Le logiciel SpaCEM^3 pour la classification de données complexes

Juliette Blanchet, Florence Forbes, Sophie Chopart, Lamiae Azizi

Equipe Mistis, INRIA Grenoble Rhône-Alpes, Laboratoire Jean Kuntzmann

Résumé Le logiciel SpaCEM^3 (Spatial Clustering with EM and Markov Models) propose une variété d'algorithmes pour la classification, supervisée ou non supervisée, de données uni ou multi-dimensionnelles en interaction, certaines de ces données pouvant être manquantes. Les structures de dépendances prises en compte sont celles pouvant être décrites par un graphe fini quelconque. Elles incluent le cas particulier des grilles régulières utilisées notamment en segmentation d'images. L'approche principale se fonde sur l'utilisation de l'algorithme EM pour une classification *floue* et sur les modèles de champs de Markov pour la modélisation des dépendances. L'estimation est basée sur des développements récents [6, 8, 7] mettant en oeuvre des techniques d'approximations variationnelles de type champ moyen.

Keywords : champs de Markov cachés, modèles de Markov triplets, données manquantes, algorithmes de type EM, champ moyen, sélection de modèles.

1 Introduction

La classification consiste à regrouper des individus en groupes homogènes par rapport aux mesures effectuées sur ces individus. Un individu au sens large peut être un pixel d'une image, un gène, un segment de texte, etc. Les mesures effectuées sur les individus peuvent être de nature variable (réelles, entières, dans l'intervalle [0, 1], etc.), uni- ou multi-dimensionnelles. L'approche probabiliste repose alors sur la donnée d'un modèle pour le couple des observations et des classes, généralement décomposé en un modèle régissant les classes et un modèle (de bruit) régissant la génération des observations lorsque les classes sont connues. Dans la pratique, des hypothèses simplificatrices sont souvent adoptées :

(1) au niveau de la modélisation, on suppose en général que les classes sont indépendantes et que le modèle de bruit se factorise sur les individus (on parle alors de bruit indépendant). Sous ces deux hypothèses, les individus sont alors implicitement supposés indépendants. Enfin, le bruit est supposé être de forme assez simple, gaussien en général, ou au moins unimodal ;

(2) au niveau des cas traités, les observations sont, en général, ou bien de dimension raisonnable, ou bien les composantes de chaque observation sont supposées indépendantes. De plus, les données utilisées sont complètes. Lorsque, pour différentes raisons, certaines observations viennent à manquer, soit ces observations ne sont pas traitées (comme si aucune mesure n'avait été faite sur l'individu correspondant), soit les valeurs manquantes sont remplacées de manière brutale (par des zéros, la moyenne, etc.).

En pratique, il existe beaucoup de cas où ces hypothèses sont mises en défaut et ne donnent pas de résultats satisfaisants. En particulier, les observations effectuées sont souvent dépendantes (les niveaux de gris des pixels d'une image par exemple). De plus, du fait des progrès des appareils de mesure et des capacités de stockage, nombre de données modernes sont en grande dimension. Sans paramétrisation particulière, le bruit doit alors

être supposé indépendant -gaussien en général- afin de limiter le nombre de paramètres à estimer. Or il est avéré qu'une telle hypothèse de bruit indépendant gaussien (ou unimodale en général) est mal adaptée à certains cas réels, par exemple pour la modélisation de textures. Enfin, il est très fréquent que certaines observations soient manquantes (certains pixels d'une image, lorsque des réponses à certaines questions d'un sondage n'ont pas été remplies, etc.). De manière générale, nous entendons dans cet article par **données complexes** des données ne suivant pas le cadre idéal des hypothèses précédemment décrites. Le logiciel SpaCEM^3 (Spatial Clustering with EM and Markov Models) offre la possibilité de mettre en œuvre des méthodes de classification pour de telles données. Le cadre sous-jacent au logiciel est celui d'une modélisation markovienne permettant de tenir compte des dépendances entre les individus. Ces dépendances sont définies à l'aide d'un système de voisinage, ou, de manière équivalente, d'un graphe (ou réseau) d'interactions. Outre les outils classiques de classification à base de mélanges gaussiens indépendants, les fonctionnalités de SpaCEM^3 incluent les points suivants :

- Classification non supervisée d'individus, basée sur une description des dépendances à l'aide d'un graphe non nécessairement régulier et un traitement basé sur les champs de Markov cachés et les modèles de mélanges (voir Section 3.1). En particulier,
  - les données peuvent être de grande dimension et les différentes dimensions être corrélées (voir Section 3.2) ;
  - certaines données peuvent être manquantes (voir Section 4.3).
- Classification supervisée d'individus lorsque le modèle de bruit n'est ni indépendant, ni unimodal. Les phases d'apprentissage et de test sont basées sur la famille de modèles de Markov triplets décrits en Section 3.3.
- Critère de sélection de modèles (BIC, ICL et leurs approximations en champ moyen) permettant de sélectionner *le meilleur* modèle de champ de Markov caché en fonction des données.
- Simulation des différents modèles (champs de Markov, champs de Markov cachés, champs de Markov triplets).

Les applications des modèles mis en oeuvre dans le logiciel sont très nombreuses. Mentionnons notamment des applications à données complexes : l'analyse d'images, médicales ou satellitaires et plus généralement la vision par ordinateur. Par exemple en télédétection, on a affaire à des images hyperspectrales. Chaque pixel correspond à un spectre de plusieurs centaines de longueurs d'onde. Aussi, en reconnaissance de textures le mélange gaussien usuel ne suffit plus. Il peut être remplacé avantageusement par une utilisation adaptée des champs de Markov triplets. Mentionnons également les applications à l'analyse de données non images. Par exemple, les travaux [16, 7] contiennent une application en génomique avec prise en compte de dépendances entre les gènes (réseaux).

2 Caractéristiques techniques

Le logiciel est développé en C++. La dernière version de SpaCEM^3 (spacem3 2.0) propose une interface graphique développée avec la librarie QT. Il est composé de 52 classes (30000 lignes de codes) pour le coeur du programme et de 20000 lignes de codes pour l'interface graphique. Le logiciel est disponible en téléchargement (http ://spacem3.gforge.inria.fr) pour les systèmes d'exploitation Linux (package .deb et .rpm), Windows et MacOs, ainsi qu'une documentation sous forme de tutorial. Le logiciel peut être

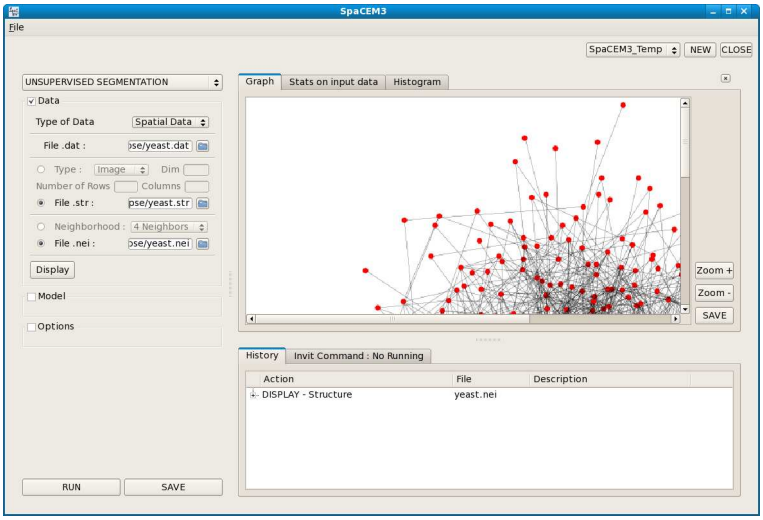


FIGURE 1 – Affichage (zoom) d'un graphe de voisinage quelconque.

utilisé de deux façons : sans interface graphique en lançant directement l'exécutable ; avec interface graphique permettant de visualiser les données, d'effectuer leur classification et de visualiser les résultats.

SpaCEM^3 accepte les données sous forme de fichier .txt ou .dat en texte ou en binaire. Chaque individu est représenté par une ligne et chaque dimension ou chaque variable de cet individu par une colonne. Les dépendances entre les individus sont modélisées sous la forme d'un graphe de voisinage. Deux types de graphes peuvent être utilisés : le type *Image* correspondant aux N plus proches voisins dans une grille régulière, et le type *Structure* pour un graphe non régulier (la liste des voisins est alors à fournir dans une fichier texte, voir Figure 1 pour illustration). Le logiciel SpaCEM^3 peut également construire certains graphes classiques (graphe de Delaunay, graphe de Gabriel, graphe de voisinage relatif, graphe des ϵ-voisins, graphe des k-voisins réciproques ; voir Figure 2).

3 Modèles pour la classification

Nous présentons ici les modèles qui sont plus spécifiquement propres à SpaCEM^3. Il s'agit des modèles de champs de Markov cachés (Section 3.1), des extensions de ce modèle utilisés pour traiter des données de grande dimension (Section 3.2), et d'une famille de champs de Markov triplet pour la classification supervisée (Section 3.3).

3.1 Modèle de champ de Markov caché

3.1.1 Champs de Markov

La définition d'un champ de Markov repose sur celle d'un système de voisinage symétrique, vu comme un graphe $\mathcal{G}$ reliant les individus. On dit que $\mathbf{Z} = \{Z_1, \cdots, Z_n\}$ est un champ de Markov associé à $\mathcal{G}$ si les deux conditions suivantes sont satisfaites :

$$\forall \mathbf{z}, \forall i \in \mathcal{I}, \quad P(Z_i = z_i | \mathbf{Z}_{\mathcal{I}\setminus\{i\}} = \mathbf{z}_{\mathcal{I}\setminus\{i\}}) = P(Z_i = z_i | \mathbf{Z}_{N_i} = \mathbf{z}_{N_i}) \tag{1}$$

$$\forall \mathbf{z}, \quad P(\mathbf{Z} = \mathbf{z}) > 0, \tag{2}$$

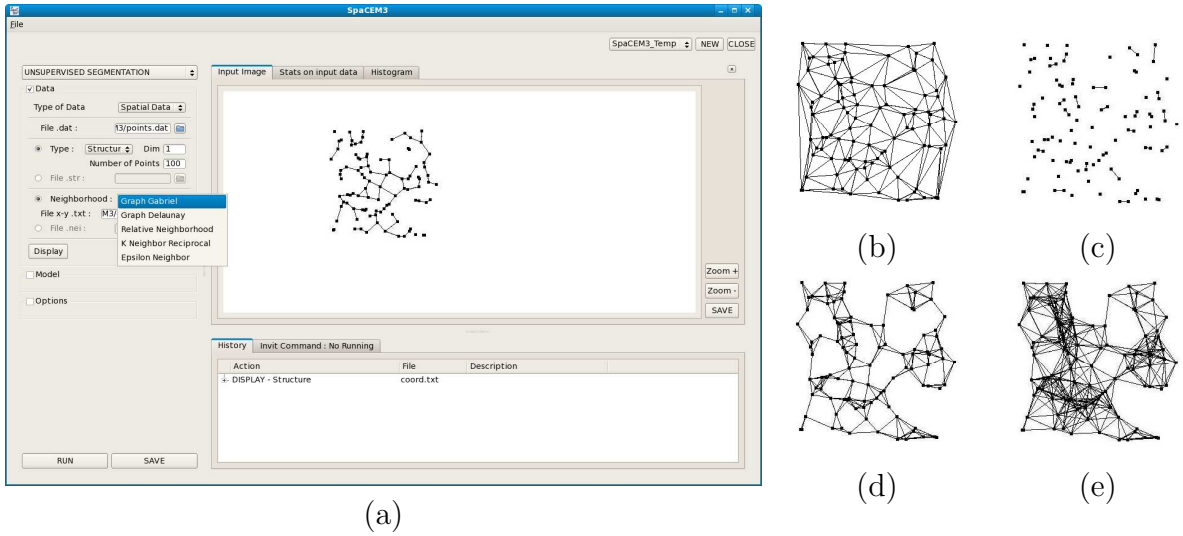


FIGURE 2 – Création des graphes de voisinage par SpaCEM^3 : (a) Graphe de Gabriel, (b) Graphe de Delaunay, (c) Graphe de voisinage relatif, (d) Graphe des ϵ-voisins (ϵ=0.2), (e) Graphe de k-voisins (k=7).

où $\mathcal{I}$ est un ensemble de site (individus) indicés par $i \in \{1 \dots n\}$, $\mathcal{I}\backslash\{i\}$ est l'ensemble des sites $\mathcal{I}$ privé du site i et N_i l'ensemble des sites voisins du site i, ie. reliés à i par une arête dans le graphe $\mathcal{G}$. En pratique, on préfère souvent caractériser un champ de Markov par une distribution jointe : un champ de Markov $\mathbf{Z}$ est de manière équivalente (théorème d'Hammersley-Clifford) une distribution de Gibbs $P_G(\mathbf{z}) = W^{-1}\exp(-H(\mathbf{z}))$ dont la fonction énergie H, définie à une constante additive près, se décompose en une somme de fonction potentiels V_c associé aux cliques c de $\mathcal{G}$, $H(\mathbf{z}) = \sum_{c\in\mathcal{C}} V_c(\mathbf{z}_c)$. W est la constante de normalisation, encore appelée fonction de partition. Il est important de noter qu'une distribution markovienne n'est pas calculable de manière exacte. Différentes approximations ont été proposées dans la littérature, dont l'approximation en champ moyen, sur laquelle nous revenons en Section 4.1.

Modèle de Potts et extensions. Le logiciel SpaCEM^3 traite le cas des champs de Markov discrets en se limitant aux potentiels sur les cliques d'ordre 1 et 2 qui sont la plupart du temps suffisants pour modéliser les dépendances spatiales. Cela correspond à une énergie de la forme :

$$H(\mathbf{z}) = \sum_i (V_i(z_i) + \sum_{j\in N_i} V_{ij}(z_i, z_j)). \tag{3}$$

Un modèle simple est alors le modèle d'Ising, dans lequel les variables Z_i sont binaires. Un modèle plus général correspond à des Z_i pouvant prendre $K > 2$ valeurs. Ces valeurs correspondent aux classes dans les problèmes de classification visés.

Potentiels sur les singletons. Les potentiels sur les singletons (les cliques d'ordre 1) $V_i(z_i)$ permettent de modéliser la probabilité d'occurrence de la classe z_i au site i considéré individuellement. Lorsque les potentiels $V_i(z_i)$ dépendent de i (et non seulement de z_i), on parle de champ externe non stationnaire. De tels potentiels non-stationnaires peuvent

être intéressants pour intégrer de l'information **a priori** visant à influencer les sites individuellement. Dans SpaCEM^3, ces fonctions potentiels sont supposées être les mêmes sur l'ensemble des sites, c'est à dire que $V_i(z_i)$ ne dépend du site i qu'à travers la valeur de z_i. Cette hypothèse correspond à un champ magnétique externe spatialement stationnaire et peut se traduire par la notation : $V_i(z_i) = -\alpha_{z_i}$ Les fonctions potentiels sur les singletons sont alors caractérisées par le vecteur des poids $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_k)$ associés aux K classes. En adoptant la notation vectorielle $\mathbf{z}_i = \mathbf{e}_{z_i}$ où $(\mathbf{e}_1, \ldots, \mathbf{e}_K)$ désigne la base canonique, et en notant $\mathbf{z}_i^{'}$ la transposée du vecteur $\mathbf{z}_i$ on a : $V_i(\mathbf{z}_i) = -\mathbf{z}_i^{'}\boldsymbol{\alpha}$.

Potentiels sur les paires. Les potentiels sur les paires (les cliques d'ordre 2) $V_{ij}(z_i, z_j)$ permettent de modéliser la dépendance entre les classes Z_i et Z_j en des sites i et j voisins. Dans SpaCEM^3, ces fonctions potentiels sont les mêmes sur l'ensemble des sites, ce qui peut se traduire par la notation : $V_{ij}(z_i, z_j) = V(z_i, z_j) = -\beta_{z_i, z_j}$. Les fonctions potentiels sur les paires sont donc caractérisées par la matrice symétrique $\boldsymbol{\beta} = (\beta_{kk'})_{k,k' \in [1,K]}$ associée aux $K \times K$ interactions entre classes. En adoptant la notation vectorielle précédente on a : $V_{ij}(z_i, z_j) = -\mathbf{z}_i^{'}\boldsymbol{\beta}\mathbf{z}_j$. Le terme $\beta_{kk'}$ peut s'interpréter comme le degré de compatibilité entre les classes k et $k^{'}$. Un cas particulier est lorsque la matrice $\boldsymbol{\beta}$ s'écrit $\beta \mathrm{I}_k$ où I_k désigne la matrice unité de dimension $K \times K$. Son énergie est alors donnée par : $H(z) = -\sum_{i \sim j} \mathbf{z}_i^{'}\boldsymbol{\beta}\mathbf{z}_j = -\beta \sum_{i \sim j} 1_{z_i = z_j} = -\beta N(\mathbf{Z})$ où $N(\mathbf{Z})$ désigne le nombre de paires homogènes pour la classification $\mathbf{z}$. C'est le **modèle de Potts**. Le logiciel SpaCEM^3 permet de considérer quatre cas de matrice d'interaction $\boldsymbol{\beta}$: matrice pleine, matrice pleine avec composants diagonaux identiques, matrice diagonale et matrice proportionnelle à l'identité. Il en résulte 8 modèles (on parlera de **modèle de Potts étendu**) possibles, selon ou non qu'on inclut des paramètres de champ externe α.

3.1.2 Champs de Markov cachés

Avant d'introduire le modèle de champ de Markov caché implémenté dans SpaCEM^3, nous précisons la notion de mélange inhérente aux modèles probabilistes de classification que nous considérons. Il s'agit de classer n observations réelles D-dimensionnelles, notées $\mathbf{x} = (x_1, \cdots, x_n)$. Notons $\mathcal{K} = [1, K]$ l'ensemble des classes. Le problème de classification est d'associer à chacune des observations x_i une classe notée $z_i \in \mathcal{K}$ qui peut être vue comme la réalisation d'une variable aléatoire discrète $Z_i \in \mathcal{K}$. En notant $\mathbf{z} = (z_1, \cdots, z_n)$, la distribution des observations est alors donnée par : $P(\mathbf{x}) = \sum_{\mathbf{z}} P(\mathbf{z})P(\mathbf{x}|\mathbf{z})$.

Distribution de mélange. On parlera de *mélange indépendant* si le couple $(\mathbf{X}, \mathbf{Z})$ suit une loi définie par :

$$P(\mathbf{z}) = \prod_{i \in \mathcal{I}} P(z_i) \tag{4}$$

$$\text{et } P(\mathbf{x}|\mathbf{z}) = \prod_{i \in \mathcal{I}} P(x_i|z_i)\ . \tag{5}$$

L'équation (4) indique que les variables cachées $\mathbf{Z}$ sont indépendantes et l'équation (5) est appelée *hypothèse de bruit indépendant*. Sous les hypothèses (4) et (5), les $x_1, \cdots, x_n$ sont alors des réalisations indépendantes et de même loi. Pour retrouver la définition classique

du mélange indépendant, il faut encore supposer que les classes Z_i sont identiquement distribuées, c'est-à-dire que $P(z_i)$ ne dépend pas de i et on peut alors noter π_k la probabilité $P(Z_i = k)$ (avec, pour tout $k \in \mathcal{K}$, $\pi_k \in [0,1]$ et $\sum_{k\in\mathcal{K}} \pi_k = 1$). De même, le distribution de la classe k, $P(.|Z_i = k)$, est supposée ne dépendre que d'un paramètre θ_k ; nous la noterons $f(.|\theta_k)$. Le logiciel SpaCEM^3 propose différents choix concernant la forme des distributions $f(.|\theta_k)$: loi gaussienne à matrice de covariance Σ_k diagonale, loi gaussienne générale (matrice Σ_k pleine), loi gaussienne adaptée à des donnnées de grande dimension (voir Section 3.2), loi de Laplace (pour données aberrantes), loi de Poisson (pour données épidémiologiques). Pour ces différentes lois, les paramètres peuvent être estimés ou fixés. On peut également mélanger ces différentes familles entre elles mais cette option n'est pas disponible dans l'interface.

Distribution de champ de Markov caché. Dans un modèle de champ de Markov caché, la classification non observée $\mathbf{z}$ est supposée être la réalisation d'un champ de Markov $\mathbf{Z}$ (Section 3.1.1). Dans le logiciel SpaCEM^3 le bruit est supposé être indépendant (équation (5)) ; on parle de **champ de Markov caché à bruit indépendant**. On a alors de manière équivalente que le couple $(\mathbf{X}, \mathbf{Z})$ est un champ de Markov d'énergie $H(\mathbf{z}; \boldsymbol{\phi}) - \sum_{i\in\mathcal{I}} \log f(x_i|\theta_{z_i})$ où $H(\mathbf{z}; \boldsymbol{\phi})$ est l'énergie du champ de Markov $\mathbf{Z}$ supposée ici dépendre d'un paramètre $\boldsymbol{\phi}$. Il s'en suit, en appliquant la règle de Bayes, que le champ $\mathbf{Z}|\mathbf{x}$ des classes $\mathbf{Z}$ conditionnellement aux observations $\mathbf{X} = \mathbf{x}$ est également un champ de Markov d'énergie : $H(\mathbf{z}|\mathbf{x}; \boldsymbol{\psi}) = H(\mathbf{z}; \boldsymbol{\phi}) - \sum_{i\in\mathcal{I}} \log f(x_i|\theta_{z_i})$.

En pratique, c'est cette distribution *a posteriori* markovienne qui est utilisée par les méthodes bayésiennes classiques pour estimer les paramètres et classer les individus. Notons néanmoins que l'hypothèse de champ de Markov caché à bruit indépendant est suffisante mais non nécessaire pour que $P(\mathbf{z}|\mathbf{x})$ soit markovienne. Une hypothèse moins forte est de supposer directement que le couple $(\mathbf{X}, \mathbf{Z})$ est markovien (sans que $\mathbf{Z}$ soit nécessairement markovien). On parle alors de champ de Markov couple [13]. Nous reviendrons sur ce modèle, ainsi que sur son extension par champ de Markov triplet dans la Section 3.3.

3.2 Modèle gaussien pour données de grande dimension

Lorsque les données observées sont en grande dimension, de nombreux algorithmes sont limités par la quantité de paramètres à estimer. Pour faire face à ce problème, des modèles parcimonieux peuvent être utilisés, comme l'un des 14 modèles particuliers proposés dans [1] pour le cas gaussien. Néanmoins de tels modèles n'ont pas été spécifiquement élaborés pour les données de grande dimension et, en particulier, ne tiennent pas compte du phénomène de l'espace vide. Une deuxième solution est d'utiliser des méthodes de réduction de dimension (ACP, sélection de variables...). En classification, ces méthodes de réduction de dimension se font au prix d'une perte d'information car, certes, toutes les variables ne sont peut être pas informatives, mais l'ensemble des variables est souvent nécessaire pour discriminer les classes les unes par rapport aux autres.

Pour classer les données de grande dimension, la méthode implémentée dans SpaCEM^3 se base donc sur un modèle différent développé dans [8] et étendu au cas markovien dans [7]. Il s'agit d'une re-paramétrisation du modèle gaussien prenant en compte le fait que les données de chaque classe vivent dans des sous-espaces différents dont les dimensions intrinsèques peuvent varier. Considérons la décomposition spectrale de la matrice de co-

variance de la classe k, $\Sigma_k = Q_k \Delta_k Q_k'$ où Q_k est la matrice orthogonale de taille $D \times D$ des vecteurs propres de Σ_k, Q_k' sa transposée et Δ_k est la matrice diagonale des valeurs propres. [8] propose de modéliser le fait que les données de chacune des classes vivent dans des sous-espaces de dimensions inférieures en écrivant Δ_k sous la forme :

$$\Delta_k = \begin{pmatrix} a_{k1} & & 0 & & \\ & \ddots & & (\mathbf{0}) & \\ 0 & & a_{kD_k} & & \\ & & & b_k & & 0 \\ & (\mathbf{0}) & & & \ddots & \\ & & & 0 & & b_k \end{pmatrix} \begin{matrix} \left.\vphantom{\begin{matrix} a \\ a \\ a \end{matrix}}\right\} D_k \\ \left.\vphantom{\begin{matrix} a \\ a \\ a \end{matrix}}\right\} (D - D_k) \end{matrix}$$

où, pour tout $d = 1, ..., D_k$, $a_{kd} > b_k$, et $D_k < D$. Notons que cela revient à supposer que les $D - D_k$ plus petites valeurs propres sont égales. Il est toujours possible de faire cette hypothèse quitte à prendre $D_k = D - 1$. Néanmoins en pratique, et c'est tout l'intérêt de cette modélisation, on a $D_k << D$. Deux de ces modèles gaussiens de grande dimension sont disponibles dans SpaCEM^3 : le modèle génèral et un modèle plus simple où $a_{k1} = a_{k2} = a_{kD_k}$ (voir illustration Figure 3).

3.3 Modèle de Markov triplet pour la classification supervisée

Dans de nombreux cas pratiques, et notamment en modélisation de textures et plus généralement de classes non unimodales, l'hypothèse largement utilisée de bruit indépendant (équation (5)) est trop restrictive et la relacher est indispensable. De manière générale, le succès des modèles de Markov cachés à bruit indépendant est du au fait que sous une telle modélisation, la distribution des classes conditionnellement aux observations (loi **a posteriori**) est aussi markovienne, ce qui rend possible l'utilisation des méthodes bayésiennes classiques pour estimer les paramètres et classer les individus. Or, la double hypothèse de champ de Markov caché (sous laquelle les classes suivent un champ de Markov) et de bruit indépendant est suffisante mais non nécessaire pour que la distribution **a posteriori** soit markovienne. A partir de cette observation fondamentale, un modèle plus général, le *champs de Markov couple* [13] a été proposé, puis étendu par la suite au champs de Markov triplet [2] permettant de modéliser un bruit plus riche avec des coûts algorithmiques similaires. Les modèles de Markov triplets disponibles dans le logiciel SpaCEM^3 sont ceux développés dans les travaux [6] différents de ceux utilisés dans [2, 3]. Les modèles implémentés ont été initialement construits avec pour objectif la classification supervisée d'individus issus de classes complexes ou soumis à des modèles de bruits non standards (non unimodaux et non indépendants). Le terme supervisé signifie que nous disposons d'individus étiquetés (nous connaissons leurs classes). A partir des observations correspondantes (formant la base d'apprentissage), nous désirons classer d'autres individus (la base de test) dans ces mêmes classes. Pour une telle problématique, le logiciel propose des modèles basés sur les modèles de Markov triplets et adaptés à un cadre supervisé. Les étapes d'apprentissage et de test implémentées sont basées sur l'application de l'algorithme de type NREM proposé dans [9] et détaillé en Section 4.2. Nous renvoyons à [6, 5] pour plus de details.

Modèle triplet. Dans le cadre d'une classification supervisée, on dispose de deux ensembles d'observations, que nous noterons $\mathcal{I}^1$ et $\mathcal{I}^2$. Les observations $\mathbf{x}^1 = (x_i)_{i \in \mathcal{I}^1}$

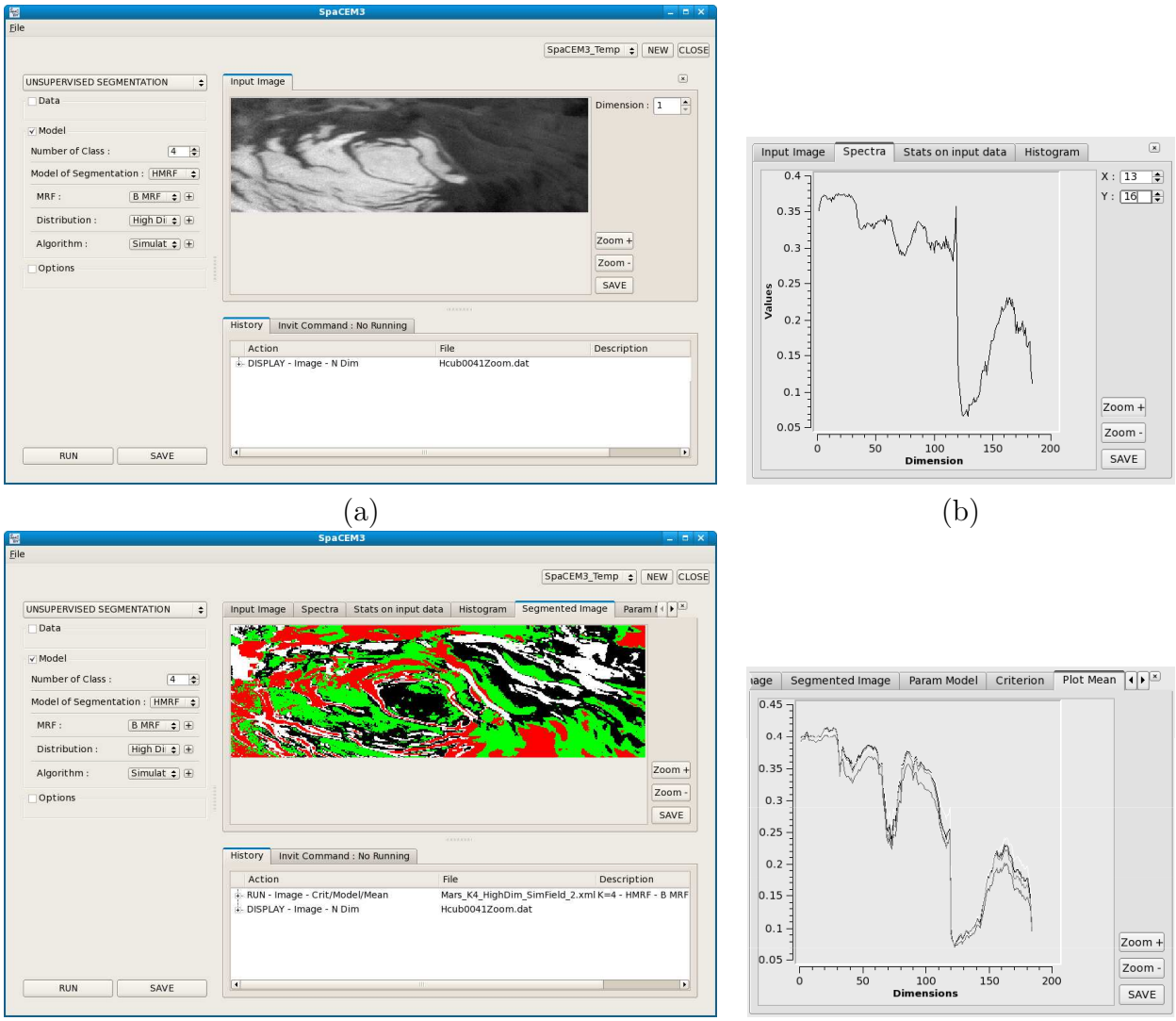


FIGURE 3 – Segmentation d'une image hyperspectrale de Mars avec SpaCEM^3 : (a) Image à segmenter (128*400 pixels de dimension 184), (b) Spectre du pixel (13,16), (c) Image segmentée (4 classes), (d) Spectres moyens des 4 classes.

de $\mathcal{I}^1$ sont étiquetées, nous connaissons donc leurs classes $\mathbf{z}^1 = (z_i)_{i\in\mathcal{I}^1}$. Les données $\mathbf{x}^2 = (x_i)_{i\in\mathcal{I}^2}$ de $\mathcal{I}^2$ sont non-étiquetées. L'objectif est alors, à partir des observations d'apprentissage $\mathbf{x}^1$ et $\mathbf{z}^1$ d'apprendre certains paramètres du modèle, de manière à pouvoir classer dans un second temps les observations de test $\mathbf{x}^2$. On suppose que les données d'apprentissage et de test suivent le même modèle et dans les deux cas, on notera $\mathbf{X}$ les observations et $\mathbf{Z}$ les classes.

Nous nous plaçons dans un cadre où le bruit $P(\mathbf{x}|\mathbf{z})$ n'est ni indépendant, ni assez simple pour être modélisé par une distribution unimodale (gaussienne par exemple). Une idée naturelle pour classer de telles données est alors de décomposer chaque classe $k \in \mathcal{K}$ en sous-classes. Supposons par exemple que chacune des K classes puisse être décomposée en L sous-classes. Pour cela, introduisons un champ auxiliaire $\mathbf{Y} = (Y_i)_{i\in\mathcal{I}}$ discret à valeurs dans $\mathcal{L}^n$ ($\forall i \in \mathcal{I},\ Y_i \in \mathcal{L} = [1, L]$). Les sous-classes de la classe $k \in \mathcal{K}$ sont alors les couples (l, k), $l \in \mathcal{L}$. L'ensemble des LK sous-classes correspond alors à l'ensemble des couples $(l, k) \in \mathcal{L} \times \mathcal{K}$.

Pour tenir compte des dépendances entre sites, à la fois pour l'apprentissage et pour le test, nous considérons un modèle de champ de Markov triplet c'est-à-dire un triplet $(\mathbf{X}, \mathbf{Y}, \mathbf{Z})$ dont la loi jointe est markovienne. Plus précisément, les triplets considérés dans SpaCEM^3 sont de la forme :

$$P_G(\mathbf{x}, \mathbf{y}, \mathbf{z}) \propto \exp(-\sum_{i\sim j} V_{ij}(y_i, z_i, y_j, z_j) + \sum_{i\in\mathcal{I}} \log f(x_i|\theta_{y_i z_i})) \tag{6}$$

où les V_{ij} sont des potentiels sur les paires. De plus, les potentiels V_{ij} sont supposés être les même sur l'ensemble des sites de sorte que nous pouvons écrire sans perte de généralité :

$$V_{ij}(y_i, z_i, y_j, z_j) = -B_{z_i z_j}(y_i, y_j) - C(z_i, z_j)$$

où les $B_{kk'}$ sont K^2 fonctions de $\mathcal{L} \times \mathcal{L} \to$ R et C est une fonction de $\mathcal{K} \times \mathcal{K} \to$ R. En utilisant la notation vectorielle $z_i = k \Leftrightarrow \mathbf{z}_i = \mathbf{e}_k$ (le k-ième vecteur canonique en dimension K) et $y_i = l \Leftrightarrow \mathbf{y}_i = e'_l$ (le l-ième vecteur canonique en dimension L), on peut encore écrire :

$$V_{ij}(\mathbf{y}_i, \mathbf{z}_i, \mathbf{y}_j, \mathbf{z}_j) = -\mathbf{y}'_i \mathrm{B}_{z_i z_j} \mathbf{y}_j - \mathbf{z}'_i \mathrm{C} \mathbf{z}_j \tag{7}$$

où les $\mathrm{B}_{kk'}$ sont K^2 matrices de taille $L \times L$ et C est une matrice de dimension $K \times K$. Remarquons que l'écriture (7) est toujours possible. Cela revient simplement à voir V_{ij} sous la forme d'une matrice V de dimension $LK \times LK$, elle-même décomposée comme $L \times L$ blocs de matrices de dimension $K \times K$. En notant $(c_{kk'})_{k,k'\in\mathcal{K}}$ les coefficients de C, la matrice V a la forme donnée dans la figure **??**. Par symétrie des interactions, V est symétrique, et donc les matrices B_{kk} le sont aussi. Dans SpaCEM^3, C est de plus supposée symétrique, si bien que toutes les matrices $\mathrm{B}_{kk'}$ le sont aussi.

Signalons que le terme $\mathbf{z}'_i \mathrm{C} \mathbf{z}_j$ de l'équation (7) aurait pu être intégré directement dans celui $\mathbf{y}'_i \mathrm{B}_{z_i z_j} \mathbf{y}_j$ mais l'intérêt d'une telle modélisation apparaîtra ultérieurement (étape de classification). La composante (l, l') de la matrice $\mathrm{B}_{kk'}$ s'interprète comme un coefficient de compatibilité entre les sous-classes l et l' des classes k et k'. Plus ce terme est grand, plus il est vraisemblable que deux sites voisins soient dans les sous-classes l et l' des classes k et k'. De même, le coefficient $c_{kk'}$ de la matrice C s'interprète comme un coefficient de compatibilité entre les classes k et k'. Plus ce terme est grand, plus il est vraisemblable que les classes k et k' soient voisines. Lorsque $\mathrm{C} = c\,\mathrm{I}_K$ est paramétré par un unique

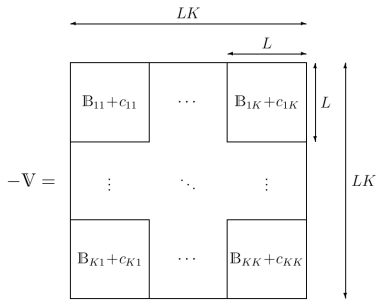


FIGURE 4 – Matrice $-\mathrm{V}$

coefficient spatial $c \in \mathrm{R}^+$, le terme $\sum_{i\sim j} \mathbf{z}_i' \mathrm{C} \mathbf{z}_j = c \sum_{i\sim j} 1_{z_i=z_j}$ agit alors comme un terme de régularisation favorisant les régions homogènes (c'est-à-dire les régions de même classe).

Remarquons que, en notant $\mathbf{U}$ le couple $(\mathbf{Y}, \mathbf{Z})$ (i.e. $\forall i \in [1, LK],\ U_i = (Y_i, Z_i)$), l'équation (6) implique que $\mathbf{U}$ est un champ de Markov et que $(\mathbf{X}, \mathbf{U})$ est un champ de Markov caché à bruit indépendant. Il sera utilisé dans la phase de classification (voir Section 4.2). Notons encore que, sous (6), $P(\mathbf{y}|\mathbf{z})$ est également markovien :

$$P(\mathbf{y}|\mathbf{z}) = \frac{1}{W(\mathbf{z})} \exp(\sum_{i\sim j} \mathbf{y}_i' \mathrm{B}_{z_i z_j} \mathbf{y}_j) \tag{8}$$

où la constante de normalisation $W(\mathbf{z})$ dépend de $\mathbf{z}$. D'après (6), on a encore

$$P_G(\mathbf{x}, \mathbf{y}|\mathbf{z}) = \frac{1}{W(\mathbf{z})} \exp(\sum_{i\sim j} \mathbf{y}_i' \mathrm{B}_{z_i z_j} \mathbf{y}_j + \log f(x_i|\theta_{y_i z_i})), \tag{9}$$

si bien que le couple $(\mathbf{X}, \mathbf{Y})$ est, conditionnellement à $\mathbf{Z} = \mathbf{z}$, un champ de Markov caché à bruit indépendant, sur lequel peuvent donc être appliquées les méthodes d'estimation et classification décrites en Section 4.1. Il sera utilisé dans la phase d'apprentissage (voir Section 4.2).

L'intérêt d'un tel modèle triplet est qu'il est bien adapté au cadre de la classification supervisée. Les traitements bayésiens classiques pourront être appliqués sur $(\mathbf{X}, \mathbf{Y})$ conditionnellement à $\mathbf{Z} = \mathbf{z}$ pour la phase d'apprentissage. La classification d'observations non étiquetées pourra ensuite être obtenue par application de ces mêmes méthodes sur $(\mathbf{X}, \mathbf{U})$. Notons que le modèle ne requiert pas que chaque classe k comporte le même nombre de sous-classe et considérer différents nombre de sous-classes est possible dans SpaCEM^3.

Illustration. Un exemple simple de champ de Markov triplet est donné par

$$P(\mathbf{x}, \mathbf{y}, \mathbf{z}) \propto \exp(b \sum_{i\sim j} 1_{y_i=y_j} 1_{z_i=z_j} + c \sum_{i\sim j} 1_{z_i=z_j} + \sum_{i\in\mathcal{I}} \log f(x_i|\theta_{y_i z_i})) \tag{10}$$

paramétré par les réels b et c, ainsi que les paramètres $\theta_{lk} = (\mu_{lk}, \Sigma_{lk})$ des distributions gaussiennes $f(.|\theta_{lk})$, pour $l \in \mathcal{L}$ et $k \in \mathcal{K}$. Le couple $(\mathbf{Y}, \mathbf{Z})$ est alors markovien, de distribution :

$$P(\mathbf{y}, \mathbf{z}) \propto \exp(b \sum_{i\sim j} 1_{y_i=y_j} 1_{z_i=z_j} + c \sum_{i\sim j} 1_{z_i=z_j}) \tag{11}$$

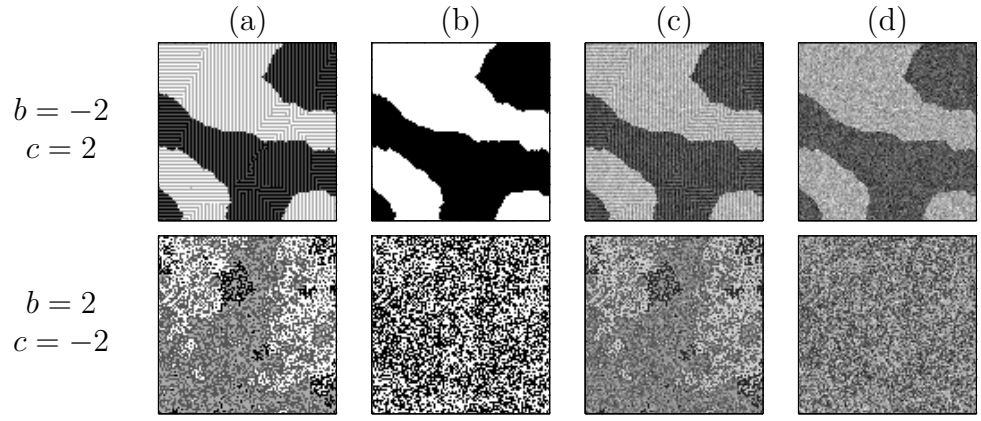


FIGURE 5 – Simulations d'un triplet à deux paramètres (b et c) défini par (10) pour $L = K = 2$ avec respectivement $b = -2$, $c = 2$ (première ligne) et $b = 2$, $c = -2$ (deuxième ligne) : (a) Réalisations de $(\mathbf{Y}, \mathbf{Z})$, (b) réalisations de $\mathbf{Z}$, (c) réalisations de $\mathbf{X}$, (d) réalisations d'un champ de Markov caché à bruit indépendant obtenu en ajoutant aux images (b) un bruit gaussien de moyenne 0 et d'écart-type 0.3.

En comparaison avec l'équation (7) cela revient à supposer que :
- pour tout $k \in \mathcal{K}$, $\mathrm{B}_{kk} = b\mathrm{I}_L$ et pour tout $k' \neq k$, $\mathrm{B}_{kk'} = 0_L$ (la matrice nulle)
- la matrice C est diagonale, ses termes diagonaux sont égaux à c.

Plusieurs cas particuliers sont à souligner :
- Pour $L = 1$, le modèle (11) est un modèle de Potts à K classes et coefficient de régularité égal à $b + c$.
- Pour $K = 1$, il s'agit d'un modèle de Potts à L classes et coefficient de régularité b.
- Pour $c = 0$, il s'agit d'un modèle de Potts à LK classes et coefficient de régularité b.

En Figure 5, on peut voir des simulations de champ de Markov triplet défini par l'équation (10) pour $K = L = 2$ et différentes valeurs de b et c. Chacune des $LK = 4$ valeurs possible du couple (y_i, z_i) est associée à un niveau de gris.

4 Algorithmes de classification

Les algorithmes disponibles dans SpaCEM[3] peuvent être répertoriés dans deux grandes classes :
- Les algorithmes dits usuels qui sont : ICM, Kmeans et l'algorithme EM et ses différentes versions (CEM,NEM et NCEM).
- Les algorithmes basés sur une approche variationnelle de l'algorithme EM sous modélisation markovienne : l'algorithme NREM (Section 4.1) et ses variantes pour modèles triplets (Section 4.2) et avec données manquantes (Section 4.3).

4.1 Algorithme NREM

Dans le modèle de champ de Markov caché à bruit indépendant, $\mathbf{Z}$ et $\mathbf{Z}|\mathbf{x}$ sont tous deux markoviens. Leurs distributions respectives $P_G(\mathbf{z}|\boldsymbol{\phi})$ et $P_G(\mathbf{z}|\mathbf{x}, \boldsymbol{\psi})$ ne peuvent donc être calculées de manière exacte et l'algorithme EM [10] appliqué au champ de Markov

caché $(\mathbf{X}, \mathbf{Z})$ n'est pas réalisable exactement. Parmi les nombreuses solutions proposées pour rendre les étapes (E) et (M) réalisables, SpaCEM^3 met en oeuvre l'algorithme NREM [9] fondé sur une approximation de type champ moyen. Il s'agit de remplacer le modèle de champ de Markov caché de loi $P_G(\mathbf{x}, \mathbf{z}|\boldsymbol{\psi})$ par une approximation de type champ moyen définie par l'équation :

$$P_G(\mathbf{x}, \mathbf{z}|\boldsymbol{\psi}) \approx \prod_{i \in \mathcal{I}} P_{\tilde{\mathbf{z}}^{\mathbf{x}}}(x_i, z_i|\boldsymbol{\psi}) = \prod_{i \in \mathcal{I}} P_G(z_i|\tilde{\mathbf{z}}^{\mathbf{x}}_{N_i}, \boldsymbol{\phi}) f(x_i|\theta_{z_i}) \tag{12}$$

où $\tilde{\mathbf{z}}^{\mathbf{x}}$ est le *champ des voisins*, déterminé conditionnellement aux observations $\mathbf{x}$. Le principe de l'algorithme NREM consiste alors à alterner une étape (NR) de choix du voisinage (*neighborhood restoration*), puis une étape (EM) d'estimation des paramètres du modèle par application de l'algorithme EM sur le mélange indépendant défini par l'approximation. Partant de valeurs initiales $\tilde{\mathbf{z}}^{\mathbf{x}}$ du champ des voisins et $\boldsymbol{\psi}^{(0)}$ des paramètres, l'itération $(q+1)$ de l'algorithme est la suivante :

(EM) **Estimation :** mettre à jour les estimateurs $\boldsymbol{\psi}^{(q+1)}$ des paramètres en appliquant l'algorithme EM sur le modèle de mélange indépendant défini par la loi jointe

$$P_{\tilde{\mathbf{z}}^{\mathbf{x}}}(\mathbf{x}, \mathbf{z}|\boldsymbol{\psi}) = \prod_{i \in \mathcal{I}} \tilde{\pi}_{iz_i} f(x_i|\theta_{z_i}) \qquad \text{où } \tilde{\pi}_{iz_i} = P_G(z_i|\tilde{\mathbf{z}}^{\mathbf{x}}_{N_i}, \boldsymbol{\phi}) \tag{13}$$

(NR) **Choix des voisins :** créer, à partir des observations $\mathbf{x}$ et de l'estimation courante $\boldsymbol{\psi}^{(q+1)}$ des paramétres, un nouveau champ des voisins $\tilde{\mathbf{z}}^{\mathbf{x}}$.

En pratique, l'étape M, dans le cas de densités gaussiennes $f(.|\theta_k) \sim \mathcal{N}(\mu_k, \Sigma_k)$, conduit à une mise à jour explicite des paramètres μ_k et Σ_k. En revanche, même dans le cas simple du modèle de Potts, il n'y a pas de formule explicite pour la mise à jour des paramètres markoviens $\boldsymbol{\phi}$. Néanmoins, dans le cas du modèle de Potts étendu, ces paramètres peuvent être obtenus par une descente de gradient.

Pour l'étape (NR), trois choix sont disponibles dans SpaCEM^3, pour la mise à jour du champ des voisins, conduisant aux algorithmes en champ moyen, en champ modal et en champ simulé :

- *Algorithme en champ moyen* : fixe $\tilde{\mathbf{z}}^{\mathbf{x}}$ à l'estimation en champ moyen de l'espérance de la distribution conditionnelle $P_G(\mathbf{z}|\mathbf{x}, \boldsymbol{\psi}^{(q+1)})$
- *Algorithme en champ modal* : fixe $\tilde{\mathbf{z}}^{\mathbf{x}}$ à l'estimation en champ modal du mode de la distribution conditionnelle $P_G(\mathbf{z}|\mathbf{x}, \boldsymbol{\psi}^{(q+1)})$
- *Algorithme en champ simulé* : simule $\tilde{\mathbf{z}}^{\mathbf{x}}$ selon la loi conditionnelle $P_G(\mathbf{z}|\mathbf{x}, \boldsymbol{\psi}^{(q+1)})$, via l'échantillonneur de Gibbs.

Classer les données suite à l'algorithme NREM. L'algorithme NREM permet donc d'estimer les paramètres d'un champ de Markov caché sous approximation de type champ moyen. Dans un second temps, comme dans le cas de l'algorithme EM pour mélange indépendant, la classification par MAP ou MPM peut-être restaurée sans calcul supplémentaire : en chaque site i, on choisit la classe la plus probable connaissant l'observation x_i :

$$\forall i \in \mathcal{I},\ z_i^{map} = \arg\max_{z_i \in \mathcal{K}} P_{\tilde{\mathbf{z}}^{\mathbf{x}}}(z_i|x_i) = \arg\max_{z_i \in \mathcal{K}} \tilde{\pi}_{iz_i} f(x_i|\theta_{z_i}). \tag{14}$$

4.2 Schéma de classification pour les modèles triplets

Plus qu'un algorithme, nous décrivons un schéma général pour traiter des données issues de classes complexes (bruit non indépendant, distributions des classes non unimodales) sous modélisation par champ de markov triplet. L'estimation des paramètres est basée sur l'algorithme NREM. La classification supervisée est effectuée en deux étapes, l'une d'apprentissage, l'autre de classification.

1) Etape d'apprentissage. Nous supposons que, pour un certain nombre de sites $i \in \mathcal{I}^1$, nous observons à la fois x_i et sa classe z_i. Avec le modèle introduit en Section 3.3, seul y_i est donc manquant. Puisque, conditionnellement à $\mathbf{Z} = \mathbf{z}$, $(\mathbf{X}, \mathbf{Y})$ est un champ de Markov caché à bruit indépendant, nous pouvons appliquer NREM pour estimer les paramètres du modèle conditionnellement aux classes $\mathbf{z}$, à savoir les matrices $\mathrm{B}_{kk'}$, $k, k' \in \mathcal{K}$ et les θ_{lk}, $l \in \mathcal{L}$ et $k \in \mathcal{K}$. Signalons néanmoins qu'il n'est pas toujours possible d'apprendre toutes les matrices $\mathrm{B}_{kk'}$. En particulier, lorsque la structure de voisinage est telle qu'il n'y a pas de voisins dans les classes k et k', les termes en $\mathrm{B}_{kk'}$ n'apparaîtront pas dans les formules du modèle et cette matrice ne pourra être estimée. C'est par exemple le cas avec les images uni-textures d'apprentissage de l'application détaillée dans [6, 5]). D'après l'équation (8), l'apprentissage de la classe k revient alors simplement à estimer un modèle de Potts étendu à L classes de matrice de compatibilité B_{kk}.

2) Etape de classification. Lors de cette phase, les observations $\mathbf{x}^2$ aux sites $i \in \mathcal{I}^2$ sont non étiquetées, si bien que les champs $\mathbf{Y}$ et $\mathbf{Z}$ sont manquants. Avec $\mathbf{U} = (\mathbf{Y}, \mathbf{Z})$, le couple $(\mathbf{X}, \mathbf{U})$ est un champ de Markov caché à bruit indépendant sur lequel on peut appliquer les traitements bayésiens. Les paramètres du modèle (équation (6)) sont alors les K^2 matrices $\mathrm{B}_{kk'}$ de dimension $L \times L$, les LK paramètres θ_{lk}, ainsi que la matrice additionnelle C de dimension $K \times K$. Lors de cette étape, les θ_{lk} doivent être fixés aux valeurs estimées lors de la phase d'apprentissage, et non réestimés, afin d'éviter un problème de *label-switching*. Concernant les $\mathrm{B}_{kk'}$, plusieurs stratégies sont envisageables selon la base d'apprentissage et du type d'interaction que l'on souhaite considérer : les fixer entièrement aux valeurs estimées, en partie, ou les ré-estimer totalement. La matrice C, elle, devra être estimée pour résoudre un problème d'identifiabilité dû au fait que les matrices $\mathrm{B}_{kk'}$ ne peuvent être estimées qu'à un coefficient additif près lors de l'apprentissage (voir équation (8)).

Illustration. La Figure 6 illustre les performances et la flexibilité des modèles de Markov triplets par rapport aux modèles de Markov cachés classiques sur une image synthétique à deux classes, bruitée selon différents bruits complexes (bruit non unimodal et/ou non indépendant). Une application à un problème de reconnaissance de textures a également été effectuée dans [6, 5].

4.3 Schéma de classification avec données manquantes

Il est très courant en pratique de ne pas disposer de toutes les mesures pour tous les individus. Par exemple, l'appareil de mesure peut être défaillant et les pixels les plus brillants de l'image ne pas être mesurés. Ou bien une expérience biologique peut avoir échoué sur certains gènes (parce qu'ils n'ont pas ou mal réagit). La méthode la plus simple

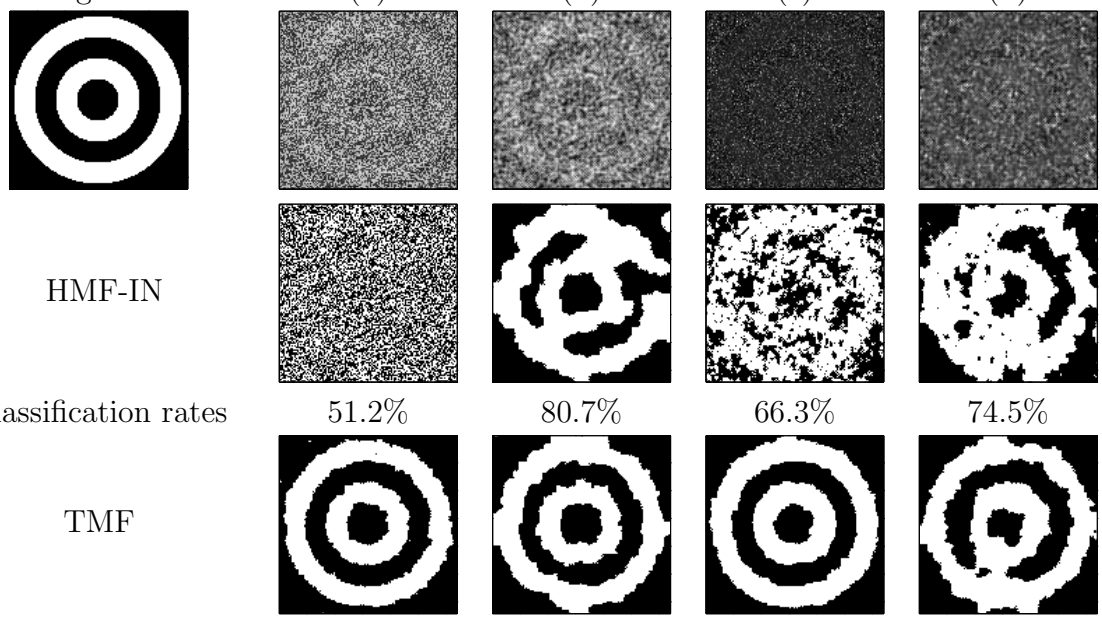


FIGURE 6 – Segmentation d'une image synthétique à l'aide d'un modèle de champ de markov caché (HMF-IN, deuxième ligne) et d'un champ de Markov triplet (TMF, troisième ligne) : vraie segmentation en 2 classes dans le coin supérieur gauche et 4 modèles de bruit différents sont considérés. Image (a) les distributions de chaque classe sont des mélanges de deux gaussiennes, en (c) les observations issues de la classe 1 sont générées à partir d'une loi Gamma(1,2) et celles de la classe 2 sont obtenues en ajoutant 1 aux réalisations d'une loi exponentielle de paramètre 1. En (b) et (d) les images bruitées sont obtenues en remplacant chaque pixel de (a) et (c) par sa moyenne avec ses 4 plus proches voisins. Les taux de classification sont donnés sous chaque segmentation. Pour le modèle triplet, la valeur de K sélectionnée à l'aide de BIC^w (voir Section 5) est donnée dans la dernière ligne.

pour faire face à un tel problème est d'éliminer brutalement les individus pour lesquels certaines observations sont manquantes. On comprend aisément qu'une telle technique soit à éviter. Une deuxième technique très populaire est de remplacer les données manquantes par des valeurs (des zéros, la moyenne, etc...) en pré-traitement (on parle encore d'imputation), puis d'effectuer la classification à partir des observations ainsi complétées. Cette technique, si elle est très couramment utilisée du fait de sa simplicité, tend à introduire un biais. De plus, concernant le choix des valeurs imputées, aucune solution universelle n'existe et des choix différents peuvent conduire à des résultats très différents. Des méthodes plus perfectionnées ont été proposées dans le cadre du modèle de mélange indépendant, notamment d'appliquer l'algorithme EM pour estimer un tel modèle (voir [14] par exemple). Dans le logiciel, les méthodes mises en oeuvre sont celles developpées dans [5, 7] pour classer de telles observations non complètes sous modélisation markovienne.

L'algorithme développé dans le logiciel SpaCEM^3 se place dans le cas où les données sont absentes aléatoirement (*Missing At Random*, MAR, voir [12]). Le fait qu'une donnée soit manquante est alors indépendante de la valeur non observée de cette donnée. Dans les application réelles, l'hypothèse MAR peut être fausse, par exemple lorsque les données sont censurées du fait des limitations de la machine de mesure. On parle alors de données manquantes non aléatoirement (*Not Missing At Random*, NMAR, voir [12]). Les méthodes basées sur l'hypothèse MAR donnent néanmoins des résultats satisfaisants dans le cas de données NMAR si les valeurs observées contiennent assez d'information pour estimer le maximum de vraisemblance.

Comme précédemment, on note $\mathcal{I}$ un ensemble de sites (individus) indicés par $i \in \{1, \cdots, n\}$ et $\mathbf{x} = \{\mathbf{x}_i \in \mathbb{R}^D\}$ la matrice $n \times D$ des observations, dont certaines valeurs sont manquantes. On note encore, pour chaque site $i \in \mathcal{I}$, $o_i \subset \{1, \ldots, D\}$ les indices correspondant aux valeurs observées x_{id} and m_i son complémentaire correspondant aux valeurs manquantes ($o_i \cup m_i = \{1, \ldots, D\}$). Le vecteur des valeurs observées au site i s'écrit alors $\mathbf{x}_i^{o_i} = \{x_{id}, d \in o_i\}$ et le vecteur des valeurs manquantes au site i, $\mathbf{x}_i^{m_i} = \{x_{id}, d \in m_i\}$. Enfin, $\mathbf{x}^o = \{\mathbf{x}_i^{o_i}, i \in \mathcal{I}\}$ dénote l'ensemble des valeurs observées sur les n individus et $\mathbf{x}^m = \{\mathbf{x}_i^{m_i}, i \in \mathcal{I}\}$ l'ensemble des valeurs manquantes.

Le but de la classification est, comme précédemment, d'assigner une valeur $z_i \in \mathcal{K} = [1, K]$ à chaque site $i \in \mathcal{I}$. On se place dans le cas d'un champs de Markov caché à bruit indépendant. Comme en Section 3.1.1, le champs caché $\mathbf{Z}$ est donc markovien et les données $\mathbf{X}$ sont indépendantes conditionnellement aux classes. Néanmoins, à la différence de la Section 3.1.1, cette distribution conditionnelle s'écrit :

$$P(\mathbf{x}^o, \mathbf{x}^m | \mathbf{z}) = \prod_{i \in \mathcal{I}} P(x_i^{o_i}, x_i^{m_i} | z_i).$$

Il y a alors 2 champs cachés à considérer : le champ markovien des classes $\mathbf{Z}$, ainsi que le champ des données manquantes $\mathbf{X}^m$.

L'algorithme de classification développé dans le logiciel SpaCEM^3 est basé sur l'algorithme NREM. La différence avec le cas de données complètes $\mathbf{x}$ de la Section 4.1 réside dans le fait que seules $\mathbf{x}^o$ est observé, et non $\mathbf{x}$ tout entier. Partant de valeurs initiales $\tilde{\mathbf{z}}^{\mathbf{x}^o}$ pour le champ des voisins (déterminé contionnellement aux valeurs observées $\mathbf{x^o}$) et $\boldsymbol{\psi}^{(0)}$ des paramètres, l'itération $(q+1)$ de l'algorithme est la suivante :

(EM) **Estimation :** Mettre à jour les estimateurs $\boldsymbol{\psi}^{(q+1)}$ des paramètres en appliquant

l'algorithme EM sur le modèle de mélange indépendant défini par la loi jointe

$$P_{\tilde{\mathbf{z}}^{\mathbf{x}^o}}(\mathbf{x^o}, \mathbf{x^m}, \mathbf{z}|\boldsymbol{\psi}) = \prod_{i \in \mathcal{I}} \tilde{\pi}^o_{iz_i} f(x_i^{o_i}, x_i^{m_i}|\theta_{z_i}) \qquad \text{où } \tilde{\pi}^o_{iz_i} = P_G(z_i|\tilde{\mathbf{z}}^{\mathbf{x}^o}_{N_i}, \boldsymbol{\phi})$$

(NR) **Choix des voisins :** créer, à partir des valeurs observés $\mathbf{x^o}$ et de l'estimation courante $\boldsymbol{\psi}^{(q+1)}$ des paramètres, un nouveau champ des voisins $\tilde{\mathbf{z}}^{\mathbf{x}^o}$.

L'application de EM sur un mélange indépendant avec observations manquantes est décrit dans [12]. Dans le cas où $f(.|\theta_k)$ est gaussien $\mathcal{N}(\mu_k, \Sigma_k)$, les estimateurs de μ_k et Σ_k sont explicites. Ils diffèrent naturellement des estimateurs obtenus avec données complètes. Pour plus de détails sur l'algorithme NREM avec données incomplètes, on pourra se référer à [5, 7].

Classer les donnés et estimer les observations manquantes suite à l'algorithme NREM. La classification par MAP ou MPM revient à classer un site i dans la classe la plus probable connaissant les valeurs observées $\mathbf{x}_i^{o_i}$ pour ce site :

$$\forall i \in \mathcal{I},\ \hat{z}_i = \arg\max_{z_i \in \mathcal{K}} P_{\tilde{\mathbf{z}}^{\mathbf{x}^o}}(z_i|x_i^{o_i}) = \arg\max_{z_i \in \mathcal{K}} \tilde{\pi}^o_{iz_i} f(x_i^{o_i}|\theta_{z_i}). \tag{15}$$

Notons que, contrairement à l'équation (14), seul $\mathbf{x^o}$ intervient dans (15). Une fois la classification $\mathbf{z}$ obtenue, les valeurs manquantes $\mathbf{x^m}$ peuvent également être reconstruites : la règle du MAP ou MPM revient à imputer les observations manquantes $\mathbf{x}_i^{m_i}$ au site i par les valeurs les plus vraisemblables conditionnellement aux valeurs observées $x_i^{o_i}$ et à la classe $\hat{z}_i$:

$$\forall i \in \mathcal{I},\ \hat{x}_i^{m_i} = \arg\max_{x_i^{m_i}} P(x_i^{m_i}|x_i^{o_i}, \hat{z}_i). \tag{16}$$

L'équation (16) peut être vue comme un imputation par la moyenne. Elle diffère néanmoins de la classique imputation par la moyenne effectuée en pré-traitement dans la mesure où elle est effectuée en post-traitement et dépend donc de la classe à laquelle appartient le site i, ainsi que de ses voisins. On peut également montrer que c'est une estimation non biaisée, contrairement à l'imputation effectuée en pré-traitement (voir [5, 7]).

Illustration. A titre d'exemple, se trouvent en Figure 7 les classifications obtenues sur une image synthétique à 4 classes bruitée par un bruit blanc gaussien en dimension $D = 4$, puis censurées à droite (les r% des valeurs les plus élevées sont manquantes, avec $r \in \{30, 50, 60, 70\}$ pour la Figure 7). Les résultats de classification sont très satisfaisants jusque 60% de données manquantes, et ce bien que les données manquantes soient de type NMAR (du fait de la censure), et non MAR comme supposé dans le modèle. Nous renvoyons à [5, 7] pour d'autres expériences, notamment pour une application à la classification de données d'expression génomique.

4.4 Utilisation des algorithmes en pratique

Pour l'utilisation pratique, deux points sont importants : le choix de l'initialisation et le choix du critère d'arrêt.

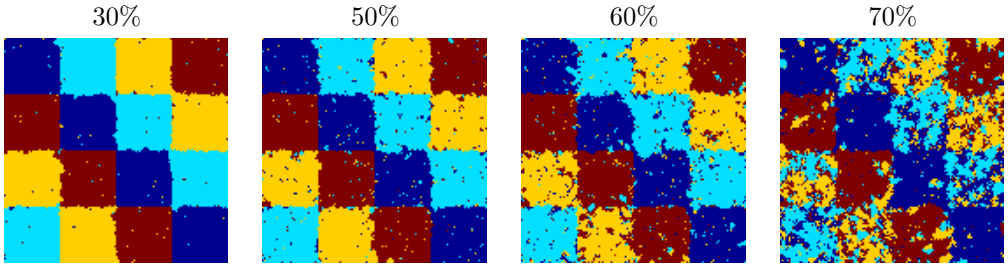


FIGURE 7 – Experience sur image simulée : visualisation des classifications obtenues pour différents pourcentages of données manquantes (30%, 50%, 60% et 70% de données censurées à droite).

Techniques d'initialisation. Dans SpaCEM^3, trois techniques d'initialisation de la classification sont proposées : une initialisation aléatoire, une initialisation par KMeans, ou encore une initialisation fixée par l'utilisateur via un fichier texte.

Critères d'arrêt. SpaCEM^3 peut calculer trois critères permettant de s'assurer de la bonne convergence des algorithmes de segmentation :

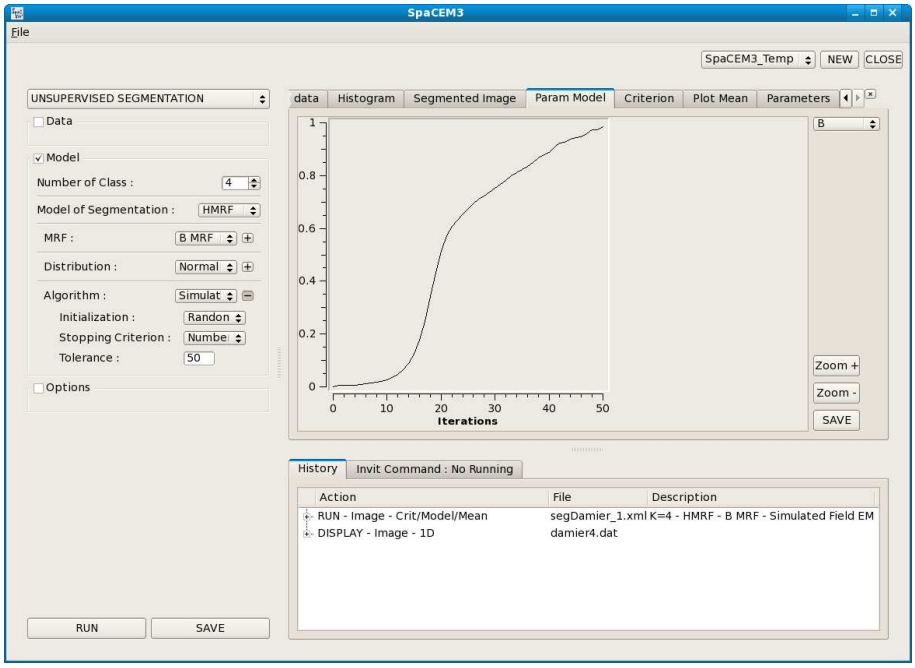
- un critère basé sur la différence des vraisemblances complétées entre deux itérations.
- un critère basé sur la plus grande différence entre les probabilités conditionnelles de classification de chaque individu, entre deux itérations successives.
- un critère basé sur la proportion d'individus pour lesquels la classification a changée entre deux itérations.

Pour arêter les algorithmes, on peut également fixer un nombre d'itérations à effectuer. SpaCEM^3 permet en outre de visualiser l'évolution de ces critères et le comportement des paramètres au cours des itérations (Figure 8).

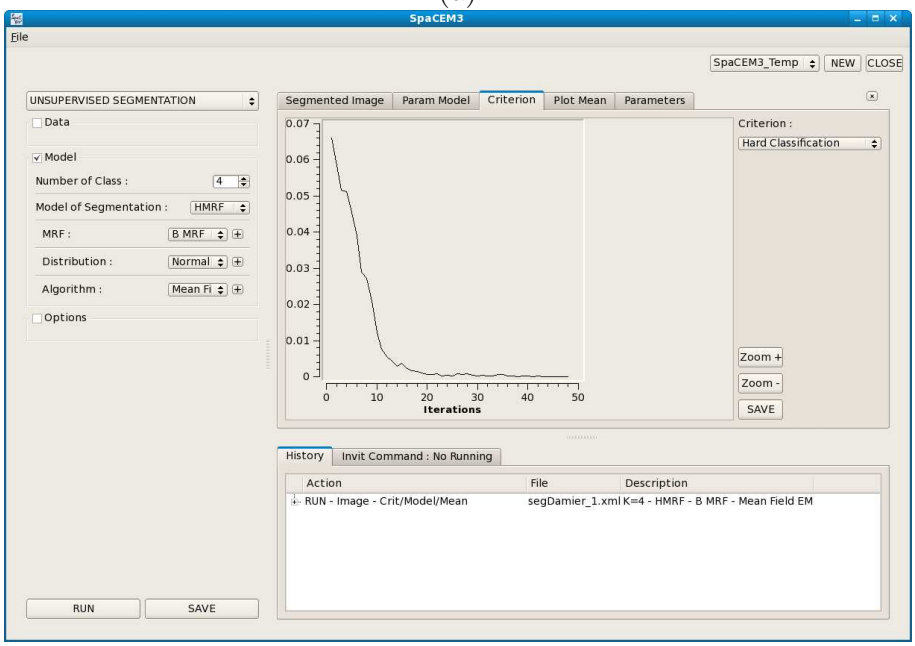
5 Sélection de modèles

Nous avons présenté un ensemble de modèles pour la classification de données. Se pose alors le problème de savoir quel modèle choisir pour modéliser et classer au mieux un jeu de données spécifique. Le "meilleur" modèle choisi devra présenter un bon compromis entre complexité et adéquation aux données. De nombreux critères ont été proposés pour choisir entre différents modèles dans un cadre non-supervisé. Les critères disponibles dans SpaCEM^3 sont Le *Bayesian Information Criterion* (BIC) [15] qui est certainement le plus répandu et le critère *Integrated Completed Likelihood* (ICL) [4] qui permet de tenir compte de la pertinence de la classification obtenue.

Cas d'une distribution markovienne. Lorsque le modèle est celui d'un champ de Markov caché, le critère BIC ne peut être calculé sans approximation. Deux approximations du critère BIC définis dans [11] sont disponibles dans le logiciel, l'une (BIC^p) utilise l'approximation en champ moyen de la distribution de Gibbs P_G, l'autre (BIC^w) l'approximation en champ moyen de la fonction de partition W. Les auteurs de [11] remarquent qu'en théorie comme en pratique, l'approximation BIC^w est plus fine que



(a)



(b)

FIGURE 8 – Visualisation des sorties dans SpaCEM^3 : (a) valeur du paramètre β au cours des itérations, (b) valeur du critère de proportion de changements au cours des itérations.

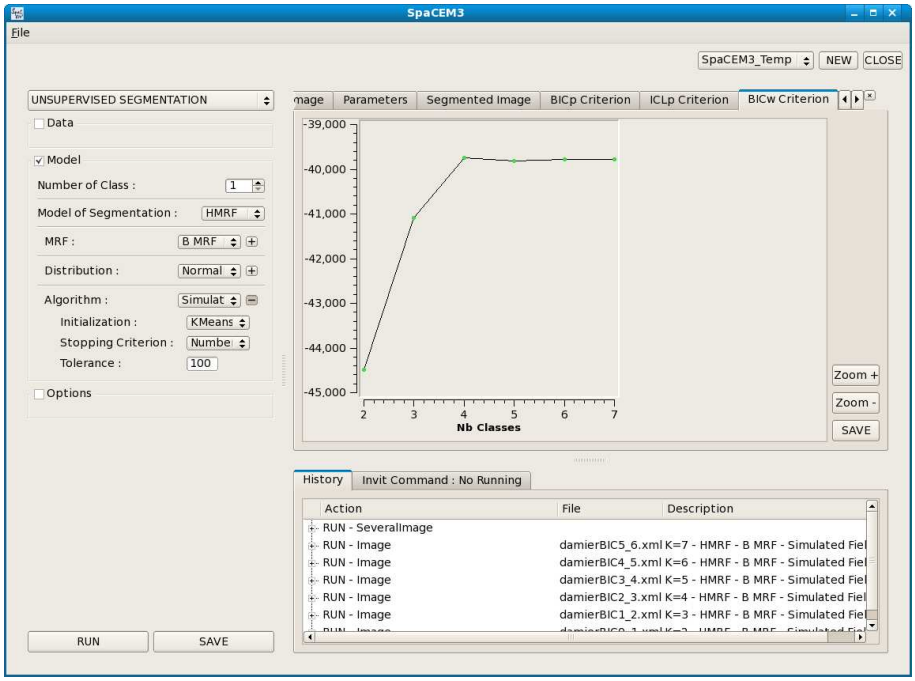


FIGURE 9 – Courbe des valeurs de BIC approchées par champ moyen pour un modèle de champ de Markov caché.

l'approximation BIC^p. Une illustration est donnée dans la Figure 9. De la même manière des approximations du critère ICL sont disponibles.

Références

[1] J. D. Banfield and A. E. Raftery. Model-based Gaussian and non Gaussian clustering. *Biometrics*, 49 :803–821, 1993.

[2] D. Benboudjema and W. Pieczynski. Unsupervised image segmentation using Triplet Markov fields. *Comput. Vision Image Underst.*, 99 :476–498, 2005.

[3] D. Benboudjema and W. Pieczynski. Unsupervised statistical segmentation of non stationary images using triplet Markov Fields. *IEEE Trans. PAMI*, 29(8) :367–1378, 2007.

[4] C. Biernacki, G. Celeux, and G. Govaert. Assessing a mixture model for clustering with the integrated complete lieklihood. *IEEE trans. PAMI*, 22 :719–725, 2000.

[5] J. Blanchet. *Modeles markoviens et extension pour la classification de donnees complexe*. PhD thesis, Universite Grenoble 1, october 2007.

[6] J. Blanchet and F. Forbes. Triplet Markov fields for the supervised classification of complex structure data. *IEEE trans. on Pattern Analyis and Machine Intelligence*, 30(6) :1055–1067, 2008.

[7] J. Blanchet and M. Vignes. A model-based approach to gene clustering with missing observation reconstruction in a Markov random field framework. *Journal of Computational Biology*, 16(3) :475–486, 2009.

[8] C. Bouveyron, S. Girard, and C. Schmid. High dimensional data clustering. *Comput. Statist. Data Analysis*, 2007.

[9] G. Celeux, F. Forbes, and N. Peyrard. EM procedures using mean field-like approximations for Markov model-based image segmentation. *Pat. Rec.*, 36(1) :131–144, 2003.

[10] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. Roy. Statist. Soc. Ser. B*, 39 :1–38, 1977.

[11] F. Forbes and N. Peyrard. Hidden Markov random field model selection criteria based on mean field-like approximations. *IEEE trans. PAMI*, 25(8), 2003.

[12] R.J. Little and D.B. Rubin. *Statistical analysis with missing data*. New-York : Wiley, second edition, 2002.

[13] W. Pieczynski and A. Tebbache. Pairwise Markov Random Fields and segmentation of textured images. *Machine Graph. Vision*, 9 :705–718, 2000.

[14] D. Rubin. Inference and missing data. *Biometrika*, 63 :581–592, 1976.

[15] G. Schwarz. Estimating the dimension of a model. *Ann. Stat.*, 6 :461–464, 1978.

[16] M. Vignes and F. Forbes. Gene clustering via integrated Markov models combining individual and pairwise features. *IEEE trans. Comput. Biol. Bioinform.*, 2007.