'algorithme SEM pour les CPMC-CPS se déroule alors de la manière suivante :

- On initialise l'algorithme avec $\theta = \theta^0$

- Pour chaque itération q on utilise θ^q pour simuler (x^q) selon (4.10). Ensuite on calcule θ^{q+1} de la manière suivante :

$$p_{ij}^{hor,(q+1)} = \frac{\sum_{(n,n+1) \in H} 1_{\{x_n^{(q)} = \omega_i, x_{n+1}^{(q)} = \omega_j\}}}{|H|} \quad (5.39)$$

$$p_{ij}^{ver,(q+1)} = \frac{\sum_{(n,n+1) \in V} 1_{\{x_n^{(q)} = \omega_i, x_{n+1}^{(q)} = \omega_j\}}}{|V|} \quad (5.40)$$

$$\mu_{ij}^{(q+1)} = \frac{\sum_{n=1}^{N-1} 1_{\{x_n^{(q)} = \omega_i, x_{n+1}^{(q)} = \omega_j\}} \begin{pmatrix} y_n \\ y_{n+1} \end{pmatrix}}{\sum_{n=1}^{N-1} 1_{\{x_n^{(q)} = \omega_i, x_{n+1}^{(q)} = \omega_j\}}} \quad (5.41)$$

$$\Gamma_{ij}^{(q+1)} = \frac{\sum_{n=1}^{N-1} 1_{\{x_n^{(q)} = \omega_i, x_{n+1}^{(q)} = \omega_j\}} \begin{pmatrix} y_n - \mu_{ij}^{1,(q+1)} \\ y_{n+1} - \mu_{ij}^{2,(q+1)} \end{pmatrix}^t \begin{pmatrix} y_n - \mu_{ij}^{1,(q+1)} \\ y_{n+1} - \mu_{ij}^{2,(q+1)} \end{pmatrix}}{\sum_{n=1}^{N-1} 1_{\{x_n^{(q)} = \omega_i, x_{n+1}^{(q)} = \omega_j\}}} \quad (5.42)$$

5.4.4 Résultats expérimentaux

Nous comparons ici la CPMC-CPS avec les PMC et le parcours de Peano classique (PMC-PS), ainsi que les CMC-CPS, dans le cadre de la segmentation d'images non-supervisée. L'idée est de vérifier si les CPMC-CPS sont capables d'améliorer simultanément les PMC-PS et les CMC-CPS. Pour cela, nous avons fait une étude de segmentations non supervisées d'images synthétiques bruitées par un bruit corrélé. En effet, les PMC, par rapport aux HMC, permettent de prendre en compte les liens entre les observations (conditionnellement aux classes) et sont donc plus efficaces pour gérer le bruit corrélé. Nous avons donc choisi de se placer dans ce cas précis pour mettre en lumière l'intérêt des CPMC-CPS. L'estimation des paramètres en non supervisé a été faite par l'algorithme SEM pour PMC-PS, CMC-CPS et CPMC-CPS. Par ailleurs, pour les CPMC-CPS et les CMC-CPS, nous n'avons pas différencié par les transitions verticales et horizontales, soit $p^{hor} = p^{ver}$. L'initialisation pour tous les modèles a été faite grâce à une segmentation à deux classes, réalisée par un algorithme k-means, celui-ci étant initialisé avec la méthode « K-moyennes++ » [76].

Le bruit corrélé a été obtenu de la manière suivante : on bruite d'abord par un bruit gaussien indépendant $N(\mu_1, \sigma_1^2)$ et $N(\mu_2, \sigma_2^2)$ pour obtenir $(W_s)_{s \in S}$. Puis on pose

$$\forall s \in S, Y_s = \frac{1}{1 + 4a} \left(W_s + a \sum_{i=1}^4 W_{s_i} \right), \quad (5.43)$$

avec s_1, \dots, s_4 les quatre plus proches voisins de s . Dans les expériences qui suivent nous avons choisi $a = 0.25$, $\mu_1 = 0$, $\sigma_1^2 = 2$, $\mu_2 = 1$ et $\sigma_2^2 = 4$

Les résultats des segmentations de chaque image par chacun des modèles sont présentés en **figures 36, 37, 38, 39, 40 et 41**. On peut remarquer que la CPMC-CPS peut améliorer les segmentations par PMC-PS avec un gain de 5% en moyenne sur le taux d'erreur, ce qui est peu par rapport aux améliorations observées précédemment. Cependant elle améliore la CMC-CPS de 14% dans le cas de bruit corrélé. En conclusion il est difficile de mettre en lumière un intérêt significatif des CPMC-CPS par rapport aux PMC-PS dans le cadre de la segmentation d'images, même si on remarque une certaine amélioration des résultats. Il faudrait des expériences supplémentaires pour approfondir le sujet.

Tableau 12. Récapitulatifs des taux d'erreurs de segmentations des modèles de PMC-PS, CMC-CPS et CPMC-CPS pour les **figures 36-41**.

Image	Bruit	PMC-PS	CMC-CPS	CPMC-CPS
figure 36	$N(0,2), N(1,4)$	14.6%	15.9%	13.9%
figure 37	$N(0,2), N(1,4)$	18.4%	26.2%	17.7%
figure 38	$N(0,2), N(1,4)$	12.8%	16.4%	12.0%
figure 39	$N(0,2), N(1,4)$	25.7%	25.0%	25.7%
figure 40	$N(0,2), N(1,4)$	20.9%	20.3%	19.5%
figure 41	$N(0,2), N(1,4)$	17.0%	18.9%	15.6%

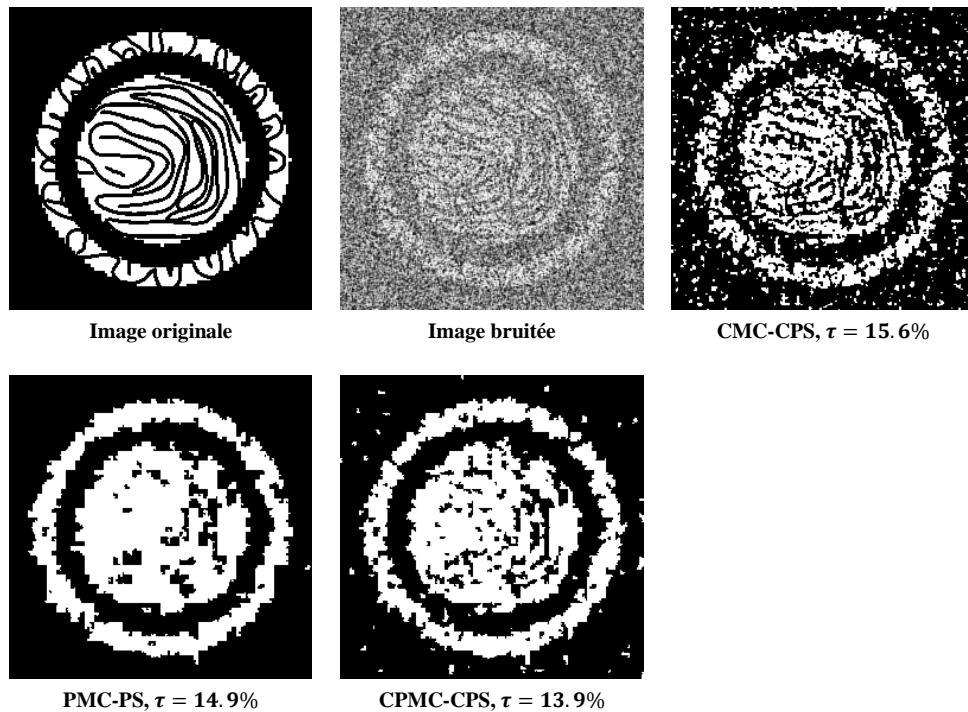


Figure 36. Segmentation d'une image a deux classes (Image originale), bruitée par un bruit corrélé $N(0,2), N(1,4)$ puis (5.43) (Image bruitée). CMC-CPS : résultat de la segmentation non supervisée avec la CMC-CPS (taux d'erreur $\tau = 15.6\%$), PMC-PS : résultat de la segmentation non supervisée avec la PMC-PS (taux d'erreur $\tau = 14.9\%$), CPMC-CPS : résultat de la segmentation non supervisée avec la CPMC-CPS (taux d'erreur $\tau = 13.9\%$).

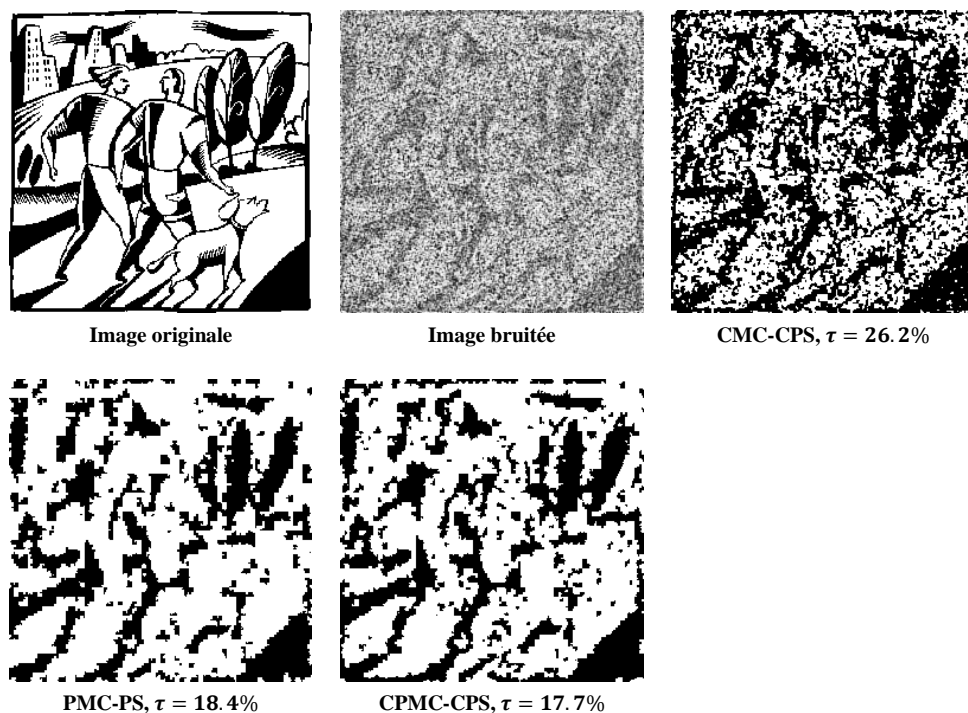


Figure 37. Segmentation d'une image a deux classes (Image originale), bruitée par un bruit corrélé $N(0, 2)$, $N(1, 4)$ puis (5.43) (Image bruitée). CMC-CPS : résultat de la segmentation non supervisée avec la CMC-CPS (taux d'erreur $\tau = 26.2\%$), PMC-PS : résultat de la segmentation non supervisée avec la PMC-PS (taux d'erreur $\tau = 18.4\%$), CPMC-CPS : résultat de la segmentation non supervisée avec la CPMC-CPS (taux d'erreur $\tau = 17.7\%$).

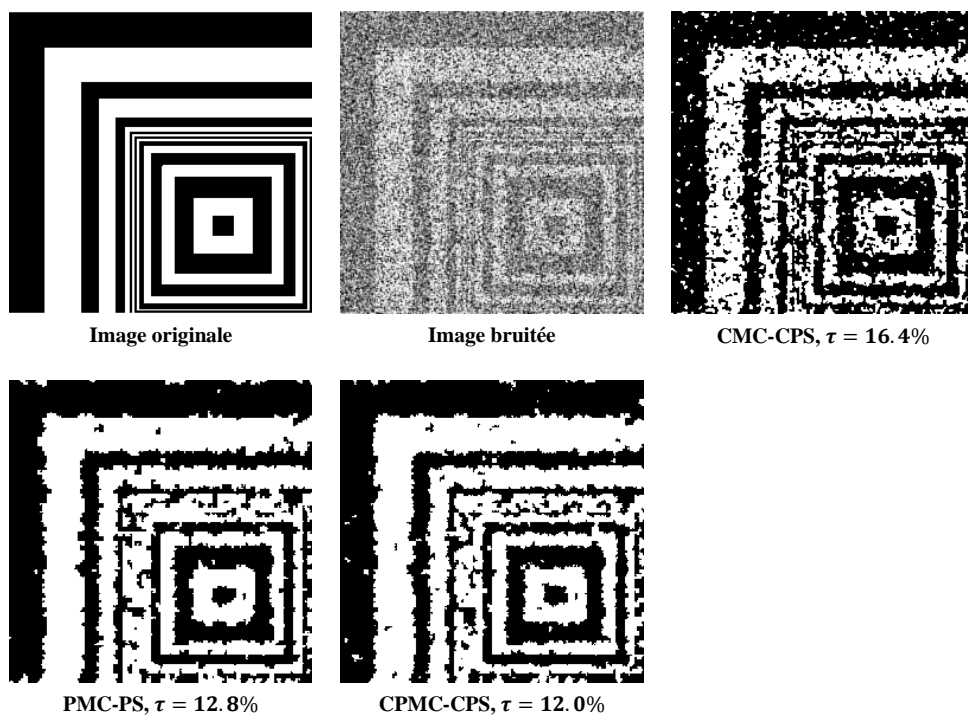


Figure 38. Segmentation d'une image a deux classes (Image originale), bruitée par un bruit corrélé $N(0, 2)$, $N(1, 4)$ puis (5.43) (Image bruitée). CMC-CPS : résultat de la segmentation non supervisée avec la CMC-CPS (taux d'erreur $\tau = 16.4\%$),

PMC-PS : résultat de la segmentation non supervisée avec la PMC-PS (taux d'erreur $\tau = 12.8\%$), CPMC-CPS : résultat de la segmentation non supervisée avec la CPMC-CPS (taux d'erreur $\tau = 12.0\%$).

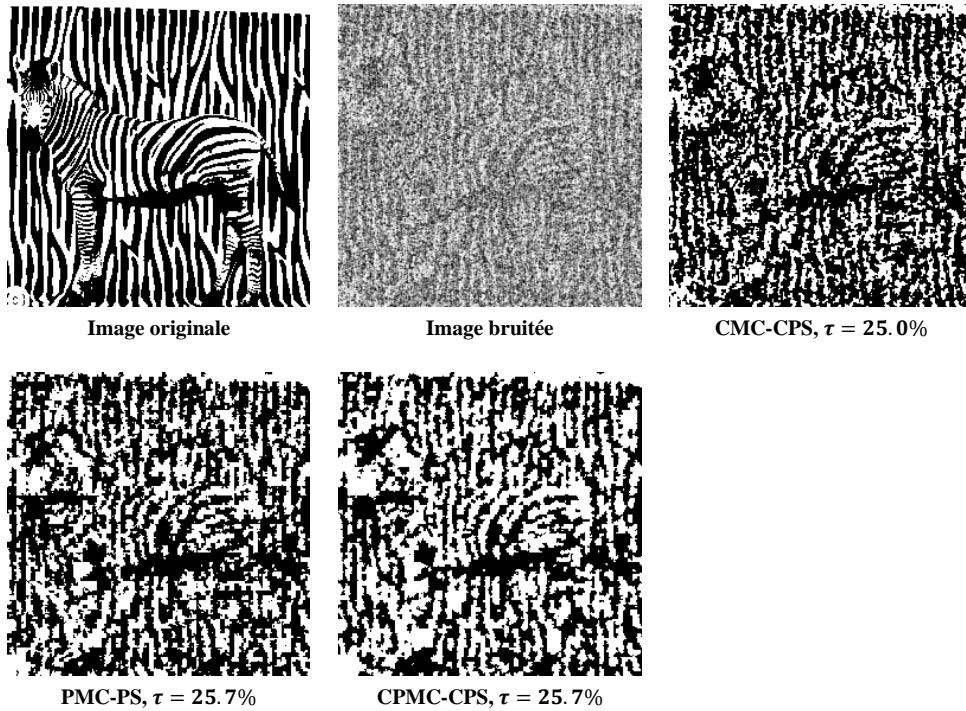


Figure 39. Segmentation d'une image a deux classes (Image originale), bruitée par un bruit corrélé $N(0, 2)$, $N(1, 4)$ puis (5.43) (Image bruitée). CMC-CPS : résultat de la segmentation non supervisée avec la CMC-CPS (taux d'erreur $\tau = 25.0\%$), PMC-PS : résultat de la segmentation non supervisée avec la PMC-PS (taux d'erreur $\tau = 25.7\%$), CPMC-CPS : résultat de la segmentation non supervisée avec la CPMC-CPS (taux d'erreur $\tau = 25.7\%$).

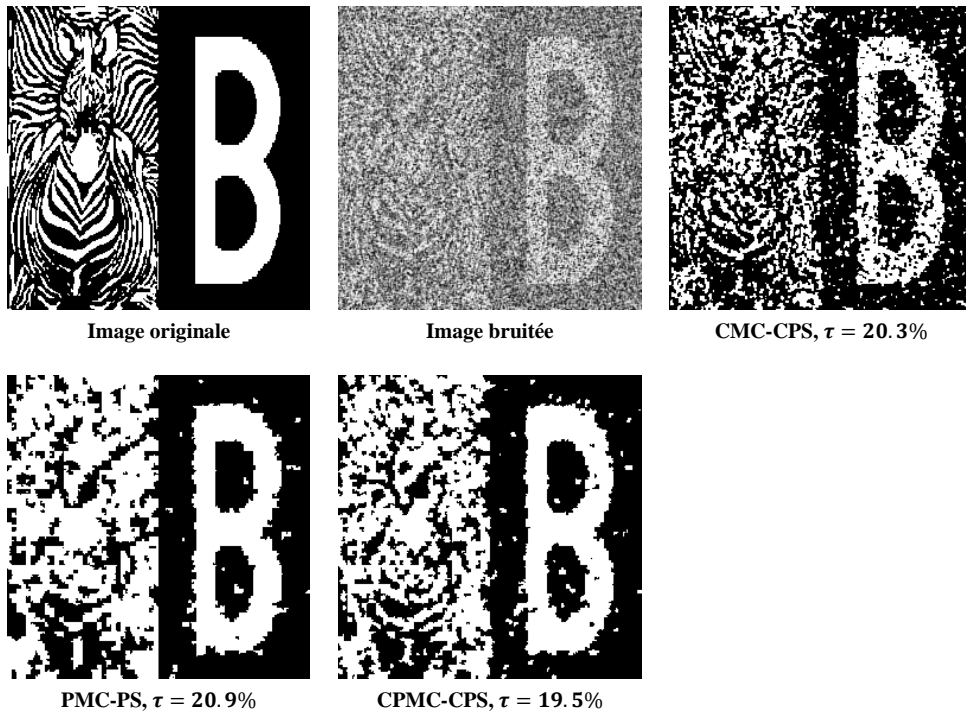


Figure 40. Segmentation d'une image a deux classes (Image originale), bruitée par un bruit corrélé $N(0, 2)$, $N(1, 4)$ puis (5.43) (Image bruitée). CMC-CPS : résultat de la segmentation non supervisée avec la CMC-CPS (taux d'erreur $\tau = 20.3\%$),

PMC-PS : résultat de la segmentation non supervisée avec la PMC-PS (taux d'erreur $\tau = 20.9\%$), CPMC-CPS : résultat de la segmentation non supervisée avec la CPMC-CPS (taux d'erreur $\tau = 19.5\%$).

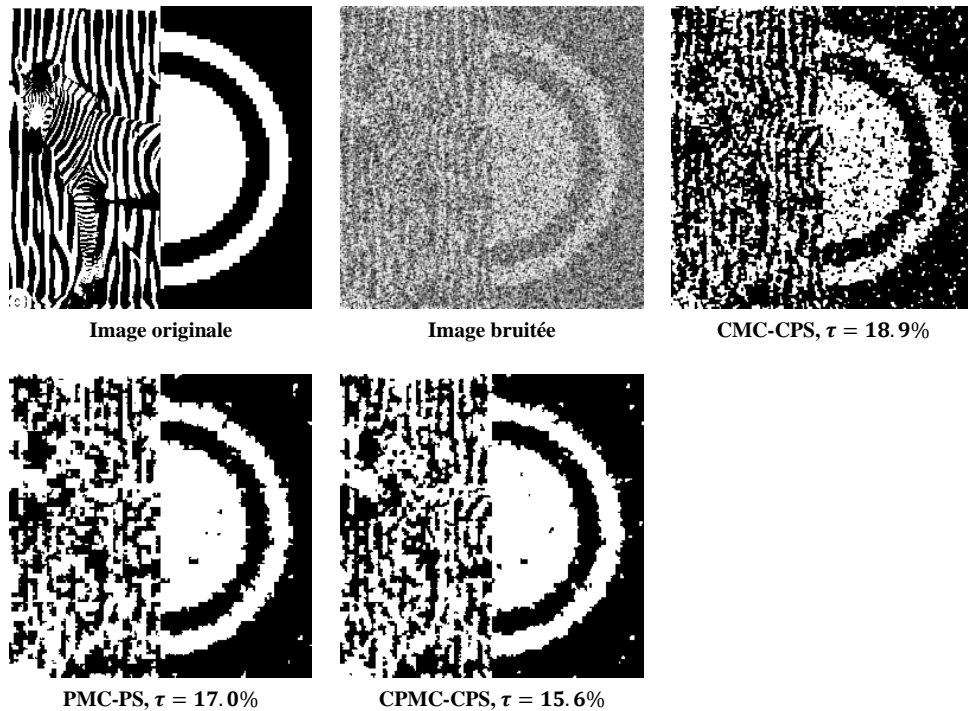


Figure 41. Segmentation d'une image a deux classes (Image originale), bruitée par un bruit corrélé $N(0, 2)$, $N(1, 4)$ puis (5.43) (Image bruitée). CMC-CPS : résultat de la segmentation non supervisée avec la CMC-CPS (taux d'erreur $\tau = 18.9\%$), PMC-PS : résultat de la segmentation non supervisée avec la PMC-PS (taux d'erreur $\tau = 17.0\%$), CPMC-CPS : résultat de la segmentation non supervisée avec la CPMC-CPS (taux d'erreur $\tau = 15.6\%$).

5.5 Conclusion

Nous avons présenté dans ce chapitre trois extensions des CMC-CPS aux modèles markoviens conditionnels couples et triplets : la CEMC-CPS, la CSMC-CPS et la CPMC-CPS. Nous avons montré pour chacun que l'on pouvait calculer l'estimateur du MPM et estimer les paramètres en non supervisé de manière efficace et rapide. Enfin nous avons présenté une étude de segmentations d'images pour chacun des trois modèles. Ceux-ci améliorent tous leur contrepartie non conditionnelle, à savoir respectivement les HEMC-PS, les HSMC-PS et les PMC-PS, dans le cadre de la segmentation d'image, et posent les bases d'extensions futures des CMC-CPS à des triplets plus généraux. Comparé aux HMF et TMF ces modèles peuvent proposer une alternative, certes un peu moins performante en moyenne, mais bien moins coûteuse en temps de calcul, leurs résultats de segmentation étant plus proches des champs markoviens que ne le sont les chaînes markoviennes classiques.

Conclusion et Perspectives

Le but principal de ces travaux était de proposer des modèles de chaînes de Markov triplets originales, généralisant les HMC, HSMC, et HEMC. Ainsi nous avons construit deux triplets markoviens combinant la théorie de l'évidence pour gérer la non-stationnarité des données et la semi-markovianité pour modéliser un temps de séjour dans un état qui ne soit pas géométrique. Ceux-ci généralisent à la fois les HSMC et les HEMC, qui eux-mêmes sont deux extensions des HMC. Ces derniers admettant un très grand nombre d'applications, Il en est donc de même des modèles proposés, qui plus est, en non-supervisé. Enfin nous avons montré expérimentalement l'excellent comportement des méthodes d'estimations dans le cadre de nos modèles. En effet dans les segmentations présentées, les deux triplets peuvent améliorer simultanément les HSMC et les HEMC, ou bien « se réduire » à l'un ou l'autre dans le cas où une seule des deux modélisations a un intérêt.

Dans le cadre de la segmentation non supervisée d'images, nous avons montré que ces deux triplets peuvent avoir de l'intérêt dans des images contenant à la fois des grandes zones homogènes et des détails fins, ainsi que du bruit non-stationnaire. Cependant, nous avons toujours un problème quant à la structure du modèle de chaîne pour segmenter une image. Intuitivement, on remarque que cette structure unidimensionnelle n'est pas adaptée à l'image, qui elle est bidimensionnelle. Au contraire, les champs de Markov cachés sont des modèles permettant de prendre en compte un voisinage bi-dimensionnel et ils ont aussi été étendus aux champs de Markov couples et triplets. Cependant, les calculs requis pour approcher la loi *a posteriori* et estimer les paramètres dans ces modèles sont très coûteux, le phénomène empirant lors du passage du caché au triplet. Nous avons alors conçu un modèle de chaîne de Markov conditionnelle, et un parcours spécifique associé, le parcours de Peano contextuel. Ces deux éléments rassemblés permettent de prendre en compte une partie du voisinage bidimensionnel, à savoir les plus proches voisins d'un pixel dans l'image qui ne sont plus voisins dans la chaîne. Nous avons montré que cette modélisation améliorerait de manière significative les chaînes de Markov cachées dans le cadre de la segmentation d'image, et pouvait même concurrencer les champs de Markov cachés dans certains cas. Finalement nous avons étendu les chaînes de Markov conditionnelles aux modèles couples ainsi qu'aux modèles triplets cachés, permettant d'introduire la semi-markovianité et la théorie de l'évidence dans celles-ci. En combinant ces modèles avec le parcours de Peano contextuel, nous avons observé que ceux-ci peuvent améliorer leur contrepartie non conditionnelle dans le cadre de la segmentation d'image, tout en posant les bases d'extensions futures à des triplets conditionnels plus généraux. Comparés

Conclusion et Perspectives

aux champs de Markov cachés, ils peuvent proposer une alternative, certes un peu moins performante en moyenne, mais pouvant les améliorer dans certain cas, et surtout bien moins coûteuse en temps de calcul. Enfin par rapport aux champs de Markov triplets, les chaînes triplets conditionnelles proposées ici permettent de mieux utiliser la généralité du triplet markovien dans le cadre de la segmentation d'image, en contournant les problèmes de temps de calcul considérables qui apparaissent lorsque l'on passe des champs de Markov cachés aux triplets.

Les modélisations proposées ouvrent la voie à d'autres extensions, d'une très grande richesse. Leur potentiel intérêt étant démultiplié par l'excellent comportement des méthodes d'estimations des paramètres, permettant des traitements non supervisés. En premier lieu, la généralité des modèles de Markov triplets ouvre de nombreuses perspectives pour des travaux futurs. En effet, il est tout à fait possible d'étendre les chaînes de semi-Markov évidentielles en rajoutant d'autres séquences aléatoires de façon à ce que le modèle reste une chaîne de Markov triplet et il peut être intéressant de le faire pour s'attaquer à divers problèmes rencontrés en segmentation d'images. Si on regarde du côté des chaînes de Markov triplets conditionnelles, celles-ci n'ont pas été explorées de fond en comble, et on voit qu'il est facile de construire de nouveaux triplets conditionnels, qui pourraient avoir un intérêt pour certains problèmes complexes de segmentation. De plus le parcours de Peano classique est, avec les chaînes de Markov cachées, couples et triplets, utilisé dans de nombreuses situations telles que les images en trois dimensions, les séquences d'images ou encore les images multi-résolutions. Ces situations, bien plus complexes que celles présentés dans ce document, ne sont pas adaptés aux méthodes fondées sur les champs de Markov qui deviennent bien trop coûteuses en temps. Ainsi la combinaison du parcours de Peano contextuel et des chaînes de Markov triplet conditionnelles est très susceptible d'avoir un intérêt dans ce genre de situations. Enfin les chaînes de Markov cachées ne sont pas utilisées qu'en segmentation d'image, et les différentes modélisations proposées ici pourraient améliorer de nombreux traitements dans les domaines où les chaînes de Markov cachées sont utilisées.

Annexes

A. Normalisation de la loi de chaîne de Markov évidentielle

Pour fixer les idées, nous allons regarder une simple chaîne de Markov évidentielle (EMC), sans, dans un premier temps, introduire de version bruitée de celle-ci. Soit $X = (X_1, \dots, X_N)$ un processus aléatoire qui prend ses valeurs à partir d'un ensemble fini de classes $\Omega = (\omega_1, \dots, \omega_K)$. Notons 2^Ω l'ensemble puissance de Ω et $2^{(\Omega^N)}$ l'ensemble puissance de (Ω^N) . Introduisons un deuxième processus aléatoire La séquence aléatoire $U = (U_1, \dots, U_N)$ à valeur dans 2^Ω . Définissons par ailleurs m comme une masse de Dempster-Shafer sur $(2^\Omega)^N$ de telle sorte que :

$$m(u_1, \dots, u_N) = m(u_1) \prod_{i=1}^{N-1} m(u_{i+1}|u_i) \quad (\text{A.1})$$

Nous prendrons m nulle sur $2^{(\Omega^N)}/(2^\Omega)^N$. Cela signifie que pour tout $n = 1, \dots, N-1$ et $u_n \in 2^\Omega$, $u_{n+1} \rightarrow m(u_{n+1}|u_n)$ est une masse de Dempster-Shafer sur 2^Ω . La distribution d'une EMC est une distribution de Markov obtenue par la normalisation de la fonction φ définie sur $\Omega^N \times (2^\Omega)^N$ telle que :

$$\varphi(x, u) = \prod_{n=1}^{N-1} \varphi_n(x_n, u_n, x_{n+1}, u_{n+1}) \quad (\text{A.2})$$

Où

$$\varphi_1(x_1, u_1, x_2, u_2) = 1_{\{(x_1, x_2) \in u_1 \times u_2\}} m(u_1) m(u_2|u_1) \quad (\text{A.3})$$

$$\begin{aligned} \forall n \in [2, N-1], \varphi_n(x_n, u_n, x_{n+1}, u_{n+1}) \\ = 1_{\{x_{n+1} \in u_{n+1}\}} m(u_{n+1}|u_n) \end{aligned} \quad (\text{A.4})$$

Pour simplifier, posons $v_n = (x_n, u_n)$. Les probabilités $p(v_1)$ et les transitions $p(v_{n+1}|v_n)$ d'une EMC obtenue avec (A.1) - (A.4) sont calculées avec les récursion « backward » [26].

En posant :

$$\begin{aligned} \beta_N^*(v_N) &= 1 \\ \beta_n^*(v_n) &= \sum_{v_{n+1}} \varphi_n(v_n, v_{n+1}) \beta_{n+1}^*(v_{n+1}) \end{aligned} \quad (\text{A.5})$$

Nous avons :

$$p(v_1) = \frac{\beta_1^*(v_1)}{\sum_{v_1} \beta_1^*(v_1)(x_1, u_1)} \quad (\text{A.6})$$

$$p(v_{n+1}|v_n) = \frac{\varphi_n(v_n, v_{n+1})\beta_{n+1}^*(v_{n+1})}{\beta_n^*(v_n)} \quad (\text{A.7})$$

Ainsi :

$$p(v_1) = \frac{\sum_{v_2} 1_{\{(x_1, x_2) \in u_1 \times u_2\}} m(u_1) m(u_2|u_1) \beta_2^*(v_2)}{\sum_{v_1} \sum_{v_2} 1_{\{(x_1, x_2) \in u_1 \times u_2\}} m(u_1) m(u_2|u_1) \beta_2^*(v_2)} \quad (\text{A.8})$$

Et :

$$p(v_{n+1}|v_n) = \frac{1_{\{x_{n+1} \in u_{n+1}\}} m(u_{n+1}|u_n) \beta_{n+1}^*(v_{n+1})}{\sum_{v_{n+1}} 1_{\{x_{n+1} \in u_{n+1}\}} m(u_{n+1}|u_n) \beta_{n+1}^*(v_{n+1})} \quad (\text{A.9})$$

On peut remarquer que pour $n = 2, \dots, N$, $\varphi_n(v_n, v_{n+1})$ ne dépend pas de x_n d'après (A.4). Et donc $\beta_n^*(v_n)$ ne dépend pas non plus de x_n d'après (A.5). (A.8) et (A.9) peuvent alors s'écrire de la manière suivante :

$$p(v_1) = \frac{1_{\{x_1 \in u_1\}} m(u_1) \sum_{u_2} m(u_2|u_1) \beta_2^*(v_2) \sum_{x_2} 1_{\{x_2 \in u_2\}}}{\sum_{u_1} \sum_{x_1} 1_{\{x_1 \in u_1\}} m(u_1) \sum_{u_2} m(u_2|u_1) \beta_2^*(v_2) \sum_{x_2} 1_{\{x_2 \in u_2\}}} \quad (\text{A.10})$$

$$p(v_{n+1}|v_n) = \frac{1_{\{x_{n+1} \in u_{n+1}\}} m(u_{n+1}|u_n) \beta_{n+1}^*(v_{n+1})}{\sum_{u_{n+1}} m(u_{n+1}|u_n) \beta_{n+1}^*(v_{n+1}) \sum_{x_{n+1}} 1_{\{x_{n+1} \in u_{n+1}\}}} \quad (\text{A.11})$$

La dernière remarque est la suivante :

$$\forall n \in [1, N], \sum_{x_n} 1_{\{x_n \in u_n\}} = |u_n| \quad (\text{A.12})$$

L'opérateur $|\cdot|$ dénotant le cardinal d'un ensemble.

Pour finir on va multiplier (A.11) par $\frac{|u_1|}{|u_1|}$ et (A.12) par $\frac{|u_{n+1}|}{|u_{n+1}|}$, $u_1 \neq \{\emptyset\}$, $u_{n+1} \neq \{\emptyset\}$, pour obtenir :

$$p(v_1) = \frac{|u_1| m(u_1) \sum_{u_2} |u_2| m(u_2|u_1) \beta_2^*(v_2)}{\sum_{u_1} |u_1| m(u_1) \sum_{u_2} |u_2| m(u_2|u_1) \beta_2^*(v_2)} \times \frac{1_{\{x_1 \in u_1\}}}{|u_1|} \quad (\text{A.13})$$

$$p(v_{n+1}|v_n) = \frac{|u_{n+1}| m(u_{n+1}|u_n) \beta_{n+1}^*(v_{n+1})}{\sum_{u_{n+1}} m(u_{n+1}|u_n) \beta_{n+1}^*(v_{n+1}) |u_{n+1}|} \times \frac{1_{\{x_{n+1} \in u_{n+1}\}}}{|u_{n+1}|} \quad (\text{A.14})$$

Les termes $\frac{1_{\{x_1 \in u_1\}}}{|u_1|}$, $\frac{|u_1| m(u_1) \sum_{u_2} |u_2| m(u_2|u_1) \beta_2^*(v_2)}{\sum_{u_1} |u_1| m(u_1) \sum_{u_2} |u_2| m(u_2|u_1) \beta_2^*(v_2)}$, $\frac{1_{\{x_{n+1} \in u_{n+1}\}}}{|u_{n+1}|}$,

$\frac{|u_{n+1}| m(u_{n+1}|u_n) \beta_{n+1}^*(v_{n+1})}{\sum_{u_{n+1}} m(u_{n+1}|u_n) \beta_{n+1}^*(v_{n+1}) |u_{n+1}|}$, sont bien des lois de probabilités, en particulier, tous somment

à 1. On obtient donc à la fin :

$$p(v_1) = p(u_1)p(x_1|u_1)$$

$$p(v_{n+1}|v_n) = p(u_{n+1}|u_n)p(x_{n+1}|u_{n+1})$$

Ce qui conclut notre démonstration.

B. Démonstration du lemme 4.1

Lemme 4.1

Soit $Z = (Z_1, \dots, Z_N)$ une chaîne stochastique prenant ses valeurs dans un ensemble fini Ω .

Alors :

- Z est markovien si et seulement s'il existe $N - 1$ fonctions $\varphi_1, \dots, \varphi_{N-1}$ de Ω^2 vers \mathbb{R}^+ telles que :

$$p(z) \propto \prod_{n=1}^{N-1} \varphi_n(z_n, z_{n+1}) \quad (B.1)$$

avec \propto exprimant la relation de proportionnalité.

- Si (B.9) est vérifiée, la probabilité initiale $p(z_1)$ et les transitions $p(z_{n+1}|z_n)$ s'expriment en fonction des $\varphi_1, \dots, \varphi_{N-1}$ de la manière suivante :

$$p(z_1) = \frac{\beta_1(z_1)}{\sum_{z_1} \beta_1(z_1)}$$

$$\text{Pour } n = 1, \dots, N - 1, \quad p(z_{n+1}|z_n) = \frac{\varphi_n(z_n, z_{n+1})\beta_{n+1}(z_{n+1})}{\beta_n(z_n)} \quad (B.2)$$

où $\beta_1(z_2), \dots, \beta_N(z_N)$ sont calculées à partir des récursions rétrogrades (« backward ») :

$$\beta_N(z_N) = 1$$

$$\beta_n(z_n) = \sum_{z_{n+1}} \varphi_{n+1}(z_n, z_{n+1})\beta_{n+1}(z_{n+1}) \quad (B.3)$$

- Une fois $p(z_1)$ et $p(z_{n+1}|z_n)$ données, chaque $p(z_n)$ peut être calculé à partir des récursion progressive (« forward ») :

$$\forall n \in [2, \dots, N], p(z_n) = \sum_{z_n} p(z_n|z_{n-1})p(z_{n-1}) \quad (B.4)$$

Preuve

1. Soit $Z = (Z_1, \dots, Z_N)$ une chaîne de Markov : $p(z_1, \dots, z_N) = p(z_1)p(z_2|z_1)p(z_3|z_2) \dots p(z_N|z_{N-1})$. Alors (B.1) est vérifiée en posant $\varphi_1(z_1, z_2) = p(z_1)p(z_2|z_1)$, $\varphi_2(z_2, z_3) = p(z_3|z_2)$, \dots , $\varphi_{N-1}(z_{N-1}, z_N) = p(z_N|z_{N-1})$.

Annexes

2. Inversement, soit $p(z_1, \dots, z_N)$ vérifiant (B.1), alors $p(z_1, \dots, z_N) = K \varphi_1(z_1, z_2) \dots \varphi_{N-1}(z_{N-1}, z_N)$, K étant une constante, ce qui implique que pour tout $n = 1, \dots, N - 1$ nous avons :

$$\begin{aligned}
 p(z_{n+1}|z_1, \dots, z_n) &= \frac{p(z_1, \dots, z_n, z_{n+1})}{p(z_1, \dots, z_n)} \\
 &= \frac{\sum_{(z_{n+2}, \dots, z_N)} \varphi_1(z_1, z_2) \dots \varphi_n(z_n, z_{n+1}) \varphi_{n+1}(z_{n+1}, z_{n+2}) \dots \varphi_{N-1}(z_{N-1}, z_N)}{\sum_{(z_{n+1}, z_{n+2}, \dots, z_N)} \varphi_1(z_1, z_2) \dots \varphi_n(z_n, z_{n+1}) \varphi_{n+1}(z_{n+1}, z_{n+2}) \dots \varphi_{N-1}(z_{N-1}, z_N)} \\
 &= \frac{\varphi_n(z_n, z_{n+1}) \sum_{(z_{n+2}, \dots, z_N)} \varphi_{n+1}(z_{n+1}, z_{n+2}) \dots \varphi_{N-1}(z_{N-1}, z_N)}{\sum_{(z_{n+1}, z_{n+2}, \dots, z_N)} \varphi_n(z_n, z_{n+1}) \varphi_{n+1}(z_{n+1}, z_{n+2}) \dots \varphi_{N-1}(z_{N-1}, z_N)} = p(z_{n+1}|z_n) \quad (B.5)
 \end{aligned}$$

ce qui montre que $p(z_1, \dots, z_N)$ est de Markov.

Par ailleurs, posons $\beta_n(z_n) = \sum_{(z_{n+1}, z_{n+2}, \dots, z_N)} \varphi_n(z_n, z_{n+1}) \dots \varphi_{N-1}(z_{N-1}, z_N)$ pour tout $n = 1, \dots, N - 1$. D'une part, nous pouvons remarquer que $\beta_n(z_n) = \sum_{z_{n+1}} \varphi_n(z_n, z_{n+1}) \beta_{n+1}(z_{n+1})$. D'autre part, selon (B.5), nous avons $p(z_{n+1}|z_n) = \frac{\varphi_n(z_n, z_{n+1}) \beta_{n+1}(z_{n+1})}{\beta_n(z_n)}$. Comme $p(z_1) = \frac{\beta_1(z_1)}{\sum_{z_1} \beta_1(z_1)}$, (B.2) et (B.3) sont vérifiées, ce qui termine la preuve.

Bibliographie

- [1] Sericola, B., Markov Chains: Theory and Applications, Applied stochastic methods series, John Wiley & Sons, 2013.
- [2] Gwet, D. , Ottesteanu, M. , Libouga, I. , Bitjoka, L. , Popa, G. 'A Review on Image Segmentation Techniques and Performance Measures'. World Academy of Science, Engineering and Technology, Open Science Index 144, International Journal of Computer and Information Engineering, 12(12), 1107 - 1117, 2018.
- [3] Mohamed El Yazid Boudaren, Modèles graphiques évidentiels, thèse en cotutelle entre l'Université USTHB, Alger, Algérie, et l'Université Paris VI, Paris, 2014.
- [4] A. P. Dempster, N. M. Laird, D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm (with discussion)," Journal of the Royal Statistical Society B, Vol. 39, pp. 1–38, 1977.
- [5] M. Broniatowski, G. Celeux, J. Diebolt, Reconnaissance de mélanges de densités par un algorithme d'apprentissage probabilistes, In Data Analysis and Informatics 3, E. Diday (Ed.), North Holland, Amsterdam, pp. 359-373, 1983.
- [6] G. Celeux, D. Diebolt, A random imputation principle: the stochastic EM algorithm, Technical report 901, INRIA Roquencourt, Sep. 1988.
- [7] G. Celeux, D. Chauveau, J. Diebolt, "Stochastic versions of the EM algorithm: an experimental study in the mixture case," J. Statistical Computation and Simulation, Vol. 55, No. 4, pp. 287–314, 1996.
- [8] W. Pieczynski, Statistical image segmentation, Machine Graphics and Vision, Vol. 1, No. 1/2 (Proceedings of GKIP0'92), pp. 261-268, 1992..
- [9] W. Pieczynski, Convergence of the iterative conditional estimation and application to mixture proportion identification, IEEE Statistical Signal Processing Workshop, SSP 2007, Madison, Wisconsin, USA, 26-29 August, 2007..
- [10] W. Pieczynski, "Champs de Markov cachés et estimation conditionnelle itérative," Traitement du Signal, Vol. 11, No. 2, pp. 141-153, 1994.

Bibliographie

- [11] W. Pieczynski, Sur la convergence de l'estimation conditionnelle itérative, Comptes Rendus de l'Académie des Sciences-Mathématique, Vol. 346, No. 7-8, pp. 457-460, Avril 2008.
- [12] L .R. Rabiner, « A tutorial on hidden Markov models and selected applications in speech recognition », Proceedings of IEEE, Vol . 77, No2, pp . 257 -286, 1989.
- [13] K. Aas, L. Eikvil, and R.B. Huseby, “Applications of hidden Markov chains in image analysis”, Pattern Recognition, vol. 32, no. 4, pp. 703-713, Apr. 1999.
- [14] W. Pieczynski, Modèles de Markov en traitement d'images, Traitement du Signal, Vol. 20, No. 3, pp. 255-278, 2003.
- [15] B. Benmiloud, W. Pieczynski, Estimation des paramètres dans les chaînes de Markov cachées et segmentation d'images, Traitement du Signal, Vol. 12, No. 5, pp. 433-454, 1995.
- [16] Bouyahia, Z., Benyoussef, L., Derrode, S., Change detection in synthetic aperture radar images with a sliding hidden Markov chain model. J. Appl. Remote Sens. 2, 023526, 2008.
- [17] M.-Y. Chen, A. Kundu, and J. Zhou, “Off-Line handwritten word recognition using a hidden Markov model type stochastic network,” IEEE Trans. Pattern Analysis and Machine Intelligence”, vol. 16, no. 5, pp. 481- 496, May 1994.
- [18] G.A. Churchill, “Hidden Markov chains and the analysis of genome structure”, Computers and Chemistry, vol. 16, no. 2, pp. 107-115, 1992.
- [19] Quoc Thong Nguyen, Yves Delignon, François Septier, Anh Thu Phan Ho. Probabilistic Modelling of Printed Dots at the Microscopic Scale. Signal Processing: Image Communication, Elsevier, 62, pp.129-138, 2018..
- [20] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," in IEEE Transactions on Information Theory, vol. 13, no. 2, pp. 260-269, 1967.
- [21] G. D. Forney Jr., “The Viterbi Algorithm,” Proc. IEEE, vol. 61, no. 3, pp. 268- 277, Mar. 1973.
- [22] L.E. Baum, T. Petrie, G. Soules, N. Weiss, “A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains,” Ann. Math. Statistics, Vol. 41, No. 1, pp. 164-171, Feb. 1970.

Bibliographie

- [23] W. Pieczynski, Pairwise Markov Chains and Bayesian Unsupervised Fusion, Proceedings of 3rd International Conference on Information Fusion, FUSION 2000, Vol. 1, Paris, France, pp. MoD4-24 - MoD4-31 , 2000.
- [24] W. Pieczynski, Pairwise Markov chains, IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 25, No. 5, pp. 634-639, 2003.
- [25] W. Pieczynski, Chaînes de Markov Triplet, Comptes Rendus de l'Académie des Sciences – Mathématique, Série I, Vol. 335, Issue 3, pp. 275-278, 2002.
- [26] M. Y. Boudaren and W. Pieczynski, “Dempster-Shafer fusion of evidential pairwise Markov chains”, IEEE Trans. on Fuzzy Systems, Vol. 24, No. 6, pp. 1598-1610, December 2016.
- [27] J. Lapuyade-Lahorgue and W. Pieczynski, “Unsupervised segmentation of hidden semi-Markov non stationary chains”, Signal Processing, vol. 92, no. 1, pp. 29–42, January 2012.
- [28] P. Lanchantin and W. Pieczynski, Unsupervised restoration of hidden non stationary Markov chain using evidential priors, IEEE Trans. on Signal Processing, vol. 53, no. 8, pp. 3091-3098, 2005.
- [29] M. Y. Boudaren, E. Monfrini, and W. Pieczynski, Unsupervised Segmentation of Random Discrete Data hidden with Switching Noise Distributions, IEEE Signal Processing Letters, Vol. 19, No. 10, pp. 619-622, October 2012.
- [30] H. Li, S. Derrode, and W. Pieczynski, “Adaptive on-line lower limb locomotion activity recognition of healthy individuals using semi-Markov model and single wearable inertial sensor”, Sensors, 19, 4242, 2019.
- [31] S. Le Cam, F. Salzenstein, and C. Collet,” Fuzzy pairwise Markov chain to segment correlated noisy data”, Signal Processing, vol. 88, no. 10, pp. 2526-2541, 2008.
- [32] Y. Lember and J. Sova, “Existence of infinite Viterbi path for pairwise Markov models”, Stochastic Processes and Their Applications, vol. 130, no. 3, pp. 1388-1425, 2020.
- [33] M. Ameer, N. Idrissi, and C. Daoudi, “Color images segmentation using Pairwise Markov Chain”, International Conference on Multimedia Computing and Systems, Rabat, Marocco, May 10-12, 2018.

Bibliographie

- [34] A. K. Atiampo and G. L. Loum, "Unsupervised image segmentation with Pairwise Markov Chains based on nonparametric estimation of copula using orthogonal polynomials", *Journal of Image and Graphics*, vol. 16, no. 4, pp. 2526-2541, 2016.
- [35] S. Derrode et W. Pieczynski, Segmentation non supervisée d'images par chaîne de Markov couple, *Ateliers Traitement et Analyse de l'Information : Méthodes et Applications (TAIMA'03)*, Hammamet, Tunisie, 26 september - 3 octobre, 2003.
- [36] E. Azeraf, E. Monfrini, E. Vignon, and W. Pieczynski, Highly fast text segmentation with pairwise Markov chains, *IEEE International Congress on Information Science and Technology, Machine Learning for Natural Language Processing*, 2020.
- [37] M. Aneur, N. Idrissi, and C. Daoudi, "Triplet Markov chain in images segmentation", *International Conference on Intelligent Systems and Computer Vision (ISCV)*, Fez, Marocco April 02-04, 2018.
- [38] S. Brick, C. Collet, J-P. Armspach, "Triplet Markov chain for 3D MRI brain segmentation using a probabilistic atlas", *3rd IEEE ISBI*, Arlington, United States, April 06-09, 2006.
- [39] Lanchantin, P., Pieczynski, W., Unsupervised non stationary image segmentation using triplet Markov chains. In: *Proc. of Advanced Concepts for Intelligent Vision Systems (ACVIS 04)*, Brussels, Belgium, August 31 September 3, 2004.
- [40] A. Ben Abbes, I. R. Farah, and V. Barra, "A non-stationary NDVI time series modelling using triplet Markov chain", *International Journal of Information and Decision Sciences*, vol. 11, no. 2, pp. 163-179, 2019.
- [41] M. Y. Boudaren, E. Monfrini, W. Pieczynski, and A. Aissani, "Phasic triplet Markov chains", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 36, no. 11, pp. 2310 - 2316, November 2014.
- [42] H. Li, S. Derrode, and W. Pieczynski, "An adaptive and on-line IMU-based locomotion activity classification method using a triplet Markov model", *Neurocomputing*, 362, pp. 94-105, 2019.
- [43] S. Chen and X. Jiang, "Modeling repayment behavior of consumer loan in portfolio across business cycle: a triplet Markov model approach", *Complexity*, Article ID 5458941, 2020.
- [44] C. Carincotte, S. Derrode, and S. Bourenane, "Unsupervised change detection on SAR images using fuzzy hidden Markov chains, *IEEE Trans. on Geosciences and Remote Sensing*, vol. 44, no. 2, pp. 432-441, Feb. 2006.

Bibliographie

- [45] F. Salzenstein, C. Collet, S. Le Cam, and M. Hatt, “Non-stationary fuzzy Markov chain”, *Pattern Recognition*, vol. 28, no. 16, pp. 2201-2208, 2007.
- [46] S. Derrode and W. Pieczynski, Signal and image segmentation using Pairwise Markov chains, *IEEE Trans. on Signal Processing*, Vol. 52, No. 9, pp. 2477-2489, 2004..
- [47] N. Brunel and W. Pieczynski, Unsupervised signal restoration using Copulas and Pairwise Markov chains, *IEEE Workshop on Statistical Signal Processing (SSP 2003)*, Saint Louis, Missouri, September 28-October 1, 2003.
- [48] W. Pieczynski, “EM and ICE in hidden and triplet Markov models”, in *Proceedings of Stochastic Modeling and Data Analysis International Conference*, Chania, 8-11 June, 2010.
- [49] G. Celeux. Bayesian inference for mixtures: the label switching problem. In *Physica-Verlag, editor, COMPSTAT 98*, pages 227–232, 1998.
- [50] Juliette Blanchet, Florence Forbes. Triplet Markov fields for the classification of complex structure data. [Research Report] RR-6356, INRIA. 2007.
- [51] T. Denoeux, 40 years of Dempster-Shafer theory, *International Journal of Approximate Reasoning*, vol. 79, pp. 1-6, 2016 .
- [52] J. Guan and D.A. Bell, “Evidence theory and its applications”, North-Holland, Amsterdam, 1991.
- [53] G. Shafer, *A Mathematical Theory of Evidence*. Princeton, NJ Princeton Univ. Press, 1976.
- [54] P. Smets, “Belief functions: the disjunctive rule of combination and the generalized Bayesian theorem”, *International Journal of Approximate Reasoning*, vol. 9, pp. 1-35, 1993.
- [55] P. Smets and R. Kennes, “The transferable belief model”, *Artificial Intelligence*, vol. 66, pp. 191-234, 1994.
- [56] E. Ramasso and T. Deneoux, “Making use of partial knowledge about hidden states in HMMs: an approach based on belief functions”, *IEEE Trans. on Fuzzy Systems*, vol. 22, no. 2, pp. 1102-1114, 2017.
- [57] E. Ramasso, “Inference and learning in evidential discrete latent Markov models”, *IEEE Trans. on Fuzzy Systems*, vol. 25, no. 5, pp. 395-405, 2014.

Bibliographie

- [58] H. Soubaras, On evidential Markov chains, *Studies in Fuzziness and Soft Computing*, 249, pp. 247-264, 2010.
- [59] H. Soubaras, An evidential measure of risk in evidential markov chains, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 5590 LNAI, pp. 863-874, 2009.
- [60] V. S. Barbu, N. Limnios, "Semi-Markov Chains and hidden semi-Markov models toward applications: Their use in reliability and DNA analysis" Springer Science & Business Media, New York, USA, Vol. 191, 2009.
- [61] S. Z. Yu and H. Kobayashi, "An efficient forward-backward algorithm for an explicit-duration hidden Markov model", *IEEE Signal Processing Letters*, vol. 10, no. 1, pp. 11-14, 2003.
- [62] S. Z. Yu, "Hidden Semi-Markov Models", *Artificial Intelligence*, vol. 174, no. 2, pp. 215-243, February 2010.
- [63] Barbu, S. and Limnios, N., editors (2016). *Semi-Markov Chains and Hidden Semi-Markov Models Toward Applications*. Springer-Verlag, New York, 2016.
- [64] Pieczynski, W., Multisensor triplet Markov chains and theory of evidence. *Int. J. Approx. Reason.* 45 (1), 1-16, 2007.
- [65] S.-Z. You, H. Kobayashi, "A hidden semi-Markov model with missing data and multiple observation sequences for mobility tracking," *Signal Processing*, vol. 83, pp. 235–250, 2003.
- [66] Benouareth, A., Ennaji, A., Sellami, M., 2008. Arabic handwritten word recognition using HMMs with explicit state duration. *EURASIP J. Adv. Signal Process.* 1-13, 2008.
- [67] Lin, D.-T., Liao, Y.-C., On-line handwritten signature verification using hidden semiMarkov model. *Commun. Comput. Inf. Sci.* 173, 584-589, 2011.
- [68] J. Lapuyade-Lahorgue and W. Pieczynski, Unsupervised segmentation of hidden semi-Markov non stationary chains, *Twenty six International Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering, MaxEnt 2006*, Paris, France, 2006.
- [69] J. Lapuyade-Lahorgue and W. Pieczynski, Partially Markov models and unsupervised segmentation of semi-Markov chains hidden with long dependence

Bibliographie

- noise, International Symposium on Applied Stochastic Models and Data Analysis, ASMDA 2007, Crete, Greece, 2007.
- [70] Yurur, O., Liu, C., Perera, C., Chen, M., Liu, X., Moreno, W., Energy-efficient and context-aware smartphone sensor employment. *IEEE Trans. Vehicular Technol*, 2014.
- [71] Joo, R., Bertrand, S., Tam, J., Fablet, R., Hidden Markov models: the best models for forager movements? *PLOS ONE* 8 (8), e71246, 2013.
- [72] Shun-Zheng Yu, Hidden Semi-Markov models theory, algorithms and applications, 2016.
- [73] N. Limnios. Estimation of the Stationary Distribution of Semi-Markov Processes with Borel State Space. *Statistics and Probability Letters*, 76(14) :1536–1542, 2006.
- [74] Donald E. Knuth, *The Art of Computer Programming*, vol. 4A : Combinatorial Algorithms, Part 1, Addison-Wesley, 2011.
- [75] Jérôme Lapuyade-Lahorgue, Sur diverses extensions des chaînes de Markov cachées avec application au traitement des signaux radar, thèse de l'Institut National des Télécommunications et de l'Université Pierre et Marie Curie (Paris 6), 2008.
- [76] A. David et V. Sergei, k-means++: the advantages of careful seeding , *ACM-SIAM symposium on Discrete algorithms*, 2007.
- [77] G. Peano, “Sur une courbe, qui remplit une aire plane,” *Math. Ann.*, Vol. 36, pp. 157- 160, 1890.
- [78] D. Hilbert, “Über die stetige Abbildung einer Linie auf ein Flächenstück,” *Math. Ann.*, Vol. 38, pp. 459-460, 1891.
- [79] W. Pieczynski, Modeling non stationary hidden semi-Markov chains with triplet Markov chains and theory of evidence, *Statistical Signal Processing (SSP2005)*, Bordeaux, France, July 17-20, 2005.
- [80] J. Besag, Spatial interaction and the statistical analysis of lattice systems, *J. R. Stat. Soc. B* 6 192–236, 1974 .
- [81] J. Marroquin, S. Mitter, and T. Poggio, Probabilistic solution of ill-posed problems in computational vision, *Journal of the American Statistical Association*, 82, pp. 76-89, 1987.
- [82] C. S. Won and R. M. Gray, *Stochastic image processing*, Springer Science & Business Media, March 2004.

Bibliographie

- [83] S Bricq, C Collet, JP Armspach, Unifying framework for multimodal brain MRI segmentation based on Hidden Markov Chains, *Medical image analysis*, 12, pp. 639–652, 2008.
- [84] S. Geman and D. Geman, Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 6, No. 6, pp. 721-741, 1984.
- [85] B. Chalmond, An iterative Gibbsian technique for reconstruction of m-ary images, *Pattern Recognition*, 22, pp. 747-761, 1989.
- [86] O. Eches, J.A Benediktsson, N. Dobigeon, J.-Y. Tourneret, Adaptive Markov random fields for joint unmixing and segmentation of hyperspectral images, *IEEE Transactions on Image Processing*, 22 (1), art. no. 6216411, pp. 5-16, 2013.
- [87] P. Chen, J. Nelson, J.-Y. Tourneret, Toward a sparse Bayesian Markov random field approach to hyperspectral unmixing and classification, *IEEE Transactions on Image Processing*, 26 (1), art. no. 7723938, pp. 426-438, 2017.
- [88] L. Younes, Parametric inference for imperfectly observed gibbsian fields, *Probability Theory and Related Fields*, 82, pp. 625-645, 1989.
- [89] O. Cappé, E. Moulines, T. Ryden, *Inference in Hidden Markov Models*, Springer, Series in Statistics, 2005.
- [90] F. Salzenstein, C. Collet, Fuzzy Markov random fields versus chains for multispectral image segmentation, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 28, No. 11, pp. 1753 - 1767, 2006.
- [91] A. Hafiane; B. Zavidovique, and S. Chaudhuri, A modified FCM with optimal Peano Scan for image segmentation, *IEEE International Conference on Image Processing Genova*, pp. III-840, 2005.
- [92] T. Koski, *Hidden Markov Models for Bioinformatics*, Springer, 2011.
- [93] J.-F. Mari and F. Le Ber, Temporal and spatial data mining with second-order hidden Markov models, *Soft Computing*, Vol. 10, 5, pp. 406-414, 2006.
- [94] R. Paroli and L. Spezia, Reversible Jump Markov chain Monte Carlo method and segmentation algorithms in hidden Markov models, *Australian and New Zealand Journal of Statistics*, vol. 52, no. 2, pp. 151-166, 2010.
- [95] J.-N. Provost, C. Collet, P. Pérez, and P. Bouthemy, Unsupervised multispectral segmentation of SPOT images applied to nautical cartography, *Proceedings of the*

Bibliographie

- IEEE-EURASIP Workshop on Nonlinear Signal and Image Processing (NSIP 99), Antalya, Turkey, 1999.
- [96] C. Fernandes, T. Monti, E. Monfrini, and W. Pieczynski, Fast Image Segmentation with Contextual Scan and Markov Chains, 29th European Signal Processing Conference (EUSIPCO), 2021.
- [97] W. Pieczynski and A.-N. Tebbache, Pairwise Markov random fields and segmentation of textured images, *Machine Graphics & Vision*, Vol. 9, No. 3, pp. 705-718, 2000.
- [98] W. Pieczynski, D. Benboudjema, and P. Lanchantin, Statistical image segmentation using Triplet Markov Fields, SPIEs International Symposium on Remote Sensing, September 22-27, Crete, Greece, 2002.
- [99] D. Benboudjema and W. Pieczynski, Unsupervised image segmentation using triplet Markov fields, *Computer Vision and Image Understanding*, Vol. 99, No. 3, pp. 476-498, 2005.
- [100] D. Benboudjema et W. Pieczynski, Segmentation non supervisées dimages non stationnaires avec champs de Markov évidentiels, GRESI 2005, Louvain-La-Neuve, Belgique, 6-9 septembre 2005.
- [101] D. Benboudjema and W. Pieczynski, Segmenting non stationary images with triplet Markov fields, International Conference on Image Processing (ICIP 2005), Genova, Italy, September 11-14, 2005.
- [102] A. Habbouchi, M. Y. Boudaren, A. Aissani, and W. Pieczynski, Unsupervised segmentation of Markov random fields corrupted by nonstationary noise, *IEEE Signal Processing Letters*, Vol. 23, No. 11, pp. 1607-1611, 2016.
- [103] M. Y. Boudaren, L. An, and W. Pieczynski, Unsupervised segmentation of SAR images using Gaussian mixture hidden evidential Markov fields, *IEEE Geoscience and Remote Sensing Letters*, Vol. 13, No. 12, pp. 1865-1869, December 2016.
- [104] P. Zhang, M. Li, Y. Wu, L. Gan, M. Liu, F. Wang, G. Liu, “Unsupervised multi-class segmentation of SAR images using fuzzy triplet Markov fields model,” *Pattern Recognition*, Vol. 45, No. 11, pp. 4018- 4033, 2012.
- [105] P. Zhang, M. Li, Y. Wu, M. Liu, F. Wang, L. Gan , “SAR Image Multiclass Segmentation Using a Multiscale TMF Model in Wavelet Domain,” *IEEE Trans. on Geoscience and Remote Sensing Letters*, Vol. 9 , No. 6, pp. 1099 –1103, 2012.

Bibliographie

- [106] L. An, M. Li, M. E. Y. Boudaren, and W. Pieczynski, Unsupervised segmentation of hidden evidential Markov fields corrupted by correlated non-Gaussian noise, *International Journal of Approximate Reasoning*, Vol. 102, pp. 41-59, November, 2018.
- [107] M. Y. Boudaren, L. An, and W. Pieczynski, Dempster-Shafer fusion of evidential pairwise Markov fields, *International Journal of Approximate Reasoning*, Vol. 74, pp. 13-29, July 2016.
- [108] X. Lian, Y. Wu, W. Zhao, F. Wang, Q. Zhang, M. Li, Unsupervised SAR image segmentation based on conditional triplet Markov fields, *IEEE Geosci. Remote Sens. Lett.* 11 (7) 1185–1189, 2014.
- [109] F. Wang, Y. Wu, Q. Zhang, P. Zhang, M. Li, Y. Lu, “Unsupervised Change Detection on SAR Images Using Triplet Markov Field Model,” *IEEE Geoscience and Remote Sensing Letters*, Vol. 10, No. 4, pp. 697–701, Jul. 2013.
- [110] F. Tupin, I. Bloch, H. Maitre, A first step toward automatic interpretation of SAR images using evidential fusion of several structure detectors. *IEEE Transactions on Geoscience and Remote Sensing*, 37(3), 1327–1343, 1999.
- [111] W. Pieczynski and D. Benboudjema, Multisensor triplet Markov fields and theory of evidence, *Image and Vision Computing*, Vol. 24, No. 1, pp. 61-69, 2006.

Liste des figures

Figure 1. Graphe de probabilité d'une chaîne de Markov cachée pour $N=3$	22
Figure 2. Graphe de probabilité d'une chaîne de Markov couple pour $N=3$	28
Figure 3. Graphe de probabilité d'une chaîne de Markov triplet pour $N=3$	32
Figure 4. Exemple de TMC, $N = 3$, où l'identifiabilité de (\mathbf{X}, \mathbf{U}) ne pose pas de problème dans le cas non supervisé.....	35
Figure 5. Graphe de probabilité d'une chaîne de Markov triplet cachée pour $N=3$	36
Figure 6. Graphe de probabilité d'une chaîne de Markov évidentielle cachée pour $N=3$	44
Figure 7. Segmentation d'une image synthétique à deux classes (Image originale) bruitée par un bruit gaussien $\mathbf{N}(0, 1, \mathbf{N}(0.4, 1)$ (Image bruitée). HMC, HSMC-2, HSMC-5, HSMC-10 et HSMC-20 : segmentation non supervisée avec les modèles en question.	51
Figure 8. Segmentation d'une image synthétique à deux classes (Image originale) bruitée par un bruit gaussien $\mathbf{N}(0, 1, \mathbf{N}(0.3, 1)$ (Image bruitée). HMC, HSMC-2, HSMC-5, HSMC-10 et HSMC-20 : segmentation non supervisée avec les modèles en question.	52
Figure 9 Segmentation d'une image synthétique à deux classes (Image originale) bruitée par un bruit gaussien $\mathbf{N}(0, 1, \mathbf{N}(0.2, 1)$ (Image bruitée). HMC, HSMC-2, HSMC-5, HSMC-10 et HSMC-20 : segmentation non supervisée avec les modèles en question.	52
Figure 10. Graphe de probabilité d'un HESMC-Z pour $N=3$	56
Figure 11 Graphe de probabilité d'une HESMC-V pour $N=3$	59
Figure 12. Variations de δn dans les expériences 1 (Exp. 1), 2 (Exp. 2), 3 (Exp. 3), 4 (Exp.4), 5 (Exp. 5).....	65
Figure 13. Segmentation d'une image a deux classes, bruitée par un bruit gaussien $\mathbf{N}(0, 1, \mathbf{N}(0.3, 1)$. HMC, HSMC, HEMC, HESMC-Z, HESMC-V et NSHSMC : résultat, avec taux d'erreur et amélioration par rapport à HMC entre parenthèses, de la segmentation non supervisée avec le modèle correspondant.	70
Figure 14. Segmentation d'une image a deux classes (Image originale), bruitée par un bruit gaussien $\mathbf{N}(0, 1, \mathbf{N}(2, 1)$ (Image bruitée). HMC, HSMC, HEMC, HESMC-Z, HESMC-V et NSHSMC : résultat, avec taux d'erreur et amélioration par rapport à HMC entre parenthèses, de la segmentation non supervisée avec le modèle en question.....	71

Liste des figures

- Figure 15. Segmentation d'une image a trois classes (Image originale), bruitée par un bruit gaussien $N(0, 1)$, $N(0.3, 1)$, $N(3, 1)$ (Image bruitée). HMC, HSMC, HEMC, HESMC-Z, HESMC-V et NSHSMC : résultat, avec taux d'erreur et amélioration par rapport à HMC entre parenthèses, de la segmentation non supervisée avec le modèle en question 71
- Figure 16. Segmentation d'une image a trois classes (Image originale), bruitée par un bruit non stationnaire $NU(-1, 1)$, 1 , $NU(1, 3)$, 1 , $NU(3, 5)$, 1 (Image bruitée). HMC, HSMC, HEMC, HESMC-Z, HESMC-V et NSHSMC : résultat, avec taux d'erreur et amélioration par rapport à HMC entre parenthèses, de la segmentation non supervisée avec le modèle en question 72
- Figure 17. Construction du balayage de Peano. 78
- Figure 18. Les six configurations spatiales possibles pour les voisins de s (pour s hors bords et coins de l'image) ajoutés par le Peano contextuel. En bleu les voisins de s , aussi voisins dans le balayage de Peano. En vert les voisins de s qui ne le sont plus dans le balayage de Peano, et dont les observations sont associées à l'observation ys sur s 78
- Figure 19. Segmentations de plusieurs images (a.) bruitées par un bruit gaussien indépendant $N(0, 1)$, $N(1, 1)$ (b.), avec HMC-PS de manière supervisée (c.), CMC-CPS de manière supervisée (d.), CMC-CPS de manière non supervisée (e.). τ est le taux d'erreur..... 84
- Figure 20. Segmentation d'une image a deux classes (Image originale), bruitée par un bruit gaussien $N(0, 1)$, $N(1, 1)$ (Image bruitée). HMC-PS : résultat de la segmentation non supervisée avec la HMC-PS (taux d'erreur $\tau = 12.2\%$), CMC-CPS : résultat de la segmentation non supervisée avec la CMC-CPS (taux d'erreur $\tau = 9.1\%$), HMF : résultat de la segmentation non supervisée avec le HMF (taux d'erreur $\tau = 8.9\%$)..... 86
- Figure 21. Segmentation d'une image a deux classes (Image originale), bruitée par un bruit gaussien $N(0, 1)$, $N(1, 1)$ (Image bruitée). HMC-PS : résultat de la segmentation non supervisée avec la HMC-PS (taux d'erreur $\tau = 15.3\%$), CMC-CPS : résultat de la segmentation non supervisée avec la CMC-CPS (taux d'erreur $\tau = 13.9\%$), HMF : résultat de la segmentation non supervisée avec le HMF (taux d'erreur $\tau = 11.6\%$). 86
- Figure 22. Segmentation d'une image a deux classes (Image originale), bruitée par un bruit gaussien $N(0, 1)$, $N(1, 1)$ (Image bruitée). HMC-PS : résultat de la segmentation non supervisée avec la HMC-PS (taux d'erreur $\tau = 9.5\%$), CMC-CPS : résultat de la segmentation non supervisée avec la CMC-CPS (taux d'erreur $\tau = 7.4\%$), HMF : résultat de la segmentation non supervisée avec le HMF (taux d'erreur $\tau = 6.1\%$)..... 87
- Figure 23. Segmentation d'une image a deux classes (Image originale), bruitée par un bruit gaussien $N(0, 1)$, $N(1, 1)$ (Image bruitée). HMC-PS : résultat de la segmentation non supervisée avec la HMC-PS (taux d'erreur $\tau = 20.1\%$), CMC-CPS : résultat de la segmentation non

Liste des figures

supervisée avec la CMC-CPS (taux d'erreur $\tau = 17.8\%$), HMF : résultat de la segmentation non supervisée avec le HMF (taux d'erreur $\tau = 18.0\%$). 87

Figure 24. Segmentation d'une image a deux classes (Image originale), bruitée par un bruit gaussien $N(0, 1)$, $N(1, 1)$ (Image bruitée). HMC-PS : résultat de la segmentation non supervisée avec la HMC-PS (taux d'erreur $\tau = 16.1\%$), CMC-CPS : résultat de la segmentation non supervisée avec la CMC-CPS (taux d'erreur $\tau = 14.1\%$), HMF : résultat de la segmentation non supervisée avec le HMF (taux d'erreur $\tau = 12.0\%$). 88

Figure 25. Segmentation d'une image a deux classes (Image originale), bruitée par un bruit gaussien $N(0, 1)$, $N(1, 1)$ (Image bruitée). HMC-PS : résultat de la segmentation non supervisée avec la HMC-PS (taux d'erreur $\tau = 13.8\%$), CMC-CPS : résultat de la segmentation non supervisée avec la CMC-CPS (taux d'erreur $\tau = 11.7\%$), HMF : résultat de la segmentation non supervisée avec le HMF (taux d'erreur $\tau = 10.5\%$). 88

Figure 26. Segmentation d'une image de fracture en niveau de gris (Image originale). HMC, CMC-CPS, HMF : résultat de la segmentation non supervisée avec le modèle en question... 89

Figure 27. Segmentation d'une image a deux classes (Image originale), bruitée par un bruit gaussien $N(0, 1)$, $N(1, 1)$ (Image bruitée). CMC-CPS : résultat de la segmentation non supervisée avec la CMC-CPS (taux d'erreur $\tau = 14.9\%$), HEMC-PS : résultat de la segmentation non supervisée avec la HEMC-PS (taux d'erreur $\tau = 17.4\%$), CEMC-CPS : résultat de la segmentation non supervisée avec la CEMC-CPS (taux d'erreur $\tau = 11.5\%$), HMF : résultat de la segmentation non supervisée avec le HMF (taux d'erreur $\tau = 8.8\%$).. 98

Figure 28. Segmentation d'une image a deux classes (Image originale), bruitée par un bruit gaussien $N(0, 1)$, $N(1, 1)$ (Image bruitée). CMC-CPS : résultat de la segmentation non supervisée avec la CMC-CPS (taux d'erreur $\tau = 15.5\%$), HEMC-PS: résultat de la segmentation non supervisée avec la HEMC-PS (taux d'erreur $\tau = 16.8\%$), CEMC-CPS : résultat de la segmentation non supervisée avec la CEMC-CPS (taux d'erreur $\tau = 12.0\%$), HMF : résultat de la segmentation non supervisée avec le HMF (taux d'erreur $\tau = 12.5\%$). 98

Figure 29. Segmentation d'une image a deux classes (Image originale), bruitée par un bruit gaussien $N(0, 1)$, $N(2, 1)$ (Image bruitée). CMC-CPS : résultat de la segmentation non supervisée avec la CMC-CPS (taux d'erreur $\tau = 15.8\%$), HEMC-PS : résultat de la segmentation non supervisée avec la HEMC-PS (taux d'erreur $\tau = 5.5\%$), CEMC-CPS : résultat de la segmentation non supervisée avec la CEMC-CPS (taux d'erreur $\tau = 5.4\%$), HMF : résultat de la segmentation non supervisée avec le HMF (taux d'erreur $\tau = 12.8\%$), HEMF : résultat de la segmentation non supervisée avec le HEMF (taux d'erreur $\tau = 6.7\%$). 99

Liste des figures

Figure 30. Segmentation d'une image a deux classes (Image originale), bruitée par un bruit gaussien $N(0,1)$, $N(1,1)$ (Image bruitée). CMC-CPS : résultat de la segmentation non supervisée avec la CMC-CPS (taux d'erreur $\tau = 7.4\%$), HEMC-PS : résultat de la segmentation non supervisée avec la HEMC-PS (taux d'erreur $\tau = 9.3\%$), CEMC-CPS : résultat de la segmentation non supervisée avec la CEMC-CPS (taux d'erreur $\tau = 8.7\%$), HMF : résultat de la segmentation non supervisée avec le HMF (taux d'erreur $\tau = 6.1\%$). 99

Figure 31. Segmentation d'une image a deux classes (Image originale), bruitée par un bruit gaussien $N(0,1)$, $N(1,1)$ (Image bruitée). CMC-CPS : résultat de la segmentation non supervisée avec la CMC-CPS (taux d'erreur $\tau = 17.8\%$), HEMC-PS : résultat de la segmentation non supervisée avec la HEMC-PS (taux d'erreur $\tau = 20.3\%$), CEMC-CPS : résultat de la segmentation non supervisée avec la CEMC-CPS (taux d'erreur $\tau = 18.7\%$), HMF : résultat de la segmentation non supervisée avec le HMF (taux d'erreur $\tau = 18.0\%$). 100

Figure 32. Segmentation d'une image a deux classes (Image originale), bruitée par un bruit gaussien $N(0,1)$, $N(1,1)$ (Image bruitée). CMC-CPS : résultat de la segmentation non supervisée avec la CMC-CPS (taux d'erreur $\tau = 9.1\%$), HSMC-PS : résultat de la segmentation non supervisée avec la HSMC-PS (taux d'erreur $\tau = 12.3\%$), CSMC-CPS : résultat de la segmentation non supervisée avec la CSMC-CPS (taux d'erreur $\tau = 9.2\%$), HMF : résultat de la segmentation non supervisée avec le HMF (taux d'erreur $\tau = 8.9\%$). 103

Figure 33. Segmentation d'une image a deux classes (Image originale), bruitée par un bruit gaussien $N(0,1)$, $N(1,1)$ (Image bruitée). CMC-CPS : résultat de la segmentation non supervisée avec la CMC-CPS (taux d'erreur $\tau = 7.4\%$), HSMC-PS : résultat de la segmentation non supervisée avec la HSMC-PS (taux d'erreur $\tau = 9.1\%$), CSMC-CPS : résultat de la segmentation non supervisée avec la CSMC-CPS (taux d'erreur $\tau = 7.2\%$), HMF : résultat de la segmentation non supervisée avec le HMF (taux d'erreur $\tau = 6.1\%$). 104

Figure 34. Segmentation d'une image a deux classes (Image originale), bruitée par un bruit gaussien $N(0,1)$, $N(1,1)$ (Image bruitée). CMC-CPS : résultat de la segmentation non supervisée avec la CMC-CPS (taux d'erreur $\tau = 17.8\%$), HSMC-PS : résultat de la segmentation non supervisée avec la HSMC-PS (taux d'erreur $\tau = 19.7\%$), CSMC-CPS : résultat de la segmentation non supervisée avec la CSMC-CPS (taux d'erreur $\tau = 16.4\%$), HMF : résultat de la segmentation non supervisée avec le HMF (taux d'erreur $\tau = 18.0\%$). 104

Figure 35. Segmentation d'une image a deux classes (Image originale), bruitée par un bruit gaussien $N(0,1)$, $N(1,1)$ (Image bruitée). CMC-CPS : résultat de la segmentation non supervisée avec la CMC-CPS (taux d'erreur $\tau = 13.3\%$), HSMC-PS : résultat de la segmentation non supervisée avec la HSMC-PS (taux d'erreur $\tau = 17.3\%$), CSMC-CPS :

Liste des figures

résultat de la segmentation non supervisée avec la CSMC-CPS (taux d'erreur $\tau = 12.5\%$),
HMF : résultat de la segmentation non supervisée avec le HMF (taux d'erreur $\tau = 11.9\%$).
..... 105

Figure 36. Segmentation d'une image a deux classes (Image originale), bruitée par un bruit
corrélé $N(0, 2)$, $N(1, 4)$ puis (5.43) (Image bruitée). CMC-CPS : résultat de la segmentation non
supervisée avec la CMC-CPS (taux d'erreur $\tau = 15.6\%$), PMC-PS : résultat de la
segmentation non supervisée avec la PMC-PS (taux d'erreur $\tau = 14.9\%$), CPMC-CPS :
résultat de la segmentation non supervisée avec la CPMC-CPS (taux d'erreur $\tau = 13.9\%$).
..... 109

Figure 37. Segmentation d'une image a deux classes (Image originale), bruitée par un bruit
corrélé $N(0, 2)$, $N(1, 4)$ puis (5.43) (Image bruitée). CMC-CPS : résultat de la segmentation non
supervisée avec la CMC-CPS (taux d'erreur $\tau = 26.2\%$), PMC-PS : résultat de la
segmentation non supervisée avec la PMC-PS (taux d'erreur $\tau = 18.4\%$), CPMC-CPS :
résultat de la segmentation non supervisée avec la CPMC-CPS (taux d'erreur $\tau = 17.7\%$).
..... 110

Figure 38. Segmentation d'une image a deux classes (Image originale), bruitée par un bruit
corrélé $N(0, 2)$, $N(1, 4)$ puis (5.43) (Image bruitée). CMC-CPS : résultat de la segmentation non
supervisée avec la CMC-CPS (taux d'erreur $\tau = 16.4\%$), PMC-PS : résultat de la
segmentation non supervisée avec la PMC-PS (taux d'erreur $\tau = 12.8\%$), CPMC-CPS :
résultat de la segmentation non supervisée avec la CPMC-CPS (taux d'erreur $\tau = 12.0\%$).
..... 110

Figure 39. Segmentation d'une image a deux classes (Image originale), bruitée par un bruit
corrélé $N(0, 2)$, $N(1, 4)$ puis (5.43) (Image bruitée). CMC-CPS : résultat de la segmentation non
supervisée avec la CMC-CPS (taux d'erreur $\tau = 25.0\%$), PMC-PS : résultat de la
segmentation non supervisée avec la PMC-PS (taux d'erreur $\tau = 25.7\%$), CPMC-CPS :
résultat de la segmentation non supervisée avec la CPMC-CPS (taux d'erreur $\tau = 25.7\%$).
..... 111

Figure 40. Segmentation d'une image a deux classes (Image originale), bruitée par un bruit
corrélé $N(0, 2)$, $N(1, 4)$ puis (5.43) (Image bruitée). CMC-CPS : résultat de la segmentation non
supervisée avec la CMC-CPS (taux d'erreur $\tau = 20.3\%$), PMC-PS : résultat de la
segmentation non supervisée avec la PMC-PS (taux d'erreur $\tau = 20.9\%$), CPMC-CPS :
résultat de la segmentation non supervisée avec la CPMC-CPS (taux d'erreur $\tau = 19.5\%$).
..... 111

Figure 41. Segmentation d'une image a deux classes (Image originale), bruitée par un bruit
corrélé $N(0, 2)$, $N(1, 4)$ puis (5.43) (Image bruitée). CMC-CPS : résultat de la segmentation non

supervisée avec la CMC-CPS (taux d'erreur $\tau = 18.9\%$), PMC-PS : résultat de la segmentation non supervisée avec la PMC-PS (taux d'erreur $\tau = 17.0\%$), CPMC-CPS : résultat de la segmentation non supervisée avec la CPMC-CPS (taux d'erreur $\tau = 15.6\%$).
 112

Liste des tableaux

Tableau 1. Taux d'erreur (en %) de la segmentation d'image synthétique à deux classes pour chaque modèle et chaque bruit considérés 51

Tableau 2. Paramètres estimés et ratio d'erreur. M : moyenne, V : Variance, ER : ratio d'erreur, pour la segmentation des cinq signaux (Exp. 1, Exp. 2, Exp. 3, Exp. 4, Exp. 5) par les six modèles (HMC, HEMC, HESMC-Z, HESMC-V, NSHSMC) 64

Tableau 3 Paramètres estimés et ratio d'erreur. M : moyenne, V : Variance, pour la segmentation de l'image présentée à la figure 13 par les six modèles (HMC, HEMC, HESMC-Z, HESMC-V, NSHSMC)..... 68

Tableau 4. Paramètres estimés et ratio d'erreur. M : moyenne, V : Variance, pour la segmentation de l'image présentée à la figure 14 par les six modèles (HMC, HEMC, HESMC-Z, HESMC-V, NSHSMC)..... 69

Tableau 5 Paramètres estimés et ratio d'erreur. M : moyenne, V : Variance, pour la segmentation de l'image présentée à la figure 15 par les six modèles (HMC, HEMC, HESMC-Z, HESMC-V, NSHSMC)..... 69

Tableau 6. Paramètres estimés et ratio d'erreur. M : moyenne, V : Variance, pour la segmentation de l'image présentée à la figure 16 par les six modèles (HMC, HEMC, HESMC-Z, HESMC-V, NSHSMC)..... 70

Tableau 7. Ordre de parcours des pixels d'une image 4×4 d'un balayage de Peano commençant dans le coin haut gauche 77

Tableau 8. Les observations associées aux pixels 1,2, 3, ..., 16 du tableau 7 par le balayage de Peano contextuel, qui sont (y_1, y_4) , (y_2, y_3) , (y_{14}, y_3, y_8) , (y_1, y_4) , (y_5, y_8) , y_6 , (y_7, y_{10}) , (y_8, y_3, y_5) , ..., (y_{16}, y_{13}) 77

Tableau 9. Récapitulatifs des taux d'erreurs de segmentations des modèles de HMC-PS, CMC-CPS et HMF pour les figures 20-25. 85

Tableau 10. Récapitulatifs des taux d'erreurs de segmentations des modèles de HEMC-PS, CMC-CPS, CEMC-CPS et HMF pour les figures 27-31 ainsi que HEMF pour la figure 29.. 97

Liste des tableaux

Tableau 11. Récapitulatifs des taux d'erreurs de segmentations des modèles de HSMC-PS, CMC-CPS, CSMC-CPS et HMF pour les figures 32-35.	103
Tableau 12. Récapitulatifs des taux d'erreurs de segmentations des modèles de PMC-PS, CMC-CPS et CPMC-CPS pour les figures 36-41.	109

Titre : Chaînes de Markov triplets et segmentation non supervisée d'images.

Mots clés : chaînes de Markov triplets, segmentation non supervisée d'images, balayage de Peano contextuel, chaînes de semi-Markov cachées, données non stationnaires, théorie de l'évidence.

Résumé : Les chaînes de Markov cachées (HMC) sont très utilisées pour la segmentation bayésienne non supervisée de données discrètes. Elles sont particulièrement robustes et, malgré leur simplicité, elles sont suffisamment efficaces dans de nombreuses situations. En particulier pour la segmentation d'image, malgré leur nature unidimensionnelle, elles sont capables, grâce à une transformation des images bidimensionnelles en séquences monodimensionnelles avec le balayage de Peano (PS), de produire des résultats satisfaisants. Cependant, dans certains cas, on peut préférer des modèles plus complexes tels que les champs de Markov cachés (HMF) malgré leur plus grande complexité en temps, pour leurs meilleurs résultats. De plus, les modèles de Markov cachés (les chaînes aussi bien que les champs) ont été étendus aux modèles de Markov couples et triplets, qui peuvent être intéressants dans des cas plus complexes. Par exemple, lorsque le temps de séjour n'est pas géométrique, les chaînes de semi-Markov cachées (HSMC) ont tendance à être plus performantes que les HMC, and on peut dire de même pour les chaînes de Markov évidentielles cachées (HEMC) dans le cas de données non-stationnaires. Dans cette thèse, nous proposons dans un premier lieu une nouvelle chaîne de Markov triplet (TMC), qui étend simultanément les HSMC et les HEMC. Basée sur les chaînes de Markov triplets cachées (HTMC), la nouvelle chaîne de semi-Markov évidentielle cachée (HESMC) peut être utilisée de manière non supervisée, les paramètres étant estimés avec l'algorithme Expectation-Maximization (EM). Nous validons l'intérêt d'un tel modèle grâce à des expériences sur des données synthétiques. Nous nous intéressons ensuite au problème de l'unidimensionnalité des HMC avec PS dans le cadre de la segmentation d'image, en construisant le balayage de Peano contextuel (CPS).

Il consiste à associer à chaque indice dans le HMC obtenu à partir du PS, deux observations sur les pixels qui sont voisins du pixel en question dans l'image considérée, mais qui ne sont pas voisins dans la HMC. On obtient donc trois observations pour chaque point du balayage de Peano, ce qui induit une nouvelle chaîne de Markov conditionnelle (CMC) avec une structure plus complexe, mais dont la loi a posteriori est toujours markovienne. Ainsi, nous pouvons appliquer la méthode classique d'estimation des paramètres : l'algorithme Stochastic Expectation-Maximization (SEM), ainsi qu'étudier la segmentation non supervisée obtenue avec l'estimateur du mode des marginales a posteriori (MPM). Les segmentations supervisées et non supervisées par MPM, basées sur la CMC avec CPS, sont comparées aux HMC avec PS et aux HMF à travers des expériences sur des images synthétiques. Elles améliorent de manière significative les premières, et peuvent même être compétitives avec ces derniers. Finalement, nous étendons les CMC-CPS aux chaînes de Markov couples conditionnelles (CPMC) et à deux chaînes de Markov triplets particulières : les chaînes de Markov évidentielles conditionnelles (CEMC) et les chaînes de semi-Markov conditionnelles (CSMC). Pour chacune de ces extensions, nous montrons qu'elles peuvent améliorer de manière notable leur contrepartie non conditionnelle, ainsi que les CMC-CPS, et peuvent même être compétitives avec les HMF. Par ailleurs, elles permettent de mieux utiliser la généralité du triplet markovien dans le cadre de la segmentation d'image, en contournant les problèmes de temps de calcul considérables qui apparaissent lorsque l'on passe des champs de Markov cachés aux triplets.



Title : Triplet Markov chains and unsupervised image segmentation.

Keywords : triplet Markov chain, unsupervised image segmentation, contextual Peano scan, hidden semi-Markov chains, non-stationary data, theory of evidence.

Abstract : Hidden Markov chains (HMC) are widely used in unsupervised Bayesian hidden discrete data restoration. They are very robust and, in spite of their simplicity, they are sufficiently efficient in many situations. In particular for image segmentation, despite their mono-dimensional nature, they are able, through a transformation of the bi-dimensional images into mono-dimensional sequences with Peano scan (PS), to give satisfying results. However, sometimes, more complex models such as hidden Markov fields (HMF) may be preferred in spite of their increased time complexity, for their better results. Moreover, hidden Markov models (the chains as well as the fields) have been extended to pairwise and triplet Markov models, which can be of interest in more complex situations. For example, when sojourn time in hidden states is not geometrical, hidden semi-Markov (HSMC) chains tend to perform better than HMC, and such is also the case for hidden evidential Markov chains (HEMC) when data are non-stationary. In this thesis, we first propose a new triplet Markov chain (TMC), which simultaneously extends HSMC and HEMC. Based on hidden triplet Markov chains (HTMC), the new hidden evidential semi-Markov chain (HESMC) model can be used in unsupervised framework, parameters being estimated with Expectation-Maximization (EM) algorithm. We validate its interest through some experiments on synthetic data. Then we address the problem of mono-dimensionality of the HMC with PS model in image segmentation by introducing the “contextual” Peano scan (CPS).

It consists in associating to each index in the HMC obtained from PS, two observations on pixels which are neighbors of the pixel considered in the image, but are not its neighbors in the HMC. This gives three observations on each point of the Peano scan, which leads to a new conditional Markov chain (CMC) with a more complex structure, but whose *posterior* law is still Markovian. Therefore, we can apply the usual parameter estimation method: Stochastic Expectation-Maximization (SEM), as well as study unsupervised segmentation Marginal Posterior Mode (MPM) so obtained. The CMC with CPS based supervised and unsupervised MPM are compared to the classic scan based HMC-PS and the HMF through experiments on artificial images. They improve notably the former, and can even compete with the latter. Finally, we extend the CMC-CPS to Pairwise Conditional Markov (CPMC) chains and two particular triplet conditional Markov chain: evidential conditional Markov chains (CEMC) and conditional semi-Markov chains (CSMC). For each of these extensions, we show through experiments on artificial images that these models can improve notably their non conditional counterpart, as well as the CMC-CPS, and can even compete with the HMF. Beside they allow the generality of Markovian triplets to better play its part in image segmentation, while avoiding the substantial time complexity of triplet Markov fields.