□ Thèse présentée pour l'obtention du grade de docteur de l'UTC
□ Spécialité : contrôle des systèmes

Approche probabiliste en classification automatique et contraintes de voisinage

par Christophe AMBROISE

Soutenance prévue le 4 octobre 1996 devant le jury composé par :

M.	G. CELEUX (Rapporteur)
MME	A. GUERIN-DUGUE (Rapporteur)
MM.	B. DUBUISSON
	G. GOVAERT
	M. SCHMITT

Remerciements

En premier lieu, je tiens à remercier Gérard Govaert, professeur à l'Université de technologie de Compiègne, qui a dirigé ces trois années de recherche. Toujours disponible pour discuter de problèmes scientifiques aussi bien que pratiques, il a été pour moi un exemple de rigueur et de sérieux. J'ai beaucoup apprécié la qualité de son encadrement et j'espère sincèrement que nous continuerons à travailler ensemble.

Gilles Celeux, directeur de recherche à l'INRIA Grenoble, a écrit de nombreux articles, sur l'approche probabiliste en classification, qui ont été une source d'information et d'inspiration pour mon travail de thèse. Ce chercheur, dont j'apprécie autant les qualités humaines que scientifiques, m'a fait l'honneur de rapporter ce mémoire et j'en suis très touché.

Anne Guérin-Dugué, maître de conférence à l'INPG (Grenoble), a eu la gentillesse de rapporter ce mémoire. Sachant le travail que cela représente, je la remercie sincèrement.

Michel Schmitt, directeur du centre de géostatistique de l'école des Mines de Paris, m'accueille dans son laboratoire pour l'année 1997. Je le remercie chaleureusement pour sa participation au jury. J'en suis très honoré .

Je suis très reconnaissant à Bernard Dubuisson, professeur à l'Université de technologie de Compiègne, d'avoir accepté de participer au jury.

Merci à Bertrand Vachon de m'avoir confié l'enseignement de travaux dirigés d'informatique. Travailler en sa compagnie a été une expérience de l'enseignement agréable et enrichissante.

Merci à Frédéric Guyon pour ses conseils éclairés en mathématique et pour son amitié.

Je remercie Mo Dang pour ses commentaires constructifs et pour son aide inestimable en programmation.

Je suis très reconnaissant à Djalil Kateb et Pierre Villon d'avoir répondu avec patience à mes questions concernant l'analyse numérique.

Merci à Stéphane Canu pour ses commentaires du "Monde", pour sa bonne humeur imperturbable et ses nombreux conseils, qui ont ponctué les innombrables allers-retours SNCF Paris Compiègne.

Mes remerciements vont à Nathalie Alexandre, Nathalie Laboureur, Jacqueline Beusnel et Dominique Porras pour leur disponibilité et efficacité face à mes nombreuses demandes.

Je remercie les membres de l'équipe logistique du département de génie informatique et en particulier Corinne Boscolo et David Lewis pour leur aide.

A Martine et Manon,

Die Tage sehen wir, die teuren, gerne schwinden,
Um etwas Teureres herangereift zu finden:
Ein seltenes Gewächs, das wir im Garten treiben,
Ein Kind, das wir erziehn, ein Büchlein, das wir schreiben.

Hermann Hesse (1943) *Das Glasperlenspiel.* Fretz & Wasmuth, Zürich

Table des matières

Notations

Mathématique

$arg\max_t f(t)$	argument qui maximise la fonction f
$arg\min_t f(t)$	argument qui minimise la fonction f
C_N^x	coefficient binomial
$I\!I_A$	fonction indicatrice (vaut 1 si A est vrai, 0 sinon)
I	matrice identité
$f(t) \propto g(t)$	f et g sont proportionnelles

Probabilité

$f(\mathbf{x}\|\theta)$	densité de $\mathbf{x}$ indicée par le paramètre θ
$I\!E_\theta[g(\mathbf{x})]$	espérance de $g(\mathbf{x})$ pour la loi de $\mathbf{x}$ indicée par le paramètre θ
$I\!E^k[h(\mathbf{x})\|\theta]$	espérance de $h(\mathbf{x})$ pour la loi conditionnelle de $\mathbf{x}$ sachant θ, $k(\mathbf{x}\|\theta)$
$x \sim f(\mathbf{x}\|\theta)$	$\mathbf{x}$ suit la loi de densité $f(\mathbf{x}\|\theta)$

Statistique et classification

$(\mathbf{x}_1, \cdots, \mathbf{x}_N)$	échantillon observé de taille N
$\ell(\theta; \mathbf{x})$	vraisemblance du paramètre θ
$L(\theta; \mathbf{x})$	log-vraisemblance du paramètre θ
$\mathbf{c}$	matrice de classification
p_k	proportion de la classe k
$\boldsymbol{\mu}_k$	vecteur moyenne de la classe k
$\boldsymbol{\Sigma}_k$	matrice de variance covariance de la classe k
$P(\mathbf{c}_i\|\mathbf{x}_i)$	probabilité a posteriori que $\mathbf{x}_i$ appartienne à la classe $\mathbf{c}_i$
$P(c_{ik} = 1\|\mathbf{x}_i)$	probabilité a posteriori que $\mathbf{x}_i$ appartienne à la classe k
$P(\mathbf{c}_i\|\mathbf{x}_i, \mathbf{c}_j, v_{ij} = 1)$	probabilité a posteriori que $\mathbf{x}_i$ appartienne à la classe $\mathbf{c}_i$, conditionnellement à la connaissance de la classe des voisins de $\mathbf{x}_i$

Introduction

Le présent mémoire traite de la prise en compte de connaissances *a priori* dans des algorithmes de classification automatique basés sur des modèles probabilistes. Pour plus de clarté, précisons, avant de rentrer dans le vif du sujet, le sens que nous attribuons au terme "classification automatique", qui peut être le sujet d'ambiguïtés.

Le petit Larousse définit la classification comme "la distribution par classe selon un certain ordre et une certaine méthode". L'ordre et la méthode choisis révèlent une conception des choses, imposent une description particulières des données brutes, et induisent parfois certaines interprétations. Aristote (322-384 avant Jésus-Christ), philosophe et homme de science grec, développa un système où toute chose existante trouvait sa place dans une classe précise. Cette vision du monde eut une profonde influence sur le développement de la science occidentale. En 1758 Linné entreprit de classifier tout ce qui vivait dans son ouvrage *Systema naturea*. Il décrivit toutes les espèces en fonction du nombre de leurs doigts, de leur taille ... Darwin interpréta cette classification en développant sa théorie de la sélection naturelle.

Sous le nom de classification automatique (ou encore taxonomie numérique) sont regroupés un ensemble d'algorithmes et d'heuristiques variés qui distribuent un ensemble d'objets dans des classes. Historiquement cette branche de l'analyse des données trouve ses origines en sociologie et en psychométrie dans les années 30 (Hartigan 1982). Elle s'est beaucoup developpée avec l'augmentation de la puissance de calcul des ordinateurs. Sokal et Sneath (1963) publiaient leur monographie *Principles of numerical taxonomy* qui fondait la classification automatique comme une discipline à part entière.

Une confusion répandue existe entre les termes classement et classification (respectivement classification et clustering en anglais). Le classement présuppose l'existence de classes dont certains objets sont connus, alors que la classification tente de découvrir une structure de classes qui soit "naturelle" aux données. Dans la littérature liée à la reconnaissance des formes, la distinction entre les deux approches est souvent désignée par les termes "apprentissage supervisé" et "non supervisé". Une classification peut avoir différentes motivations : compresser des informations, décrire de manière simplifiée de grandes masses de données, structurer un ensemble

de connaissances, révéler des structures, des causes cachées, réaliser un diagnostic...

Les méthodes de classification ont été développées pour la plupart par des praticiens, plutôt que des théoriciens, pour résoudre des problèmes concrets. Benzécri (1973) écrivait dans son deuxième tome de *L'analyse des données*:

> Statistique n'est pas probabilité. Sous le nom de statisitique mathématique, des auteurs (...) ont édifié une pompeuse discipline, riche en hypothèses qui ne sont jamais satisfaites dans la pratique. Ce n'est pas de ces auteurs qu'il faut attendre la solution de nos problèmes typologiques.

Les controverses entre les tenants des statistiques et ceux d'une analyse de données basée sur des considérations algébriques et géométriques se sont particulièrement développées dans les années 1970. Depuis ces conflits, les deux approches coexistent plus pacifiquement et sont parfois devenues complémentaires. Ainsi, les modèles probabilistes se sont avérés trés utiles en classification automatique. L'approche probabiliste a fourni un cadre théorique et des outils puissants qui ont permis de généraliser, d'interpréter certaines méthodes de classification purement géométriques (Celeux 1992).

L'approche probabiliste est la toile de fond de ce mémoire. Nous avons essayé d'apporter quelques réponses à une question pratique : "Comment prendre en compte des *a priori* sur la structure des classes, dans les méthodes, qui recherchent des partitions d'objets décrits par un ensemble de variables ?" Deux types d'*a priori* sont considérés, qui correspondent à des problèmes qui ont été posés respectivement dans le domaine des réseaux de neurones et en statistique spatiale :

- Le principe de conservation topologique : supposons définie une relation de voisinage entre les classes, on veut que deux objets similaires soient distribués dans des classes proches. Ce principe, issu de constatations biologiques, a donné naissance à de nombreux développements fructueux en reconnaissance des formes, dont les cartes auto-organisatrices de Kohonen (1982).

- Le principe de connexité : supposons que parmi toutes les variables qui décrivent les objets considérés, certaines soient relatives à une position "géographique", on veut alors que les classes représentées dans l'espace géographique ne soient pas trop morcelées (le plus connexe possible). Ce principe traduit l'intuition suivante : deux objets géographiquement proches ont plus de chance d'appartenir à une même classe que deux objets éloignés.

Le plan est structuré en quatre parties. Au premier chapitre, les approches probabilistes en classification (Bock 1989), en particulier le modèle de mélange fini de lois de probabilités (Banfield et Raftery 1993, Celeux et Govaert 1992, Celeux et Govaert 1995, Symons 1981) sont abordées. L'estimation des paramètres d'un mélange

de lois gaussiennes par l'algorithme EM (Dempster *et al.* 1977, Redner et Walker 1984, Hathaway 1986) constitue l'axe central du chapitre.

Le deuxième chapitre introduit la notion de conservation topologique ou encore cartographie associative (Amari 1980, Fritzke 1993, Kohonen 1982, Kohonen 1991), et présente les utilisations de ce principe en représentation (Ambroise et Trautmann 1994, Trautmann 1995, Ultsch 1990, Zhao 1992), et en classification (Luttrell 1990, Tomasini 1993). Des algorithmes originaux, versions modifiées de l'algorithme EM (Ambroise et Govaert 1995, Ambroise et Govaert 1996, Ambroise et Govaert à paraître), sont proposés et comparés. Ces algorithmes permettent d'intégrer le principe de conservation topologique dans un cadre statistique.

Le problème de la classification spatiale est le sujet du troisième chapitre. Après une revue des méthodes existantes de classification adaptées à des données "spatiales" (Geman et Geman 1984, Lebart 1978, Legendre 1987), des algorithmes basés sur l'algorithme EM sont proposés (Ambroise *et al.* To appear). Une relation formelle est établie avec les statistiques bayésiennes.

Les méthodes proposées dans ce mémoire sont illustrées, dans le quatrième chapitre, par trois applications :

- la classification automatique de données géologiques,
- la projection de données multidimensionnelles relatives à des distances-horaires SNCF, dans le plan,
- la segmentation d'images de cultures cellulaires.

Les deux premières applications concernent une version modifiée de l'algorithme EM, proposée au chapitre 2, qui intègre le principe de conservation topologique. La dernière application illustre les capacités de la méthode de classification spatiale, exposée au chapitre 3.

Chapitre 1

Approche probabiliste en classification automatique

Statisticians are often interested in defining homogeneous groups.

Fisher (1958)

L'approche probabiliste en classification automatique considère que les objets à classer sont les réalisations indépendantes d'une variable aléatoire suivant une certaine distribution f. La connaissance de la loi f induit un partitionnement des données.

Disposer d'un modèle explicite entraîne des avantages pratiques en classification. Cela permet de trouver facilement de nouveaux critères, d'aider l'interprétation, de fixer certains paramètres du modèle et surtout de disposer du cadre formel solide de la statistique.

Ce chapitre présente essentiellement l'estimation par maximum de vraisemblance des paramètres d'un mélange de lois gaussiennes, par l'algorithme EM et quelques unes de ses nombreuses variantes. Des liens sont mis en évidence entre cette approche, la classification floue, et la physique statistique. Une solution bayésienne de ce problème d'estimation est également introduite.

1.1 Un cadre pour une classification formelle

Une définition formelle de la classification, qui puisse servir de base à un processus automatisé, amène à se poser les questions suivantes :

- Comment les objets à classer sont-ils définis ?
- Comment définir la notion de ressemblance entre objets ?

- Qu'est-ce qu'une classe?
- Comment sont structurées les classes?
- Comment juger une classification par rapport à une autre?

A toutes ces questions, des réponses exhaustives, présentées simplement, peuvent être trouvées dans l'excellent ouvrage de Gordon (1980). Dans la suite de ce document nous serons amenés à préciser dans quel cadre de classification formelle nous nous plaçons (c'est-à-dire quelles réponses aux questions précédentes nous considérons).

Dans la majorité des cas, les données à classer se présentent sous la forme d'un tableau de N individus décrits par d variables, qui peuvent être de nature différente (qualitatives, binaire, quantivative). Dans ce document, nous nous intéresserons uniquement à des individus décrits par des variables quantitatives. Dans ce cas, les individus (encore appelés objets) sont des vecteurs de $\mathbb{R}^d$.

Marque de cigarette	Nicotine (mg)	Goudron (mg)
Royal anis	0.45	4.9
Rothmans	1.1	14
Chesterfield Lights	0.6	8
Benson & Hedges	1.1	13
Peter Stuyvesant	1	12.7
Gitanes	1	12
Malboro	1	14
Lucky Strike	0.9	14

TAB. 1.1 - : Exemple d'un tableau individus-variables

Pour mesurer la ressemblance entre deux objets, deux démarches sont envisageables:

- On peut dire que deux objets sont semblables s'ils partagent une certaine caractéristique. Considérons le nombre de doigts d'un être vivant et comparons le singe et l'homme: sur ce critère de comparaison (et sur bien d'autres) les deux espèces seront jugées semblables. Ce genre de démarche aboutit à une classification *monothétique* base de l'approche aristotélicienne (Sutcliffe 1994). Tous les objets d'une même classe partagent alors un certain nombre de caractéristiques (e.g.: "Tous les hommes sont mortels").
- On peut aussi mesurer la ressemblance en utilisant une mesure de proximité (distance, dissimilarité). Dans ce cas la notion de ressemblance est mesurée de façon plus floue et deux objets d'une même classe possèderont des caractérsitiques "proches" au sens de la mesure utilisée. Cette démarche est dite *polythétique*.

Exemple 1.1 Classification polythéthique : considérons l'ensemble des marques de cigarettes vendues dans les bureaux de tabac français. Le tableau 1.1 est un échantillon de cet ensemble. Un problème de classification possible est la séparation de cet ensemble en deux classes : cigarettes légères et cigarettes fortes. Si l'on décrit une marque par sa teneur en nicotine et en goudron, une mesure de ressemblance entre deux marques peut être donnée par la distance euclidienne entre les vecteurs descripteurs de chaque marque :

$$d(cigaretteA, cigaretteB) = \sqrt{(nicotineA - nicotineB)^2 + (goudronA - goudronB)^2}$$

△

Nous concentrerons notre attention sur l'approche polythétique et plus particulièrement sur les méthodes de classification qui mesure la ressemblance à l'aide d'une distance.

Une classification amène à répartir l'ensemble des données en différentes classes. La définition d'une classe et les relations entre classes peuvent être très variées. Les principales structures de classification sont la hiérarchie et la partition. Nous étudierons uniquement la partition.

1.1.1 Partition dure

Définition 1.1 *Ω étant un ensemble fini, un ensemble $P = (P_1, P_2, ..., P_K)$ de parties non vides de Ω est une partition si :*

1. $\forall i \neq j,\ P_i \cap P_j = \emptyset$,
2. $\cup_i P_i = \Omega$.

Exemple 1.2 Soit l'ensemble des marques de cigarettes listées dans le tableau 1.1. Une partition possible est la suivante :

- $P_1 = \{RoyalAnis, Chesterfield\}$,
- $P_2 = \{Benson, Malboro, Gitanes, Lucky, Peter, Rothmans\}$.

△

Dans un ensemble $\Omega = (\mathbf{x}_1, ..., \mathbf{x}_N)$ partitionné en K classes, chaque élément de l'ensemble appartient à une classe et une seule. Une manière pratique de décrire cette partition P consiste à utiliser une notation matricielle. Soit $\mathbf{c}(P)$ la matrice caractéristique de la partition $P = (P_1, P_2, ..., P_K)$ (ou matrice de classification) :

$$\mathbf{c}(P) = \mathbf{c} = \begin{pmatrix} c_{11} & \cdots & c_{1K} \\ \vdots & \ddots & \vdots \\ c_{N1} & \cdots & c_{NK} \end{pmatrix}$$

où $c_{ik} = 1$ si et seulement si $\mathbf{x}_i \in P_k$, et $c_{ik} = 0$ sinon. Remarquons que la somme de la i^{e} ligne est égale à 1 (un élément appartient à une seule classe) et la somme des valeurs de la k^{e} colonne vaut n_k le nombre d'éléments de la classe P_k. On a donc $\sum_{k=1}^{K} n_k = N$.

1.1.2 Partition floue

La notion de partition dure repose sur une conception ensembliste classique. Considérant les travaux de Zadeh (1965) sur les ensembles flous, une définition du concept de partition floue semble "naturelle". La classification floue, développée au début des années 1970 (Ruspini 1969), généralise une approche classique en classification en élargissant la notion d'appartenance à une classe.

Quelques définitions sont nécessaires à l'explicitation de la fonction d'appartenance qui est le concept de base des ensembles flous :

Définition 1.2 *L'ensemble de tous les objets possibles dans un contexte particulier est appelé univers du discours (ou référentiel).*

Exemple 1.3 L'ensemble de toutes les marques de cigarettes existant sur le marché constitue un référentiel fini.

△

Définition 1.3 *Un (sous-)ensemble classique $A \in \Omega$ est une collection d'éléments dont l'appartenance est parfaitement définie. Soit x un élément quelconque, on peut affirmer que $x \in A$ ou $x \notin A$.*

Lorsqu'il est fini, un ensemble classique peut être défini par l'énumération de ses éléments.

Définition 1.4 *Un (sous-)ensemble flou $A \subset \Omega$ est une collection de couples*

$$\{(x, \mu_A(x)) | x \in \Omega\}$$

tels que $\mu_A(x) \in [0, 1]$.

μ_A est une fonction de Ω dans $[0, 1]$. A chaque élément x du référentiel Ω, elle fait correspondre un degré d'appartenance. Cette fonction d'appartenance définit complètement l'ensemble flou A. À partir de cette définition d'un ensemble flou, il est possible de redéfinir les opérations ensemblistes de base telles que l'intersection, l'union, l'inclusion, la complémentarité.... (Zadeh 1965).

Exemple 1.4 Considérant l'univers du discours des marques de cigarettes du tableau 1.1, définissons l'ensemble H des cigarettes à haute teneur en goudron dont la fonction d'appartenance est :

$$\mu_H(x_i) = \begin{cases} 0 & \text{if } x_i < 8; \\ \frac{x_i}{3} - \frac{8}{3} & \text{if } x_i \in [8, 11]; \\ 1 & \text{if } x_i > 11, \end{cases}$$

où x_i est la teneur en goudron de la cigarette i (Figure 1.1).

△

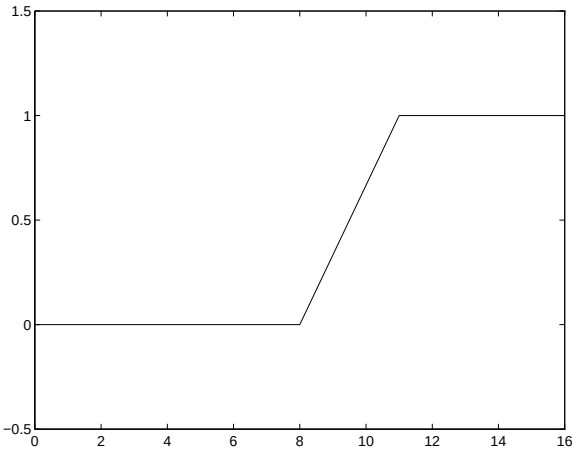


FIG. 1.1 - : Fonction d'appartenance floue de l'ensemble "cigarette forte"

En classification automatique, la fonction d'appartenance est utilisée pour quantifier le degré d'appartenance d'un individu à une classe. Un classe P_k d'un ensemble d'individus Ω peut alors être définie comme un sous-ensemble flou de fonction μ_k du référentiel Ω.

Définition 1.5 *$\Omega = (\boldsymbol{x}_1, ..., \boldsymbol{x}_N)$ étant un ensemble fini, K sous-ensemble flous de fonction d'appartenance respective μ_1,...,μ_K définissent une partition floue ssi:*

1. $\forall k = 1..K,\ 0 < \sum_{i=1}^{N} \mu_k(\boldsymbol{x}_i) < N$,
2. $\forall \boldsymbol{x}_i \in \Omega,\ \sum_{k=1}^{K} \mu_k(\boldsymbol{x}_i) = 1$.

La première condition traduit le fait qu'aucune classe ne doit être vide et la seconde condition exprime le concept d'appartenance totale. Pour garder une notation cohérente nous écrirons les matrices de partition dure et matrices de partition floue en utilisant les même notations. Nous noterons donc l'appartenance de l'individu $\mathbf{x}_i$ à la classe k, c_{ik} au lieu de $\mu_k(\mathbf{x}_i)$ dans la suite de ce document.

1.1.3 Point de vue algorithmique

Les concepts de partition et de classification polythétique étant précisés, la question suivante émerge: comment trouver une partition optimale d'un ensemble de données, lorsque la ressemblance entre deux individus est évaluée par une mesure de proximité?

La première chose à faire consiste à clarifier formellement le sens du mot optimal. La solution généralement adoptée est de choisir une mesure numérique de la qualité d'une partition. Cette mesure est parfois appelée critère, fonctionnelle, ou bien encore fonction d'énergie. L'objectif d'une procedure de classification est donc de trouver la partition ou les partitions qui donnent la meilleure valeur (la plus petite ou la plus grande) pour un critère donné.

Mais le nombre de partitions possibles, même pour un problème de taille raisonnable, est énorme. En effet si l'on considère un ensemble de N objets à partitionner en K classes, le nombre de partitions possibles est :

$$NP(N,K) = \frac{1}{K!}\sum_{k=0}^{K}(-1)^{k-1}\cdot C_k^K \cdot k^N. \tag{1.1}$$

Exemple 1.5 Soit un ensemble de 8 objets que l'on désire partager en 4 classes. Il existe 1701 partitions possibles !

△

Plutôt que de chercher la meilleure partition, celle qui donne la valeur optimale du critère, on utilise des méthodes plus rapides qui convergent vers des optima "locaux" du critère. Les partitions ainsi trouvées sont souvent satisfaisantes.

Quel critère choisir pour mesurer la qualité de la partition ? De nombreux critères existent (Gordon 1980). Certains peuvent être liés, comme nous le verrons dans la suite, au choix d'un modèle pour l'ensemble des données. L'une des fonctions les plus utilisée est la somme des variances intra-classes :

$$W(\boldsymbol{\mu}, \mathbf{c}) = \sum_{k=1}^{K}\sum_{i=1}^{N} c_{ik}\|\mathbf{x}_i - \boldsymbol{\mu}_k\|^2 \tag{1.2}$$

où les $\boldsymbol{\mu}_k$ sont les prototypes (centres) des classes et les c_{ik} sont les éléments d'une matrice de partition dure. Le problème posé est alors un probème d'optimisation sous contraintes (liées aux c_{ik}) :

$$(\hat{\mathbf{c}}, \hat{\boldsymbol{\mu}}) = arg \min_{(\mathbf{c},\boldsymbol{\mu})} W(\mathbf{c}, \boldsymbol{\mu}) \tag{1.3}$$

L'algorithme des centres mobiles

Un algorithme très répandu pour résoudre ce probème est celui des *k-means* ou centres mobiles. Historiquement, cet algorithme date des années soixante. Il a été proposé par plusieurs chercheurs dans différents domaines à des dates proches (Edwards et Cavalli-Sforza 1965, Lloyd 1957). Cet algorithme basé sur des considérations géométriques doit certainement son succés à sa simplicité et son efficacité :

1. Initialisation des centres : une méthode répandue consiste à initialiser les centres avec les coordonnées de K points choisis au hasard.

2. Ensuite les itérations possèdent la forme alternée suivante :

   - étant donné $\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_K$, choisir les c_{ik} qui minimisent W,
   - étant donné $\mathbf{c} = \{c_{ik}\}$, minimiser W par rapport aux prototypes $\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_K$.

La première étape affecte chaque $\mathbf{x}_i$ au prototype le plus proche, et la seconde étape recalcule la position des prototypes en considérant que le prototype de la classe i devient son vecteur moyenne. Il est possible de montrer que chaque itération fait décroître le critère mais aucune garantie de convergence vers un maximum global n'existe en général. Si le critère des k-means est considéré du point de vue de la recherche d'une partition floue, c'est-à-dire si les contraintes sur les c_{ik} sont relachées et deviennent $c_{ik} \in [0, 1]$ à la place de $c_{ik} \in \{0, 1\}$, la partition optimale au sens du nouveau critère est celle qui est optimale pour le critère classique (Selim et Ismail 1984). En d'autre terme, il n'y a aucun intérêt à considérer des partitions floues, lorsqu'on travaille avec le critère des k-means.

Cette forme d'algorithme alterné où un certain critère est optimisé, alternativement par rapport aux variables d'appartenance aux classes, puis par rapport aux paramètres définissant ces classes a été intensivement exploité. Citons entre autre *les nuées dynamiques* de Diday (Diday 1971) et l'algorithme des *fuzzy c-means* (Bezdek 1974).

Notons que Webster Fisher (1958) (à ne pas confondre avec Ronald Fisher) avait proposé un algorithme trouvant la partition optimale, au sens de la variance intra-classe, d'un ensemble de N données unidimensionelles en $O(N \cdot K^2)$ opérations en utilisant des méthodes issues de la progammation dynamique.

Une version adaptative des centres mobiles

Une autre version des *k-means* (Macqueen 1967) consiste à modifier les prototypes des classes en considérant les données une à une. On parle alors d'algorithme adaptatif :

1. Les K prototypes sont tirés au hasard parmi les N points.

2. A l'itération q, un individu $\mathbf{x}_i$ est choisi au hasard.

   - Détermination du prototype le plus proche de $\mathbf{x}_i$:

     $$\boldsymbol{\mu}_k^q = \min_j \|\mathbf{x}_i - \boldsymbol{\mu}_j^q\|.$$

     L'individu est affecté à la classe k.
   - Modification du prototype $\boldsymbol{\mu}_k^q$:

     $$\boldsymbol{\mu}_k^{q+1} = \frac{\mathbf{x}_i + n_k^q \cdot \boldsymbol{\mu}_j^q}{n_k^q + 1},$$

et

$$n_k^{q+1} = n_k^q + 1$$

où n_k^q représente l'effectif de la classe k à l'itération q.

Les algorithmes adaptatifs sont particulièrement adéquats lorsque toutes les données à classer ne sont pas disponibles à l'avance. Les paramètres définissant les classes peuvent alors être ajustés à l'apparition de chaque nouvelle donnée sans trop de calculs.

De nombreux autres algorithmes et heuristiques qui optimisent d'autres critères ou qui simplement produisent une partition, existent. Nous concentrerons notre attention sur l'approche probabiliste du problème de partitionnement. Cette approche offre un cadre formel bien défini, dispose de méthodes puissantes, et permet surtout de donner une interprétation à la démarche de classification.

1.2 Modèles paramétriques et partitions

L'approche probabiliste de la recherche de partitions fait l'hypothèse que l'ensemble des données $\mathbf{x} = (\mathbf{x}_1, ..., \mathbf{x}_N)$ est la réalisation d'un échantillon de n variables aléatoires indépendantes de même loi f, prenant leurs valeurs dans $\mathbb{R}^d$. La connaissance de cette loi f doit permettre de séparer "naturellement" les N observations en K classes. Deux approches statistiques sont envisageables pour modéliser la loi f :

(a) Modélisation par la méthode de noyaux (b) Modèle gaussien

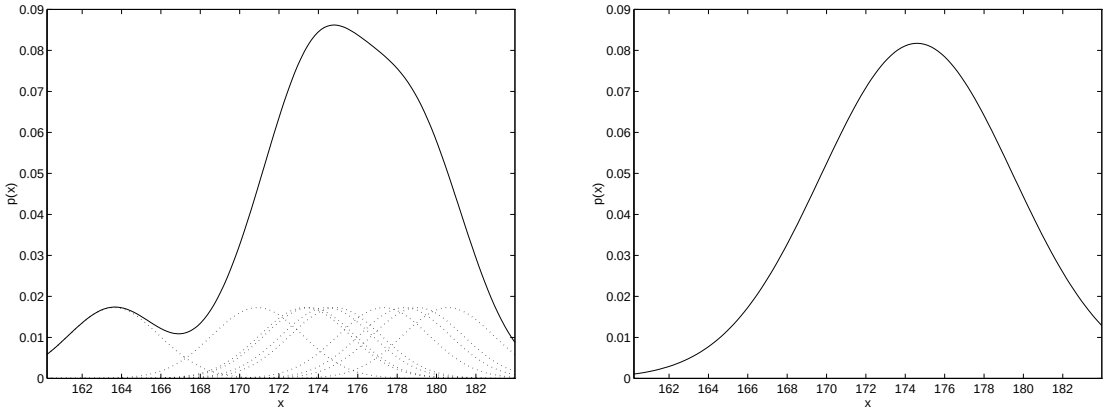


FIG. 1.2 - : Répartition de la taille d'adultes de sexes masculins modélisée par une méthode non paramétrique (a) et une méthode paramétrique (b) à partir de 10 mesures.

– L'approche non paramétrique ne fait pratiquement aucune hypotèse restrictive sur la nature de la distribution. Les connaissances a priori sur les données

n'influencent pas le modèle. Dans ce contexte, la recherche de partition s'effectue en deux étapes :

1. détermination de la distribution par une méthode non paramétrique (méthode des noyaux, méthode des k plus proche voisins).
2. calcul de la partition par la recherche des modes ou par la méthode des classes à haute densité (Bock 1989).

– L'approche paramétrique suppose que la loi cherchée appartient à une famille de loi connues $f(\mathbf{x}|\theta)$ et que cette loi est complètement déterminée par la connaissance de θ, un vecteur de paramètres inconnus de dimension fini. La recherche d'une partition à l'aide de modèles statistiques paramétriques comporte trois étapes principales :

1. le choix d'un modèle ;
2. l'estimation des paramètres de ce modèle ;
3. l'obtention d'une partition à partir du modèle.

Exemple 1.6 La taille d'un homme peut être considérée comme une variable aléatoire. La Figure 1.2 montre la distribution de probabilité de la taille, obtenue par une méthode paramétrique et une méthode non paramétrique, à partir d'un échantillon de taille 10.

△

Dans la section suivante, le modèle de mélange et la distribution de Gibbs sont présentés. Ces deux modèles servent de base à toutes les méthodes de recherche de partition présentées dans ce mémoire.

1.2.1 Modèle de mélange gaussien

En 1894, Karl Pearson publiait un article sur l'estimation par la méthode des moments des cinq paramètres d'une densité mélange de deux distribution normales univariées (Pearson 1894). Depuis, ce genre de modèle connait un certain succès et a été à l'origine de nombreuses applications.

D'une manière très générale, les mélanges de densité sont des distributions de probabilité de la forme suivante :

$$f(\mathbf{x}) = \int h(\theta) \cdot f(\mathbf{x}|\theta) d\theta \tag{1.4}$$

où $f(\mathbf{x}|\theta)$ est une densité paramétrique conditionnelle définie par le paramètre θ et $h(\theta)$ est la densité de mélange.

Lorsque la densité de probabilité $h(\theta)$ est discrète et prend ses valeurs sur un ensemble fini $(\theta_1, \cdots, \theta_K)$ avec les probabilités $(p_1, \cdots, p_K)$ (avec $\sum_{k=1}^{K} p_k = 1$), la densité f s'écrit

$$f(\mathbf{x}) = \sum_{k=1}^{K} p_k f_k(\mathbf{x}|\theta_k), \tag{1.5}$$

et on parle de mélange fini. Ce genre de densité apparait naturellement lorsque la population considérée est formée de plusieurs sous-populations qui ont des densités différentes.

Exemple 1.7 En mécanique, l'étude d'un matériau passe souvent par une phase pratique d'essais de traction. On tire sur le matériau pour observer la déformation et temps de rupture. En pratique, une distribution de Weibull est souvent un bon modèle statistique du temps de rupture. Comme le matériau peut se rompre pour diverses raisons, un mélange de distributions de Weibull permet de modéliser le phénomène. Dans ce cas, il y aura autant de composants que de raisons de rupture.

△

Notons aussi que les mélanges finis peuvent modéliser des distributions de probabilités "biscornues" dont les modes ne correspondent pas forcement à la présence d'une sous-population. Les mélanges sont un intermédiaire, un compromis entre approche paramétrique et non paramétrique.

En classification automatique, le modèle de mélange fini de densité est l'un plus étudié. Dans ce document, nous nous concentrerons sur le modèle de mélange gaussien (Figure 1.3), qui est de loin le plus populaire. En effet, lorsque peu d'information a priori est disponible sur une population et les éventuelles sous-populations qui la composent, l'hypothèse de normalité des sous-populations parait souvent raisonnable. Dans ce contexte, on considère un échantillon $(\mathbf{x}_1, ..., \mathbf{x}_N)$ d'une variable aléatoire à valeurs dans $\mathbb{R}^d$ d'une densité de mélange à K composantes f_k de la forme

$$f_k(\mathbf{x}_i|\theta_k) = (2\pi)^{-\frac{d}{2}} \det|\mathbf{\Sigma}_k|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_k)^t \mathbf{\Sigma}_k^{-1}(\mathbf{x} - \boldsymbol{\mu}_k)\right) \tag{1.6}$$

avec $\boldsymbol{\mu}_k$ le vecteur moyenne, $\mathbf{\Sigma}_k$ la matrice de variance covariance, et d la dimension des vecteurs $\mathbf{x}_i$.

Le problème consiste à estimer les paramètres du mélange. Avant de résoudre ce genre de problème, il faut s'assurer que le problème est bien posé, c'est-à-dire qu'il admet une solution unique et donc que les composants du mélange sont effectivement identifiables.

Exemple 1.8 Le mélange de deux lois uniformes n'est pas identifiable. Prenons par exemple les deux distributions suivantes :

$$f(x) = \frac{1}{3}U[-1, 1] + \frac{2}{3}U[-2, 2]$$
$$f(x) = \frac{1}{2}U[-2, 1] + \frac{1}{2}U[-1, 2]$$

(a) Trois distributions gaussiennes (b) Mélange de distributions

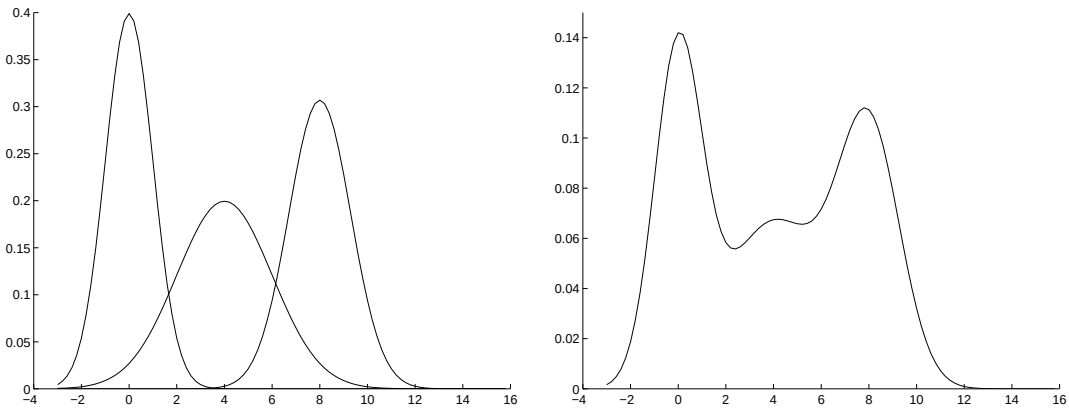


FIG. 1.3 - : Exemple d'un mélange gaussien

Elles sont identiques et il existe même une infinité de mélanges de lois uniformes qui sont identiques aux deux densités précédentes.

△

On peut montrer que les mélanges gaussiens (ainsi que les mélanges exponentiels, de Poisson et de Cauchy) sont identifiables. Dans la suite du document, nous détaillerons plusieurs méthodes d'estimation adaptées à ce genre de modèles de mélange identifiable.

1.2.2 Distribution de Gibbs-Boltzmann

Mécanique statistique

Ludwig Boltzmann (1844–1906) se suicida, désespéré par la réaction de la communauté scientifique face aux idées novatrices qu'il avait développées. Ce chercheur autrichien, à la suite des travaux de Maxwell (1839–1903), tentait d'expliquer les propriétés de la matière, observées au niveau macroscopique, à partir des propriétés d'hypothétiques particules microscopiques la composant (hypothétique car la structure atomique de la matière ne fut démontrée de manière directe qu'en 1908 par Perrin).

Avec W. Gibbs, L. Boltzmann est un des fondateurs de la mécanique statistique. Cette théorie scientifique combine l'utilisation des lois de la mécanique classique ou bien quantique avec la statistique. En effet les échantillons considérés sont composés d'approximativement 10^{23} atomes ou molécules, et une approche déterministe qui tiendrait compte des caractéristiques exactes (i.e. position et vitesse d'une molécule dans un gaz) de chaque élément aboutirait à des calculs fantastiquement compliqués.

La mécanique statistique se contente donc de calculer des quantités globales et permet une très bonne approximation de la réalité. Une bonne introduction, illustrée d'exemples, est donnée par les cours de physique de Berkeley (Reif 1972).

A l'équilibre, un système thermodynamique ouvert en contact avec une source de chaleur à température T possède une énergie E qui fluctue légèrement en fonction de la configuration ω du système. Cette configuration est la réalisation d'une variable aléatoire qui suit une distribution de Gibbs-Boltzmann (ou encore distribution canonique) :

$$\pi(\omega) = \frac{1}{Z(T)} e^{-E(\omega)/kT}, \tag{1.7}$$

où $Z(T)$ un facteur de normalisation aussi appelé fonction de partition est défini par :

$$Z(T) = \int e^{-E(\omega)/kT} d\omega, \tag{1.8}$$

et k est la constante de Boltzmann.

Exemple 1.9 Dans le cadre de la mécanique statistique classique, l'énergie d'un gaz parfait (gaz où les interactions entre les molécules sont négligeables), où chacune des N molécule possède une vitesse v_i, se résume à la somme de l'énergie cinétique de chaque molécule :

$$E(\omega) = \sum_{i=1}^{N} \frac{1}{2} m v_i^2. \tag{1.9}$$

△

La distribution de Gibbs permet de replacer de nombreux problèmes de minimisation d'une fonction d'énergie dans le cadre probabilité-statistique car minimiser $E(\omega)$ revient à trouver la configuration ω la plus probable au sens de la distribution de Gibbs correspondante. Cette transposition permet d'utiliser des outils propres à la statistique (calculs de Monte Carlo, estimation) pour résoudre des problèmes d'optimisation numérique.

1.3 Principes d'estimation

Une fois le modèle probabiliste choisi, au vu des observations et des a priori de l'analyste, il reste à estimer les paramètres de ce modèle. L'estimation d'un paramètre soulève deux types de problèmes :

- Comment obtenir un estimateur ?
- Comment juger de la qualité de cet estimateur ?

Avant d'ébaucher une réponse à ces questions, définissons plus formellement la notion d'estimateur et d'estimation :

Définition 1.6 *Soit un échantillon i.i.d. $X = \{X_1, \cdots, X_N\}$ de loi parente dépendant d'un paramètre θ. On appelle estimateur du paramètre θ toute fonction de l'échantillon $X_1, \cdots, X_N$.*

Un estimateur est donc une variable aléatoire, fonction de variables aléatoires. La réalisation $\hat{\theta}(\mathbf{x})$ d'un estimateur $\hat{\theta}(X)$ est une estimation. Quand plusieurs estimateurs sont disponibles, il faut pouvoir les comparer. En théorie de la décision, l'approche fréquentiste consiste alors à considérer le risque fréquentiste :

$$R(\theta, \hat{\theta}) = I\!E_\theta[C(\theta, \hat{\theta}(X))] = \int_{\mathcal{X}} C(\theta, \hat{\theta}(\mathbf{x})) f(\mathbf{x}|\theta) d\mathbf{x}, \tag{1.10}$$

où $C(\theta, \hat{\theta}(\mathbf{x}))$ est le coût de choisir $\hat{\theta}(\mathbf{x})$ alors que la valeur du paramètre est θ.

Ce risque est une fonction de θ, qui permet de définir certaines qualités d'un estimateur : un estimateur minimax minimisera par exemple le risque maximum. Un estimateur $\hat{\theta}$ sera inadmissible s'il existe un meilleur estimateur $\hat{\theta}'$ tel que pour tout θ, $R(\theta, \hat{\theta}') \leq R(\theta, \hat{\theta})$ et pour au moins une valeur de θ, $R(\theta, \hat{\theta}') < R(\theta, \hat{\theta})$.

Exemple 1.10 Un risque classique est celui qui utilise un coût quadratique

$$C(\theta, \hat{\theta}) = (\theta - \hat{\theta})^2. \tag{1.11}$$

Dans ce cas, le risque peut être formuler comme la somme de la variance et du biais de l'estimateur :

$$R(\theta, \hat{\theta}) = I\!E_\theta[(\theta - \hat{\theta})^2] = Var[\hat{\theta}] + (I\!E_\theta[\hat{\theta}] - \theta)^2 \tag{1.12}$$

Ainsi le "meilleur" estimateur au sens de ce risque, parmi les estimateurs sans biais est celui qui a la variance minimum.

△

La théorie de l'estimation est une branche très importante de la statistique et nous présentons dans la suite seulement deux types d'estimateurs qui nous seront utiles pour la présentation de l'approche probabiliste en classification.

1.3.1 Estimation par le maximum de vraisemblance

Entre autres contributions à la statistique, R. Fisher (1890–1962) a introduit le concept de vraisemblance en 1912 dans un article intitulé "On absolute criterion for fitting frequency curves" (Fisher 1912). Aujourd'hui (1996), les estimateurs du maximum de vraisemblance jouent un rôle central dans la théorie de l'estimation.

Définition 1.7 *Soit la réalisation d'un échantillon i.i.d. $\boldsymbol{x} = (\boldsymbol{x}_1, ..., \boldsymbol{x}_n)$ d'une variable aléatoire de densité f dépendant d'un paramètre θ. On note $f(\boldsymbol{x}|\theta) = \prod_{i=1}^n f(\boldsymbol{x}_i|\theta)$ la densité de l'échantillon et $\ell(\theta; \boldsymbol{x}) = f(\boldsymbol{x}|\theta)$ la vraisemblance du paramètre θ.*

La notion de vraisemblance amène à réécrire la densité de l'échantillon en considérant le paramètre θ comme fonction d'un échantillon observé, $\mathbf{x}$. Cette définition du concept de vraisemblance conduit naturellement à la définition de l'estimateur du maximum de vraisemblance :

Définition 1.8 *L'estimateur du maximum de vraisemblance $\hat{\theta}_{MV}$ de θ est tel que $\hat{\theta}_{MV} = arg\max_\theta \ell(\theta; \boldsymbol{x})$.*

Il est souvent plus avantageux de maximiser la log-vraisemblance $L(\theta; \mathbf{x}) = \log \ell(\theta; \mathbf{x})$ plutôt que la vraisemblance. Dans le cas particulier où la log-vraisemblance est deux fois différentiable et le paramètre est un scalaire, $\hat{\theta}_{MV}$ est la solution du système :

$$\begin{cases} \frac{\partial L(\theta;\mathbf{X})}{\partial \theta} = 0 \\ \frac{\partial^2 L(\theta;\mathbf{X})}{\partial \theta^2} < 0 \end{cases}$$

Exemple 1.11 Soit un billard, et une boule de billard A. La boule A est lancée perpendiculairement à un bord de référence et s'arrête à une distance l de ce bord du billard. Un seconde boule B est lancée n fois et l'on note X le nombre de fois où B s'arrête à une distance l' du bord telle que $l' > l$. On cherche à estimer la proportion p du nombre de fois où $l' > l$ sachant que $X = x$. On suppose que $X \sim \mathcal{B}(n, p)$ (loi binomiale). Dans ce cas on dispose d'un échantillon de taille 1 et la vraisemblance du paramètre p s'écrit :

$$\ell(p; x) = C_n^x p^x (1-p)^{n-x},$$

et en annulant la dérivée première de la log-vraisemblance, on obtient

$$\hat{p}_{MV} = arg\max_p \ell(p; \mathbf{x}) = \frac{x}{n}$$

Si de nombreux lancés ont été effectués, l'estimation de p sera satisfaisante. Par contre si un seul lancé est observé, on trouve :

- soit $x = 0$, ce qui donne $\hat{p} = 0$;
- soit $x = 1$, ce qui donne $\hat{p} = 1$.

Dans les deux cas l'estimation paraît intuitivement de très mauvaise qualité, mais nous constaterons dans le prochain exemple, qu'il existe d'autres méthodes statistiques qui produisent des résultats sensés pour des échantillons de petite taille.

△

Les estimateurs du maximum de vraisemblance possèdent un coté intuitif séduisant et de plus ont de très bonnes propriétés asymptotiques :

- Pour presque toutes les distributions, $\hat{\theta}_{MV}$ estimateur du maximum de vraisemblance de θ est consistant, c'est-à-dire que $\hat{\theta}_{MV}$ converge en probabilité vers θ :

$$\forall \epsilon > 0, \ \lim_{n\to\infty} P(|\hat{\theta}_{MV} - \theta| > \epsilon) = 0.$$

- $\hat{\theta}_{MV}$ est asymptotiquement gaussien (sous certaines conditions).
- $\hat{\theta}_{MV}$ est asymptotiquement efficace (sous certaines conditions). Quand la taille de l'échantillon est très grande, la variance de l'estimateur du m.v. est plus petite ou égale à celle de tous les autres estimateurs vérifiant les conditions de Cramer.

Notons que la méthode du m.v. fournit de très bons estimateurs pour de grands échantillons, mais que ces bonnes qualités ne sont plus vraies lorsque des échantillons de taille réduite sont considérés.

Le calcul des estimateurs du maximum de vraisemblance n'est pas toujours aussi simple que dans l'exemple 1.11 et nécessite souvent l'utilisation d'algorithmes itératifs. De nombreuses techniques d'optimisation existent pour résoudre ce problème de maximisation du critère de vraisemblance : descente de gradient, gradient conjugué, algorithme de Newton, ... Dans la suite, nous détaillerons particulièrement une méthode très en faveur parmi les statisticiens : l'algorithme EM.

1.3.2 Estimation bayésienne

Deux ans après la mort du révérend T. Bayes (1701-1761), un ami de celui-ci, publiait son *essai en vue de résoudre la doctrine des chances* (Bayes 1763). Dans ce petit livret, qui est à l'origine de l'inférence statistique bayésienne moderne, paramètres dirigeant un phénomène aléatoire et observations de ce phénomène ne sont pas considérés comme des quantités fondamentalement différentes. Les paramètres ne sont plus traités comme des quantités déterministes mais aléatoires tout comme les observations. Le rôle dual des paramètres θ et des observations $\mathbf{x}$ est décrit grâce au conditionnement par le théorème de Bayes :

Théorème 1.1 *(Théorème de Bayes) Pour une loi (dite* a priori*) π sur le paramètre θ, et une observation $\boldsymbol{x}$ de densité $f(\boldsymbol{x}|\theta)$, la distribution de θ conditionnellement à $\boldsymbol{x}$ a pour densité*

$$\pi(\theta|\boldsymbol{x}) = \frac{f(\boldsymbol{x}|\theta)\pi(\theta)}{\int f(\boldsymbol{x}|\theta)\pi(\theta)d\theta}.$$

L'innovation principale du modèle statistique bayésien est la loi π sur les paramètres du modèle.

Ainsi, dans le contexte d'une approche statistique bayésienne trois fonctions doivent être spécifiées :

1. la loi sur les observations, $f(\mathbf{x}|\theta)$;
2. la distribution a priori sur les paramètres, $\pi(\theta)$;
3. le coût associé à la décision δ pour les paramètres θ.

Le coût est une mesure numérique de la qualité d'une décision.

Définition 1.9 *On appelle estimateur de Bayes associé à une loi a priori π et à un coût C, tout estimateur δ^π qui, étant donné un vecteur d'observation $\boldsymbol{x}$, minimise le coût a posteriori*

$$\rho(\pi, \delta|\boldsymbol{x}) = I\!E^\pi[C(\theta, \delta)|\boldsymbol{x}] = \int_\theta C(\theta, \delta)\pi(\theta|\boldsymbol{x})d\theta.$$

Exemple 1.12 Reprenons l'exemple 1.11 et cherchons un estimateur de Bayes. La première chose à faire consiste à préciser le cadre bayésien de l'analyse :

- on suppose que la loi sur les observations est une binomiale, $X \sim \mathcal{B}(n, p)$;
- on suppose que la boule A peut s'arrêter équiprobablement à n'importe qu'elle distance du bord. D'où $p \sim U[0, 1]$;
- on choisit un coût quadratique : $C(p, \delta) = (p - \delta)^2$.

Dans ce cas,

$$\begin{aligned}\pi(p|X = x) &= \frac{C_n^x p^x (1-p)^{n-x} I\!I_{\{p\in[0,1]\}}}{\int_0^1 C_n^x p^x (1-p)^{n-x} dp} \\ &= \frac{p^x (1-p)^{n-x} I\!I_{\{p\in[0,1]\}}}{\int_0^1 p^x (1-p)^{n-x} dp}.\end{aligned}$$

La loi a posteriori est donc une loi bêta, $\mathcal{B}e(x + 1, n - x + 1)$. Il est facile de montrer que l'estimateur de Bayes associé à une loi π et un coût quadratique est la moyenne a posteriori

$$\delta^\pi(x) = I\!E^\pi[p|x] = \int p \cdot \pi(p|x)dp.$$

L'espérance d'une variable aléatoire X suivant une loi bêta, $\mathcal{B}e(\alpha, \beta)$ est donnée par

$$I\!E[X] = \frac{\alpha}{\alpha + \beta}.$$

L'estimateur de Bayes associé au coût quadratique s'écrit donc

$$\delta^\pi(x) = \frac{x + 1}{n + 2}.$$

Si de nombreux lancés ont été effectués, l'estimation de p par cette procédure bayésienne sera très proche de l'estimateur du maximum de vraisemblance. Par contre si un seul lancé est observé, on trouve :

- soit $x = 0$, ce qui donne $\hat{p} = \frac{1}{3}$;
- soit $x = 1$, ce qui donne $\hat{p} = \frac{2}{3}$.

Ces deux résultats semblent "raisonnables".

Notons qu'en prenant un coût qui vaut 0 si la décision est correcte et 1 sinon (coût "0-1"), l'estimateur de Bayes est, dans ce cas, le même que celui obtenu par la méthode du maximum de vraisemblance.

△

Remarquons que les estimateurs de Bayes sont justifiés pour une taille d'échantillon finie, contrairement aux estimateurs du maximum de vraisemblance qui n'ont que des propriétés asymptotiques. Pour une présentation approfondie de l'analyse statistique bayésienne nous conseillons vivement le livre de Robert (1992).

1.4 Obtention d'une partition

1.4.1 L'algorithme EM et la classification automatique

Le principe d'information manquante

Dans certains problèmes, l'échantillon de données disponible ne permet pas de cacluler facilement les estimateurs du maximum de vraisemblance. C'est par exemple le cas pour l'estimation des paramètres d'un mélange fini de densités de probabilité.

Exemple 1.13 (Redner et Walker 1984) La taille d'un flétan (poisson de la mer baltique) d'un âge donné est distribuée suivant un mélange de deux lois gaussiennes correspondant aux deux distributions relatives aux mâles et femelles :

$$f(\mathbf{x}|\Phi) = p_1 f_1(\mathbf{x}|\mu_1, \sigma_1) + p_2 f_2(\mathbf{x}|\mu_2, \sigma_2) \tag{1.13}$$

où les p_k sont les proportions du mélange ($0 < p_k < 1$, pour $k = 1,2$ et $\sum_k p_k = 1$), $f_k(\mathbf{x}|\mu_k, \sigma_k)$ est une loi de Gauss de moyenne μ_k et d'écart type σ_k, et $\Phi = (p_1, \mu_1, \sigma_1, \mu_2, \sigma_2)$. L'estimation de Φ est un problème simple si les mesures prises spécifient la taille et le sexe de chaque poisson considéré. Malheureusement le sexe du flétan est difficile à déterminer et il faut estimer le vecteur Φ à l'aide de données incomplètes.

△

Pour maximiser la vraisemblance de ce type de données, qualifiées de données incomplètes, il est souvent avantageux de poser le problème pour un jeu hypothétique de données complètes. Cette facon d'aborder le problème conduit à la formulation d'un algorithme itératif qui permet de calculer des estimateurs des paramètres inconnus. Dempster, Laird et Rubin (1977) ont baptisé cet algorithme, basé sur le principe de l'information manquante, algorithme EM (Expectation Maximization), et ont donné des nombreux exemples de son application à des problèmes aussi variés que le calcul

des estimateurs du m.v. des paramètres d'une loi multinomiale, d'un mélange fini de densités ou l'estimation d'hyperparamètres dans un cadre bayésien.

D'une manière très générale deux espaces mesurables sont considérés : $\mathcal{X}$, l'espace des données observées (ou données incomplètes) et $\mathcal{Y}$ l'espace des données complètes. Soient deux vecteurs $\mathbf{x} \in \mathcal{X}$ et $\mathbf{y} \in \mathcal{Y}$, de densité respective $f(\mathbf{x}|\Phi)$ et $g(\mathbf{y}|\Phi)$.

Le but de l'algorithme est de calculer l'estimateur du m.v. du vecteur de paramètres inconnus, Φ, en utilisant les relations qui existent entre $\mathbf{x}$ et $\mathbf{y}$.

En pratique $\mathbf{y}$ n'est pas observé et contient des données manquantes, des paramètres inconnus, des données inobservables (e.g. le sexe des flétans dans l'exemple précédent).

On note $k(\mathbf{y}|\mathbf{x}, \Phi)$ la densité conditionnelle des données complètes connaissant les données observées :

$$f(\mathbf{x}|\Phi) = \frac{g(\mathbf{y}|\Phi)}{k(\mathbf{y}|\mathbf{x}, \Phi)}. \tag{1.14}$$

En prenant le logarithme, on obtient :

$$L(\Phi; \mathbf{x}) = L(\Phi; \mathbf{y}) - L(\Phi; \mathbf{y}|\mathbf{x}), \tag{1.15}$$

où $L(\Phi; \mathbf{y})$ et $L(\Phi; \mathbf{x})$ sont les log-vraisemblances de Φ en considérant respectivement les données complètes et les données observées. De même $L(\Phi; \mathbf{y}|\mathbf{x})$ représente la log-vraisemblance de Φ tenant compte de la densité conditionnelle de $\mathbf{y}$ sachant $\mathbf{x}$.

Considérons Φ_d une valeur donnée du vecteur Φ. En prenant de chaque coté de l'équation 1.15, l'espérance pour la loi $k(\mathbf{y}|\mathbf{x}, \Phi_d)$, on peut écrire :

$$L(\Phi; \mathbf{x}) = Q(\Phi|\Phi_d) - H(\Phi|\Phi_d), \tag{1.16}$$

où

$$Q(\Phi|\Phi_d) = I\!E^k[L(\Phi; \mathbf{y})|\mathbf{x}, \Phi_d];$$
$$H(\Phi|\Phi_d) = I\!E^k[L(\Phi; \mathbf{y}|\mathbf{x})|\mathbf{x}, \Phi_d].$$

Notons que l'inégalité de Jensen (Dempster *et al.* 1977) permet de montrer que la valeur de Φ, qui maximise $H(\Phi|\Phi_d)$, est Φ_d. La valeur Φ^+ de Φ qui maximise $Q(\Phi|\Phi_d)$ est une fonction de Φ_d :

$$\Phi^+ = M(\Phi_d). \tag{1.17}$$

Soit Φ^* le maximum de vraisemblance cherché. Si l'on pose

$$\Phi_d = \Phi^*$$

il est alors évident que la valeur Φ^* maximise

$$L(\Phi; \mathbf{x}) + H(\Phi|\Phi^*).$$

De cette constatation, on déduit que Φ^* maximise $Q(\Phi|\Phi^*)$. Ainsi, Φ^* est un point fixe de la fonction $M(\Phi)$, et ceci suggère un algorithme itératif de type point fixe qui calcule le paramètre Φ^{q+1} à partir d'une valeur Φ^q :

- **Etape d'Estimation :** Déterminer $Q(\Phi|\Phi^q) = I\!E^k[L(\Phi; \mathbf{y})|\mathbf{x}, \Phi^q]$

– **Etape de Maximisation :** Calculer $\Phi^{q+1} = M(\Phi^q)$. Φ^{q+1} vérifie alors

$$\Phi^{q+1} = arg\max_{\Phi} Q(\Phi|\Phi^q)$$

La propriété fondamentale de l'algorithme EM est que chaque itération augmente la vraisemblance des paramètres à estimer. En effet, suite à l'étape de maximisation on a

$$Q(\Phi^{q+1}|\Phi^q) \geq Q(\Phi^q|\Phi^q)$$

et d'après l'inégalité de Jensen (Dempster *et al.* 1977) :

$$H(\Phi^{q+1}|\Phi^q) \leq H(\Phi^q|\Phi^q),$$

donc

$$L(\Phi^{q+1};\mathbf{x}) \geq L(\Phi^q;\mathbf{x}).$$

Dans un cadre général la convergence de l'algorithme n'est pas démontrée (la démonstration de Dempster, Laird et Rubin en 1977 était fausse) et si l'algorithme converge vers un point fixe, on est seulement sûr que c'est un point stationnaire de la vraisemblance et pas obligatoirement un maximum local, mais dans le cadre de l'estimation des paramètres d'un mélange fini, qui nous intéresse particulièrement, Redner et Walker (1984) ont démontré le théorème de convergence locale suivant :

Théorème 1.2 *(Redner et Walker 1984) Soit un mélange de densités exponentielles, supposons que $I(\Phi)$, la matrice d'information de Fisher associée aux paramètres du mélange est définie positive pour Φ^* les vraies valeurs des paramètres, si les proportions sont positives, alors pour n suffisamment grand, l'unique solution presque sûrement consistante Φ_n des équations de vraisemblance existe presque sûrement, et la suite $\{\Phi^q\}$ des itérés de l'algorithme EM converge vers Φ_n pourvu que la position initiale Φ^0 soit suffisamment proche de Φ_n; de plus il existe une norme sur l'espace des paramètres pour laquelle il existe λ, $0 \leq \lambda < 1$, pour laquelle :*

$$\|\Phi^{q+1} - \Phi_n\| \leq \lambda\|\Phi^q - \Phi_n\|, \forall q \geq 0.$$

D'après ce théorème, et avec un peu de pratique, on s'aperçoit que l'initialisation de l'algorithme conditionne la qualité du résultat. Si la position initiale choisie est très "éloignée" de la vraie valeur des paramètres, l'algorithme EM risque de converger vers une solution singulière.

L'algorithme EM converge linéairement et dans certaines situations peut s'avérer particulièrement lent. Ainsi, lorsque les composants du mélange sont mal séparés, le coefficient λ sera proche de 1 et un grand nombre d'itérations sera nécessaire à la convergence.

Pour pallier ce problème de vitesse de convergence, Redner et Walker (1984) ont suggéré l'utilisation de méthodes d'optimisation qui ont une meilleure vitesse de

convergence comme celle de Newton. La méthode de Newton est itérative. A partir d'une position initiale Φ^0, une suite d'itérés est calculée comme suit :

$$\Phi^{q+1} = \Phi^q - H(\Phi^q)^{-1}\nabla_\Phi L(\Phi^q; \mathbf{x}), \tag{1.18}$$

où $H(\Phi^q)$ est la matrice hessienne de $L(\Phi_q; \mathbf{x})$. Cette méthode a une vitesse de convergence quadratique ; c'est-à-dire qu'il existe une constante λ, telle que :

$$\|\Phi^{q+1} - \Phi_n\| \leq \lambda \|\Phi^q - \Phi_n\|^2.$$

La convergence quadratique est beaucoup plus rapide que la convergence linéaire mais le calcul de l'inverse de la matrice hessienne est très coûteux. Une autre méthode possible est celle de quasi Newton, qui approxime la matrice hessienne et réduit ainsi la complexité algorithmique de la méthode de Newton tout en ayant une convergence supra-linéaire, donc supérieure à celle de l'algorithme EM.

Malgré les qualités des méthodes de Newton, l'algorithme EM reste très utilisé pour plusieurs raisons. En effet, chaque itération nécessite peu de calculs et même dans les cas où la convergence vers les vraies valeurs des paramètres est lente, la convergence de la vraisemblance reste très rapide (Redner et Walker 1984). Ainsi les premières itérations produisent des bonnes valeurs des paramètres et les nombreuses autres augmentent peu la vraisemblance. Xu et Jordan (1995) remarquent :

> In the context of the current litterature on learning, in which the predictive aspects of data modeling is emphasized at the expense of the traditonal Fisherian statistician's concern over the "true" value of the parameters, such rapid convergence in likelihood is a major desiratum of learning algorithm and undercuts the critique of EM as a "slow" algorithm.

Lorsque la convergence de l'algorithme EM est lente (composantes du mélange mal séparées), les matrices Hessiennes sont mal conditionnées et les méthodes superlinéaires et de quadratiques ont aussi des problèmes. De plus dans le cas des modèles de mélange gaussien, l'algorithme EM peut être considéré comme une montée de gradient projeté (Xu et Jordan 1995) ; les deux étapes de l'algorithme se résument à l'équation suivante :

$$\Phi^{q+1} = \Phi^q - P(\Phi^q)\nabla_\Phi L(\Phi^q; \mathbf{x}) \tag{1.19}$$

où $P(\Phi^q)$ est une matrice de projection calculée à chaque itération. Il est alors possible de montrer que sous certaines conditions l'algorithme EM approxime une méthode superlinéaire.

Application au modèle de mélange

Pour toutes les raisons mentionnées précédemment, l'algorithme EM est très utilisé pour l'estimation des paramètres d'un modèle de mélange de densité de probabilité. Dans ce contexte, son utilisation est bien antérieure à l'article de Dempster, Laird

et Rubin (1977) : Day (1969) proposait déjà un algorithme identique pour identifier les paramètres d'un mélange de deux gaussiennes multidimensionnelles. Wolfe (1970), de manière indépendante, décrivait un algorithme de classification automatique probabiliste destiné à l'estimation des paramètres de mélanges de K lois de Bernoulli, ou de Gauss multivariées. Cet article remarquable introduisait ainsi l'algorithme EM pour obtenir une partition floue, alors que la notion de flou en classification ne se développa qu'à partir de 1974 principalement sous l'impulsion de Bezdek (1974).

Dans le cadre d'un modèle de mélange, le problème d'estimation des paramètres se pose comme suit : on dispose d'un échantillon $(\mathbf{x}_1, ..., \mathbf{x}_N)$ d'une variable aléatoire à valeurs dans $\mathbb{R}^d$ de densité :

$$f(\mathbf{x}_i|\Phi) = \sum_{k=1}^{K} p_k f_k(\mathbf{x}_i|\theta_k), \tag{1.20}$$

où les p_k sont les proportions du mélange ($0 < p_k < 1$, pour $k = 1, .., K$ et $\sum_k p_k = 1$) et $f_k(\mathbf{x}|\theta_k)$ est une loi complètement déterminée par la connaissance du vecteur θ_k.

Posons le problème de l'estimation de $\Phi = (p_1, ..., p_k, \theta_1, ..., \theta_K)$ sous une forme traitable par le principe d'information manquante. Considérons que l'échantillon observé $\mathbf{x} = (\mathbf{x}_1, ..., \mathbf{x}_N)$ est incomplet. L'échantillon complet s'écrit $\mathbf{y} = (\mathbf{y}_1, ..., \mathbf{y}_N)$ avec $\mathbf{y}_i = (\mathbf{x}_i, \mathbf{z}_i)$. $\mathbf{z}_i = (z_{ik}, k = 1, ..., K)$ est un vecteur qui indique de quelle composante du mélange est issu $\mathbf{x}_i$ ($z_{ik} \in \{0, 1\}$ et $\sum_{k=1}^{K} z_{ik} = 1$) : $z_{ik} = 1$ signifie que $\mathbf{x}_i$ provient de la k^{e} composante. Indiçons les paramètres à estimer par $\mathbf{z}_i$ à la place de k lorsque $z_{ik} = 1$ et écrivons les densités des deux échantillons $\mathbf{x}$ et $\mathbf{y}$:

$$f(\mathbf{x}|\Phi) = \prod_{i=1}^{N} f(\mathbf{x}_i|\Phi) = \prod_{i=1}^{N} \sum_{k=1}^{K} p_k f_k(\mathbf{x}_i|\theta_k), \tag{1.21}$$

et

$$g(\mathbf{y}|\Phi) = \prod_{i=1}^{N} p_{\mathbf{z}_i} f_{\mathbf{z}_i}(\mathbf{x}_i|\theta_{\mathbf{z}_i}). \tag{1.22}$$

$k(\mathbf{y}|\mathbf{x}, \Phi)$ la densité conditionnelle des données complètes connaissant les données observées s'exprime par :

$$k(\mathbf{y}|\mathbf{x}, \Phi) = \prod_{i=1}^{N} k(\mathbf{y}_i|\mathbf{x}_i; \Phi) = \prod_{i=1}^{N} \frac{p_{\mathbf{z}_i} f_{\mathbf{z}_i}(\mathbf{x}_i|\theta_{\mathbf{z}_i})}{\sum_{k=1}^{K} p_k f_k(\mathbf{x}_i|\theta_k)}. \tag{1.23}$$

Ainsi dans le cas particulier des modèles de mélanges les quantités Q et H de l'équation 1.16 deviennent :

$$\begin{aligned} Q(\Phi|\Phi^q) &= \mathbb{E}^k[L(\Phi; \mathbf{y})|\mathbf{x}, \Phi^q] = \mathbb{E}^k[\log \prod_{i=1}^{N} p_{\mathbf{z}_i} f_{\mathbf{z}_i}(\mathbf{x}_i|\theta_{\mathbf{z}_i})|\mathbf{x}, \Phi^q] \\ &= \sum_{i=1}^{N} \mathbb{E}^k[\log p_{\mathbf{z}_i} f_{\mathbf{z}_i}(\mathbf{x}_i|\theta_{\mathbf{z}_i})|\mathbf{x}, \Phi^q] \end{aligned}$$

$$= \sum_{i=1}^{N}\sum_{k=1}^{K} t_k(\mathbf{x}_i)^q \log p_k f_k(\mathbf{x}_i|\theta k)$$

$$H(\Phi|\Phi^q) = I\!E^k[L(\Phi;\mathbf{y}|\mathbf{x})|\mathbf{x},\Phi_d] = I\!E^k[\log \prod_{i=1}^{N} k(\mathbf{y}_i|\mathbf{x}_i;\Phi)|\mathbf{x},\Phi^q]$$

$$= \sum_{i=1}^{N} I\!E^k[\log k(\mathbf{y}_i|\mathbf{x}_i;\Phi)|\mathbf{x},\Phi^q]$$

$$= \sum_{i=1}^{N}\sum_{k=1}^{K} t_k(\mathbf{x}_i)^q \log t_k(\mathbf{x}_i)$$

avec

$$t_k(\mathbf{x}_i)^q = \frac{p_k^q f_k(\mathbf{x}_i|\theta_k^q)}{f(\mathbf{x}_i)}. \tag{1.24}$$

Dans la suite de cette section, nous considérons uniquement le cas des mélanges gaussiens qui sont de loin les plus utilisés en classification automatique. Le mélange est alors paramétré par le vecteur $\Phi^q = (p_1^q, ..., p_{K-1}^q, \theta_1^q, ..., \theta_K^q)$ où $\theta_k = (\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ et les deux étapes de l'algorithme EM s'écrivent :

- **Etape E :** Calcul des probabilités $t_k(\mathbf{x}_i)^q$ en utilisant $\Phi^q = (p_1^q, ..., p_k^q, \theta_1^q, ..., \theta_K^q)$.
- **Etape M :** Calcul de Φ^{q+1} qui maximise

$$Q(\Phi|\Phi^q) = \sum_{i=1}^{N}\sum_{k=1}^{K} t_k(\mathbf{x}_i)^q \log p_k f_k(\mathbf{x}_i|\theta_k).$$

Les estimateurs du maximum de vraisemblance s'écrivent alors :

$$\boldsymbol{\mu}_k^{q+1} = \frac{\sum_{i=1}^{N} t_k(\mathbf{x}_i)^q \cdot \mathbf{x}_i}{n_k^q}; \tag{1.25}$$

$$\boldsymbol{\Sigma}_k^{q+1} = \frac{1}{n_k}\sum_{i=1}^{N} t_k(\mathbf{x}_i)^q (\mathbf{x}_i - \boldsymbol{\mu}_k^{q+1})(\mathbf{x}_i - \boldsymbol{\mu}_k^{q+1})^t; \tag{1.26}$$

$$p_k^{q+1} = \frac{n_k^q}{N}, \tag{1.27}$$

où $n_k^q = \sum_{i=1}^{N} t_k(\mathbf{x}_i)^q$.

A la convergence, l'algorithme EM fournit une estimation des paramètres du mélange et aussi des probabilités $k((\mathbf{x}_i, \mathbf{z}_i)|\mathbf{x}_i)$. Dans une optique classificatoire, $k((\mathbf{x}_i, \mathbf{z}_i = k)|\mathbf{x}_i)$ est interprété comme la probabilité a posteriori que $\mathbf{x}_i$ appartienne à la classe k. Ces probabilités permettent de calculer une partition dure de l'échantillon $\mathbf{x}$, en affectant chaque $\mathbf{x}_i$ à la classe la plus probable a posteriori (Principe du Maximum A Posteriori, MAP). Notons que dans le cadre de la théorie bayésienne de la décision,

cette stratégie revient à considérer chaque individu séparément et prendre la décision "δ est la classe de x_i" qui minimise le coût a posteriori :

$$\rho(\delta, t_{\mathbf{Z}_i}|\mathbf{x}_i) = I\!E^k[I\!I_{\{\delta \neq \mathbf{Z}_i\}}|\mathbf{x}_i, \Phi] = \sum_{k=1}^{K} t_k(\mathbf{x}_i) I\!I_{\{\delta \neq k\}} = 1 - t_\delta(\mathbf{x}_i). \tag{1.28}$$

Exemple 1.14 Pour illustrer les perforances de l'algorithme EM, nous avons défini un mélange de trois gaussiennes bidimensionnelles (Figure 1.4a). A partir de ce modèle, un échantillon de 300 individus a été tiré au hasard (Figure 1.4b). En utilisant l'algorithme EM pour estimer les paramètres du mélange à partir des 300 individus disponibles, on a obtenu la figure 1.4c. La figure 1.4d montre la partition obtenue par le principe de MAP, en utilisant les paramètres trouvés par l'algorithme EM, des 300 individus. On peut calculer que si les individus avaient été classés par le principe du MAP en considérant les paramètres réels du mélange, 16 % des individus auraient été classés différemment.

△

Dans un contexte de classification automatique, le modèle gaussien estimé par l'algorithme précédent est le plus général possible, dans le sens où la seule hypothèse posée est le nombre de classes K. Dans des espaces de grandes dimensions et lorsqu'il existe peu d'observations, le nombre de paramètres à estimer d'un modèle général est trop important pour la quantité d'information disponible. De plus d'après Dang (1994) :

> quand les distributions des classes sont simples, adopter un modèle plus complexe que le modèle sous-jacent introduit des degrés de liberté artificiels, et conduit à des estimations faussées.

Dans certaines applications, l'analyste peut avoir des a priori sur la structure de classe du jeu de données et réduire ainsi le nombre de paramètres à estimer en en fixant certains selon ses a priori.

Exemple 1.15 Reprenons l'exemple de la répartition de la taille de flétans (exemple 1.13). Une hypothèse simplificatrice possible consiste à supposer qu'il y a autant de mâles que de femelles. Dans ce cas les deux classes ont approximativement la même taille et les proportions du mélange peuvent être fixées à $\frac{1}{2}$. La distribution à estimer

$$f(\mathbf{x}|\Phi) = \frac{1}{2}\{f_1(\mathbf{x}|\mu_1, \sigma_1) + f_2(\mathbf{x}|\mu_2, \sigma_2)\} \tag{1.29}$$

possède alors un paramètre en moins (p_1).

△

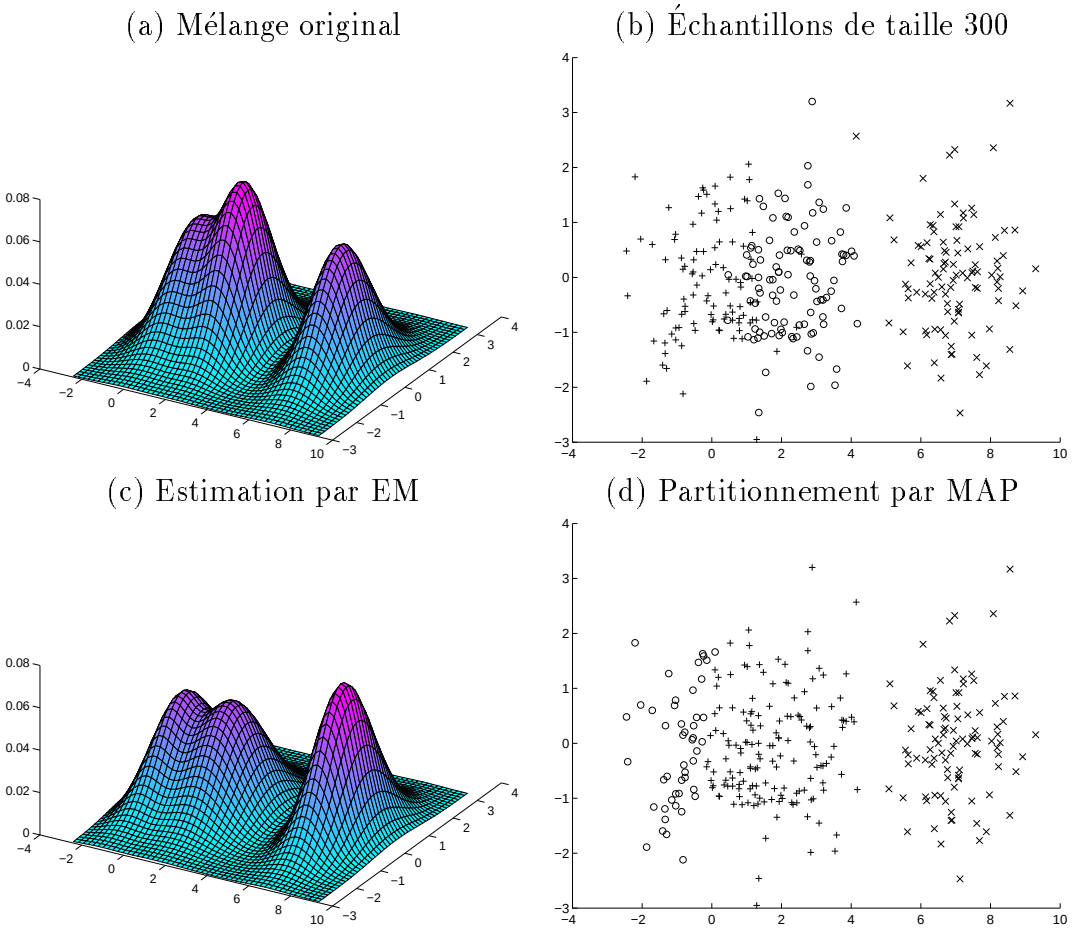
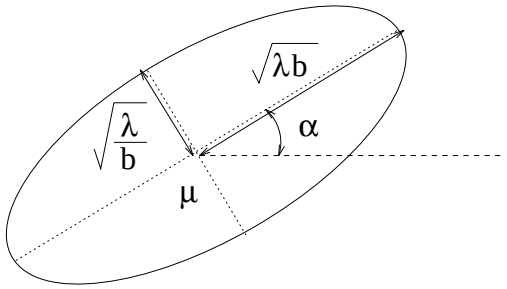


FIG. 1.4 - : Partitionnement d'un jeu de données simulées (b) en utilisant l'algorithme EM et le principe du Maximum a Posteriori : 16 % de mal classés.



$$B = \begin{bmatrix} b & 0 \\ 0 & 1/b \end{bmatrix} \qquad D = \begin{bmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{bmatrix}$$

FIG. 1.5 - : Paramétrisation de la matrice de variance dans le cas bidimensionnel

La paramétrisation de la matrice de variance, proposée par Banfield et Raftery (1993) puis utilisée en relation avec l'algorithme EM par Celeux et Govaert (1995), met en évidence des paramètres intuitifs qui facilitent la traduction d'a priori dans un langage plus formel (Figure 1.5). Cette méthode consiste à décomposer la matrice de variance dans sa base de vecteurs propres :

$$\boldsymbol{\Sigma}_k = \lambda_k \cdot \boldsymbol{D}_k \cdot \boldsymbol{B}_k \cdot \boldsymbol{D}_k^t \tag{1.30}$$

où

- $\lambda_k = \det |\boldsymbol{\Sigma}_k|^{\frac{1}{d}}$ est interprété comme le volume de la k^{e} classe. En effet plus λ_k est grand plus la classe occupera une place importante dans l'espace $\mathbb{R}^d$. Cette notion ne doit pas être confondue avec le nombre d'individus de la classe qui est relatif à la proportion p_k; ce n'est pas parce qu'une classe occupe un grand volume qu'elle contient forcément beaucoup d'individus.
- $\boldsymbol{B}_k$ est la matrice diagonale des valeurs propres. Elle caractérise la forme de la classe k. Plus une valeur propre est importante plus l'envoloppe de la classe est "allongée" dans la direction du vecteur propre correspondant.
- $\boldsymbol{D}_k$ est la matrice des vecteurs propres. Elle représente l'orientation de la classe k. C'est une matrice orthogonale de changement de base. Par rapport aux axes de référence, la base de vecteurs propres est obtenue par rotation.

Liens avec la classification floue

Dans le cadre de la reconnaissance des formes, l'algorithme EM pour les modèles de mélanges peut être interprété comme un algorithme d'optimisation alternée d'un certain critère (Hathaway 1986, Celeux et Govaert 1994).

Si les probabilités a posteriori $t_k(\mathbf{x}_i)$ sont considérées comme des variables notées c_{ik}, la log-vraisemblance $L(\Phi; \mathbf{x})$ devient une fonction du vecteur Φ et des c_{ik} que nous noterons :

$$L(\mathbf{c}, \Phi) = \sum_{i=1}^{N} \sum_{k=1}^{K} c_{ik} \log p_k f_k(\mathbf{x}_i|\theta_k) - \sum_{i=1}^{N} \sum_{k=1}^{K} c_{ik} \log c_{ik}, \tag{1.31}$$

avec $\mathbf{c} = \{c_{ik} : 0 \leq c_{ik} \leq 1, \sum_{k=1}^{K} c_{ik} = 1, \sum_{i=1}^{N} c_{ik} > 0 (1 \leq i \leq N, 1 \leq k \leq K)\}$.

Considérons le problème qui consiste à maximiser $L(\mathbf{c}, \Phi)$ par rapport aux variables Φ et $\mathbf{c}$. Il s'agit d'un problème classique d'optimisation sous contraintes. Une méthode d'optimisation possible consiste à séparer les variables en deux groupes et à optimiser le critère alternativement par rapport à un groupe en gardant fixe les valeurs des variable de l'autre groupe. Dans le cas du critère $L(\mathbf{c}, \Phi)$, pour la q^{e} itération il est possible d'optimiser alternativement par rapport à $\mathbf{c}$ puis à Φ :

1. Maximisons $L(\mathbf{c}, \Phi)$ par rapport à $\mathbf{c}$: le lagrangien s'écrit

$$\mathcal{L}(\mathbf{c}) = L(\mathbf{c}, \Phi) + \sum_{i=1}^{N} \lambda_i (\sum_{k=1}^{K} (c_{ik} - 1)), \tag{1.32}$$

où les λ_i sont les coefficients de Lagrange correspondant aux contraintes

$$\sum_{k=1}^{K} c_{ik} = 1.$$

Les conditions nécessaires d'optimalité amènent les équations suivantes :

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial c_{ik}} = \log\left(p_k f_k(\mathbf{x}_i|\theta_k)\right) - 1 - \log c_{ik} + \lambda_i = 0; \\ \sum_{k=1}^{K} c_{ik} = 1; \end{cases}$$

ce qui donne,

$$\begin{cases} c_{ik} = \exp\left\{\log(p_k f_k(\mathbf{x}_i|\theta_k)) - 1 + \lambda_i\right\}; \\ \sum_{k=1}^{K} \exp\left\{\log(p_k f_k(\mathbf{x}_i|\theta_k)) - 1 + \lambda_i\right\} = 1; \end{cases}$$

Ainsi les nouvelles valeurs des c_{ik} sont :

$$c_{ik} = \frac{p_k f_k(\mathbf{x}_i|\theta_k)}{f(\mathbf{x}_i)}. \tag{1.33}$$

2. La maximisation de $L(\mathbf{c}, \Phi)$ par rapport à Φ est équivalente l'étape M de l'algorithme EM.

Ces deux étapes qui visent à maximiser le critère $L(\mathbf{c}, \Phi)$ sont identiques aux deux étapes de l'algorithme EM appliqué à un mélange de distribution de probabilité.

Si l'on considère $\mathbf{c}$ comme une matrice de classification floue (elle en a toutes les caractéristiques), l'algorithme EM peut être interprété comme un algorithme de classification floue.

Remarquons que le critère optimisé s'écrit comme la somme de deux termes :

- Dans la terminologie utilisée en classication automatique, le premier est appelé "vraisemblance classifiante floue" (avec proportions libres). Plusieurs algorithmes de classification automatique existent qui visent à trouver la partition dure qui optimise la vraisemblance classifiante.

- Le second terme peut être considéré comme une entropie, ou encore une mesure de floue de la partition. Ce second terme est maximum si la partition obtenue est complètement floue et minimum (nul en l'occurence) $\mathbf{c}$ est une matrice de partition dure.

Au vu des remarques précédentes, l'algorithme EM peut être considéré comme un algorithme de classification flou qui optimise un critère de classification pénalisé par une entropie.

L'algorithme EM et la mécanique statistique

L'algorithme EM peut aussi être interprété dans le cadre de la mécanique statistique. Un avantage de cette interprétation est qu'elle suggère l'utilisation du recuit simulé pour améliorer les résultats de l'algorithme.

En mécanique statistique un système isolé décrit par une variable aléatoire ω possède une énergie $E(\omega)$. Cette variable aléatoire ω suit une distribution de Gibbs :

$$\pi(\omega) = \frac{e^{-E(\omega)/kT}}{Z(T)}, \tag{1.34}$$

où T est la température, k une constante, et Z la fonction de partition :

$$Z = \sum_{\omega \in \Omega} e^{-E(\omega)/kT}. \tag{1.35}$$

Il existe une relation entre $\overline{E}$ l'énergie moyenne du système, F son énergie libre (dite énergie de Helmotz) et S l'entropie :

$$\frac{F}{T} = \frac{\overline{E}}{T} - S = -k \log Z. \tag{1.36}$$

Remarquons que lorsque la température du système tend vers zéro, l'énergie libre du système tend vers l'énergie moyenne totale.

Suivons Rose *et al.* (1990) et mettons en parallèle le problème de partitionnement de N individus en K classes, et la recherche de la configuration d'entropie maximale (ou d'énergie libre minimale) d'un système donné en mécanique statistique :

Soit un système composé de N individus $(\mathbf{x}_1, ..., \mathbf{x}_N)$. Si un individu appartient à la classe m, il possède une énergie $E_{\mathbf{X}_i}(m)$. La probabilité que $\mathbf{x}_i$ appartienne à la classe m est définie par une distribution de Gibbs :

$$P(\mathbf{x}_i \in C_m) = \frac{\exp(-E_{\mathbf{X}_i}(m)/kT)}{Z_{\mathbf{X}_i}}$$

L'énergie totale du système est :

$$E = \sum_{i=1}^{N} \sum_{m=1}^{K} c_{im} E_{\mathbf{X}_i}(m), \tag{1.37}$$

avec $\mathbf{c} = \{c_{im}\}_{i=1,...,N;m=1,...,K}$ une matrice de classification dure. Il est possible de montrer que l'énergie moyenne du système est

$$\overline{E} = \sum_{i=1}^{N} \sum_{m=1}^{K} P(\mathbf{x}_i \in C_m) E_{\mathbf{X}_i}(m),$$

et l'énergie libre,

$$F = -kT \cdot \sum_{i=1}^{N} \log \sum_{m=1}^{K} exp(-E_{\mathbf{X}_i}(m)/kT).$$

Dans leur article Rose *et al.* (1990) utilisent une énergie quadratique de la forme :

$$E_{\mathbf{X}_i}(m) = |\mathbf{x}_i - \boldsymbol{\mu}_m|^2 \tag{1.38}$$

où les $\boldsymbol{\mu}_m$ sont les prototypes des classes. Pour trouver la configuration d'énergie libre maximum, les auteurs proposent un algorithme itératif de recuit déterministe :

1. Initialiser $1/kT = 0$ ($T = \infty$).
2. Initialiser $\Phi = (\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_K)$, les centres des classes.
3. Itérer jusqu'à la convergence, en optimisant successivement F par rapport à $\Phi = (\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_K)$, puis F par rapport à la matrice de classification floue, $\mathbf{c}_{floue} = \{P(\mathbf{x}_i \in C_m)\}_{i=1,\ldots,N;m=1,\ldots,K}$.
4. Diminuer la température
5. Si $T > Tmin$ alors recommencer à partir de l'étape 3.

Cet algorithme s'interprète dans le cadre probabiliste des modèles de mélange gaussien. En effet, l'étape trois (recherche de l'énergie moyenne), revient à optimiser la vraisemblance d'un mélange de K gaussiennes, mélangées dans des proportions identiques et de matrice de variance-covariance commune, $\boldsymbol{\Sigma}_m = kT \cdot I$. De même la diminution de la température correspond à imposer une réduction du volume commun des classes. Ce parallèle n'est évidemment valable que parce que les auteurs ont choisi une forme quadratique pour l'énergie.

En utilisant une terminologie issue de la classification automatique, on peut aussi ajouter que l'énergie moyenne du système précédent correspond à la vraisemblance classifiante floue. Ainsi la recherche d'une partition avec l'algorithme EM peut s'interpréter comme un problème d'approximation de l'énergie moyenne ("mean field approximation") d'un certain système en mécanique statistique. Cette relation a été soulignée par plusieurs auteurs (Yuille 1990, Yuille *et al.* 1993, Simic 1990).

1.4.2 Les algorithmes SEM, SAEM, CEM et CAEM

En ce qui concerne l'estimation des paramètres d'un modèle de mélange, et donc indirectement l'obtention d'une partition, l'algorithme EM donne des résultats satisfaisants (Celeux 1992) si :

- le nombre de composants du mélange est connu,
- les proportions du mélange ne sont pas trop différentes les unes des autres,
- la position initiale Φ^0 n'est pas trop loin d'un optimum local.

Le principe d'affectation stochastique

Pour dépasser ces limitations, Celeux et Diebolt (1986) ont proposé une version stochastique de EM, baptisée *Stochastic EM algorithm*, qui introduit une étape intermédiaire de tirage aléatoire de l'information manquante. Les quantités manquantes (composante d'origine de chaque observation) sont tirées au hasard suivant la loi d'appartenance a posteriori. Une autre particularité de l'algorithme SEM est la procédure qui consiste en la détermination automatique de K, le nombre de composantes du mélange.

Une itération prend alors la forme suivante :

- **Etape E (estimation) :** Calcul des probabilités $t_k(\mathbf{x}_i)^q$ pour chaque $\mathbf{x}_i$.
- **Etape S (stochastique) :** En chaque $\mathbf{x}_i$, tirage au hasard de $\mathbf{z}_i = (z_{ik}, k = 1, ..., K)$, avec $z_{ik} = 1$ si $\mathbf{x}_i$ appartient à la k^{e} classe et $z_{ik} = 0$ sinon, suivant la loi multinomiale de paramètres $(t_1(\mathbf{x}_i)^q, \cdots, t_K(\mathbf{x}_i)^q)$. La matrice $\mathbf{z} = \{z_{ik}; i = 1, \cdots, n \text{ et } k = 1, \cdots, K\}$ définit une partition dure de l'échantillon $\mathbf{x}$.

  Si l'effectif d'une classe est plus petit que le nombre de variables alors l'algorithme est réinitialisé sur la base de $(K-1)$ classes.
- **Etape M (maximisation) :** Calcul des estimateurs du m.v. des paramètres du mélange sur la base des sous-échantillons mis en évidence par la matrice $\mathbf{z}$:

$$\boldsymbol{\mu}_k^{q+1} = \frac{\sum_{i=1}^N z_{ik} \cdot \mathbf{x}_i}{n_k^q}; \tag{1.39}$$

$$\boldsymbol{\Sigma}_k^{q+1} = \frac{1}{{n_k}^q} \sum_{i=1}^N z_{ik} (\mathbf{x}_i - \boldsymbol{\mu}_k^{q+1})(\mathbf{x}_i - \boldsymbol{\mu}_k^{q+1})^t; \tag{1.40}$$

$$p_k^{q+1} = \frac{n_k^q}{N}. \tag{1.41}$$

où $n_k^q = \sum_{i=1}^N z_{ik}$

La suite de paramètres du mélange $\{\Phi^q\}$ générée par l'algorithme SEM converge en loi vers une distribution de probabilité stationnaire. De plus, sous certaines conditions de régularité, on montre que l'écart entre les valeurs de la suite et l'unique solution convergente des équations de vraisemblance suit asymptotiquement (lorsque la taille de l'échantillon $\mathbf{x}$ tend vers l'infini) une loi normale de moyenne nulle (Celeux et Diebolt 1986).

En pratique l'algorithme SEM permet d'estimer correctement le nombre de composantes du mélange et surtout il ne reste pas bloqué sur un col de vraisemblance car il génère une suite statistique et pas un unique résultat. C'est donc une méthode qui, en théorie ne dépend pas des conditions initiales.

Empiriquement Celeux et Diebolt ont montré que l'algorithme SEM était moins performant avec des petits échantillons (moins d'une vingtaine d'individus par classes) que l'algorithme EM, car dans ce cas particulier, "les perturbations aléatoires de l'étape S prennent trop d'importance".

Une version de type recuit simulé de l'algorithme EM

En 1990, Celeux et Diebolt ont proposé une version de type recuit simulé de l'algorithme EM (Celeux et Diebolt 1990): *Simulated Annealing EM algorithm*. Cette version utilise le principe d'affectation aléatoire de l'algorithme SEM mais propose une estimation ponctuelle des paramètres tout comme EM. L'idée de base consiste à profiter des perturbations aléatoires pour éviter de stagner autour d'un col de vraisemblance, et de réduire l'importance des perturbations pour finir en fait par des itérations du type EM. L'importance de la composante stochastique décroit donc avec le nombre d'itérations et est contrôlée par un paramètre de température. A chaque itération les paramètres du mélange sont estimés par une quantité qui est une combinaison convexe des estimateurs produits par SEM et EM.

Au départ les probabilités d'appartenance des observations aux classes sont tirées au hasard, une température initiale, γ^0, ainsi que le nombre de composantes du mélange sont fixés, puis on itère jusqu'à la convergence. Une itération se décompose ainsi :

- **Etape E (estimation) :** Calcul des probabilités $t_k(\mathbf{x}_i)^q$ pour chaque $\mathbf{x}_i$.
- **Etape S (stochastique) :** Tirage stochastique de la matrice $\mathbf{z} = \{z_{ik};\ i = 1, \cdots, n \text{ et } k = 1, \cdots, K\}$ qui définit une partition dure de l'échantillon $\mathbf{x}$.

  Si l'effectif d'une classe est plus petit que le nombre de variables alors l'algorithme est réinitialisé sur la base de $K - 1$ classes.
- **Etape A (annealing) :** Calcul des quantités $c_{ik} = t_k(\mathbf{x}_i)^q + \gamma^q(z_{ik} - t_k(\mathbf{x}_i)^q)$, pour $i = 1, \cdots, n$ et $k = 1, \cdots, K$.

  Une nouvelle température γ^{q+1} est calculée.
- **Etape M (maximisation) :** Calcul des estimateurs du m.v. de Φ^{q+1} en considérant artificellement les c_{ik} comme les probabilités a posteriori d'appartenance de $\mathbf{x}_i$ à la classe k.

La suite des températures $\{\gamma^q\}$ converge vers 0. La façon de faire décroître la température est importante pour le comportement pratique et théorique de l'algorithme SAEM. Par exemple, on pourra opter pour une décroissance lente avec une décroissance du type

$$\gamma^q = \cos \frac{q\pi}{2q_{max}},$$

ou bien une décroissance linéaire plus rapide avec

$$\gamma^q = \frac{q_{max}+1}{q_{max}} - \frac{q+1}{q_{max}},$$

avec q_{max} le nombre d'itérations maximum caractérisé par $\gamma^{q_{max}} = 0$.

Pour un modèle simplifié de l'algorithme SAEM, Celeux et Diebolt (1990) ont montré que l'algorithme converge presque sûrement vers un maximum local de la vraisemblance.

En pratique, sur des mélanges gaussiens, l'algorithme SAEM avec un mode de décroissance lent permet de retrouver le nombre de classes du modèle et gère bien les problèmes liés aux échantillons de petite taille. Il évite mieux que EM les solutions singulières mais nécessite beaucoup plus d'itérations. Un nombre d'itérations classique pour l'algorithme SAEM est 200 alors que pour de nombreux problèmes l'algorithme EM converge en moins de 10 itérations.

Approche classification et algorithme EM

Nous avons vu que le problème statistique de l'estimation des paramètres d'un mélange fini de densités de probabilités permet indirectement d'obtenir une partition dure de l'échantillon en utilisant le principe du MAP (maximum a posteriori). Une autre approche possible, dans une optique de partitionnement de l'échantillon $\mathbf{x}$, consiste à considérer directement la partition comme le paramètre inconnu. Les pionniers de cette approche sont Scott et Symons (1971) et Schroeder (1976). Dans ce contexte le problème à résoudre peut être formulé comme suit : étant donné un échantillon de taille N, $\mathbf{x} = (\mathbf{x}_1, ..., \mathbf{x}_N)$, rechercher une partition dure $P = (P_1, \cdots, P_K)$, K étant supposé connu, telle que chaque classe P_k soit assimilable à un sous-échantillon suivant la loi $f_k(.|\theta_k)$.

Le critère considéré alors n'est plus la vraisemblance de l'échantillon, mais la vraisemblance classifiante,

$$CML(P, \Phi) = \sum_{k=1}^{K} \sum_{i=1}^{n} c_{ik} \log \{f_k(\mathbf{x}_i|\theta_k\} \tag{1.42}$$

avec $\Phi = (\theta_1, \cdots, \theta_K)$ et $\mathbf{c} = \{c_{ik}\}$ une matrice de partition dure qui définit K classes (ou sous échantillons). Ce critère est la log-vraisemblance associée à K échantillons séparés de taille fixée.

La vraisemblance classifiante ne fait pas apparaître explicitement la notion de proportions entre les différentes sous-populations et tend en pratique à produire des partitions où les classes sont de tailles comparables. En fait le critère suppose implicitement que toute les sous-populations sont de même taille. Cette limitation a incité Symons (1981) à pénaliser la vraisemblance classifiante avec un terme prenant en compte les proportions $(p_1, \cdots, p_K)$ des différents sous-échatillons :

$$CML'(P, \Phi') = CML(P, \Phi) + \sum_{k=1}^{K} n_k \log p_k \tag{1.43}$$

où $\Phi' = (\Phi, p_1, \cdots, p_K)$ et n_k est l'effectif de la k^{e} classe. Notons qu'en introduisant les variables c_{ik} dans le terme de pénalité, la vraisemblance classifiante pénalisée s'écrit

$$\begin{aligned} CML'(P, \Phi') &= CML(P, \Phi) + \sum_{k=1}^{K} \sum_{i=1}^{N} c_{ik} \log p_k \\ &= \sum_{k=1}^{K} \sum_{i=1}^{N} c_{ik} \log \{p_k f_k(\mathbf{x}_i|\theta_k)\}, \end{aligned}$$

et s'interprète comme la log-vraisemblance d'un échantillon aléatoire sur une population entière, les taille des sous populations étant la réalisation d'une loi multinomiale de paramètres $(p_1, \cdots, p_K)$.

Ces deux critères peuvent être maximisés par une version classificatoire de l'algorithme EM : *Classification EM algorithm.* L'algorithme CEM a été proposé par Celeux et Govaert (1992).

Une itération de l'algorithme CEM se décompose ainsi :

- **Etape E (estimation) :** Calcul des probabilités $t_k(\mathbf{x}_i)^q$ pour chaque $\mathbf{x}_i$.
- **Etape C (classification) :** Chaque $\mathbf{x}_i$ est affecté à la composante du mélange de plus forte probabilité a posteriori. Une partition P^{q+1} est donc définie caractérisée par la matrice $\mathbf{c} = \{c_{ik}\}$ avec $c_{ik} = 1$ si $k = arg\max_\ell t_\ell(x_i)^q$ et $c_{ik} = 0$ sinon.
- **Etape M (maximisation) :** Calcul des estimateurs du m.v. de Φ^{q+1} sur la base des sous-échantillons précisés par la matrice de classification dure $\mathbf{c}$.

L'algorithme CEM génère une suite $CML'(\Phi^q, P^q)$ croissante qui atteint son maximum en un nombre fini d'itérations (Celeux et Govaert 1992).

L'algorithme CEM est un algorithme très général de classification qui permet d'optimiser de nombreux critères de classification de type inertiels suivant les modèles gaussiens considérés. Prenons par exemple le modèle gaussien le moins contraint, pour lequel les classes sont de tailles différentes et possèdent une matrice de variance covariance quelconque, l'algorithme CEM maximise alors le critère de vraisemblance classifiante pénalisée. Si toutes les proportions sont fixées égales, le critère optimisé est alors simplement la vraisemblance classifiante. Un autre cas particulier intéressant est celui où les densités $f_k(.|\theta_k)$ du mélange sont des gaussiennes de vecteur moyenne $\boldsymbol{\mu}_k$ et de matrice de variance covariance $\boldsymbol{\Sigma}_k = \lambda \cdot I$, mélangés en proportions égales. En effet le critère optimisé est la somme des variances intra-classes,

$$\begin{aligned} CML(\Phi, P) &= \sum_{k=1}^{K} \sum_{i=1}^{N} c_{ik} \log \left(2\pi \det |\boldsymbol{\Sigma}_k|\right)^{-\frac{d}{2}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_k)^t (\lambda \cdot I)^{-1} (\mathbf{x} - \boldsymbol{\mu}_k)\right) \\ &= -\frac{1}{2\lambda} \sum_{k=1}^{K} \sum_{i=1}^{N} c_{ik} (\mathbf{x} - \boldsymbol{\mu}_k)^t (\mathbf{x} - \boldsymbol{\mu}_k) + Cst, \end{aligned}$$

et l'algorithme CEM, avec ce modèle, est exactement l'algorithme des centres mobiles présenté dans la section 1.1.3.

Différentes études (Celeux 1992) ont montré que l'approche classification introduisait un biais dans l'estimation des paramètres. En effet, cette approche estime les paramètres du mélange sur la base des classes, alors que les classes sont disjointes et constituent en fait des échantillons tronqués des composantes du mélange. Ce phénomène a tendance à surestimer les différences entre les moyennes, et à sous estimer les variances et les différences entre proportions. Ces inconvénients ne sont pas rédhibitoires si les classes sont bien séparées et les proportions du même ordre de grandeur.

Notons qu'il existe aussi une version recuit simulé de l'algorithme CAEM : *Classification Annealing EM Algorithm* (Celeux et Govaert 1992). Dans ce cas l'idée de base consiste à commencer par des itérations de type SEM pour s'orienter, lorsque la température baisse vers des itérations de type CEM. Comme pour l'algorithme SAEM, on espère que l'introduction de perturbations aléatoires permettra d'éviter de "mauvais" minima locaux et que les dernières itérations deterministes produiront une "bonne" estimation ponctuelle.

- **Etape E (estimation) :** Calcul des probabilités $t_k(\mathbf{x}_i)^q$ pour chaque $\mathbf{x}_i$.

- **Etape A (annealing) :** Calcul des quantités $s_{ik} = \frac{(t_k(\mathbf{x}_i)^q)^{\frac{1}{\gamma^q}}}{\sum_{\ell=1}^{K}(t_\ell(\mathbf{x}_i)^q)^{\frac{1}{\gamma^q}}}$, pour $i = 1, \cdots, N$ et $k = 1, \cdots, K$, avec γ^q un scalaire compris entre 0 et 1, appelé température, jouant le même rôle que dans l'algorithme SAEM.

  Une nouvelle température γ^{q+1} est calculée.

- **Etape C (classification) :** Chaque $\mathbf{x}_i$ est affecté au hasard à une classe k, suivant la loi multinomiale de paramètres $(s_{i1}, \cdots, s_{iK})$.Une partition dure est ainsi définie que l'on caractérise par une matrice de classification $\mathbf{c}$.

- **Etape M (maximisation) :** Calcul des estimateurs du m.v. de Φ^{q+1} sur la base des sous-échantillons précisés par la matrice de classification dure $\mathbf{c}$.

D'une version à l'autre

L'algorithme EM est un algorithme "robuste", qui semble donner des résultats cohérents même lorsque l'une des deux étapes qui constituent chaque itération est modifiée. Toutes les versions présentées dans cette section visent à surmonter certains inconvénients liés à l'algorithme EM ou bien à adapter celui-ci à certains problèmes particuliers. Chacun de ces algorithmes proposent une modification de l'étape d'estimation des probabilités a posteriori nécessaires au calcul des estimateurs du maximum de vraisemblance. Ces modifications sont récapitulées par le tableau 1.2

TAB. 1.2 - : Tableau récapitulatif des algorithmes dérivés de EM : Détails du calcul des quantités c_{ik} nécessaires à l'estimation par m.v des paramètres du mélange. Les quantités z_{ik} sont tirées au hasard suivant une loi multinomiale de paramètres $(t_1(\mathbf{x}_i)^q, \cdots, t_K(\mathbf{x}_i)^q)$

Algorithme	Etape d'estimation
EM	$c_{ik} = t_k(\mathbf{x}_i)^q$
SEM	$c_{ik} = z_{ik}$
SAEM	$c_{ik} = t_k(\mathbf{x}_i)^q + \gamma^q(z_{ik} - t_k(\mathbf{x}_i)^q)$
CEM	$c_{ik} = 1$ si $k = arg\max_\ell t_\ell(x_i)^q$, $c_{ik} = 0$ sinon
CAEM	c_{ik} est tiré suivant la loi multinomiale $(s_{i1}, \cdots, s_{iK})$ avec $s_{ik} = \frac{(t_k(\mathbf{x}_i)^q)^{\frac{1}{\gamma^q}}}{\sum_{\ell=1}^K (t_\ell(\mathbf{x}_i)^q)^{\frac{1}{\gamma^q}}}$

1.4.3 Approche bayésienne

L'analyse statistique bayésienne propose des solutions élégantes en classification automatique. A notre connaissance, deux types d'approches bayésiennes sont couramment utilisées :

- la première utilise un mélange de densités comme modèle probabiliste de la loi des observations ;
- la seconde trouve son origine dans les statistiques spatiales et est très utilisée en traitement d'image. Les observations suivent dans ce cas une distribution de Gibbs. Ce cas sera examiné dans le chapitre consacré à la classification spatiale.

La stratégie bayésienne vise à estimer les paramètres du modèle, qui minimisent le coût a posteriori

$$\rho(\pi, \delta|\mathbf{x}) = I\!E^\pi[L(\theta, \delta)|\mathbf{x}] = \int_\theta L(\theta, \delta)\pi(\theta|\mathbf{x})d\theta.$$

Ainsi, le calcul d'estimateurs bayésiens nécessite en général le calcul de la loi $\pi(\theta|\mathbf{x})$. Les techniques d'échantillonnage permettent de simuler la loi a posteriori losque celle-ci necessite des calculs trop fastidieux. Ces techniques sont particulièrement utiles dans le contexte de la classification automatique basée sur les modèles de mélange ou sur les distributions de Gibbs.

Notons que l'échantillonnage oblige à se servir d'un coût quadratique ou 0–1. En effet, les techniques d'échantillonnage ne produisent que des réalisation de variables aléatoires, sans la connaisance de la probabilité de ces réalisation :

- Dans le cas d'un coût quadratique, l'estimateur produit est celui de la moyenne a posteriori. Les réalisations simulées du paramètre recherché permettent d'approcher la vraie valeur de la moyenne a posteriori par la moyenne empirique a posteriori.

– Dans le cas d'un coût 0–1, l'estimateur recherché est le maximum a posteriori et des techniques de recuit simulé (Geman et Geman 1984) permettent de produire des réalisations qui approchent la valeur du MAP.

Échantillonnage

L'échantillonnage est une méthode de Monte-Carlo qui permet de générer des variables aléatoires suivant la loi a posteriori $\pi(\theta|\mathbf{x})$. Deux technique sont communément utilisées :

– l'échantillonnage bayésien,

– l'échantillonnage de Gibbs.

Dans le cas où l'introduction d'un paramètre λ permet d'obtenir les distributions de $\pi(\theta|\mathbf{x},\lambda)$ et $\pi(\lambda|\mathbf{x},\theta)$ sous forme explicite la méthode appelée *data augmentation* (Tanner et Wong 1987) ou échantillonnage bayésien peut être utilisée. Cette méthode est itérative. Partant d'une position initiale λ^0, une itération se décompose comme suit :

– générer θ^q suivant $\pi(\theta|\mathbf{x},\lambda^{q-1})$,

– générer λ^q suivant $\pi(\lambda|\mathbf{x},\theta^q)$.

Théorème 1.3 *Si la variable aléatoire θ (resp. λ) est, a posteriori, à valeurs dans un espace fini Θ (resp. Λ) et si $\pi(\theta|\boldsymbol{x},\lambda) > 0$ sur Θ (resp. $\pi(\lambda|\boldsymbol{x},\theta) > 0$ sur Λ, les suites (θ^q) et (λ^q) forment des chaines de Markov ergodique dont les uniques lois invariantes sont respectivement $\pi(\theta|\boldsymbol{x})$ et $\pi(\lambda|\boldsymbol{x})$.*

L'échantillonnage de Gibbs est une alternative à l'échantillonnage bayésien. Il ne considère que des paramètres unidimensionnels et les génère donc conditionnellement à tous les autres et aux données. Ce type d'échantillonnage ne respecte pas une structure hiérarchique.

Dans certains textes (Bensmail *et al.* 1995), la distinction entre les deux types d'échantillonnage n'est pas faite et le terme méthode MCMC (Markov Chain Monte Carlo) est utilisé de manière générique.

Application aux mélanges gaussiens

L'estimation des paramètres d'un modèle de mélange gaussien par échantillonnage baysésien est basée sur le principe de l'information manquante (Dempster *et al.* 1977) introduit dans ce contexte par Tanner et Wong (1987).

On dispose d'un échantillon $(\mathbf{x}_1,...,\mathbf{x}_n)$ d'une loi de mélange gaussien :

$$f(\mathbf{x}_i|\Phi) = \sum_{k=1}^{K} p_k f_k(\mathbf{x}_i|\theta_k), \tag{1.44}$$

Il est possible d'introduire les quantités manquantes z_i ($1 \leq i \leq N$), qui indiquent la provenance de l'individu i ($z_i = k$ si $\mathbf{x}_i$ provient de la classe k). L'introduction de ces nouvelles variables permet de considérer la densité $f(\mathbf{x})$ comme une structure hiérarchique où :

$$\mathbf{x}_i | z_i \sim f_{z_i}(\mathbf{x}_i|\theta_{z_i}), \quad z_i \sim p_1 I\!I(z_i = 1) + \cdots + p_K I\!I(z_i = K)$$

Une loi a priori conjuguée est choisie sur $\Phi = (p_1, \cdots, p_{K-1}, \boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_K, \boldsymbol{\Sigma}_1, \cdots, \boldsymbol{\Sigma}_K)$:

$$\pi(\Phi) = \pi(p_1, \cdots, p_K) \cdot \prod_{k=1}^{K} \pi(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{1.45}$$

avec

- $\pi(p_1, \cdots, p_K)$ est une distribution de Dirichlet, $\mathcal{D}(\alpha_1, \cdots, \alpha_K)$.
- $\pi(\boldsymbol{\mu}_k|\boldsymbol{\Sigma}_k)$ est une distribution gaussienne, $\mathcal{N}(\xi_k, \boldsymbol{\Sigma}_k/\tau_k)$
- $\pi(\boldsymbol{\Sigma}_k)$ dépend de la paramétrisation choisie pour la matrice de variance-covariance.

Une itération du processus d'échantillonnage se décompose alors de la façon suivante :

1. Simulation des z_i suivant leur probabilité a posteriori :

$$t_k(\mathbf{x}_i) = \pi(z_i = k|\mathbf{x}_i, \Phi) = \frac{p_k f_k(\mathbf{x}_i|\theta_k)}{f(\mathbf{x}_i)}$$

   Remarquons que cette étape est identique à l'etape *Stochastique* de l'algorithme SEM.

2. Simulation des proportions suivant leur distribution a posteriori, conditionnellement aux z_i :

$$p_1, \cdots, p_K \sim \mathcal{D}(\alpha_1 + \sum_{i=1}^{N} I\!I(z_i = 1), \cdots, \alpha_K + \sum_{i=1}^{N} I\!I(z_i = K))$$

3. Simulations des moyennes et matrices de variances-covariances suivant leur distribution a posteriori, conditonnellement aux z_i. Les lois de ces simulations dépendent elles aussi de la paramétrisation de la matrice de variance (Bensmail *et al.* 1995).

Le choix du nombre d'itérations nécessaire pour atteindre la convergence de ce type d'échantilloneur est un problème de recherche actuel (1996). Les avis sont divers et variés : Robert (1996) estime par exemple, que pour des données unidimensionnelles, 5000 est un nombre d'itérations raisonnable.

Chapitre 2

Cartographie associative

Whereas eye, retina and lateral geniculate body transform the images in a "photographic" way, i.e. preserving essentially the spatial arrangement of the retinal image, the cortex transforms this geometry into a space of concepts.

Von der Malsburg (1973)

Classification automatique en analyse des données et apprentissage non supervisé par réseaux de neurones résolvent des problèmes similaires. D'origine bien distincte, il nous semble que les deux approches peuvent s'enrichir mutuellement. Les réseaux de neurones peuvent apporter des idées originales en analyse de données, et la classification automatique met à disposition un cadre formel et une grande bibliothèque d'idées éprouvées.

Dans le domaine des réseaux de neurones, deux courants d'intérêt se côtoient. Le plus vieux provient de la biologie et tente de comprendre et modéliser les mécanismes biologiques réels. L'autre courant a pour origine des considérations d'ingénieurs qui empruntent des concepts à la biologie pour développer des nouveaux paradigmes en algorithmique et reconnaissance de formes. L'apprentissage non supervisé, plus encore que l'apprentissage supervisé (réseaux de neurones à couches), porte la marque des modèles biologiques (Hertz *et al.* 1991).

Depuis le milieu des années 1980, les recherches à propos des réseaux de neurones ont pris une orientation plus mathématique. Les nouveaux algorithmes et techniques développés n'ont plus de justifications biologiques. Une volonté de théorisation succède aux méthodes empiriques des débuts. Des statisticiens, des mathématiciens s'investissent pour comprendre ce que font les réseaux de neurones, et déterminer à quelles autres méthodes déjà existantes ces réseaux peuvent s'apparenter.

Dans ce courant de pensée, nous avons tenté de rapprocher les cartes auto-organisatrices de Kohonen –modèle connexioniste d'apprentissage non supervisé– de l'approche probabiliste en classification automatique (voir Chapitre 1). Ce chapitre propose deux algorithmes originaux, basés sur les modèles de mélanges gaussiens, qui intègrent le concept de conservation topologique des cartes de Kohonen.

2.1 De l'apprentissage compétitif aux cartes de Kohonen

2.1.1 Des origines

Comment les signaux sensoriels sont-ils encodés dans le cerveau ? Ce problème de représentation interne n'est pas encore résolu par les biologistes, mais quelques éléments de réponses existent. Ainsi, il semble que la localisation d'un neurone dans le cerveau soit en relation avec le type de stimulations qui activent ce neurone. L'ensemble des stimulations qui stimulent un neurone constituent le *champ récepteur* (receptive field) de ce neurone et deux neurones spatialement proches possèdent des champs récepteurs proches. Ceci implique une sorte de continuité spatiale de la distribution du champ récepteur dans le cerveau.

De part cette continuité, certaines aires du cerveau sont des projections topographiques d'un ensemble donné de signaux sensoriels. Il est d'usage de dire que ce type de projection conserve la topologie. C'est-à-dire que deux signaux similaires activeront deux neurones spatialement proches.

Exemple 2.1 (Kohonen 1984) Dans le cerveau des chats, une aire est spécialisée dans le traitement des signaux acoustiques. Cette aire spécifique est constituée de neurones ordonnés suivant les fréquences acoustiques auxquelles ils sont sensibles. Ainsi les neurones activés par des hautes fréquences sont localisés à l'opposé des neurones sensibles aux basses fréquences. Cette zone est appelée la carte tonotopique.

△

Les cartes neuronales structurées sont communes dans le cerveau. La question de l'origine de cette "auto-organisation" émergea dans les années soixante dix. Quelle quantité et quel genre d'information est nécessaire pour induire un phénomène d'auto-organisation ? Est-ce que des mécanismes simples sont en mesure d'expliquer cette étonnante faculté du cerveau, ou bien toute l'information est-elle contenue dans le détail au niveau des gènes ? Des chercheurs ont eu l'idée d'aborder le problème en testant leurs hypothèses sur des modèles hypersimplifiés de réseaux de neurones. Leurs idées et modèles furent ensuite repris en reconnaissance des formes et font l'objet de ce chapitre.

La première modélisation du neurone a été suggérée dans les années quarante par Mac Culloch and Pitts (Davalo et Naim 1992). C'était une unité qui en fonction

de plusieurs signaux transmettait une réponse binaire. D'une manière générale, un neurone formel possède des "dendrites" qui recoivent le signal d'entrée et un "axone" qui transmet le signal de sortie : le signal d'entrée $\mathbf{x}$ est un vecteur appartenant le plus souvent à $\mathbb{R}^d$ ou $\{0,1\}^d$. Les dendrites sont caractérisées par un vecteur poids $\boldsymbol{\mu}^1$ de même dimension que les vecteurs d'entrée. La sortie est une fonction de $\mathbf{x}$ et $\boldsymbol{\mu}$, qui est la composition d'une fonction d'entrée, $h(\mathbf{x}, \boldsymbol{\mu})$ et d'une fonction de sortie (ou d'activation), $f(h)$ (Figure 2.1). La plupart du temps la fonction d'entrée est un simple produit scalaire :

$$h(\mathbf{x}, \boldsymbol{\mu}) = (\mathbf{x} \cdot \boldsymbol{\mu}) \tag{2.1}$$

Les fonctions d'activation sont diverses mais appartiennent à de grandes familles (fonctions à bases radiales, fonctions sigmoïdes...).

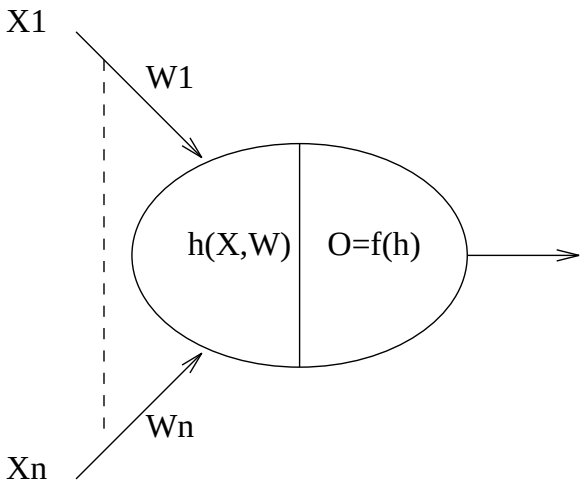


FIG. 2.1 - : Le neurone formel (le vecteur poids est noté W)

Exemple 2.2 Une fonction d'activation typique est par exemple :

$$f(\mathbf{x}, \boldsymbol{\mu}) = \alpha \cdot \frac{\exp\{(\mathbf{x} \cdot \boldsymbol{\mu})\} - 1}{\exp\{(\mathbf{x} \cdot \boldsymbol{\mu})\} + 1}. \tag{2.2}$$

Les fonctions qui possèdent cette allure sont dites sigmoïdes (Figure 2.2). Le lecteur interessé pourra consulter Davalo et Naim (1992) pour plus de détails.

△

Des neurones peuvent être connectés les uns aux autres et forment alors un réseau. L'apprentissage consiste à régler les paramètres libres du réseau en fonction du but désiré, c'est-à-dire à calculer les valeurs des vecteurs poids en fonction des entrées.

[1]La notation conventionnelle dans le domaine des réseaux de neurones utilise $\mathbf{w}$ comme symbole d'un vecteur poids ("weight vector" en anglais). Nous dérogeons à cette convention pour une raison de cohérence avec la notation usuelle de l'approche probabiliste en classification présentée dans le chapitre 1, où $\boldsymbol{\mu}$ dénotait les centres des classes.

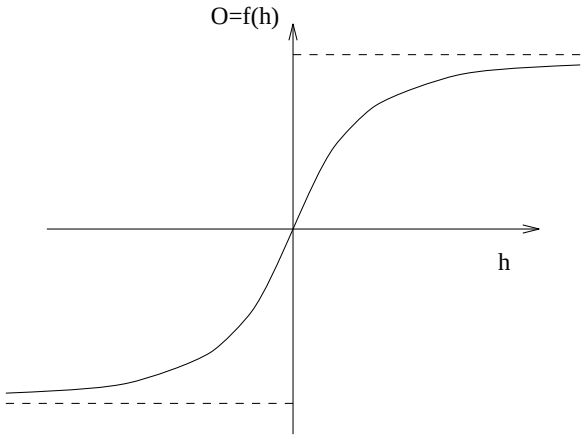


FIG. 2.2 - : Une fonction sigmoïde

2.1.2 Apprentissage compétitif

Dans le cerveau des mammifères, les cellules interagissent latéralement. Lorsqu'une cellule est excitée, elle transmet son excitation à ses cellules voisines dans un rayon de 50 à 100 μm et inhibe les cellules situées dans un rayon de 200 à 5000 μm (Kohonen 1984). Le degré d'interaction latérale est une fonction de la distance, qui ressemble à un chapeau mexicain (Figure 2.3). Ces interactions latérales engendrent des réponses de groupes de neurones répartis autour du maximum local d'excitation. Ce phénomène a été modélisé par Von der Malsburg (1973), Grossberg (1976*a*), Grossberg (1976*b*) et Kohonen (1984). Les algorithmes d'apprentissage compétitif sont des simplifications algorithmiques qui utilisent l'idée de réponses localisées et d'interactions latérales.

La règle du "Winner Take All"

La forme la plus simple d'apprentissage compétitif modifie seulement le vecteur poids du "meilleur" neurone à chaque étape de l'apprentissage. En fait, à chaque présentation d'une entrée (un vecteur de l'ensemble d'apprentissage), deux étapes sont effectuées :

1. choisir le meilleur neurone, c'est-à-dire celui qui manifeste la sortie la plus importante ;

2. modifier les coordonnées du vecteur poids de ce neurone.

Lorsque la fonction d'activation est croissante (ce qui n'est pas vrai pour les fonctions à base radiale), le neurone gagnant est celui qui produit la plus grande valeur de fonction d'entrée. Si nous considérons un produit scalaire comme fonction

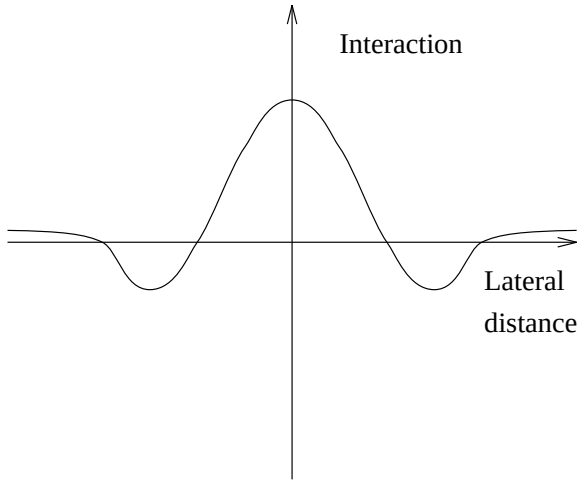


FIG. 2.3 - : Intensité de l'intéraction latérale en fonction de la distance

d'entrée, le vecteur poids du gagnant, i^*,vérifie :

$$\forall i,\ (\boldsymbol{\mu}_{i^*} \cdot \mathbf{x}) \geq (\boldsymbol{\mu}_i \cdot \mathbf{x}). \tag{2.3}$$

Et si *les vecteurs poids sont normalisés*, le gagnant est le neurone qui possède le vecteur poids, le plus proche de l'entrée $\mathbf{x}$, au sens de la distance euclidienne :

$$\begin{aligned} \|\boldsymbol{\mu}_{i^*} - \mathbf{x}\|^2 &\leq \|\boldsymbol{\mu}_i - \mathbf{x}\|^2, \\ \|\boldsymbol{\mu}_{i^*}\|^2 - 2\boldsymbol{\mu}_{i^*} \cdot \mathbf{x} + \|\mathbf{x}\|^2 &\leq \|\boldsymbol{\mu}_i\|^2 - 2\boldsymbol{\mu}_i \cdot \mathbf{x} + \|\mathbf{x}\|^2, \\ \boldsymbol{\mu}_{i^*} \cdot \mathbf{x} &\geq \boldsymbol{\mu}_i \cdot \mathbf{x}. \end{aligned}$$

Les coordonnées du vecteur poids du gagnant sont actualisées en utilisant une règle du type suivant :

$$\boldsymbol{\mu}_{i*}(t+1) = \boldsymbol{\mu}_{i^*}(t) + \alpha(t) \cdot (\mathbf{x} - \boldsymbol{\mu}_{i^*}(t)),\ \alpha(t) \leq 1 \tag{2.4}$$

où $\alpha(t)$ est le pas d'apprentissage à l'itération t.

Ce type d'apprentissage est baptisé "Winner Take All", car seul le neurone sélectionné apprend. Notons que la règle 2.4 donne des résultats plus rapides et meilleurs lorsque les vecteurs d'entrée sont normalisés (Hertz *et al.* 1991).

Remarquons qu'en remplacant respectivement les mots neurone par prototype et vecteur d'entrée par individu, pour utiliser la terminologie de l'analyse des données, il est évident que la règle d'apprentissage 2.4 est la même que celle utilisée par l'algorithme des k-means. Dans ce cas précis, les réseaux de neurones ne nous semblent pas apporter de grandes nouveautés, mais dans les sections suivantes nous présentons quelques développements de l'apprentissage compétitif qui possèdent des aspects réellement originaux.

Le problème des neurones morts

Si l'on utilise la règle du "Winner Take All", il se peut que certaines unités soient trop éloignées de toutes les données et ne gagnent jamais. Ces unités ne supportent aucune information et sont parfaitement inutiles. De nombreuses solutions sont envisageables pour pallier ce problème (Hertz *et al.* 1991) :

- les vecteurs poids peuvent être initialisés avec les entrées elles même, ce qui assure que, au moins au début de l'algorithme, tous les neurones apprennent ;
- les vecteurs poids des neurones perdants peuvent être modifiés aussi mais dans des proportions moindres que les gagnants ;
- les vecteurs d'entrée peuvent être bruités ;
- ...

La cartographie associative (traduction de "feature mapping") propose une solution, d'inspiration biologique, aux problème des unités mortes.

Cartographie associative

L'apprentissage compétitif, dans sa version la plus simple, ne tient aucun compte d'interactions latérales entre les neurones. La cartographie associative utilise cette idée d'interactions et postule des relations de voisinage *a priori* entre les unités. Ainsi chaque unité, posséde un ensemble d'unités voisines, qui constituent son voisinage.

Les relations de voisinage entre les neurones définissent une topologie sur l'ensemble des unités et donc un nouvel espace. La caractéristique principale de la cartographie associative tient ainsi à la prise en compte de deux espaces bien distincts :

1. Un espace des entrées, dans lequel peuvent être représentés les données et les vecteurs poids des neurones.
2. Un espace de sortie (ou carte), qui contient l'ensemble des neurones et sur lequel une topologie a été définie. Cet espace possède le plus souvent une ou deux dimensions.

Utilisons une image biologique, pour illustrer la différence entre ces deux espaces : l'espace des entrées correspond à l'ensemble des signaux perçus par le cerveau, muni d'une distance, et l'espace des sorties est le cerveau lui même, la topologie traduisant l'existence des connections latérales entre les neurones...

Le but de la cartographie associative consiste à associer chaque vecteur d'entrée à un neurone de la carte. De manière très intuitive, on espère que la topologie de la carte permettra, en fin d'apprentissage, de transmettre plus d'information sur l'espace des entrées. C'est-à-dire que deux vecteurs d'entrée proches, au sens d'une distance dans l'espace des entrées, exciteront deux neurones proches sur la carte.

Notons que si l'on dispose d'un espace muni d'une métrique d, on peut lui donner une structure d'espace topologique en définissant le voisinage V_k de l'unité $\boldsymbol{\mu}_k$ comme l'ensemble des unités $\boldsymbol{\mu}_i$ contenus dans une boule de rayon non nul centrée en $\boldsymbol{\mu}_k$ (Bertrandias 1970) :

$$V_k = \{\boldsymbol{\mu}_i / d(i,k) < \sigma,\ \sigma > 0\} \tag{2.5}$$

Ainsi en pratique, la carte peut être spécifiée en répartissant les neurones régulièrement dans l'espace de sortie muni d'une distance adéquate (euclidienne ou autre).

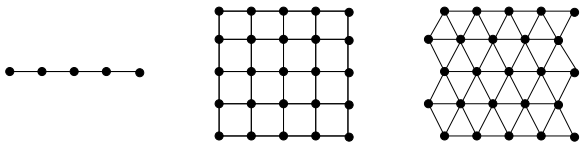


FIG. 2.4 - : Trois architectures courantes de carte

Trois architectures sont couramment utilisées (voir Figure 2.4) :

- les cartes unidimensionnelles où chaque neurone possède deux voisins ;
- les cartes bidimensionnelles à voisinage rectangulaire où chaque neurone est entouré de quatre voisins ;
- les cartes bidimensionnelles à voisinage hexagonale où chaque neurone est entouré de six voisins.

La plupart des exemples "d'écoles" traitent des données qui sont décrites par deux variables. Dans ce cas, il est possible de visualiser les données et les prototypes dans le plan. Les relations de voisinage se traduisent par des segments tracés entre tous les neurones voisins.

2.1.3 Les cartes auto-organisatrices de Kohonen

En 1982, Kohonen proposait un algorithme d'apprentissage pour la cartographie associative qui faisait suite aux travaux de Von der Malsburg (1973), Amari (1980) : *The Self-organizing Feature Map Algorithm*(SOM).

La forme de l'algorithme a évoluée au cours du temps, sous l'influence de Kohonen (Kohonen 1984, Kohonen 1988*a*, Kohonen 1988*b*, Kohonen 1991) et d'autres chercheurs. En 1984, Kohonen publiait la première version de son livre *Self Organization and Associative Memory* qui est remis à jour régulièrement. Dans cet ouvrage, l'algorithme se présentait comme suit :

1. **Initialisation :** L'architecture de la carte (ou encore grille, réseau) est spécifiée, ce qui revient à choisir le nombre de neurones et définir les relations de voisinage. Les vecteurs poids sont alors initialisés.

2. **A chaque itération**, un vecteur d'entrée est choisi au hasard. Ce vecteur $\mathbf{x}$ modifie les poids du réseau de la façon suivante :

   (a) Localisation du vecteur poids gagnant, $\boldsymbol{\mu}_{k^*}$, qui vérifie la condition :

   $$\|\mathbf{x} - \boldsymbol{\mu}_{k^*}\| = \min_k \|\mathbf{x} - \boldsymbol{\mu}_k\|. \tag{2.6}$$

   (b) Modification des vecteurs poids de l'unité gagnante et de ses voisines :

   $$\boldsymbol{\mu}_k(t+1) = \boldsymbol{\mu}_k(t) + \alpha(t) \cdot (\mathbf{x} - \boldsymbol{\mu}_k(t)),\ 0 \leq \alpha(t) \leq 1,\ \forall k \in V_{k^*}, \tag{2.7}$$

   avec V_{k^*} l'ensemble des voisins de l'unité k^* (voir Équation 2.5). Le pas d'apprentissage $\alpha(t)$ satisfait les conditions de l'approximation stochastique :

   $$\sum_{t=0}^{\infty} \alpha(t) = \infty,\ \sum_{t=0}^{\infty} \alpha(t)^2 < \infty.$$

Sans justification autre que pratique, Kohonen (1984) conseille de faire décroître la taille du voisinage au cours des itérations, pour améliorer les performances de l'algorithme.

Notons que l'équation 2.7 est la règle d'apprentissage compétitif standard. L'originalité et l'efficacité de l'algorithme tiennent à la façon de faire intervenir les relations de voisinage entre neurones dans la règle de modification des poids.

Ritter et Shulten (1986) proposaient de modifier légèrement l'algorithme en introduisant une *fonction de voisinage*. La loi d'adaptation des poids s'exprime alors comme :

$$\boldsymbol{\mu}_k(t+1) = \boldsymbol{\mu}_k(t) + \alpha(t) \cdot h(k, k^*) \cdot (\mathbf{x} - \boldsymbol{\mu}_k(t)), \tag{2.8}$$

où $h(k, k^*)$ est la fonction de voisinage (ou d'interaction), qui est une fonction de la distance $d(k, k^*)$ entre les unités k et k^* sur la carte. Cette fonction vaut 1 lorsque $d(k, k^*) = 0$ et décroit quand la distance augmente.

Exemple 2.3 Une fonction d'interaction souvent utilisée est la gaussienne :

$$h(k, k^*) = \exp\{-\frac{d(k, k^*)^2}{2 \cdot \sigma(t)^2}\}. \tag{2.9}$$

△

Lo et Bavarian (1991) ont montré que l'utilisation d'une fonction de voisinage plutôt que d'un ensemble fini de voisins permettait d'améliorer la vitesse de convergence de l'algorithme.

L'algorithme des cartes de Kohonen soulève beaucoup de questions. C'est un algorithme simple à programmer qui donne des résultats visualisables et qui fonctionne très bien en pratique. Par contre, au niveau théorique beaucoup d'analyses ont été

publiées sans parvenir à démontrer la conservation de la topologie dans le cas général. En effet, Cottrell et Fort (1987) et Bouton et Pages (1992) ont énoncé des théorèmes sur l'ordonnancement des unités dans le cas où les vecteurs d'entrée sont de dimension 1 et les unités forment une carte unidimensionnelle, mais dans le cas multidimensionnel, la simple définition du concept de conservation topologique pose problème. Beaucoup d'efforts ont aussi été investis pour tenter de trouver si l'algorithme SOM optimise un critère quelconque. Cette direction de recherche constitue le sujet de la prochaine section.

2.2 Critères et algorithmes pour la cartographie associative

Hertz *et al.* (1991) notent que dans le domaine des réseaux de neurones :

> In practice most authors have defined an algorithm and then looked after the fact (if at all) at what it optimized.

Les cartes de Kohonen appartiennent à cette catégorie d'algorithmes qui sont issus de la modélisation d'un processus biologique et non de l'optimisation d'un critère[2].

Pour juger de manière objective des résultats d'un algorithme, pour pouvoir comparer avec d'autres méthodes non supervisées, il nous semble que l'existence d'un critère numérique est souhaitable.

Dans la suite, nous allons examiner la possibilité de dériver l'algorithme de Kohonen d'une fonction d'énergie par la méthode de l'approximation stochastique, puis nous donnerons quelques exemples d'algorithmes de cartographie associative qui sont issus de l'optimisation d'une fonctionnelle bien définie.

2.2.1 Algorithmes adaptatifs et approximation stochastique

Les algorithmes adaptatifs sont utiles pour l'identification des paramètres d'un système en traitement du signal et en reconnaissance des formes. Notons que presque tous les algorithmes dits neuronaux appartiennent à cette catégorie. L'objectif de tels algorithmes est d'adapter un vecteur de paramètres Φ au fur et à mesure de la présentation de vecteurs de mesures $\mathbf{x}_1, \cdots, \mathbf{x}_N$. Les vecteurs de mesures sont supposés être les réalisations d'un vecteur aléatoire de loi f. Dans le contexte des réseaux de neurones, les composantes du vecteur Φ sont les poids des connections et les vecteurs de mesures forment ce qui est appelé l'ensemble d'apprentissage (ou encore, les vecteurs d'entrée).

Les algorithmes adaptatifs modifient le vecteur paramètre en fonction de l'information apportée par un seul vecteur de mesure. A ce titre, ils sont parfois qualifiés

[2]Les termes fonctionnelle ou fonction d'énergie sont couramment utilisés comme synonymes du mot critère.

d'algorithmes "en ligne" (on-line) par opposition aux algorithmes 'hors ligne" (off-line), qui eux prennent en compte l'information apportée par tous les vecteurs de mesures disponibles pour identifier le vecteur de paramètres Φ.

D'après Benveniste *et al.* (1987), un algorithme adaptatif s'exprime sous la forme générale suivante :

$$\Phi_n = \Phi_{n-1} + \alpha_n \cdot H(\Phi_{n-1}, \mathbf{x}_n) + \alpha_n^2 \cdot \epsilon_n(\Phi_{n-1}, \mathbf{x}_n) \tag{2.10}$$

où H est une fonction qui décrit l'ajustement du vecteur paramètre, α_n est le gain, qui est variable au cours des itérations, et ϵ_n définit une perturbation. Ce dernier terme est souvent négligé.

Une distinction existe entre les algorithmes à gain constant,

$$\alpha_n \geq 0, \quad \lim_{n \to \infty} \alpha_n = \alpha > 0$$

et les algorithmes à gain décroissant,

$$\sum_{n=0}^{\infty} \alpha_n = \infty, \quad \sum_{n=0}^{\infty} \alpha_n^2 < \infty.$$

Les premiers sont dédiés à l'estimation de paramètres changeant lentement au cours du temps et les seconds à l'estimation de paramètres stables.

La suite des vecteurs paramètres calculés étant une suite aléatoire, ces algorithmes convergeront au mieux presque sûrement. Dans le cas où il y a convergence, la vitesse de convergence sera d'intérêt. Pour une présentation détaillée des algorithmes adaptatifs et leur théorie, nous renvoyons le lecteur à la monographie de Benveniste *et al.* (1987).

Les algorithmes de gradient stochastique sont des algorithmes adaptatifs qui visent à minimiser une fonction du paramètre Φ, $C(\Phi)$, qui peut se mettre sous la forme d'une espérance sur l'ensemble des mesures, issue d'une distribution de probabilité notée f :

$$C(\Phi) = I\!E^f[\mathcal{J}(\Phi, \mathbf{x})] = \int \mathcal{J}(\Phi, \mathbf{x}) df(\mathbf{x}).$$

où $\mathcal{J}(\Phi, \mathbf{x})$ est une fonction de la mesure aléatoire $\mathbf{x}$ et de Φ. L'idée de la descente de gradient stochastique consiste à considérer à chaque itération, une direction de descente liée au seul vecteur de mesure disponible, $\mathbf{x}_n$. Le gradient considéré s'exprime alors comme :

$$\frac{\partial \mathcal{J}(\Phi, \mathbf{x}_n)}{\partial \Phi} = H(\Phi, \mathbf{x}_n),$$

et l'algorithme adaptatif correspondant est donné par :

$$\Phi_n = \Phi_{n-1} + \alpha_n \cdot H(\Phi_{n-1}, \mathbf{x}_n). \tag{2.11}$$

2.2.2 Les cartes de Kohonen optimisent-elles un critère?

L'algorithme de cartes de Kohonen optimise-t-il un critère? La réponse à cette question nécessite de distinguer deux cas:

1. les données sont issues d'une distribution de probabilité discrète;
2. les données sont les réalisations d'une distribution de probabilité continue.

Le premier cas semble être un modèle correct, lorsque l'ensemble d'apprentissage est fini et disponible dans son entier. Les vecteurs d'entrée peuvent alors être considérés comme étant tous équiprobables et la loi f dont ils sont la réalisation est une loi uniforme discrète. Dans ce cas, on suppose que les paramètres cherchés possèdent une valeur fixée.

Le second cas est une généralisation du premier et nous parait adapté à une situation dans laquelle l'ensemble d'apprentissage est un échantillon d'une population de taille infini.

En pratique, considérer le premier ou le second cas, se résume à un choix de modèle. D'un point de vue théorique, l'utilisation de l'algorithme de Kohonen trouve plus de justifications dans le cas discret que dans le cas continu.

Distribution discrète de l'ensemble d'apprentissage

Considérons un ensemble d'apprentissage fini, $S = \{\mathbf{x}_1, \cdots, \mathbf{x}_N\}$ avec $\mathbf{x}_i \in I\!R^d$, où chaque réalisation a une probabilité $P(\mathbf{x}_i)$. Dans ce cas, l'algorithme de Kohonen minimise la fonction (Ritter et Schulten 1988):

$$W = \frac{1}{2} \sum_{i=1}^{N} \sum_{k=1}^{K} h(k, k^*) \cdot \|\mathbf{x}_i - \boldsymbol{\mu}_k\|^2 \cdot P(\mathbf{x}_i), \tag{2.12}$$

où k^* est l'indice de l'unité gagnante pour l'exemple $\mathbf{x}_i$:

$$k^* = arg \min_k \|\mathbf{x}_i - \boldsymbol{\mu}_k\|,$$

et h, la fonction de voisinage. Le critère W peut être exprimé comme une espérance (Kohonen 1991),

$$\begin{aligned} W &= I\!E^P[\frac{1}{2} \sum_{k=1}^{K} h(k, k^*) \cdot \|\mathbf{x}_i - \boldsymbol{\mu}_k\|^2], \\ &= I\!E^P[\mathcal{J}(\mathbf{x}_i, \boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_K)], \\ &= I\!E^P[\mathcal{J}(\mathbf{x}_i, \Phi)]. \end{aligned}$$

Le critère W est défini et unique presque partout, sauf aux endroits de l'espace où la définition de la fonction de voisinage h est ambigüe. En effet sur les bords du pavage de Voronoï défini par les vecteurs $\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_K$, l'indice $k^* = arg \min_k \|\mathbf{x}_i - \boldsymbol{\mu}_k\|$ n'est pas

défini de manière unique ce qui entraine des points de discontinuité pour la fonction de voisinage $h(k, k^*)$.

Considérons l'espace d'entrée sans les bords du pavage de Voronoï. Dans cet espace, le critère W est parfaitement dérivable par rapport aux vecteurs $\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_K$. Ainsi, dans le cas où la fonction de voisinage ne dépend pas du temps, il apparait que l'algorithme de Kohonen est exactement une descente de gradient stochastique qui minimise la fonction de coût W. En effet, la mise à jour du paramètre $\Phi = \{\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_K\}$, suivant un gradient stochastique, met en évidence la loi d'adaptation de Kohonen :

$$\boldsymbol{\mu}_k(t+1) = \boldsymbol{\mu}_k(t) + \alpha(t) \cdot \frac{\partial \mathcal{J}(\Phi, \mathbf{x}_i)}{\partial \boldsymbol{\mu}_k},$$
$$\boldsymbol{\mu}_k(t+1) = \boldsymbol{\mu}_k(t) + \alpha(t) \cdot h(k, k^*) \cdot (\mathbf{x} - \boldsymbol{\mu}_k(t)).$$

Distribution continue de l'ensemble d'apprentissage

Si l'on considère le cas plus général où les données sont les réalisations d'une distribution de probabilité continue, f, la fonction d'énergie 2.12 devient

$$W = \int_{\mathbf{x}} \sum_{k=1}^{K} h(k, k^*) \cdot \|\mathbf{x} - \boldsymbol{\mu}_k\|^2 \cdot f(\mathbf{x}) dx. \tag{2.13}$$

Dans ce cas, il semble douteux de négliger les bords du pavage de Voronoï, et une descente de gradient stochastique n'aboutit pas à la forme connue de l'algorithme SOM.

Par contre, une descente de gradient stochastique sur ce critère (Kohonen 1991) amène à la formulation d'une nouvelle règle d'apprentissage, qui est la règle de base de l'algorithme SOM agrémenté d'un terme additif. Cette nouvelle règle semble produire de bons résultats, avec une rapidité plus grande, mais Kohonen objecte qu'elle n'a plus de justification biologique (Dans le cas où la règle initiale serait justifiée biologiquement...).

Ce problème de critère pour les cartes de Kohonen a été solutionné dans le cas général par Erwin *et al.* (1992), qui a démontré que la règle d'apprentissage originale ne peut être déduite, comme descente de gradient stochastique, d'aucun critère.

Une autre idée (Tolat 1990) consiste à tenter d'associer une fonction d'énergie à chaque neurone et de dériver la règle d'apprentissage de ce système de fonctions d'énergie. La proposition de Tolat est inexacte (Erwin *et al.* 1992), mais Erwin a montré qu'il est effectivement possible de déduire l'algorithme de Kohonen d'un système de fonctions d'énergie. Notons que ces fonctions n'ont pas une forme simple.

2.2.3 Critères de qualité pour la cartographie associative

Comment juger la qualité du résultat produit par l'algorithme SOM ? Cette question en amène une autre, qui est de savoir ce qui est cherché lorsqu'on utilise l'algo-

rithme SOM. Selon moi, deux caractéristiques rendent l'algorithme intéressant pour une utilisation en analyse de données :

1. La conservation de la topologie, qui permet de "projeter" sur la carte un certain type d'information contenue dans les données. Cette propriété constitue l'originalité de ce que nous appellons la cartographie associative. Elle se définit de manière assez intuitive en disant qu'il est souhaitable que deux vecteurs d'entrée proches dans l'espace d'entrée activent deux neurones proches dans l'espace de sortie. Nous verrons en fin de chapitre comment cette propriété peut être mise à profit pour réduire la dimension des données comme le font les techniques de positionnement multidimensionnel ou l'analyse en composantes principales.

2. Le résumé d'un grand ensemble de données par un nombre plus petit de vecteurs prototypes (cellules, neurones). Cette caractéristique rejoint les problèmatiques que l'on peut trouver en classification automatique ou bien en quantification vectorielle.

Ainsi l'algorithme SOM cherche à satisfaire deux exigences. D'une part obtenir une carte qui préserve la topologie "le mieux possible" et d'autre part trouver des vecteurs prototypes qui résument "au mieux" les données. Ces deux desirata sont bien traduit numériquement par le critère 2.12, qui peut être considéré comme la fonction optimisée par SOM, lorsque les données sont issues d'une distribution de probabilité discrète. En effet, le critère est la somme du produit de deux facteurs. Le premier facteur, $h(k, k^*)$, contribue à la conservation de la topologie et le second, $\|\mathbf{x}_i - \boldsymbol{\mu}_k\|^2$, à une bonne représentation des entrées par les prototypes.

D'une manière générale, le critère,

$$W = \frac{1}{2N} \sum_{i=1}^{N} \sum_{k=1}^{K} h(k, k^*) \cdot \|\mathbf{x}_i - \boldsymbol{\mu}_k\|^2 \tag{2.14}$$

peut être utilisé comme mesure de qualité, mais suscite plusieurs problèmes :

- Le critère juge un résultat d'ensemble et ne permet pas de faire la part entre conservation de la topologie et résumé des données.

- Dans le cas général (les données sont la réalisation d'une distribution de probabilité continue), ce critère n'est pas exactement optimisé par l'algorithme SOM et il n'y a pas de raison de l'utiliser comme mesure de qualité plutôt qu'un autre.

Dans les sections suivantes, des mesures de qualité qui traitent séparement les deux caractéristiques des algorithmes de cartographie associative sont présentées.

Critères d'approximation

Les critères d'approximation quantifient numériquement la qualité du placement des prototypes dans l'espace des entrées. De manière plus ou moins intuitive, Fritzke (1993) définit l'erreur de distribution et l'erreur de discrétisation :

- **Erreur de distribution :** La définition de cette erreur part de l'*a priori* que chaque cellule devrait avoir la même probabilité d'être activée. Le nombre d'entrée activant une cellule peut être considéré comme une v.a. La variance de cette variable aléatoire mesure l'erreur de distribution. Ainsi, si tous les prototypes représentent ("sont activés par") le même nombre de vecteurs d'entrée, l'erreur de distribution est nulle. Le but recherché sera naturellement d'obtenir la plus petite erreur de distribution possible. Ce critère pénalisent les solutions qui contiennent des classes vides.

- **Erreur de discrétisation :** L'estimateur de cette erreur (Fritzke 1993) est le critère bien connu de la variance intra classe (ou critère des k-means) très utilisé en classification automatique. Ce parallèle montre les liens forts qui existent entre cartographie associative et classification automatique.

Tomasini (1993) considère que les entrées sont issues d'une distribution f et que l'on cherche à approximer cette distribution par g, un mélange de K gaussiennes de même variance, dont les moyennes sont les vecteur prototypes, $\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_K$. Dans le cadre de la théorie de l'information, le contraste de Kullback est une mesure de proximité entre deux distribution :

$$W = \int f(\mathbf{x}) \cdot \log \frac{f(\mathbf{x})}{g(\mathbf{x})} dx. \tag{2.15}$$

Tomasini utilise donc cette fonction pour mesurer la qualité de l'approximation.

Critères topologiques

Le concept d'ordre est bien défini dans le cas unidimensionel, c'est-à-dire lorsque les vecteurs d'entrée appartiennent à l'ensemble des réels et que la structure du réseau est linéaire. Dans ce cas précis, la conservation de la topologie peut être quantifiée par le critère :

$$J(\Phi) = \sum_{i=2}^{K} \|\boldsymbol{\mu}_i - \boldsymbol{\mu}_{i-1}\| - \|\boldsymbol{\mu}_K - \boldsymbol{\mu}_1\|.$$

La topologie est alors parfaitement conseervée lorsque $J = 0$.

Dans le cas général, le concept de conservation de la topologie est moins évident, et relève de la traduction numérique des deux principes suivants :

- deux vecteurs d'entrée proches dans l'espace des entrées devraient activer deux cellules proches sur la carte. On parle alors de préservation du voisinage ;

– deux cellules voisines sur la carte devraient être proches dans l'espace des entrées. C'est la préservation du voisinage inverse.

De nombreux auteurs ont proposé des critères de mesure de préservation de la topologie. Citons entre autres, Bauer et Pawelzik (1992) pour le produit topographique, Zhao (1992) pour le Glogal D-Display, Zrehen et Blayo (1992) pour les critères Bmap et Dmap.

Dans ce mémoire, nous nous intéresserons en particulier au critère de la longueur moyenne des arêtes proposé par Durbin et Willshaw (1987) pour l'algorithme de *l'elastic net* et repris par Durbin et Mitchinson (1990) pour le modèle des cartes corticales. Tomasini (1993) a proposé une généralisation de ce critère qui, traduite dans le formalisme que nous utilisons, s'exprime comme :

$$MEL = \sum_{i=1}^{K} \sum_{j=1}^{K} h(i,j) \cdot \|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\|^2, \tag{2.16}$$

où h est la fonction de voisinage.

2.2.4 Autres approches en cartographie associative

Dans le cas général, l'algorithme SOM n'est pas une descente de gradient stochastique qui minimise un critère bien défini. Malgré cet inconvénient, cet algorithme a suscité beaucoup d'intérêt et a donné lieu à quelques applications pratiques.

D'autres auteurs, originaires de domaines de recherche diverses, ont proposé des algorithmes optimisant un critère, et produisant des résultats similaires à ceux de Kohonen. Luttrell (1990) a proposé un algorithme basé sur les principes de la quantification vectorielle. L'algorithme de Durbin et Willshaw (1987) est présenté comme une alternative pour résoudre le problème du voyageur de commerce. Durbin et Mitchinson (1990) suggère un algorithme, parent de *l'elastic net*, qui est une autre tentative de modélisation du phénomène de conservation topologique. Tomasini (1993) dérive un algorithme ayant des propriétés similaires à partir d'une mesure d'information. Toutes ces propositions ont pour mérite d'éclairer d'un point de vue algorithmique les principes de la conservation de topologie.

Le principe de robustesse de la couche cachée

L'article de Luttrell (Luttrell 1990) présente les propriétés de conservation topologique du point de vue de la quantification vectorielle. L'algorithme de Kohonen est déduit, après quelques approximations, d'un critère de quantification robuste.

En quantification vectorielle, un signal stochastique, $\mathbf{x} \in \mathbb{R}^d$, est associé à un code $\boldsymbol{\mu} \in \mathbb{R}^d$. L'ensemble des vecteurs code est fini. Le but consiste donc à coder un ensemble de signaux stochastiques, en un nombre fini de codes. Ceci est réalisé en minimisant une certaine fonction de coût. L'algorithme Linde-Buzzo-Gray, mimimise par exemple

$$E = \int \|\boldsymbol{\mu}_{\mathbf{y}(\mathbf{x})} - \mathbf{x}\|^2 \cdot f(\mathbf{x}) dx, \tag{2.17}$$

où

- $f(\mathbf{x})$ est la distribution de probabilité des signaux d'entrée
- $\mathbf{y}$ est l'application qui associe chaque vecteur d'entrée à un code

Luttrell introduit une étape intermédiaire dans le processus de quantification. Il suppose que l'étape de codage est perturbée par un bruit. Dans ce cas, la fonction coût précédente devient :

$$E_1 = \int\int \|\boldsymbol{\mu}_{(h(x)+n)} - \mathbf{x}\|^2 \cdot \pi(n) \cdot f(\mathbf{x}) dn d\mathbf{x} \tag{2.18}$$

avec n, le bruit distribué suivant la loi $\pi(n)$

En minimisant E_1 par une méthode de gradient stochastique, et en faisant quelques approximations, Luttrell déduit l'algorithme de Kohonen. Erwin *et al.* (1992) ont objecté que

> the approximation is only reasonable in the least intersecting case, i.e when the map is already well ordered. Although the approximative energy function 2.18 can be used to describe the dynamics of the algorithm after an ordered, or mostly ordered map has formed, it is not useful for describing the ordering of an initially highly disordered map,...

Malgré des approximations abusives, l'article de Luttrell est intéressant car il propose une interprétation du phénomène de conservation topologique : définir des relations topologiques revient à définir *a priori* des relations de proximités entre les classes. C'est-à-dire, prendre en compte le fait que lorsque un individu $\mathbf{x}_i$ est classé dans la classe k de manière erronnée, il appartient sûrement à une classe voisine de k. La définition de ces relations *a priori* entre les classes entraîne une certaine robustesse en quantification vectorielle.

Optimisation multi-critère

Dans la présentation des critères de qualité en cartographie associative, nous avons mis en évidence deux types de mesures numériques :

- certaines étaient liées à la conservation de la topologie. Nous les indiquerons par la lettre T (Topologie) ;
- d'autres relevaient de l'approximation des données par un ensemble de prototypes. Dans la suite elles seront symbolisées par la lettre Q (Quantification).

De nombreux critères de cartographie associative peuvent être obtenus en combinant ces deux types de mesure :

$$C = Q + \beta T \tag{2.19}$$

où β est un paramètre qui pondère l'importance d'un terme par rapport à l'autre. Généralement, le coefficient β décroît au cours des itérations, ce qui a pour effet de

donner de moins en moins d'importance au terme induisant la conservation de la topologie. Notons la similitude de ce procédé avec la réduction de la taille du voisinage au cours des itétations dans l'algorithme SOM. Dans certains cas (voir chapitre 1), cette décroissance peut être interprétée comme l'abaissement du paramètre température dans des algorithmes de recuit simulé.

Le double critère C étant défini, reste à trouver un algorithme pour trouver les paramètres $\Phi = \{\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_K\}$ optimum. Dans le domaine des réseaux de neurones la tendance naturelle consiste à choisir une descente de gradient stochastique, mais rien n'empêche de recourir à d'autres méthodes d'optimisation.

Exemple 2.4 (Durbin et Willshaw 1987) L'algorithme de l'*elastic net* est destiné à résoudre le problème du voyageur de commerce : cela consiste à trouver le plus court chemin passant par un ensemble de N villes de coordonnées $\mathbf{x}_1, \cdots, \mathbf{x}_N$. Chaque ville doit être visitée une seule fois. Pour définir la structure du chemin, Durbin et Willshaw proposent d'utiliser un polygone de K sommets de coordonnées $\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_K$. Les sommets du polygone sont plus nombreux que les villes. En optimisant le critère

$$C = -\alpha \cdot \sigma \sum_{i=1}^{N} \log \sum_{j=1}^{K} f_j(\mathbf{x}_i | \boldsymbol{\mu}_j, \sigma) + \beta \sum_{j=1}^{K} \|\boldsymbol{\mu}_{j+1} - \boldsymbol{\mu}_j\|^2 \; avec \; K > N \qquad (2.20)$$

où f_j est une gaussienne de moyenne $\boldsymbol{\mu}_j$ est de matrice de variance $\boldsymbol{\Sigma} = \sigma \cdot I$, les sommets $\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_K$ forment un polygone qui passe par toutes les villes (première partie du critère) et qui a tendance à avoir le périmètre le plus petit possible (second terme du critère), ce qui donne une solution du problème du voyageur de commerce. L'évolution des contours du périmètre faisant penser à aux déformations successives d'un élastique que l'on essaierait de faire passer par toutes les villes, l'algorithme justifie son nom de baptême.

△

Exemple 2.5 (Tomasini 1993) Dans son mémoire de thèse, Tomasini propose une fonction d'énergie pour la cartographie associative qui est une généralisation de celle proposée par Durbin et Willshaw (1987) :

$$C = \int_{\mathbf{X}} f(\mathbf{x}) \cdot \log \sum_{j=1}^{K} f_j(\mathbf{x}_i | \boldsymbol{\mu}_j, \sigma) d\mathbf{x} + \beta \sum_{i=1}^{K} \sum_{j=1}^{K} h(i,j) \|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\|^2, \qquad (2.21)$$

avec f la densité inconnue des données, f_j une gaussienne de moyenne $\boldsymbol{\mu}_j$ est de matrice de variance $\boldsymbol{\Sigma} = \sigma \cdot I$, et h la fonction de voisinage de la carte. Ce critère est une généralisation de l'*elastic net* dans le sens où il permet d'obtenir des cartes de dimension supérieure à un. Le fait que la densité f ne soit pas connue est contourné par l'utilisation d'un algorithme de gradient stochastique.

△

2.3 Modèle de mélange et cartes de Kohonen

L'approche de la cartographie associative par l'optimisation conjointe de deux critères

$$C = Q + \beta T$$

peut être vue comme une optimisation sous contrainte. Dans ce cas, C est le lagrangien qui correspond au problème :

$$\min Q \quad avec \quad T \leq c_1, \tag{2.22}$$

ou bien au problème dual :

$$\min T \quad avec \quad Q \leq c_2. \tag{2.23}$$

La minimisation d'un critère de "résumé des données" sous contrainte de conservation topologique nous a interessé car ce problème peut se transposer dans le cadre probabiliste de la classification automatique présenté au premier chapitre. Dans ce contexte, on considère que les données sont la réalisation d'un échantillon i.i.d. issu d'un mélange de gaussiennes $f_k(\mathbf{x}_i|\theta_k)$ de paramètres θ_k. Ce modèle permet de suggérer d'une part un critère Q (vraisemblance classifiante ou vraisemblance) et d'autre part des algorithmes efficaces pour optimiser ce critère (EM, CEM ...). Notons que les algorithmes de la famille de EM ne prennent pas en compte les contraintes topologiques. Adapter ce genre d'algorithmes pour la cartographie associative consiste donc à intégrer des contraintes supplémentaires. Dans cette section, deux manières originales de considérer les contraintes topologiques dans le cadre des modèles de mélanges sont présentées :

- ajout d'une étape supplémentaire dans l'algorithme EM, pour forcer le respect des contraintes topologiques ;
- adaptation de l'algorithme EM pour optimiser une vraisemblance pénalisée.

2.3.1 Contraindre la matrice de classification

Imposer des contraintes sur le type de partition recherchée revient à supposer que la matrice de classification appartient à un sous-ensemble donné de l'ensemble des matrices de classification floue. Dans cette section nous proposons une solution d'adaptation de l'algorithme EM qui permet d'intégrer certaines contraintes.

Nous noterons $\mathcal{C}_{floue}$, l'ensemble des matrices de classification floue décrivant une partition de N individus en K classes :

$$\mathcal{C}_{floue} = \{\mathbf{c} = \{c_{ik}\}_{(i;k)=(1,..,N;1,..,K)} / c_{ik} \in [0,1], \forall i \sum_{k=1}^{K} c_{ik} = 1\}$$

L'ensemble des matrices de classification, $\mathcal{C}_{dure}$, est le sous-ensemble de $\mathcal{C}_{floue}$ tel que :

$$\mathcal{C}_{dure} = \{\mathbf{c} \in \mathcal{C}_{floue} / c_{ik} \in \{0,1\}\}$$

Définissons aussi d la distance sur $\mathcal{C}_{floue}$ telle que :

$$d(\mathbf{c}, \mathbf{c}') = \sum_{i=1}^{N} \sum_{k=1}^{K} (c_{ik} - c_{ik}')^2$$

Une interprétation de l'algorithme CEM

Du point de vue de la classification, nous avons vu dans le premier chapitre que l'algorithme EM pouvait être considéré comme un algorithme de classification floue, optimisant un certain critère L, qui est la vraisemblance (Hathaway 1986, Celeux et Govaert 1994) :

$$L(\mathbf{c}, \Phi) = \sum_{i=1}^{n} \sum_{k=1}^{K} c_{ik} \log p_k f_k(\mathbf{x}_i|\theta_k) - \sum_{i=1}^{n} \sum_{k=1}^{K} c_{ik} \log c_{ik}, \tag{2.24}$$

La matrice $\mathbf{c} = \{c_{ik}\}_{(i;k)=(1,..,N;1,..,K)}$ obtenue appartient à $\mathcal{C}_{floue}$.

Lorsqu'une matrice de classification $\mathbf{c}$ appartenant à $\mathcal{C}_{dure}$ est recherchée, le critère L devient :

$$L(\mathbf{c}, \Phi) = \sum_{i=1}^{n} \sum_{k=1}^{K} c_{ik} \log p_k f_k(\mathbf{x}_i|\theta_k)$$

Pour optimiser ce nouveau critère, appelé "vraisemblance classifiante", l'algorithme CEM (voir Chapitre 1) peut être utilisé avantageusement. Celui-ci, basé sur l'algorithme EM introduit une étape intermédiaire, dite de classification. Cette étape de classification peut être interprétée comme la transformation de la matrice de classification floue, $\mathbf{c}^f$, obtenue à l'étape E, en une matrice de classification dure, $\mathbf{c}^d$, par une projection de $\mathcal{C}_{floue}$ dans $\mathcal{C}_{dure}$:

- **Etape C :** Transformation de la matrice $\mathbf{c}^f$ de classification floue, en matrice $\mathbf{c}^d$ de classification dure. La nouvelle matrice $\mathbf{c}^d$ est l'élément de $\mathcal{C}_{dure}$ le plus proche de $\mathbf{c}^f$ au sens de d :

  $$\mathbf{c}^d = arg \min_{\mathbf{c} \in \mathcal{C}_{dure}} d(\mathbf{c}, \mathbf{c}^f)$$

  Si l'on interprète c_{ik} comme la probabilité que l'individu i appartienne à la classe k (cf. Chapitre 1), cette étape revient à affecter chaque élément à la classe la plus probable :

  $$\begin{aligned}
  \mathbf{c}^d &= arg \min_{\mathbf{c} \in \mathcal{C}_{dure}} \sum_{i=1}^{N} \sum_{k=1}^{K} (c_{ik}^f - c_{ik})^2 \\
  \mathbf{c}^d &= arg \min_{\mathbf{c} \in \mathcal{C}_{dure}} \sum_{i=1}^{N} \sum_{k=1}^{K} (c_{ik}^f)^2 - 2 \cdot c_{ik}^f \cdot c_{ik} + 1 \\
  \mathbf{c}^d &= arg \max_{\mathbf{c} \in \mathcal{C}_{dure}} \sum_{i=1}^{N} \sum_{k=1}^{K} c_{ik}^f \cdot c_{ik}
  \end{aligned}$$

d'où

$$c^d_{ik^*} = 1, \ si \ c^f_{ik^*} = \max_k c^f_{ik}$$

De la même manière l'algorithme EM peut être modifié pour devenir un algorithme de cartographie associative. L'étape intermédiaire consiste alors à introduire les contraintes de conservations topologiques, c'est-à-dire, à projeter la matrice de classification floue, obtenue à l'étape E, dans l'espace des matrice de classification qui respectent les contraintes topologiques. Cette idée est à la base de l'algorithme *Topology Preserving EM algorithm* (Ambroise et Govaert 1996).

L'algorithme TPEM

Comment introduire les contraintes de conservations topologiques dans l'expression de la matrice de classification?

La solution proposée dans la suite de cette section s'inspire de la forme du critère optimisé dans le cas discret par l'algorithme de Kohonen :

$$W = \frac{1}{2} \sum_{i=1}^{N} \sum_{k=1}^{K} h(k, k^*) \cdot \|\mathbf{x}_i - \boldsymbol{\mu}_k\|^2 \tag{2.25}$$

où k^* est l'indice de l'unité gagnante pour l'exemple $\mathbf{x}_i$:

$$k^* = arg \min_k \|\mathbf{x}_i - \boldsymbol{\mu}_k\|,$$

et h, la fonction de voisinage.

Ce critère offre de nombreuses similitudes avec la vraisemblance classifiante lorsque les matrices de variance des composantes du mélange gaussien sont supposées être toutes sphériques, et de même volume ($\boldsymbol{\Sigma}_k = \sigma \cdot I$). En effet, dans ce cas la vraisemblance classifiante est équivalente au critère des k-means :

$$CML(\mathbf{c}, \Phi) = \sum_{i=1}^{n} \sum_{k=1}^{K} c_{ik} \cdot \|\mathbf{x}_i - \boldsymbol{\mu}_k\|^2.$$

Les critères W et CML ne diffèrent que par les coefficients pondérateurs de leur somme respective (les c_{ik} et les $h(k, k^*)$). Cette remarque amène à penser que le critère W peut être interprété comme un critère de classification, où les coefficients $h(k, k^*)$ décrivent une partition qui tient compte des contraintes topologiques. Ceci suggère l'utilisation des coefficients $h(k, k^*)$ pour construire une matrice de classification prenant en compte les contraintes topologiques. Cette idée est à la base de l'algorithme baptisé Topology Preserving EM.

Notons $\mathcal{C}_\sigma$ l'ensemble des matrices de classification de N lignes et K colonnes, qui intègrent les contraintes topologique pour une largeur de voisinage σ. Chaque ligne $\mathbf{c}_i$ d'une matrice de cet ensemble est de la forme :

$$\mathbf{c}_i = [c_{i1}, c_{i2}, \cdots, c_{iK}]$$

où $c_{ik} = h(k, k^*(i))$.

Exemple 2.6 Considérons 5 classes réparties régulièrement sur une même ligne dans l'espace de sortie :

$$1 - - - - 2 - - - - 3 - - - - 4 - - - - 5$$

Une matrice des distances inter-classes possible est :

$$\mathbf{D} = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 \\ 1 & 0 & 1 & 2 & 3 \\ 2 & 1 & 0 & 1 & 2 \\ 3 & 2 & 1 & 0 & 1 \\ 4 & 3 & 2 & 1 & 0 \end{pmatrix}.$$

Si on prend $\sigma = 1$ comme largeur de voisinage, deux classes k et ℓ sont dites voisines si

$$d_{k\ell} \leq 1.$$

Dans ce cas les relations de voisinages peuvent être résumées par une matrice binaire :

$$\mathbf{V} = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix}.$$

où $v_{k\ell} = 1$ si les classes k et ℓ sont voisines, et $v_{k\ell} = 0$ sinon.

Chaque ligne des matrices de $\mathcal{C}_\sigma$ sera choisie parmi les 5 lignes suivantes :

$$\begin{pmatrix} 0.5 & 0.5 & 0 & 0 & 0 \\ 0.33 & 0.33 & 0.33 & 0 & 0 \\ 0 & 0.33 & 0.33 & 0.33 & 0 \\ 0 & 0 & 0.33 & 0.33 & 0.33 \\ 0 & 0 & 0 & 0.5 & 0.5 \end{pmatrix}.$$

Une matrice de classification décrivant la partition de 10 individus en 5 classes et appartenant à $\mathcal{C}_\sigma$ pourrait être :

$$\mathbf{c} = \begin{pmatrix} 0 & 0 & 0.33 & 0.33 & 0.33 \\ 0 & 0 & 0.33 & 0.33 & 0.33 \\ 0.5 & 0.5 & 0 & 0 & 0 \\ 0.5 & 0.5 & 0 & 0 & 0 \\ 0.5 & 0.5 & 0 & 0 & 0 \\ 0.33 & 0.33 & 0.33 & 0 & 0 \\ 0 & 0.33 & 0.33 & 0.33 & 0 \\ 0 & 0 & 0.33 & 0.33 & 0.33 \\ 0 & 0 & 0.33 & 0.33 & 0.33 \\ 0 & 0 & 0 & 0.5 & 0.5 \end{pmatrix}.$$

△

L'algorithme TPEM est une généralisation de l'algorithme "Batch Map" (Kohonen 1993) : lorsque les matrices de variance-covariance des composantes du mélange sont contraintes à être toutes de la forme $\mathbf{\Sigma}_k = \sigma \cdot I$, l'algorithme TPEM se réduit à l'algorithme de Kohonen non séquentiel (batch). Comme nous le verrons cet algorithme peut être utilisé soit pour réaliser de la classification robuste, soit pour réduire la dimension des données, tout comme les cartes de Kohonen. Il s'articule comme suit :

1. **Initialisation :**
   - Une architecture de la carte dans l'espace de sortie est choisie. Celle-ci modèlise certains *a priori* sur les relations entre les classes.
   - σ_0, une largeur initiale de voisinage, est choisie.
   - Φ, les paramètres du mélange sont initialisés.
2. **Itération :** A chaque itération, les 3 étapes suivantes sont exécutées. Le processus stoppe lorsque la matrice de classification reste inchangée entre deux itérations :
   - **Etape E :** C'est l'étape E de l'algorithme EM, qui permet d'obtenir une matrice $\mathbf{c}^f$ de classification floue.
   - **Etape TP :** Transformation de la matrice $\mathbf{c}^f$ de classification floue, en matrice $\mathbf{c}^\sigma$ appartenant à $\mathcal{C}_\sigma$, c'est-à-dire intégrant les contraintes topologiques.

     On commence par chercher la matrice $\mathbf{c}^d$, la plus proche de $\mathbf{c}^f$ au sens de la distance d :
     $$\mathbf{c}^d = arg \min_{\mathbf{c} \in \mathcal{C}_{dure}} d(\mathbf{c}, \mathbf{c}^f).$$
     Ensuite on calcule la matrice de classification $\mathbf{c}^\sigma$ qui prend en compte les contraintes topologique pour une largeur de voisinage σ,
     $$\mathbf{c}^\sigma = arg \min_{\mathbf{c} \in \mathcal{C}_\sigma} d(\mathbf{c}, \mathbf{c}^d).$$
     Intuitivement, cela revient à affecter chaque individu à la classe la plus probable a posteriori, ainsi qu'aux classes voisines (dans l'espace de sortie). La largeur de voisinage σ est réduite.
   - **Etape M :** Calcul de nouveaux paramètres Φ^+ avec $\mathbf{c}^d$ fixée :
     $$\Phi^+ = arg \max_{\Phi} L(\mathbf{c}^\sigma, \Phi).$$

Décrivons le fonctionnement de cet algorithme par quelques remarques :

1. La préservation de la topologie est induite par la matrice de voisinage $\mathbf{H}$. Kohonen (1984) suggère de commencer avec un voisinage large et de laisser la taille

du voisinage décroitre au cours des itérations. De nombreux modes de décroissance de la taille de voisinage peuvent être pris en compte. Dans la suite, la fonction de décroissance utilisée est toujours linéaire :

$$\sigma_{m+1} = a\sigma_m \ , 0 < a \leq 1$$

où m le nombre d'itérations.

2. Dans l'évolution du processus deux phases sont distinguables :

   (a) De manière qualitative, une phase d'auto-organisation des prototypes les uns par rapport aux autres se déroule en premier. Pendant cette phase, la fonction de voisinage influence la calcul de la matrice de classification de façon non négligeable. Quantitativement parlant, la largeur du voisinage influent est supérieur à l'unité ($\sigma_m > 1$).

   (b) Une phase de classification succède au processus auto-organisateur. Lorsque la largeur du voisinage est strictement inférieur à un, c'est-à-dire lorsque le nombre d'itérations $m \geq \{\frac{\log \sigma_0}{loga} + 1\}$, l'algorithme se réduit alors à l'algorithme CEM.

3. D'un point de vue classification, une matrice de classification floue résulte de chaque itération durant la phase d'auto-organisation. Aussitôt que la phase de classification commence, les itérations successives produisent des matrices de classification dure.

De la seconde remarque, on déduit que l'algorithme converge en un nombre fini d'itérations. En effet, l'algorithme TPEM se réduit à l'algorithme CEM au bout d'un certain nombre d'itérations, et d'après Celeux et Govaert (1992), l'algorithme CEM génère une sequence convergente des paramètres $(\mathbf{c}_m, p^m, \boldsymbol{\mu}^m, \Sigma^m)$.

Exemple 2.7 La figure 2.5 donne un exemple d'évolution de l'algorithme TPEM. Le jeu de données est composé de 500 individus issus d'une distribution uniforme supportée par un carré. La topologie est définie dans l'espace de sortie par une grille de 5 par 5. L'algorithme commence sa phase de classification à la 12^{e} itération.

△

D'un point de vue pratique, la solution fournie par l'algorithme TPEM dépend de la vitesse de décroissance de la largeur de voisinage influent, et aussi de la partition initiale. Il arrive que l'algorithme s'arrête dans la phase d'auto-organisation et produisent alors une valeur médiocre du critère CML (comparé au résultats obtenus avec CEM). Pour remédier à ce problème plusieurs solutions sont envisageables :

– la manière la plus simple consiste à lancer l'algorithme plusieurs fois à partir d'initialisations différentes ;

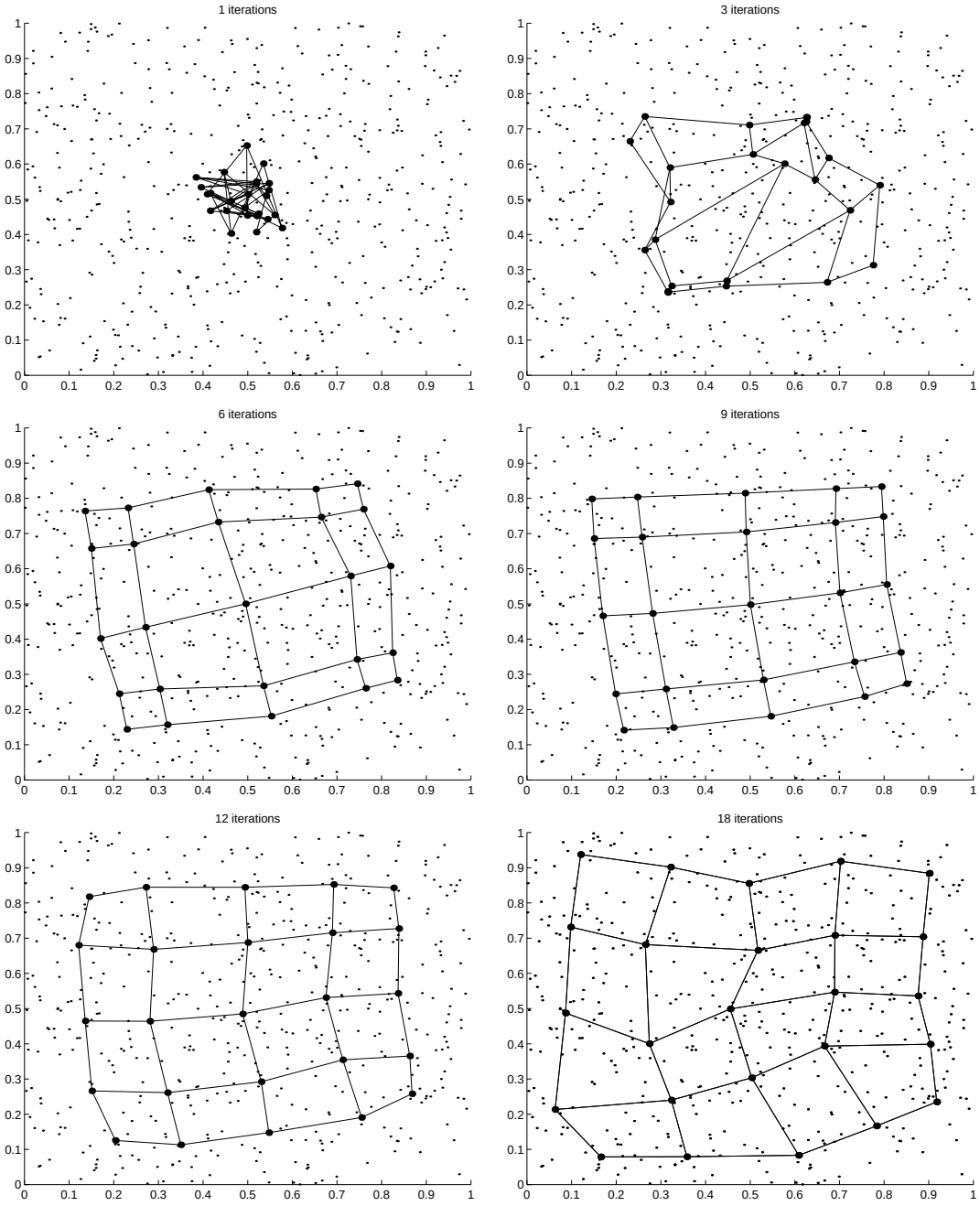


FIG. 2.5 - : Evolution de l'algorithme TPEM

- Kohonen (1994) conseille pour son algorithme "Batch Map", d'initaliser les prototypes sur le premier plan d'une Analyse en Composantes Principales en respectant les relations de voisinages décrites dans l'espace de sortie ;

- une autre solution possible consiste à introduire un facteur aléatoire dans le fonctionnement de l'algorithme. Cette démarche fait l'objet de la section suivante.

Une version stochastique de l'algorithme TPEM

L'algorithme Stochastic Topology Preserving EM (STPEM) vise à limiter les effets de la configuration de départ sur le résultat produit à la convergence. Cet algorithme utilise le principe d'affectation stochastique (Random Input Principle, RIP) proposé par Celeux et Diebolt (1985) et qui est à la base de l'algorithme SEM (Stochastic EM algorithm). La structure de l'algorithme est identique à celle de TPEM, si l'on excepte cette étape supplémentaire d'affectation stochastique :

1. **Initialisation** identique à celle de TPEM.

2. **Itérations :** A chaque itération, les 4 étapes suivantes sont executées. Le processus stoppe lorsque la matrice de classification reste inchangée entre deux itérations.

   (a) **Etape E.**

   (b) **Etape TP.** Une matrice de classification $\mathbf{c}^{\sigma}$, qui intègre les les contraintes topologiques, est calculée.

   (c) **Etape S** dite d'affectation stochastique. Une nouvelle matrice de classification $\mathbf{c}^{s}$ est déduite de la matrice $\mathbf{c}^{\sigma}$. Chaque individu $\mathbf{x}_i$ est affecté à une unique classe selon un tirage au hasard suivant la distribution multinomiale de paramètre $(c_{i1}^{\sigma}, \cdots, c_{iK}^{\sigma})$.

   (d) **Etape M**. Comme dans l'algorithme TPEM, les paramètres du mélange sont calculés mais sur la base des coefficients de la matrices $\mathbf{c}^{s}$.

L'algorithme STPEM possède quelques similitudes avec le recuit simulé (Klein et Dubes 1989) utilisé en classification automatique. Il est en effet possible d'établir un parallèle entre la largeur du voisinage influent σ_m dans STPEM et la température dans les algorithmes de recuit simulé. Plus la température (respectivement, la largeur du voisinage) devient petite, moins l'effet stochastique est important. Ainsi, lorsque cette largeur est plus petite que l'unité, l'algorithme STPEM se réduit à CEM. Comme dans le cas de l'algorithme TPEM, STPEM comprend deux phases distinctes. La première phase est une phase d'auto-organisation, durant laquelle le critère n'augmente pas forcement à chaque itération et le facteur aléatoire joue un rôle non négligeable. La deuxième phase est une succession d'itérations CEM, ce qui entraîne la convergence.

L'algorithme TPEM et les tableaux de dissimilarités

Parfois les données, au lieu de se présenter sous la forme d'un tableau de N objets décrits par d variables, sont disponibles sous la forme d'une matrice de dissimilarités. Les dissimilarités mesurent les différences entre toutes les paires d'objets. Ainsi les dissimilarités entre N objets sont spécifiées par une matrice $N \times N$, $\delta = \{\delta_{ij}\}_{i,j=1..N}$. Ce genre de matrices peut avoir différentes origines :

- Le jugement humain peut souvent être traduit par des mesures de dissimilarités. Une personne amenée à quantifier la différence de confort entre deux voitures pourra, par exemple, choisir une chiffre entre 1 et 10 qui correspond à l'intensité de la différence perçue.

- Les données telles que les temps de transports entre des paires de villes se présentent naturellement sous la forme d'une matrice de dissimilarités.

- Enfin, notons qu'il est toujours possible de dériver une matrice de dissimilarités (les distances sont des dissimilarités) d'une structure individus/variables.

Dans certain cas, la transformation de la matrice de dissimilarités en matrice de distances (qui permet ensuite de passer à un tableau individus/variables par une analyse factorielle) n'entraîne pas une grande perte d'informations et des méthodes d'analyse de données "classiques" sont applicables (ACP, k-means...). Avec d'autres matrices de dissimilarités, il est préférable de ne pas utiliser de prétraitement et de faire appel à des méthodes spécifiques à ce genre de données. Les techniques de positionnement multidimensionel ("Multi Dimensional Scaling" ou MDS en anglais) offrent une alternative qui permet de représenter les données dans le plan. Parmi ces techniques de positionnement multidimensionel, les approches métriques sont couramment opposées aux approches non métriques. Les premières produisent des représentations préservant au mieux l'information quantitative (Sammon 1969) contenue dans les données, alors que les secondes privilégient l'information qualitative (Kruskal 1964).

Adapter les algorithmes TPEM et STPEM pour traiter des matrices de dissimilarités offre une nouvelle alternative pour classer ou bien représenter ce type de données (Ambroise et Govaert 1996). Quelles sont les moyens et les conséquences de cette adaptation ? Un ensemble d'objets muni d'une mesure de dissimilarités ne permet pas de calculer un centre de gravité (calcul des prototypes) d'un sous ensemble de ces objets. De plus, aucun produit scalaire n'étant défini, les matrices de variances covariances ne peuvent être calculées. En conséquence, il est impossible de conserver l'algorithme initial ainsi que l'interprétation liée au modèle gaussien. Par contre, d'un point de vue algorithmique, il est possible d'utiliser une nouvelle définition du prototype qui permet de faire tourner l'algorithme et d'obtenir une partition. Le prototype d'une classe est le meilleur représentant de cette classe. La définition numérique de ce concept, utilisant des dissimilarités, consiste à prendre comme prototype d'une

classe, l'objet de cette classe qui est le plus proche en moyenne de tous les autres objets de cette classe. L'objet i^* est le prototype de la classe k si :

$$i^* = arg \min_i \sum_{j \in Classe k} \delta_{ij} \tag{2.26}$$

Cette définition permet de proposer l'algorithme STPEM suivant :

1. **Initialisation** : Identique à celle de TPEM sauf en ce qui concerne les prototypes. Dans ce cas, K objets sont choisis au hasard comme prototypes.

2. **Itérations :** A chaque itération, les 3 étapes suivantes sont exécutées. Le processus stoppe lorsque la matrice de classification reste inchangée entre deux itérations.

   (a) **Etape E.** La matrice, $\mathbf{D} = \{d_{ik}\}_{(i;k)=(1,..,N;1,..,K)}$, des dissimilarités (dans l'espace des entrées) entre les N objets et les K prototypes est calculée.

   (b) **Etape TP.** Chacun des N objets est affecté au prototype le plus proche. Une matrice de classification dure $\mathbf{c}^d$ résulte de ce partitionnement.

   Une matrice de classification $\mathbf{c}_h$, qui intègre les les contraintes topologiques, est calculée :

   $$\mathbf{c}^\sigma = arg \min_{\mathbf{c} \in \mathcal{C}_\sigma} d(\mathbf{c}, \mathbf{c}^d).$$

   (c) **Etape S** dite d'affectation stochastique. Une nouvelle matrice de classification $\mathbf{c}_s$ est déduite de la matrice $\mathbf{c}_h$ suivant le principe d'affectation stochastique.

   (d) **Etape M**. De nouveaux prototypes sont calculés sur la base de la partition décrite par $\mathbf{c}_s$:

2.3.2 Cartographie associative et vraisemblance pénalisée

L'idée de vraisemblance pénalisée a été introduite par Good et Gaskins (1971) dans le cadre de l'estimation non paramétrique de densité de probabilité. Le terme de pénalisation devant permettre d'obtenir une densité plus régulière. Cette approche a été reprise entre autre par Akaike (1978) et Silverman (1982).

Comme nous l'avons déjà mentionné, les contraintes topologiques peuvent être introduites sous la forme d'un terme pénalisant. C'est l'approche de Durbin et Mitchinson (1987) et de Tomasini (1993). L'originalité de la section suivante repose sur le choix de la vraisemblance, comme critère principal (ce qui permet de faire un lien avec les modèles de mélange), ainsi que sur le fait d'utiliser l'algorithme EM pour optimiser le critère pénalisé obtenu.

Un terme de pénalisation topologique

La log-vraisemblance pénalisée, dans le cadre d'un modèle de mélange, que nous considérons, s'exprime de la façon suivante :

$$U(\mathbf{c}, \Phi) = L(\mathbf{c}, \Phi) - \beta \cdot \sum_{k=1}^{K} \sum_{l=1}^{K} h_{k\ell} \cdot \|\boldsymbol{\mu}_k - \boldsymbol{\mu}_l\|^2, \tag{2.27}$$

où $L(\mathbf{c}, \Phi)$ est la log-vraisemblance, $\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_K$ les vecteurs moyennes des K gaussiennes composant le mélange, et $h_{k\ell} = h(k, \ell)$ la fonction de voisinage utilisée par l'algorithme TPEM.

Dans le cadre des modèles de mélange, l'algorithme EM peut être interprété comme un algorithme d'optimisation alternée (Hathaway 1986). Si nous utilisons cette technique d'optimisation pour trouver les paramètres qui maximisent le critère U, chaque itération de l'algorithme alterne les deux étape suivantes.

- **Etape E.** Calcul de la nouvelle matrice de classification :

  $$\mathbf{c}^+ = arg \max_{\mathbf{c}} U(\mathbf{c}, \Phi).$$

  Comme le terme pénalisant n'est pas fonction de la matrice de classification, cette étape est identique à celle de l'algorithme EM classique.

- **Etape M.** Décroissance du paramètre de pénalisation :

  $$\beta = a \cdot \beta, \tag{2.28}$$

  avec $a \in [0, 1]$, et calcul de nouveaux paramètres du mélange :

  $$\Phi^+ = arg \max_{\Phi} U(\mathbf{c}^+, \Phi).$$

  Les conditions nécessaires d'optimalité amènent les équations suivantes, pour chaque composante du mélange :

  $$\begin{cases} p_k^+ = \frac{\sum_{i=1}^{N} c_{ik}^+}{N} \\ \boldsymbol{\Sigma}_k^+ = \sum_{k=1}^{K} \sum_{i=1}^{n} c_{ik}^+ (\mathbf{x}_i - \boldsymbol{\mu}_k)(\mathbf{x}_i - \boldsymbol{\mu}_k)^t \\ \sum_{i=1}^{N} c_{ik}^+ (\boldsymbol{\mu}_k^+ - \mathbf{x}_i) + (4 \cdot \beta \boldsymbol{\Sigma}_k^+) \cdot \sum_{\ell=1}^{K} h_{k\ell} \cdot (\boldsymbol{\mu}_k^+ - \boldsymbol{\mu}_\ell^+) = 0 \end{cases}$$

  Ainsi, seul le calcul des vecteurs moyennes n'est pas immédiat et nécessite la résolution d'un système de K équations avec K vecteurs inconnus. C'est un système linéaire à $d \times K$ inconnues (d est la dimension des vecteurs $\boldsymbol{\mu}_k$), qui peut s'écrire de la façon suivante :

  $$\begin{cases} \mathbf{A}_{11} \boldsymbol{\mu}_1^+ + \cdots + \mathbf{A}_{1K} \boldsymbol{\mu}_K^+ = \mathbf{b}_1 \\ \vdots \\ \mathbf{A}_{K1} \boldsymbol{\mu}_1^+ + \cdots + \mathbf{A}_{KK} \boldsymbol{\mu}_K^+ = \mathbf{b}_K \end{cases}$$

  où

  - $\mathbf{A}_{kk} = I \cdot \sum_{i=1}^{N} c_{ik}^+ + 4 \cdot \beta \boldsymbol{\Sigma}_k^+ \cdot \sum_{\ell \neq k}^{K} h_{k\ell}$,

- $\mathbf{A}_{k\ell} = -4 \cdot \beta \cdot h_{k\ell} \boldsymbol{\Sigma}_k^+$ lorsque $k \neq \ell$,
- $\mathbf{b}_k = \sum_{i=1}^N c_{ik}^+ \mathbf{x}_i$.

Notons que lorsque le coefficient β approche de zéro, le terme pénalisant devient négligeable et les itérations de l'algorithme deviennent des itérations EM classique où :

$$\boldsymbol{\mu}_k^+ = \frac{\sum_{i=1}^N c_{ik}^+ \mathbf{x}_i}{\sum_{i=1}^N c_{ik}^+}.$$

Lorsque le nombre d'itérations est grand, le critère optimisé est la vraisemblance. Ainsi cet algorithme, que par commodité nous noterons EMP, ressemble de plus en plus à l'algorithme EM.

Comme dans le cas des cartes auto-organisatrices de Kohonen, le problème du réglage des paramètres se pose. Ainsi l'initialisation de la largeur de voisinage (qui détermine la fonction $h_{k\ell}$), de la valeur du paramètre β et la vitesse de décroissance de ce paramètre influencent fortement la qualité des résultats.

Par rapport aux algorithmes SOFM, TPEM et STPEM, cet algorithme pose un problème lorsque, initialisé avec un grand nombre de prototypes, on tente de trouver une partition d'un jeu de données qui ne possède pas de structure de classe. C'est par exemple le cas lorsque le jeu de données est la réalisation d'un distribution uniforme, ou d'une seule gaussienne. Dans ce cas, la matrice de classification a tendance à devenir totalement floue, et chaque individu appartient avec la même probabilité à toutes les classes. Nous n'avons malheureusement pas d'explication rigoureuse de ce phénomène.

Exemple 2.8 La figure 2.6 montre l'évolution de l'algorithme EMP utilisé pour optimiser une vraisemblance avec pénalisation topologique, lorsque le jeu de données est la réalisation d'une distribution uniforme supportée par un triangle.

On observe que tous les prototypes convergent vers le centre de gravité de la figure. Ce résultat est inexploitable pour une réduction de dimension des données et indique seulement que la distribution de probabilité dont est issue l'échantillon est difficilement approchée par un mélange de K gaussiennes distinctes.

Par contre, si l'on considére un jeu de données réalisation d'un mélange de K gaussiennes bi-dimensionnelles raisonnablement séparée (Figure 2.7), le résultat obtenu par l'algorithme semble cohérent et fournit une information sur les relations de voisinage entre les unités.

△

Pour retrouver des résultats similaires à ceux de l'algorithme SOFM, on peut forcer la matrice de classification à être une matrice de partition dure, comme dans l'algorithme CEM. Cette variante de l'algorithme, que nous noterons CEMP dans la suite, comporte alors trois étapes par itération.

Exemple 2.9 La figure 2.8 montre l'évolution de l'algorithme CEM utilisé pour optimiser une vraisemblance avec pénalisation topologique sur le jeu de données utilisé

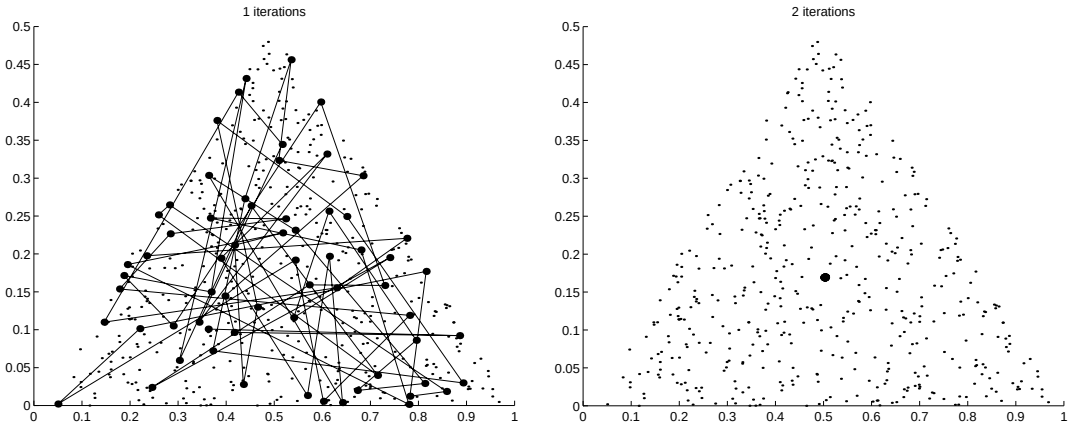


FIG. 2.6 - : Évolution de l'algorithme EMP. Le jeu de données est composé de 500 individus issus d'une distribution uniforme supportée par un triangle. La topologie est définie dans l'espace de sortie par un graphe linéaire de 64 unités.

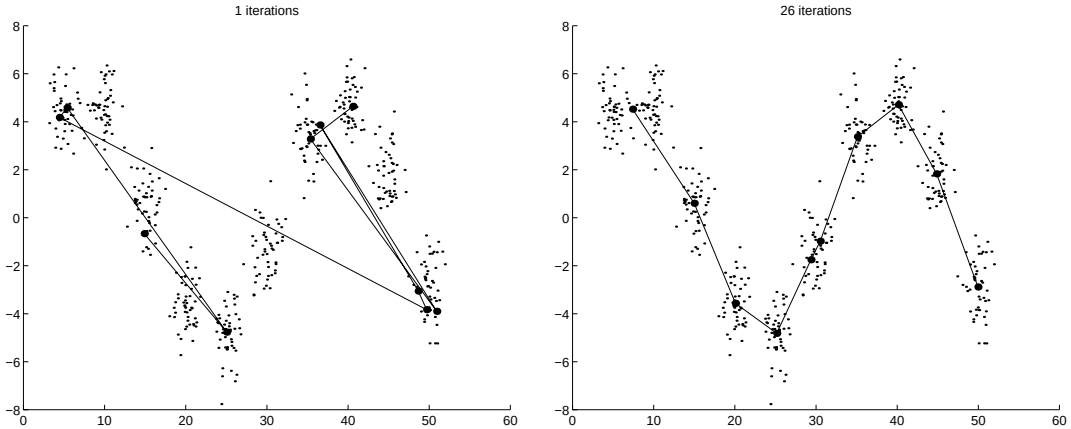


FIG. 2.7 - : Évolution de l'algorithme EMP. Le jeu de données est composé de 500 individus issus d'un mélange de 10 gaussiennes bidimensinelles, dont les moyennes sont situés sur une sinusoïde. La topologie est définie dans l'espace de sortie par un graphe linéaire de 10 unités.

dans l'exemple 2.8. On peut constater que les résultats obtenus sont similaires à ceux de l'algorithme SOFM.

△

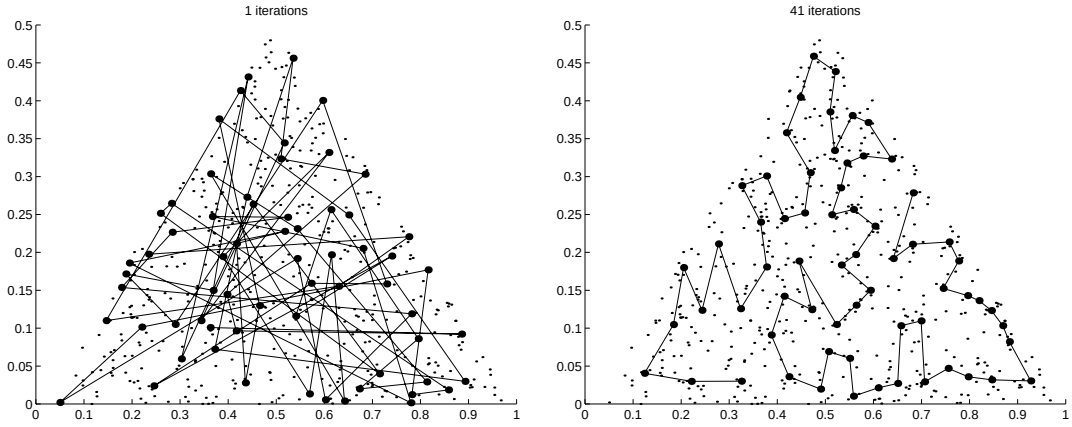


FIG. 2.8 - : Évolution de l'algorithme CEMP. Le jeu de données est composé de 500 individus issus d'une distribution uniforme supportée par un triangle. La topologie est définie dans l'espace de sortie par un graphe linéaire de 64 unités.

Notons que l'algorithme CEMP part d'une idée séduisante, mais nécessite encore de nombreuses investigations, qui permettraient peut être de comprendre et d'améliorer son comportement.

2.4 De l'intérêt de la cartographie associative

Les principales applications de la cartographie associative sont :

- la représentation ;
- la classification automatique.

Quelques articles (Angeniol *et al.* 1988, Fort 1988) utilisent aussi des algorithmes de cartographie associative pour obtenir une solution approchée du problème du voyageur de commerce, où il s'agit de trouver le plus court chemin reliant un certain nombre de villes.

2.4.1 Représentation et réduction de dimension

Les algorithmes de cartographie associative sont le plus souvent utilisés pour représenter dans le plan des données multidimensionnelles. A ce titre, ces techniques

s'intègrent dans le vaste ensemble des méthodes de mapping (Siedlecki *et al.* 1988). Ces méthodes permettent à l'analyste de percevoir visuellement un ensemble de vecteur de dimension d, et ainsi de détecter des structures telles qu'une partition. Elles peuvent aussi servir à visualiser et comparer les résultats obtenues par différentes méthodes de classification.

Parmi les méthodes de mapping, on distingue les méthodes linéaires des méthodes non linéaires. Les premières, dont l'analyse en composantes principales (ACP) réalisent une projection des données dans un sous-espace alors que les secondes font subir aux données une transformation non linéaire. Dans ce contexte, l'algorithme des cartes de Kohonen est parfois qualifié "d'ACP non linéaire", et l'on parle, par analogie, de "projection non linéaire". Notons que les techniques de "multidimensional scaling" traitent des tableaux de dissimilarités et englobent les méthodes de mapping, qui sont concernées par des ensembles d'individus décrits par un certain nombre de variables.

La représentation des données à l'aide d'un algorithme de cartographie associative utilise le fait qu'un grand nombre de vecteurs de grande dimension est représenté par un nombre plus restreint de prototypes qui possédent une représentation dans un espace de faible dimension (l'espace de sortie).

La représentation graphique des données peut prendre diverses formes, suivant le but recherché et la technique utilisée. Des comparaisons entre ces nombreuses méthodes semblent délicates et restent surtout basées sur un jugement visuel subjectif.

Visualisation des individus

La plupart des techniques de visualisation des individus tentent de trouver une représentation cohérente des vecteurs de la base de donnée dans l'espace de sortie (le plus souvent une grille bidimensionnelle). Nous nommons de manière imagée mais abusive, ce genre de manipulation "projection".

La projection la plus simple consiste à positionner, dans l'espace de sortie, chaque vecteur d'entrée à la place du prototype qui le représente, c'est-à-dire le plus proche dans l'espace des entrées. Cet type de représentation est peu lisible si un prototype représente un grand nombre de vecteurs.

Dans le cas où les données sont étiquetées (chaque individu appartient à une classe connue), cette représentation peut être simplifée, en indiquant quelle étiquette (classe) est la plus représentée parmi les vecteurs qui sont ratachés à chaque prototype. On voit alors comment les classes sont réparties les unes par rapport aux autres sur la carte, dans l'espace de sortie.

Exemple 2.10 Les iris d'Anderson sont un jeu de données constitué de 150 individus (des fleurs) décrits par quatre variables :

- largeur de pétale ;
- longueur de pétale ;

- largeur de sépale ;

- longueur de sépale.

Les 150 fleurs regroupent 3 espèces d'iris différentes. Chaque espèce est représentée par 50 individus.

La Figure 2.9 montre une carte étiquettée obtenue à partir des iris d'Anderson avec l'algorithme STPEM. Chaque unité est représenté par un rectangle de couleur. Quatre couleurs coexistent. Le noir signifie que le prototype ne représente aucun individu. Les trois autres couleurs correspondent aux trois espèces d'iris. Chaque unité possède la couleur de l'espèce la plus représentée. On peut observer que les classes semblent bien séparées par la carte, mais ce jugement n'est pas très précis car une unité est susceptible de représenter des fleurs appartenant à différentes espèces.

△

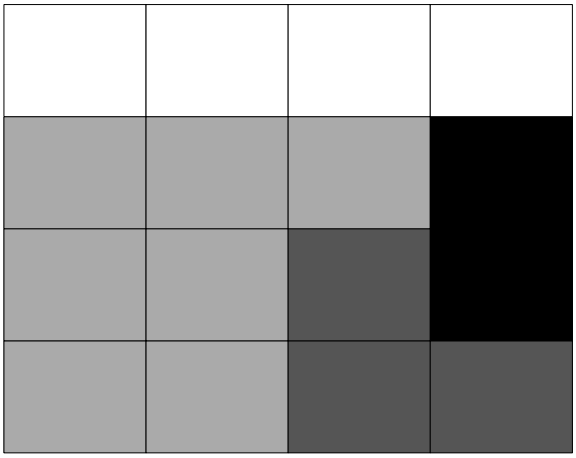


FIG. 2.9 - : Carte étiquettée : cette carte a été obtenue aprés convergence de l'agorithme STPEM, initialisé avec une grille de 4×4 unités, utilisant un voisinage rectangulaire, avec les iris.

Ce type de représentation possède plusieurs inconvénients :

- Tous les vecteurs d'un même groupe sont représentés à la même place dans l'espace de sortie. Cela implique que les distances, dans l'espace des entrées, entre les vecteurs d'un même groupe, ne sont pas prises en compte.

- Les prototypes sont répartis régulièrement dans l'espace de sortie, alors que leur répartition dans l'espace des entrées est souvent très irrégulière.

Pour dépasser ces limites, des solutions diverses ont été proposées (Ambroise et Trautmann 1994, Ambroise et Govaert 1996, Finch et Austin 1994)

Ainsi "la projection des meilleurs voisins" (Ambroise et Trautmann 1994) permet de représenter les données sur toute la surface de la grille dans l'espace de sortie et non plus seulement sur les nœuds. Chaque vecteur $\mathbf{x}_i$ est traité de la manière suivante :

1. Localisation du prototype de plus proche ("Best Matching Unit"), $\boldsymbol{\mu}_{bmu}$:

$$\|x_i - \boldsymbol{\mu}_{bmu}\| = \min_k \|x_i - \boldsymbol{\mu}_k\|. \tag{2.29}$$

2. Parmi les prototypes voisins de $\boldsymbol{\mu}_{bmu}$, trouver les m prototypes les plus proches de $\mathbf{x}_i$, que nous notons : $\boldsymbol{\mu}_{bmu1}, \cdots, \boldsymbol{\mu}_{bmum}$.

3. Dans l'espace de sortie, $\mathbf{x}_i$ est alors positionné en $\mathbf{x}_i{}'$, avec :

$$\mathbf{x}_i{}' = \frac{\frac{\boldsymbol{\mu}_{bmu}{}'}{d(\boldsymbol{\mu}_{bmu},\mathbf{X}_i)} + \frac{\boldsymbol{\mu}_{bmu1}{}'}{d(\boldsymbol{\mu}_{bmu1},\mathbf{X}_i)} + \cdots + \frac{\boldsymbol{\mu}_{bmum}{}'}{d(\boldsymbol{\mu}_{bmum},\mathbf{X}_i)}}{\frac{1}{d(\boldsymbol{\mu}_{bmu},\mathbf{X}_i)} + \frac{1}{d(\boldsymbol{\mu}_{bmu1},\mathbf{X}_i)} + \cdots + \frac{1}{d(\boldsymbol{\mu}_{bmum},\mathbf{X}_i)}}, \tag{2.30}$$

où $\boldsymbol{\mu}_{bmu}{}'$, $\boldsymbol{\mu}_{bmu1}{}', \cdots, \boldsymbol{\mu}_{bmum}{}'$ sont les coordonnées des prototypes dans l'espace de sortie et $d()$ la distance euclidienne dans l'espace des entrées.

Une autre technique assez similaire a été proposée par Finch et Austin (1994). Elle fait intervenir une étape intermédiaire de projection orthogonale dans l'espace des entrées :

1. Localisation du prototype de plus proche ("Best Matching Unit"), $\boldsymbol{\mu}_{bmu}$.

2. Parmi les prototypes voisins de $\boldsymbol{\mu}_{bmu}$, trouver les 2 prototypes les plus proches de $\mathbf{x}_i$, que nous notons : $\boldsymbol{\mu}_{bmu1}, \boldsymbol{\mu}_{bmu2}$.

3. $\mathbf{x}_i$ est projeté orthogonalement sur les vecteurs $\boldsymbol{\mu}_{bmu} - \boldsymbol{\mu}_{bmu1}$, et $\boldsymbol{\mu}_{bmu} - \boldsymbol{\mu}_{bmu2}$ et les coefficients p_1 et p_2 sont calculés :

$$p_k = \frac{(\mathbf{x}_i - \boldsymbol{\mu}_{bmu}, \boldsymbol{\mu}_{bmu} - \boldsymbol{\mu}_{bmuk})}{(\boldsymbol{\mu}_{bmu} - \boldsymbol{\mu}_{bmuk}, \boldsymbol{\mu}_{bmu} - \boldsymbol{\mu}_{bmuk})}. \tag{2.31}$$

4. Dans l'espace de sortie, $\mathbf{x}_i$ est alors positionné en $\mathbf{x}_i{}'$, avec :

$$\mathbf{x}_i{}' = \boldsymbol{\mu}_{bmu}{}' + \frac{1}{2}[p_1 \cdot (\boldsymbol{\mu}_{bmu2}{}' - \boldsymbol{\mu}_{bmu}{}') + p_2 \cdot (\boldsymbol{\mu}_{bmu2}{}' - \boldsymbol{\mu}_{bmu}{}')], \tag{2.32}$$

où $\boldsymbol{\mu}_{bmu}{}'$, $\boldsymbol{\mu}_{bmu1}{}', et \boldsymbol{\mu}_{bmu2}{}'$ sont les coordonnées des prototypes dans l'espace de sortie.

Finch et Austin remarquent que les bords de la carte posent un problème car les prototypes formant ces bords possèdent moins de voisins que les autres prototypes de la carte. Ainsi si l'on considère une carte bi-dimensionnelle à voisinage rectangulaire, où chaque prototype possède 4 voisins directs, les unités placés dans les coins de la carte n'ont que 2 voisins et les autres unités des bords, seulement 3 voisins. La solution proposée consiste à créer des unités virtuelles à l'extérieur de la carte qui sont les images par symétrie centrale des voisins directe des unités des bords. Le centre de chaque symétrie étant une unité du bord de la carte.

Exemple 2.11 Les figures 2.10 et 2.11 illustrent les deux méthodes de projections présentées ci-dessus. Chaque fleur est représentée sur les cartes par le symbole de son espèce.

Ce genre de représentation permet d'obtenir plus d'information qu'une simple carte étiquetée. Ainsi, il apparait que l'espèce + peut facilement être distinguée des deux autres espeéces (o et x). On peut voir aussi que certains prototypes représentent à la fois des fleurs o et des fleurs x.

Notons que les deux méthodes de projection donnent des résultats semblables.

△

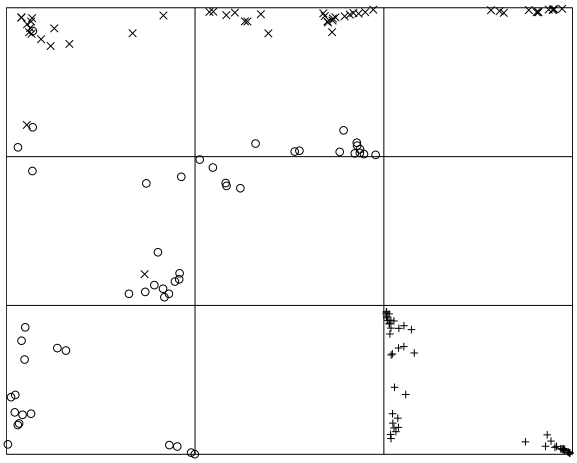


FIG. 2.10 - : Illustration de la projection des meilleurs voisins avec les iris.

Ambroise et Govaert (1996) ont proposé une méthode de projection qui permet à la fois de représenter les individus sur toute la surface de la carte et en plus de déformer cette carte habituellement rigide pour améliorer la préservation de la topologie. Cette projection s'inspire d'une idée de Lowe et Tipping (1995) utilisant la projection de Sammon (Sammon 1969). La projection de Sammon trouve $\mathbf{x}_1{}',...,\mathbf{x}_N{}'$, une configuration de N points dans le plan, qui minimise le critère suivant :

$$S = \frac{1}{\sum_{i>j} \delta_{ij}} \sum_{i>j} \frac{(\delta_{ij} - d_{ij})^2}{\delta_{ij}}, \tag{2.33}$$

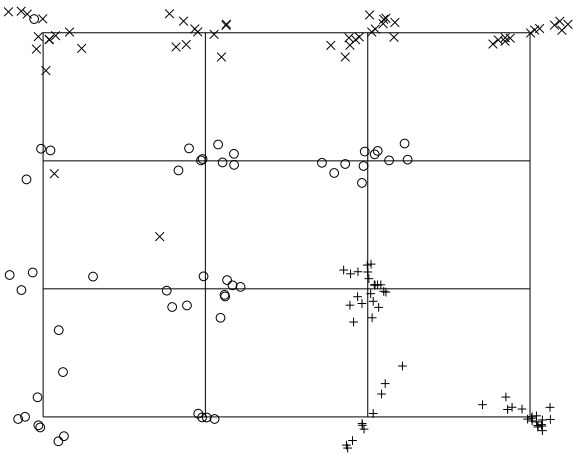


FIG. 2.11 - : Illustration de la projection de Finch avec les iris.

où d_{ij} est la distance euclidienne entre $\mathbf{x}_i{}'$ and $\mathbf{x}_j{}'$, et δ_{ij}, est la mesure de dissimilarité dans l'espace des entrées entre les individus i et j.

Une projection de Sammon, qui utilise simplement les informations d'une carte de Kohonen peut aisement être obtenue en remplacant, dans le critère, les dissimilarités initiales δ_{ij}, par δ^*_{ij} :

$$\delta^*_{ij} = (1-\alpha)\cdot\delta_{ij} + \alpha\cdot d(\boldsymbol{\mu}_i{}', \boldsymbol{\mu}_j{}'), \tag{2.34}$$

où $d(\boldsymbol{\mu}_i{}', \boldsymbol{\mu}_j{}')$ est la distance entre les prototypes des objets i et j dans l'espace de sortie, et α un coefficient à définir appartenant à $[0,1]$.

De manière intuitive, cette projection hybride Kohonen-Sammon, tente de préserver les relations de voisinage définies *a priori* par la carte, ainsi que les dissimilarités (souvent des distances) entre les individus dans l'espace des entrées. Le compromis étant géré par le coefficient α. Si α vaut zéro, cette projection est simplement une projection de Sammon et ne tient pas compte de l'information apportée par l'algorithme de cartographie associative. Par contre si α vaut un, seules les relations de voisinage sont préservées et cette projection revient à localiser tous les individus à l'emplacement de leur prototype le plus proche dans l'espace des entrées.

Exemple 2.12 Lorsque une carte est représentée en utilisant la projection hybride, les variations du coefficient α influencent beaucoup le résultat (Figure 2.12). Plus α est grand et plus la carte possède un aspect déployée. La descente de gradient utilisée pour trouver le minimum du critère S est un algorithme coûteux en temps de calcul, ce qui rend cette méthode mal adaptée à des jeux de données importants (plus de mille individus). De même il semble que les résultats soient plus concluant avec de très petite cartes.

Cette méthode est avantageuse lorsque les relations de voisinages traduisent réellement un *a priori* sur la structure des données, et qu'un prototype de la carte représente

une classe. Dans ce cas la méthode permet d'obtenir une représentation qui sépare les classes plus nettement qu'une projection de Sammon classique (Figure 2.13).

△

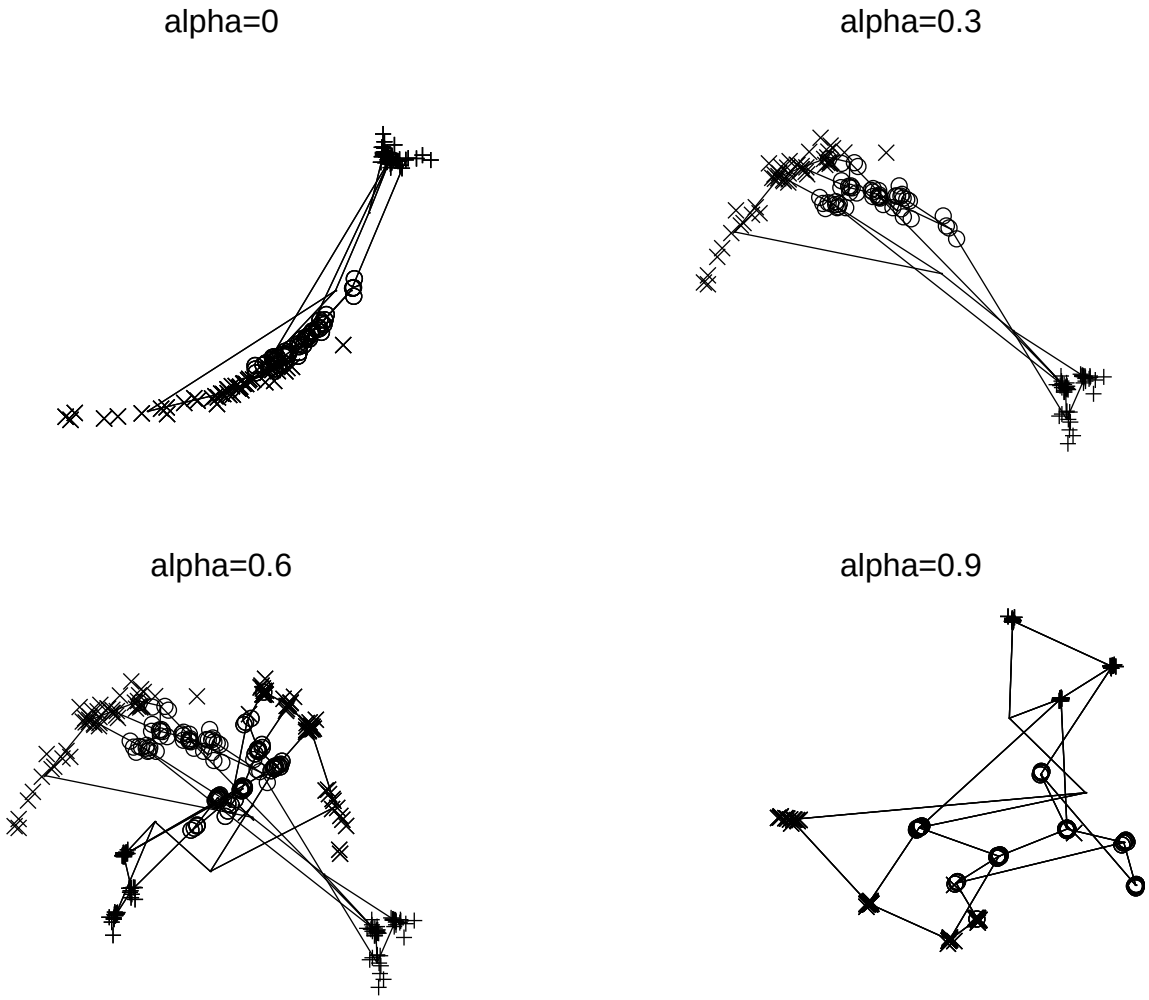


FIG. 2.12 - : Projection hybride des iris sur une grille de 4×4 unités, avec quatre valeurs différentes de α

Représentation des variables

Une représentation des variables dans l'espace de sortie est possible. Une méthode simple consiste à considérer les variables séparément et à visualiser une carte par variable.

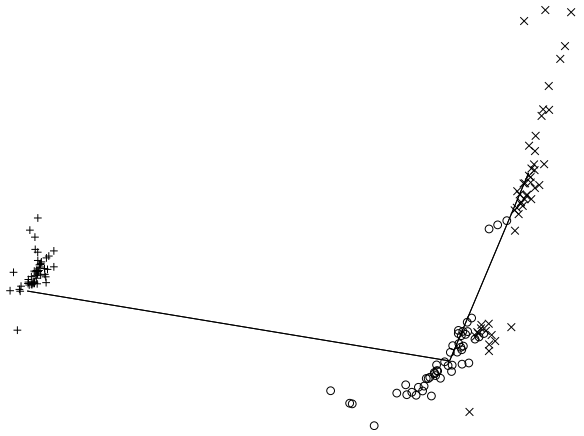


FIG. 2.13 - : Projection hybride avec 3 prototypes sur les iris

Si chaque prototype est caractérisé par d variables, dessiner une carte de la variable j consiste à représenter la valeur de cette variable pour chaque prototype à la place de ce prototype dans l'espace de sortie. Pour obtenir une carte visuellement plus exploitable, on peut faire correspondre les valeurs des variables à une échelle de couleurs et pointer les couleurs plutôt que les valeurs.

Ce type de représentation permet d'aider l'interprétation des projections des individus, et de savoir comment une zone de la carte réagit aux différentes variables.

Exemple 2.13 La représentation des variables (Figure 2.14) de la carte utilisée dans les exemples précédents permet de déceler que les fleurs de la classe + (dans le coin inférieur droit) se distingue nettement des autres fleurs par une petite largeur de sépale et une petite longueur de pétale (plus la couleur est sombre plus la valeur de la variable est petite).

△

Visualisation des distances inter-prototypes

Une autre possibilité de représentation consiste à visualiser les distances inter-prototypes dans l'espace des entrées, sur la carte. Ce genre de représentation part de l'observation empirique suivante : lorsque l'algorithme de Kohonen a convergé, les prototypes sont répartis de manière inégale dans l'espace des entrées. En effet, les zones vides de données contiennent peu ou pas de prototypes et les zones denses contiennent de nombreux prototypes. Ainsi si deux prototypes sont voisins dans l'espace de sortie et séparés par une grande distance (relativement aux autres distances) dans l'espace des entrées, il se peut que cela soit révélateur d'une zone à faible densité dans l'espace des entrées et donc d'une frontière de classe, si l'on admet que

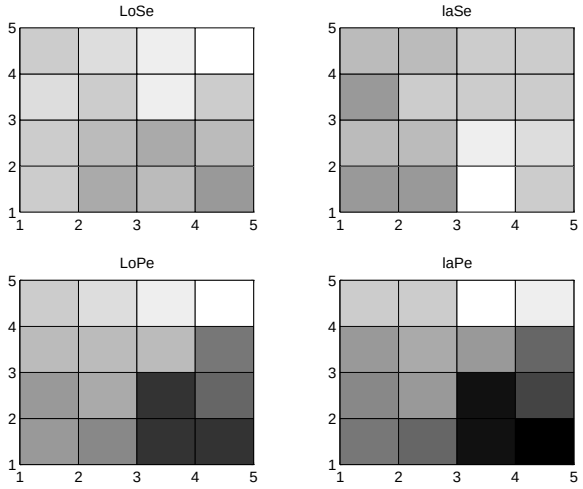


FIG. 2.14 - : Les quatre cartes des variables obtenues avec les iris d'Anderson

les classes sont des ensembles denses séparés plus ou moins nettement par des zones moins denses.

Cette observation géométrique peut être confirmée si l'on considère le cadre des modèles de mélange : en effet, l'algorithme des k-means, très proche de l'algorithme des cartes de Kohonen, cherche les paramètres (les centres) d'un mélange de K gaussiennes qui sont mélangées dans des proportions identiques, et possèdent la même matrice de variance-covariance "sphérique". Ainsi chaque prototype devrait représenter à peu près le même nombre d'individus, et les zones vides ne devraient donc pas contenir de prototypes.

Zhao (1992) propose une représentation tridimensionnelle. Dans l'espace de sortie, chaque prototype $\boldsymbol{\mu}_k$ possède une position que nous avons précedement noté $\boldsymbol{\mu}_k{}'$. En chacune de ces positions une hauteur entière h_k est calculée :

$$h_k = E(L \cdot \exp\{\frac{-d_k}{\overline{d}}\}), \tag{2.35}$$

où

- L, la hauteur maximum souhaitée,
- d_k la moyenne des distances, aux voisines sur la carte, dans l'espace des entrées :

$$d_k = \frac{1}{n_k} \sum_{\ell \ voisin \ de \ k} d(\boldsymbol{\mu}_k, \boldsymbol{\mu}_\ell), \tag{2.36}$$

  où n_k est le nombre de voisins de l'unité k,
- $\overline{d}$ est la moyenne des distances d_k.

Ultsch (Ultsch 1990, 1992, 1993) propose une méthode de visualisation 'de visualisation baptisée méthode des matrices des distances unifiées (U-matrices). Une U-matrice $\mathbf{U}\{u_{ij}\}_{i,j=1..K}$ contient les informations nécessaires au dessin d'une carte tridimensionnelle. Chaque élément u_{ij} de la matrice indique la hauteur du relief au point de coordonnées (i, j).

Considérons une carte avec un système de voisinage rectangulaire. Pour chaque unité de coordonnée $\boldsymbol{\mu}'$ sur la carte, huit unités voisines (au maximum) sont prises en compte :

$$\begin{array}{ccccc} \boldsymbol{\mu}_{-11}{}' & - & \boldsymbol{\mu}_{01}{}' & - & \boldsymbol{\mu}_{11}{}' \\ | & & | & & | \\ \boldsymbol{\mu}_{-10}{}' & - & \boldsymbol{\mu}' & - & \boldsymbol{\mu}_{01}{}' \\ | & & | & & | \\ \boldsymbol{\mu}_{-1-1}{}' & - & \boldsymbol{\mu}_{0-1}{}' & - & \boldsymbol{\mu}_{1-1}{}' \end{array}$$

Dans $\mathbb{R}^p$, l'espace des entrées, les distances du neurone $\boldsymbol{\mu}$ à ses plus proches voisins sur la grille sont calculées. La matrice des distances unifiées est construite avec ces distances. Si $\boldsymbol{\mu}$ a pour coordonnées $\boldsymbol{\mu}' = (i, j)$, une partie de la matrice est calculée :

U-matrice	2j-1	2j	2j+1
2i-1	$\frac{1}{2}\{d(\boldsymbol{\mu}, \boldsymbol{\mu}_{-11}) + d(\boldsymbol{\mu}_{-10}, \boldsymbol{\mu}_{01})\}$	$d(\boldsymbol{\mu}, \boldsymbol{\mu}_{01})$	$\frac{1}{2}\{d(\boldsymbol{\mu}, \boldsymbol{\mu}_{11}) + d(\boldsymbol{\mu}_{10}, \boldsymbol{\mu}_{01})\}$
2i	$d(\boldsymbol{\mu}, \boldsymbol{\mu}_{-10})$	$\overline{d}$	$d(\boldsymbol{\mu}, \boldsymbol{\mu}_{-10})$
2i+1	$\frac{1}{2}\{d(\boldsymbol{\mu}, \boldsymbol{\mu}_{-1-1}) + d(\boldsymbol{\mu}_{-10}, \boldsymbol{\mu}_{0-1})\}$	$d(\boldsymbol{\mu}, \boldsymbol{\mu}_{0-1})$	$\frac{1}{2}\{d(\boldsymbol{\mu}, \boldsymbol{\mu}_{1-1}) + d(\boldsymbol{\mu}_{10}, \boldsymbol{\mu}_{0-1})\}$

où d est la distance euclidienne et

$$\overline{d} = \frac{1}{4}\{d(\boldsymbol{\mu}, \boldsymbol{\mu}_{01}) + d(\boldsymbol{\mu}, \boldsymbol{\mu}_{10}) + d(\boldsymbol{\mu}, \boldsymbol{\mu}_{0-1}) + d(\boldsymbol{\mu}, \boldsymbol{\mu}_{-10})\}.$$

La carte produite par la méthode des U-matrices reflète la répartition des données dans l'espace $\mathbb{R}^p$. Lorsque des points sont répartis de façon uniforme dans une zone définie de l'espace, on observe une plateau sur la carte. Dans le cas d'une distribution de probabilité uniforme dans un hypercube, la U-matrice obtenue, est un unique plateau.

La topologie de la U-matrice peut s'interpréter comme suit : Les plateaux correspondent à des classes homogènes et les "murs" marquent la séparation entre ces classes.

Exemple 2.14 Les figures 2.15 et 2.16 illustrent les deux méthodes de visualisation utilisant les distances inter-prototypes présentées précédement. Le "local display" ne permet pas de distinguer la structure de classe des iris et nous semble apporter peu d'informations sur cet exemple. La visualisation de la U-matrice montre un mur qui isole le coin inférieur droit de la carte. Cette séparation correspond toujours à la classe de fleurs qui est mise en évidence par presque toutes les méthodes de visualisation.

△

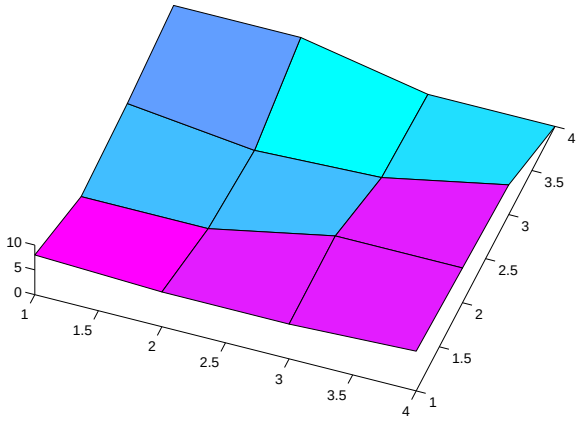


FIG. 2.15 - : "Local Display" des iris

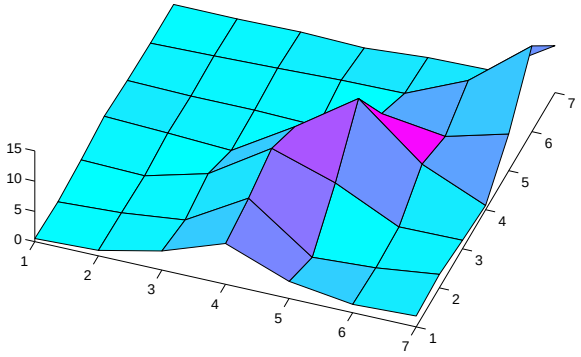


FIG. 2.16 - : U-matrice des iris

Exemple 2.15 Considérons un échantillon de taille 1000 dans $\mathbb{R}^5$, réalisation d'un mélange de 2 gaussiennes, de moyennes très différentes et de variance I.

A partir d'une carte 6×6, obtenue avec STPEM, les données sont projetées de quatre manières différentes. Alors que la structure de classe est évidente la méthode de Zhao produit un résultat délicat à interpréter. La matrice des distances unifiées donne un résultat sensiblement identique mais permet quand même de retrouver une séparation entre ces deux classes. Les deux autres méthodes de projection semblent plus facilement interprétable dans ce cas.

△

D'après l'exemple 2.14 les méthodes de visualisation utilisant les distances inter prototypes semblent peu adaptées pour déceler une quelconque structure de classe sauf si ces classes sont séparées de manière très nette (Exemple 2.15).

De manière générale, les méthodes de cartographie associative, ne sont pas des outils extrêmement efficaces pour détecter une structure de classe dans un jeu de données pour lequel on ne dispose d'aucune information *a priori*. Par contre, si les données sont déjà étiquetées, les algorithmes de cartographie associative présentés dans ce chapitre permettent de se faire une idée sur la situation des classes les unes par rapport aux autres.

2.4.2 Classification robuste

Les algorithmes de cartographie associative peuvent aussi être utilisés dans le but de faire de la classification automatique. Pour la représentation, les algorithmes de cartographie associative sont utilisés avec un grand nombre de prototypes. Les prototypes ne représentent pas vraiment des classes dans le sens utilisé dans ce mémoire, mais permettent une dissection de l'ensemble des données en petits sous-ensembles.

Au contraire, dans une optique de classification, chaque prototype est associé à une classe. L'utilisation de ce genre d'algorithme à la place des algorithmes classiques de classification non hiérarchique (chapitre 1) entraîne certains avantages :

- Introduction de connaissance *a priori* : en classification ou en quantification vectorielle, on peut disposer d'information sur des relations entre certaines classes. Ces relations peuvent se traduire par un graphe de voisinage qui stipule que certaines classes seront voisines alors que d'autres sont très différentes. Ces connaissances peuvent être traduites sous la forme d'une carte topologique. L'utilisation d'un algorithme de cartographie associative initialisé avec cette topologie particulière permet d'obtenir une partition que l'on peut qualifier de robuste (Luttrell 1990) : supposons qu'il existe une partition naturelle d'un jeu de données. Si l'algorithme de cartographie associative affecte un individu à une classe k erronnée, il y a de grandes chances pour que la vraie classe de cet individu soit une classe voisine de k.

  Exemple 2.16 Dans le cas des iris d'Anderson, les fleurs proviennent réellement de trois espèces. Un algorithme de classification automatique peut être

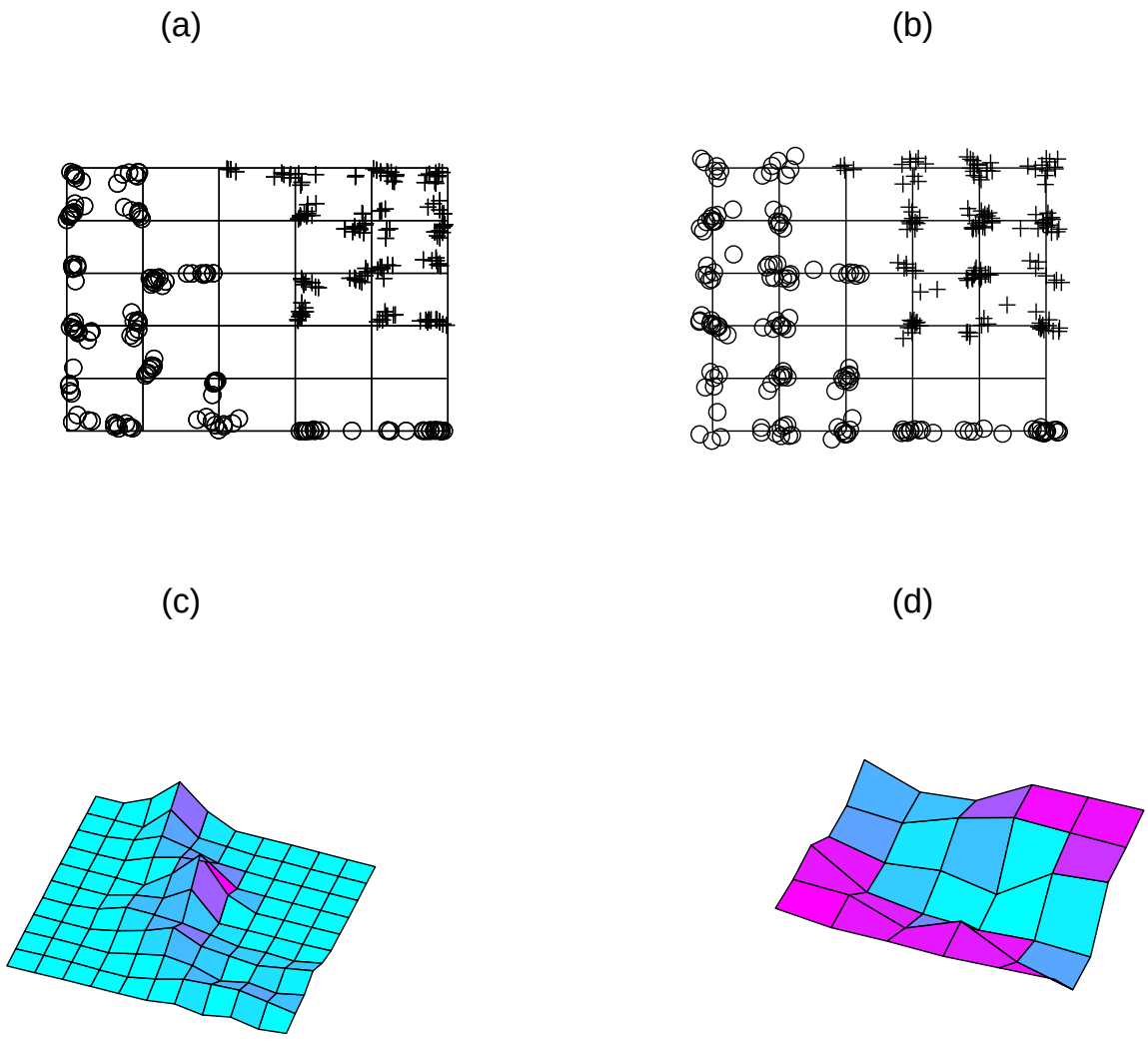


FIG. 2.17 - : Quatre type de projections utilisant une carte 6×6 avec un mélange de 2 gaussiennes: (a) projection des meilleurs voisins, (b) projection de Finch-Austin, (c) u-matrice, (d) local display.

utilisé pour retrouver cette partition naturelle. Un *a priori* possible, consiste à considérer que la classe 1 est voisine de la classe 2, et la classe 2 est voisine de la classe 3 :

$$1 - - - - - - - - - - - 2 - - - - - - - - - - - 3$$

Dans ce cas, si une fleur est affectée par erreur à la classe 3, il y a plus de chance qu'elle appartienne à la classe 2, plutôt qu'à la classe 1.

△

– Interprétation graphique : en segmentation non supervisée d'images, les individus à classer sont les pixels et les variables sont les niveaux de gris ou les proportions de trois couleurs. Une segmentation simplifie l'image, qui devient un patchwork de k couleurs (une couleur par classe). L'utilisation d'algorithme de cartographie associative permet d'associer à deux classes proches des couleurs proches. Cette simple aide au choix des couleurs peut faciliter l'interprétation de l'image segmentée.

Exemple 2.17 En géologie, des études sismiques sont parfois utiles pour déterminer à moindre coût la structure géologique du sous-sol. Ces études sismiques permettent d'obtenir des images de coupes du sous-sol étudié. Des algorithmes de segmentation peuvent être utilisés pour retrouver les différentes strates. Si deux roches proches du point de vue des traces sismiques sont coloriées avec deux couleurs proches, le spécialiste trouve généralement l'image plus évidente à interpréter, car il utilise lui aussi ce genre de convention pour dessiner des cartes géologiques à la main.

△

– Indépendance par rapport aux conditions d'initialisation : il est possible de faire un parallèle entre la largeur de voisinage dans les algorithmes de cartographie associative et la température dans les algorithmes de recuit simulé. Ces deux approches tendent à éviter les minima locaux. Cette affirmation est illustrée dans la section suivante.

2.5 Comparaisons des différentes approches

Pour comparer les performances des différents algorithmes présentés dans ce chapitre (SOFM, CEM, TPEM, STPEM et CEM pénalisé) deux approches sont possibles. Une comparaison qualitative, basée sur une visualisation des résultats, et une comparaison quantitative, fondée sur la valeur de critères numériques caractérisant les résultats obtenus par les différentes approches proposées.

Les deux propriétés, communes à tous ces algorithmes (sauf CEM), qui sont investiguées, sont la capacité à trouver une partition et la préservation de la topologie définie *a priori* entre les classes.

Des données réelles et simulées ont été utilisées pour établir cette comparaison. Le jeu de données réelles considéré est celui des iris d'Anderson déjà utilisé abondamment dans plusieurs exemples de ce chapitre. Le deuxième ensemble de données est constitué de trois classes de 50 individus réalisation de trois gaussiennes bidimensionnelles et sphériques de variance 1 et de moyennes respectives (0,0) (3,0) et (2,2). Les troisième et quatrième jeux de données sont deux échantillons de 400 individus issus de distributions uniformes bidimensionnelles qui ont respectivement pour support un carré et un triangle. Ces quatre ensembles de tests seront notés respectivement IRIS, GAUSS, UNI_SQR et UNI_TRI. Notons que les deux premiers peuvent être partitionnés de façon évidente en trois classes distinctes et seront plutot illustratifs de la capacité à classifier alors que les deux autres ne possèdent aucune structure de classes et seront plutôt utilisés pour tester l'aspect préservation de la topologie.

Pour pouvoir comparer les six algorithmes sur une base commune, quelques hypothèses ont été adoptées concernant les matrices de variances covariances recherchées par les algorithmes dérivés de l'algorithme EM. En effet, l'algorithme SOFM suppose de manière implicite que toutes les classes sont gaussiennes et ont toutes même matrice de variance covariance sphérique ($\mathbf{\Sigma}_k = \sigma \cdot I,\ \forall k$). Ainsi ces mêmes contraintes ont été imposées à l'étape M des algorithmes CEM, TPEM, STPEM et CEM pénalisé. Notons que dans ce cas l'algorithme CEM devient l'algorithme des k-means, et l'algorithme STPEM est alors identique à l'algorithme "'Batch Map" (Kohonen 1993).

L'initialisation des différents algorithmes a été réalisée suivant le même tirage aléatoire. SOFM, TPEM et STPEM utilisent une largeur de voisinage initiale égale à la moitié de la plus grande distance entre classes dans l'espace de sortie :

$$\sigma_0 = \max_{k,\ell} \frac{d_{k\ell}}{2} \tag{2.37}$$

où $d_{k\ell}$ est la distance entre les classes k et ℓ dans l'espace de sortie. La décroissance de la largeur de voisinage σ est linéaire :

$$\sigma_{m+1} = a \cdot \sigma_m$$

avec $a = 0.98$. Dans le cas de l'algorithme CEMP qui optimise une vraisemblance pénalisée, le facteur de pénalisation β est initialisé par la valeur 50, et la décroissance de ce facteur est aussi linéaire avec un facteur de décroissance de 0.95. Ces différentes valeurs ont été choisies empiriquement.

2.5.1 Comparaison visuelle

Une comparaison visuelle ne concerne naturellement que les applications de la cartographie associative à la représentation. Dans ce cas, de nombreux prototypes

sont utilisés. Ainsi une grille de 5×5 prototypes a été utilisé pour les IRIS, une grille de 8×8 pour UNI_SQR, et une ligne de 64 pour UNI_TRI.

Les jeux de données UNI_SQR et UNI_TRI, étant bidimensionnels, la visualisation des résultats obtenus par les quatre algorithmes est directe (Figures 2.18,2.19). Les iris sont décrites par quatre variables. Les résultats peuvent alors être visualisés en utilisant la projection des plus proches voisins (Figure 2.20).

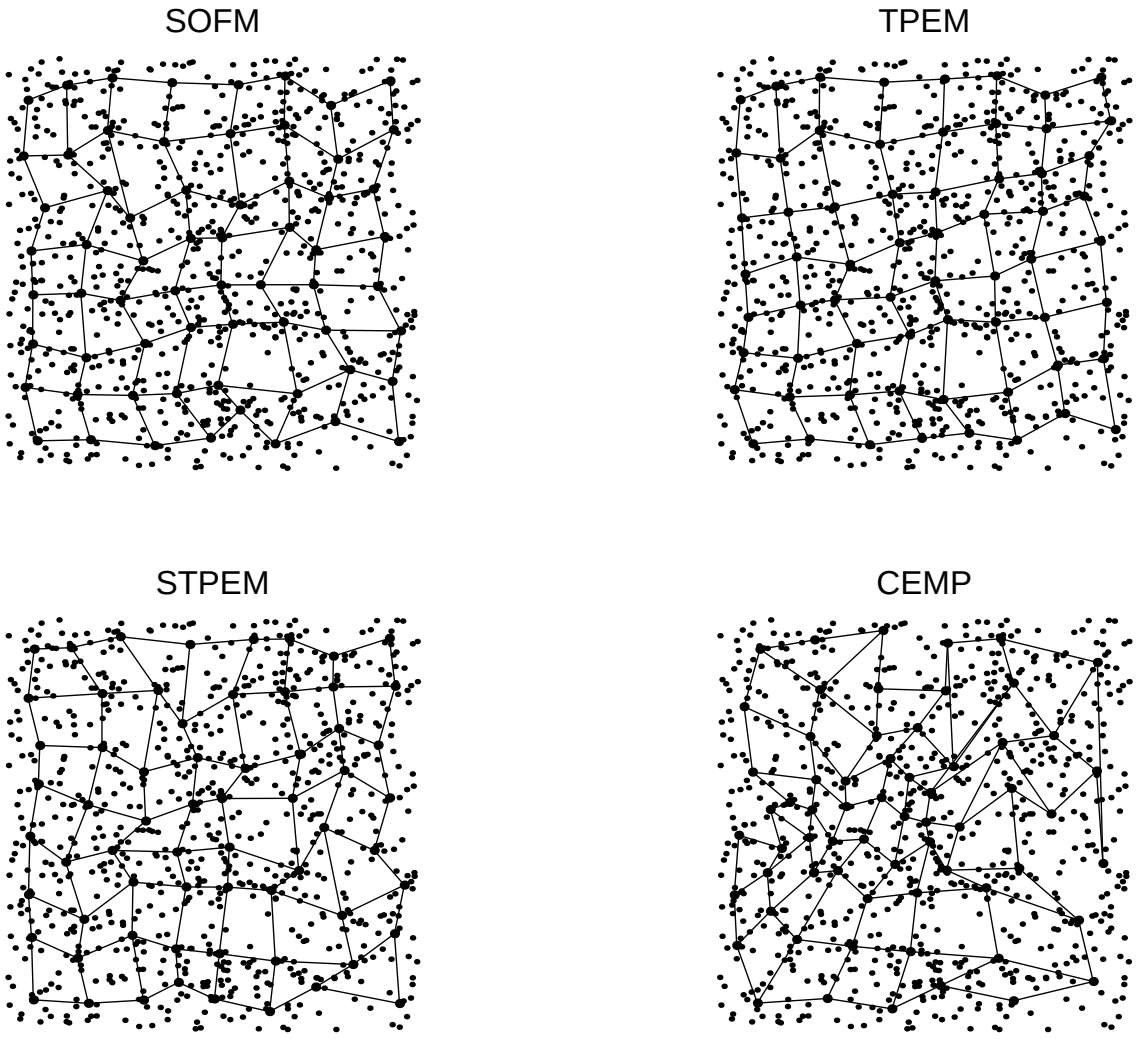


FIG. 2.18 - : Quatre algorithmes pour déployer un grille de 8×8 unités sur une distribution uniforme supportée par un carré.

Les résultats observés amènent les deux réflexions suivantes :

– Les algorithmes SOFM, TPEM et STPEM semblent avoir des comportements très semblables, même si les résultats ne sont pas strictement identiques. Dans

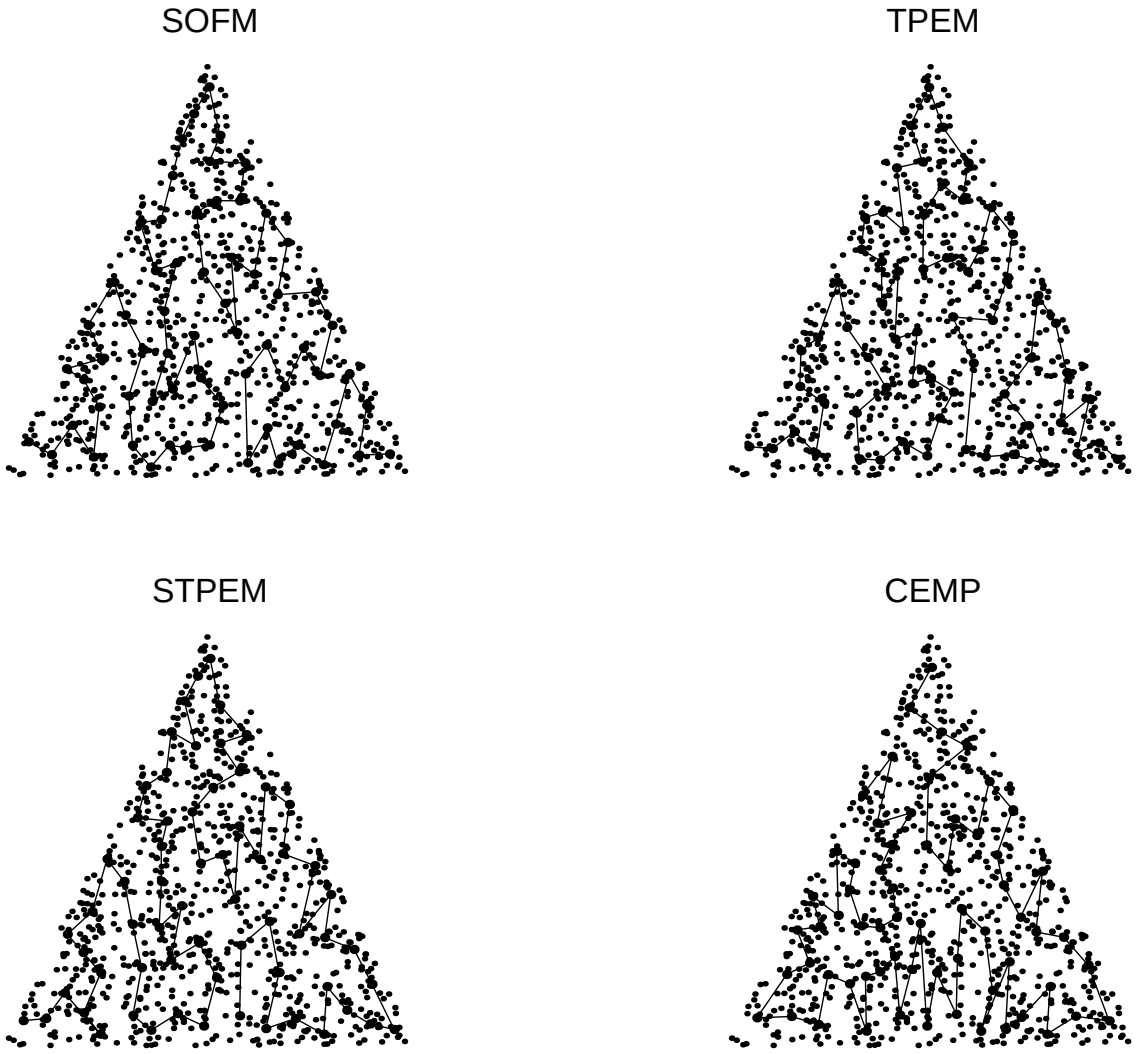


FIG. 2.19 - : Quatre algorithmes pour déployer un grille de 64×1 unités sur une distribution uniforme supportée par un triangle.

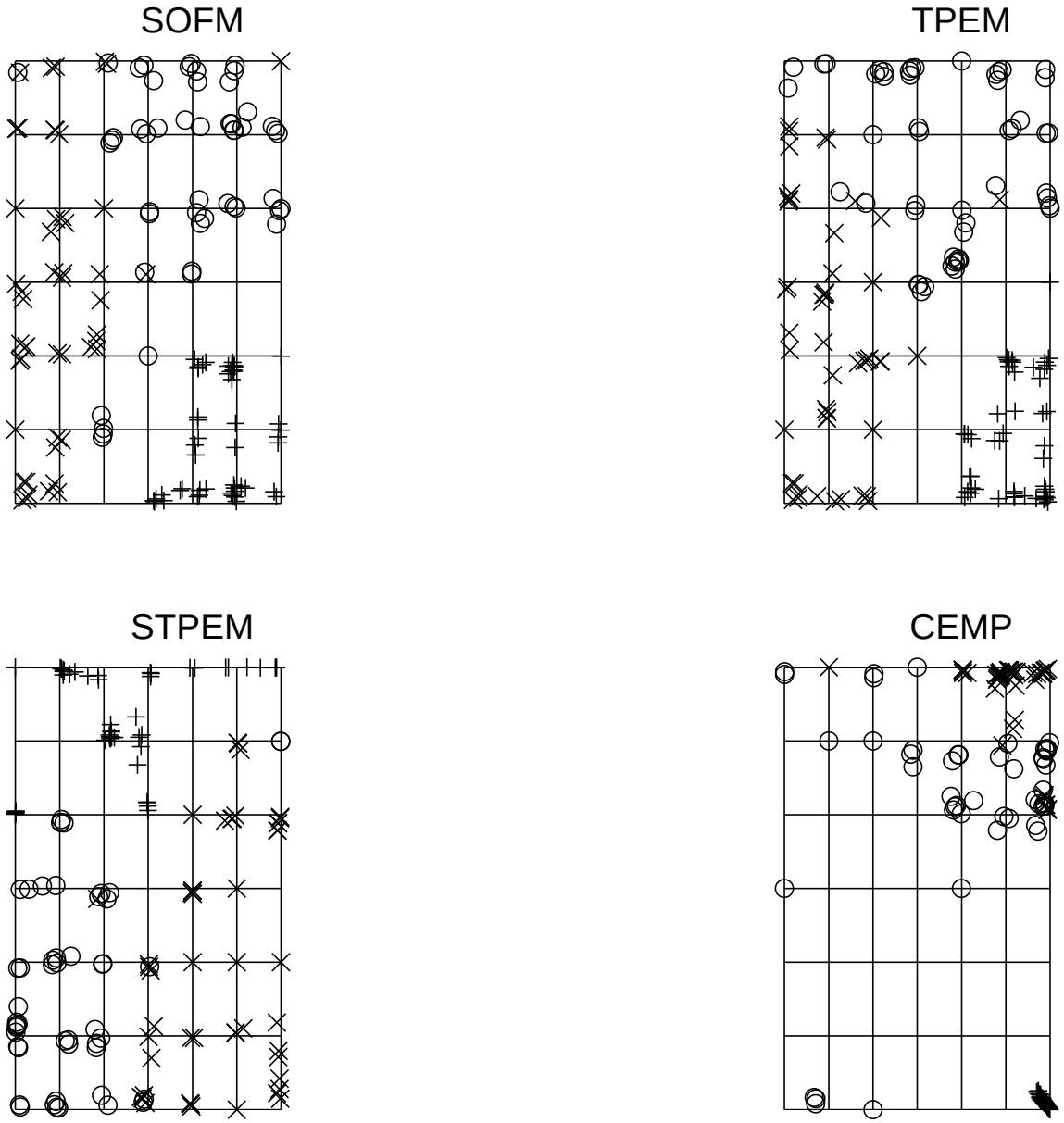


FIG. 2.20 - : Projection par la méthode des meilleurs voisins des iris d'Anderson sur une grille de 7×7 unités obtenue par quatre algorithmes de cartographie associative.

le cas des iris, les trois classes occupent chacune une zone différente de la carte.

- L'algorithme CEMP donne visuellement de moins bons résultats que les trois autres. En effet, la carte obtenue par CEMP sur la distribution uniforme supportée par un carré, est moins bien déployée. Dans le cas des iris, les trois espèces ne sont pas bien séparées et de nombreux prototypes représentent plusieur espèces à la fois.

2.5.2 Comparaison numérique

Pour la comparaison numérique, deux critères sont utilisés. La qualité de la classification est mesurée par le critère la vraissemblance classifiante, qui est le critère de la somme des variances intra-classes sous les hypothèses restrictives que nous avons formulées :

$$W = \sum_{i=1}^{N} \sum_{k=1}^{K} c_{ik} \cdot \|\mathbf{x}_i - \boldsymbol{\mu}_k\|^2.$$

La mesure considérée pour la préservation de la topologie est la somme de la longueur des arêtes, que nous avons déjà utilisée dans le cadre de la définition de l'algorithme CEMP :

$$MEL = \sum_{k=1}^{K} \sum_{\ell=1}^{K} h_{k,\ell} \cdot \|\boldsymbol{\mu}_k - \boldsymbol{\mu}_\ell\|^2,$$

Les quatre jeux de données ont été traités de la façon suivante :

- GAUSS : Réseau linéaire de trois prototypes
- IRIS : Réseau linéaire de trois prototypes
- UNI_SQR : Grille de 6×6 prototypes
- UNI_TRI : Grille de 6×6 prototypes

Comme les résultats obtenus par chacun de ces algorithmes sont dépendant de l'initialisation, chaque algorithme a été exécuté 30 fois avec différentes initialisations. Les critères W et MEL produits par ces multiples exécutions sont résumés dans le tableau 2.1 qui contient des statistiques relatives aux expériences numériques, et par l'intermédiaire d'histogrammes bidimensionnels.

Remarques

- Si l'algorithme TPEM stoppe pendant la phase "d'initialisation", cela produit une très petite valeur du critère MEL et une très grande valeur du critère W (comparée aux autres exécutions). Un exemple de ce comportement est visible sur les histogrammes relatifs aux jeux de données IRIS et GAUSS. Ces résultats avortés sont inintéressant pour la classification et la réduction de dimension car des classes sont alors vidées et l'ensemble des prototypes représente mal l'ensemble des individus.

TAB. 2.1 - : Statistiques résultant des simulations

		W criterion				MEL criterion			
		best value	worst value	mean	Std	best value	worst value	mean	Std
IRIS	SOFM	79.14	84.54	81.69	1.33	24.88	31.45	29.01	1.53
	k-means	78.85	142.75	87.38	22.09	28.84	72.89	53.57	17.90
	CEMP	78.86	394.64	118.49	82.18	3.2	40.09	25.95	9.37
	TPEM	78.86	504.45	288.33	213.16	11.03	28.84	19.93	9.05
	STPEM	78.86	78.86	78.86	0	28.84	28.84	28.84	0
GAUSS	SOFM	280.02	300.02	287.84	5.37	34.06	51.67	41.43	4.16
	k-means	274.08	278.73	274.28	0.85	39.41	97.79	81.34	23.46
	CEMP	274.08	390.54	278.13	21.23	29.1	97.41	40.77	10.87
	TPEM	274.08	926.55	402.11	260.47	16.83	39.87	35.26	9.37
	STPEM	225.98	225.98	225.98	0	41.83	41.83	41.83	0
UNI_TRI	SOFM	0.46	0.49	0.47	0.01	0.51	0.66	0.58	0.04
	k-means	0.63	0.85	0.73	0.05	3.50	6.31	4.85	0.65
	CEMP	0.503	0.91	0.608	0.097	0.177	0.503	0.384	0.079
	TPEM	0.47	0.56	0.50	0.02	0.49	0.59	0.54	0.03
	STPEM	0.47	0.52	0.50	0.01	0.47	0.63	0.54	0.03
UNI_CAR	SOFM	1.89	1.96	1.92	0.02	3.57	4.23	3.78	0.15
	k-means	2.04	2.74	2.41	0.16	24.32	37.70	29.17	3.05
	CEMP	2.094	2.718	2.384	0.171	2.771	7.92	4.129	1.082
	TPEM	1.93	2.09	1.99	0.04	3.38	5.48	3.72	0.51
	STPEM	1.94	2.09	2.00	0.03	3.38	4.21	3.71	0.19

- Quand l'algorithme TPEM ne stoppe pas, les résulats obtenus sont statistiquement très proches de ceux de l'algorithme STPEM.

- Sur tous les jeux de données, l'algorithme STPEM produit toujours de meilleurs valeurs des critères que l'algorithme des k-means, avec une variance moindre. Par exemple, avec les jeux de données IRIS et GAUSS, qui possèdent une forte structure de classe, STPEM produit 30 fois la même solution.

  On peut donc penser que la phase d'initialisation, pendant laquelle les prototypes s'organisent pour préserver la topologie, réduit le nombre de solutions possibles et guide l'algorithme près de certains minima locaux du critère W.

- Les résultats obtenus avec l'algorithme CEMP comparés aux autres algorithmes, sont les plus mauvais en terme de variance des critères. L'introduction du terme pénalisant ne semble donc pas améliorer la qualité des performances de l'algorithmes CEM.

- L'algorithme des cartes de Kohonen produit des valeurs différentes des critères à chaque exécution, mais la variance de ces résultats est très faible. Si les valeurs des critères sont différentes, c'est parce que l'algorithme SOFM est un algorithme adaptatif, qui converge après un nombre infini d'itérations (s'il y a convergence, ce qui n'est pas encore prouvé en 1996). D'un point de vue pratique, cela signifie que c'est un algorithme relativement robuste, qui semble convergé vers un petit nombre d'optima.

Mis à part l'algorithme CEMP, les algorithmes de cartographie associative présentés dans ce chapitre sont des algorithmes robustes qui sont moins sensibles aux conditions d'intialisation que l'algorithme des k-means. Le comportement de ces algorithmes ressemble à celui des algorithmes de recuit simulé, qui convergent vers un nombre réduit de solutions.

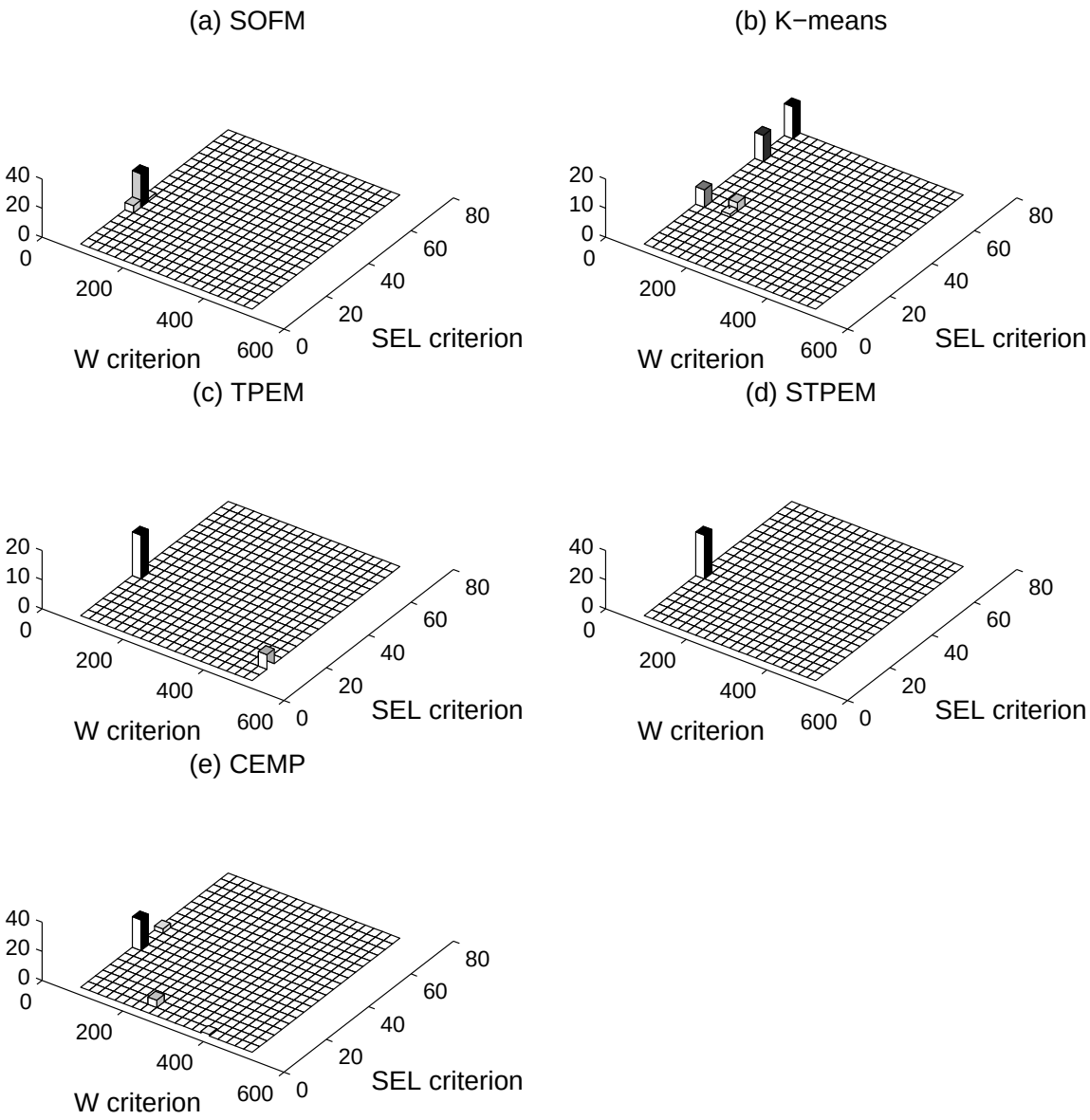


FIG. 2.21 - : Résultats obtenus avec le jeu de données IRIS

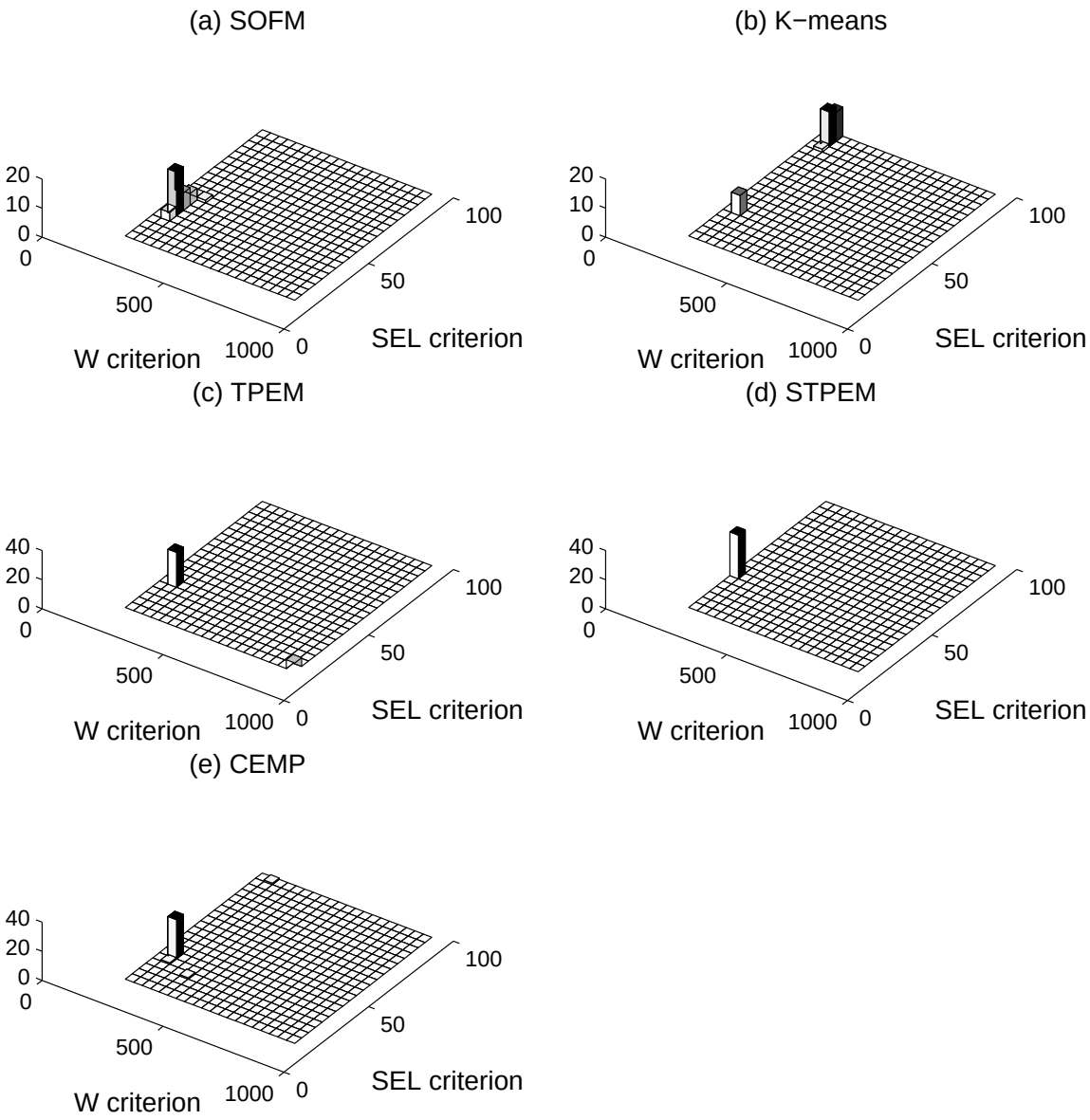


FIG. 2.22 - : Résultats obtenus avec le jeu de données GAUSS

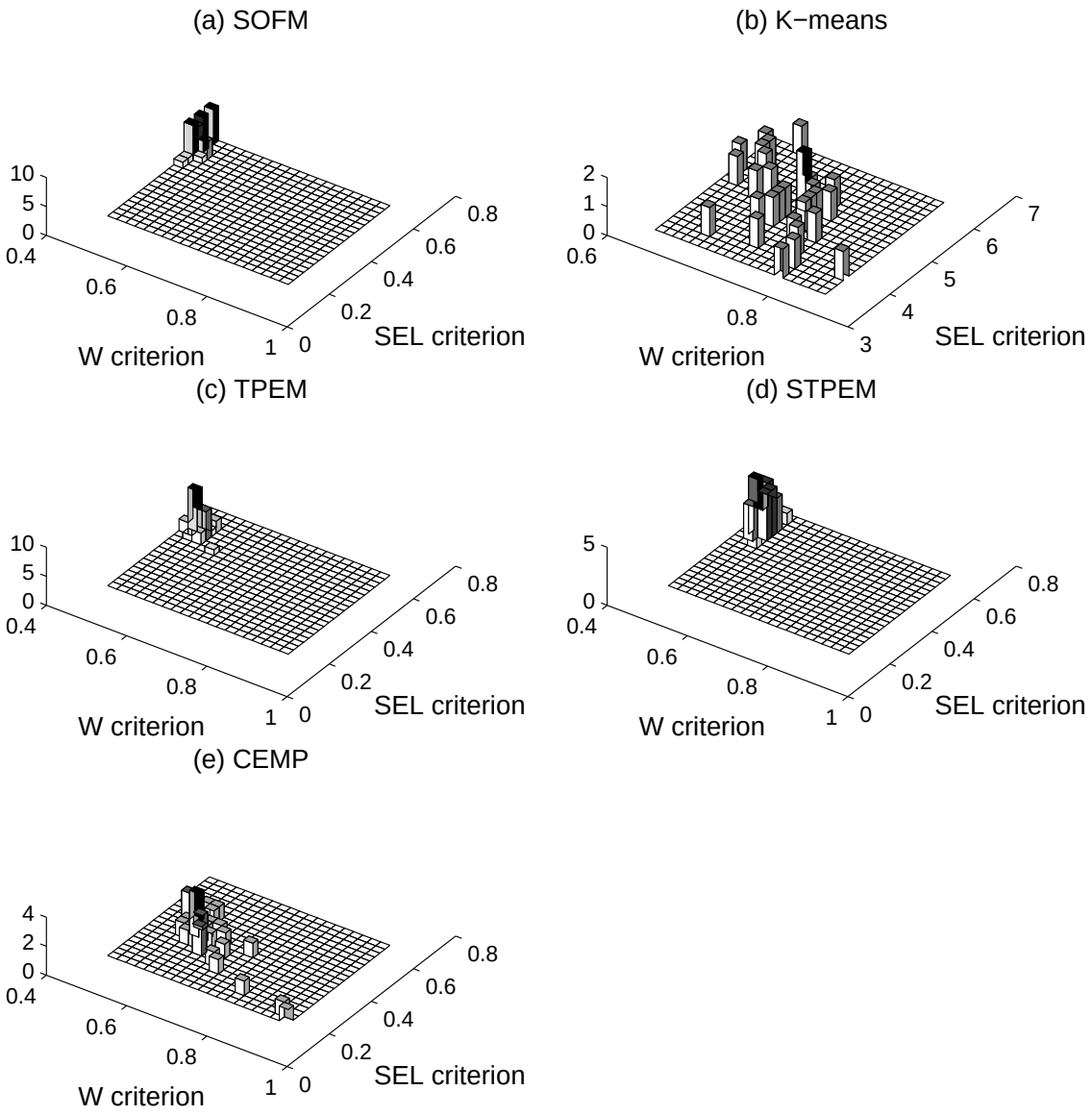


FIG. 2.23 - : Résultats obtenus avec le jeu de données UNI_TRI

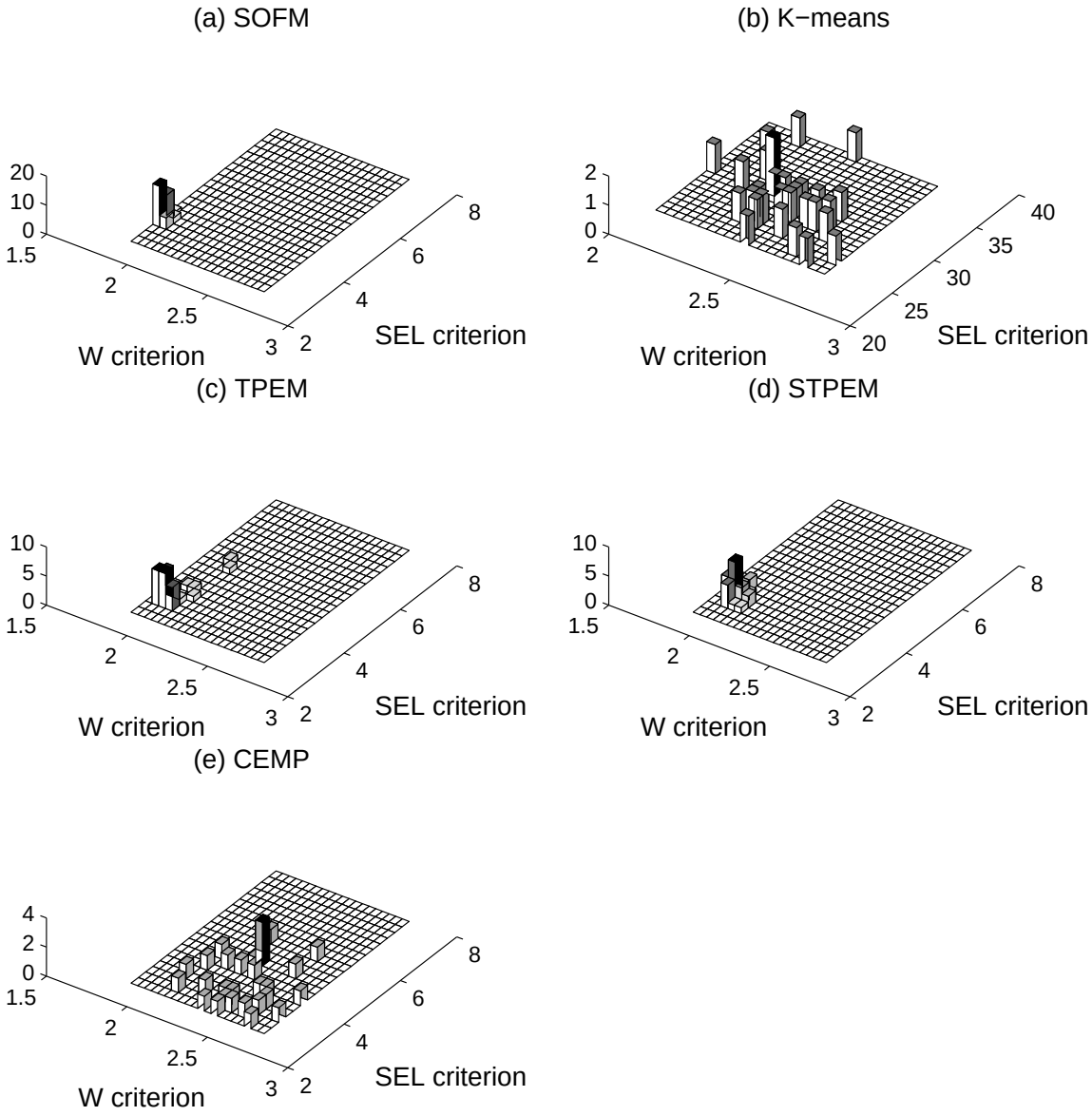


FIG. 2.24 - : Résultats obtenus avec le jeu de données UNI_CAR

Les algorithmes SOFM, TPEM et STPEM produisent des résultats comparables à tout point de vue.

Chapitre 3

Classification automatique de données spatiales

One sense of the word "statistics" is a collection of numbers, and spatial statistics includes "spatial data analysis", the reduction of spatial patterns to a few clear useful summaries.

Ripley (1981)

La classification spatiale vise à former des classes qui contiennent des sites ayant des propriétés similaires et qui sont géographiquement proches. Si des algorithmes classiques de classification automatiques sont utilisés pour partitionner des données spatiales, la classification obtenue sera en général très morcelée. Pour éviter ce morcellement, il faut considérer l'information spatiale des données. Plusieurs méthodes existent pour prendre en compte numériquement cette information :

- modifier des *algorithmes* de classification automatique existant en intégrant l'information spatiale ;
- intégrer l'information spatiale dans *les données*. Cela revient donc à considérer un nouveau jeu de données ;
- choisir un *modèle* qui prenne en compte l'aspect spatial. En pratique cela se traduit par la définition de critères intégrant les contraintes spatiales. Ce type d'approche est très utilisé en segmentation statistique d'image.

Dans les sections suivantes nous allons décrire brièvement un exemple de chaque approche. Nous accorderons une attention particulière à la modélisation stochastique de l'information spatiale, qui nous semble être l'approche la plus rigoureuse.

3.1 Heuristiques et algorithmes pour la classification spatiale

3.1.1 Adaptation d'algorithmes existants

Au cours d'un processus de classification agglomératif, il est possible de restreindre les regroupement aux entités qui sont géographiquement voisines. Les contraintes de contiguïté sont alors respectées de façon absolue et les classes produites sont connexes, c'est-à-dire qu'une classe forme une seule région géographique, un seul bloc (Legendre 1987, Lebart 1978, Openshaw 1977). Ces procédures de classification rangent dans des classes séparées deux sites qui sont spatialement très éloignés même s'ils sont très similaires au niveau des variables non géographiques. L'information spatiale joue alors un rôle prépondérant. Ce genre d'approche n'autorise pas la variation de "la quantité d'information spatiale" utilisée dans le processus de classification.

Produire des classes géographiquement connexes exige de définir au préalable quel individu est spatialement voisin de quel autre. La définition des rapports de voisinage est équivalente à la construction d'un graphe non orienté où chaque nœud est un élément de l'ensemble des données et chaque arête figure une relation de voisinage.

Une classification automatique avec contrainte de contiguïté absolue peut être partagée en deux étapes :

1. La définition d'un graphe de voisinage. Ceci peut être réalisé par une triangulation de Delaunay, par un graphe de Gabriel, par une grille...

2. La classification avec contraintes. Il est possible de modifier certains algorithmes classiques pour respecter les contraintes résumées par le graphe.

Exemple 3.1 (Lebart 1978) La classification hiérarchique ascendante est une méthode de classification simple et utilisable pour des ensembles de données de taille raisonnable (moins de 10000 individus). L'ajout de contraintes spatiales peut se faire de manière naturelle :

- **Initialisation :** calcul du graphe de voisinage et des distances entre individus deux à deux (Chaque individu est considéré comme une classe).

- **Itérer :** tant que le nombre de classes est supérieur à un :

  - regrouper les deux classes qui sont les plus proches au sens d'un certain critère d'agrégation parmi les classes voisines au sens du graphe de voisinage,
  - recalculer la matrice des distances et le graphe de voisinage entre les nouvelles classes.

Le fait de chercher seulement parmi les voisins géographiques quelles sont les classes les plus proches réduit de beaucoup l'espace de recherche et accélère la procédure.

△

3.1.2 Modification des données

Utilisation des variables spatiales

Traiter les variables de positions spatiales au même titre que les autres variables décrivant les sites, semble être une idée naturelle. Les coordonnées spatiales peuvent être pondérées pour contrôler la quantité d'information spatiale qui sera prise en compte par l'algorithme de classification automatique utilisé. Cette idée remonte à Berry (1966) et a aussi été utilisée en segmentation d'image par Jain et Farrokhnia (1991). D'après Oliver et Webster (1989) ce genre d'approche souffre du même défaut que la précédente : elle tend à séparer dans des classes différentes deux sites qui sont très similaires mais éloignés géographiquement.

Exemple 3.2 Considérons l'image simulée de la figure 3.1. Cette image comporte des pixels de deux types. En brouillant et bruitant l'image 3.1, on obtient une image dégradée codée en 256 niveaux de gris (Figure 3.2). On peut utiliser l'algorithme EM pour retrouver l'image originale, à partir de l'image dégradée. Chaque pixel est décrit par trois variables :

- son niveau de gris (0 à 256),
- $\alpha \cdot l$, où l est le numéro de ligne à laquelle appartient le pixel,
- $\alpha \cdot c$, où c est le numéro de colonne à laquelle appartient le pixel.

avec α un coefficient pondérateur. Avec quatre valeurs différentes de α, on essaie de retrouver les classes initiales en utilisant l'algorithme EM. La figure 3.3 montre les quatre images obtenues. On constate que loin d'améliorer la qualité de la reconstitution, la prise en compte des variables de position entraine une dégradation des résultats sur cet exemple.

△

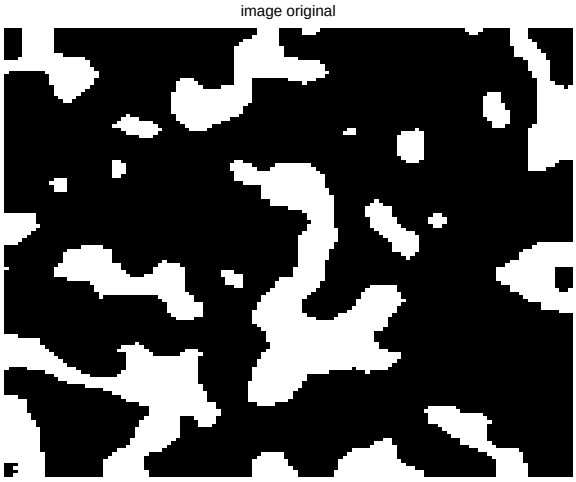


FIG. 3.1 - : Image binaire de 126×126 pixels, simulée avec un échantillonneur de Gibbs

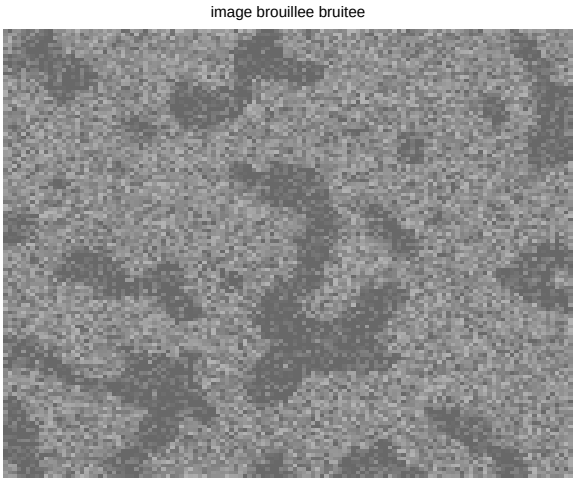


FIG. 3.2 - : Image brouillée, bruitée

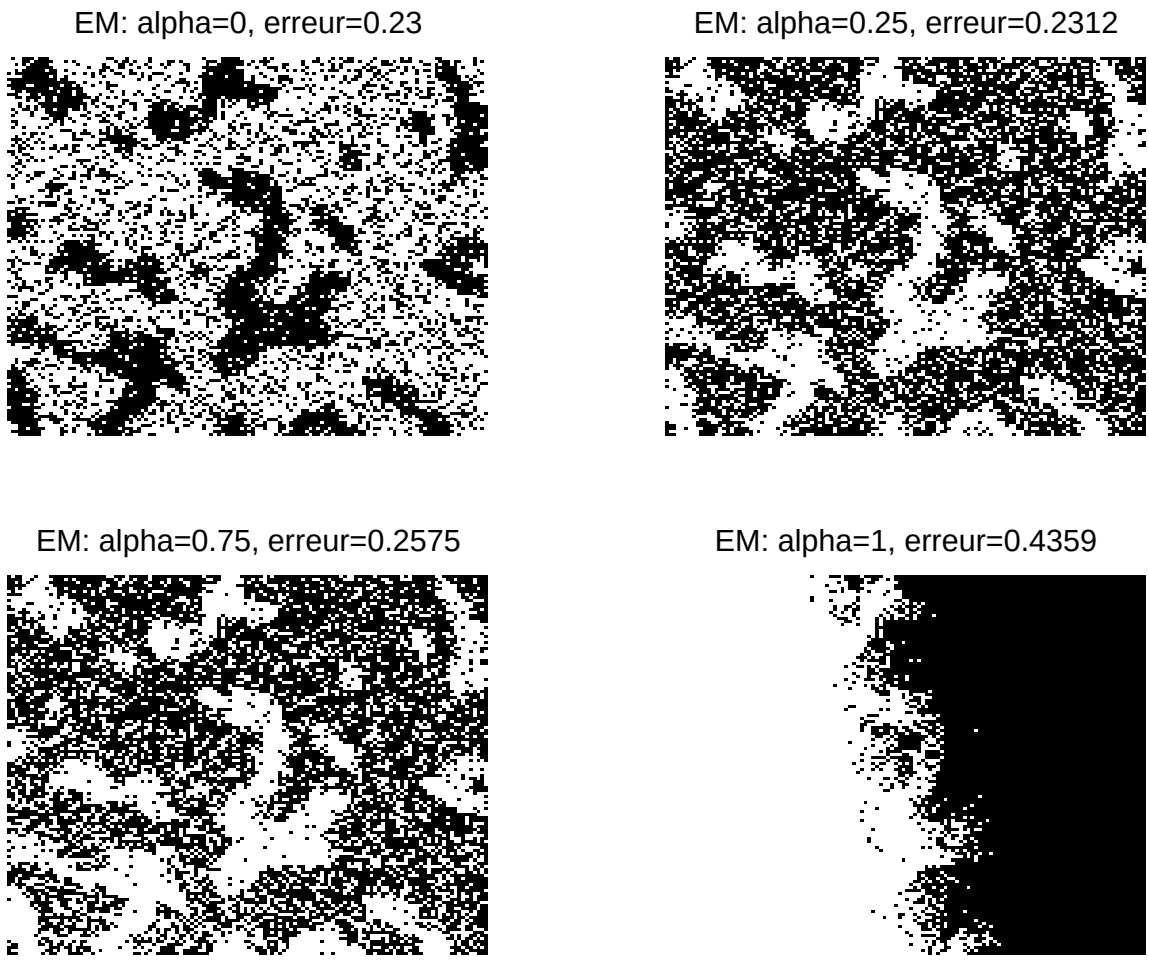


FIG. 3.3 - : Reconstruction de l'image par l'algorithme EM en prenant en compte la position des pixels dans l'image.

Transformation des variables

Au lieu de travailler sur le tableau individus/variables initial, il est possible de commencer par une phase de prétraitement. Cette phase de prétraitement a pour but d'extraire de nouvelles variables qui contiennent l'information spatiale.

Exemple 3.3 Une possibilité consiste à définir une taille de fenêtre géographique et à remplacer les variables initiales d'un individu par les valeurs moyennes des variables des voisins géographiques (c.-à-d., à l'intérieur de la fenêtre centrée sur cet individu).

△

Utilisation de la matrice des distances spatiales

Si les données initiales prennent la forme d'un tableau de distances individu/individu, on peut transformer cette matrice pour intégrer l'information spatiale.

Oliver et Webster (1989) proposent d'utiliser les distances géographiques g_{ij} entre les sites i et j pour modifier les distances ou dissimilarités d_{ij} calculées avec les variables non géographiques. Il résulte de cette opération que la méthode de classification utilisée, part d'une nouvelle matrice de dissimilarité $D^* = \{d^*_{ij}\}$ qui mélange informations géographiques et non géographiques. Les algorithmes de classifications usuels (non contraints) partitionnent les données.

Exemple 3.4 (Oliver et Webster 1989) Partant d'une matrice de dissimilarités $D = \{d_{ij}\}$, la démarche suivante fournit une partition des données qui évite un trop grand morcellement géographique sans toutefois produire des classes totalement connexes :

- Modification de la matrice des dissimilarités :

$$d^*_{ij} = d_{ij} \cdot [1 - \exp(-g_{ij}/W)]$$

  avec W un coefficient arbitraire. Plus W est grand, plus la nouvelle matrice des dissimilarités D^* est influencée par les distances géographiques et moins la partition sera fragmentée.

- Transformation de la matrice D^* en un tableau individus/variables par une analyse factorielle.

- Partitionnement du nouveau tableau de données par l'algorithme des k-means.

△

3.2 Modèles de processus spatiaux

3.2.1 Généralités

Les statistiques spatiales trouvent leurs applications dans tous les domaines où les données à traiter sont localisées spatialement comme en astronomie, exploitation minière, écologie, géographie et archéologie... Cette branche de la statistique vise à répondre à des questions aussi diverses que :

- Comment résumer un ensemble de données spatiales par des statistiques et des graphiques pertinents?
- Est-ce que les arbres d'une forêt sont répartis au "hasard", ou bien existe-t-il une structure sous-jacente?
- Tel modèle statistique explique-t-il mieux que tel autre la répartition spatiale des données observées?
- Quelle température fait-il à Paris, connaissant la température de certaines villes voisines?

Suivant le types de données considérées, les intérêts et les méthodes sont différents (Ripley 1982). Si la localisation des individus (individu au sens de l'analyse des données) et les distances entre individus sont le phénomène de première importance, les données analysées seront des points dans l'espace. Par contre, si des mesures localisées spatialement sont la matière première de l'analyse statistique, les individus étudiés seront des vecteurs localisés.

Dans tous les cas, l'ensemble des données est considéré comme la réalisation d'un processus stochastique :

Définition 3.1 *Un processus stochastique* $\{\boldsymbol{X}_t\}$ *est une suite de v.a. indicées sur un sous-ensemble de* $\mathbb{R}^d$.

Souvent l'indice représente le temps. Dans le cadre des statistiques spatiales, l'indice figure les coordonnées dans l'espace (la plupart du temps le plan) et la variable aléatoire $\mathbf{X}_t$ peut signifier l'absence ($\mathbf{X}_t = 0$) ou la présence ($X_t = 1$) d'un point en t, la température à l'endroit t...

Le type de données détermine la classe de modèles statistiques pris en compte. Ainsi plusieurs classes de processus peuvent être distinguées :

- Les processus stochastiques générateurs de points (Stochastic Point Processes en anglais), qui modélisent la répartion spatiale de points.

  Exemple 3.5 Dans le cas où l'on s'intéresse uniquement à la répartition spatiale d'un ensemble d'individus (des arbres, des villes...) à l'intérieur d'une zone géographique définie, les processus stochastiques générateurs de points sont des modèles statistiques adaptés.

Ainsi, la première étape de l'analyse consiste à déterminer si les points sont répartis au hasard ou forment une structure plus complexe. Ce problème relève de la théorie des tests d'hypothèse et fait intervenir le processus de Poisson (voir par exemple la Figure 3.4) :

H_0 : Les données sont la réalisation d'un processus de Poisson
H_1 : Les données ne sont pas réparties au "hasard"

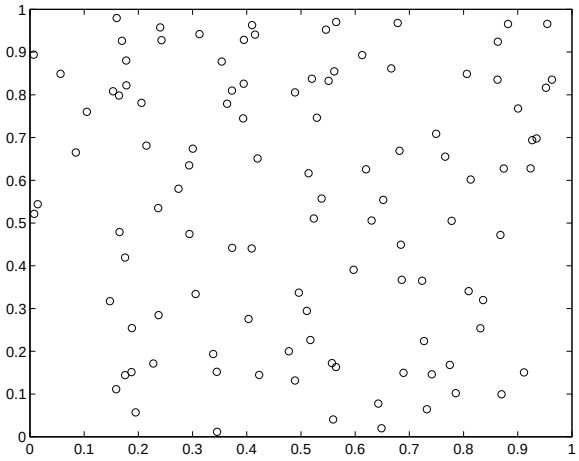


FIG. 3.4 - : Processus de Poisson homogène d'intensité $\lambda = 100$

△

- Les processus générateurs de variables régionalisées, qui modélisent des phénomènes spatialement continus (par exemple, la température en France). Ce genre de processus sert essentiellement dans la théorie de l'interpolation spatiale.

- Les séries spatiales qui sont une sorte de généralisation des série temporelles, qui prennent en compte des données localisées, en un certain nombre de sites (Par exemple, la hauteur des arbres dans une forêt).

Dans la suite nous nous concentrerons sur les séries spatiales qui peuvent apporter des solutions en classification spatiale telle que nous l'avons définie. Une présentation détaillée des processus stochastiques spatiaux peut être trouvée dans Cliff et Ord (1981) et Ripley (1981).

3.2.2 Séries spatiales

Dans le cas où le processus $\{\mathbf{X}_t\}$ est défini en un certain nombre de sites spécifiques, et prend ses valeurs sur un ensemble discret ou continu, il existe des modèles, qui possèdent beaucoup de points commun avec les séries temporelles.

Ces modèles, comme dans le cas des processus générateurs de points, sont des alternatives à l'hypothèse d'absence d'autocorrélation spatiale (Ord 1982).

La classe de modèles la plus répandue dans ce contexte est représentée par les champs aléatoires de Markov.

Champs aléatoire de Markov

Les chaînes de Markov sont des processus stochastiques les plus simples pour tenir compte de la non indépendance des v.a. $\mathbf{X}_n$ par rapport à un indice discret n :

Définition 3.2 *Un processus stochastique* $\{\boldsymbol{X}_n : n = 1, 2, ...\}$ *prenant ses valeurs dans un espace fini est une chaîne de Markov si la réalisation de* $\boldsymbol{X}_n$ *sachant toutes les réalisations passées ne dépend que de la dernière valeur prise :*

$$P(\boldsymbol{X}_n = \boldsymbol{x}_n | \boldsymbol{X}_{n-1} = \boldsymbol{x}_{n-1}, ..., \boldsymbol{X}_1 = \boldsymbol{x}_1) = P(\boldsymbol{X}_n = \boldsymbol{x}_n | \boldsymbol{X}_{n-1} = \boldsymbol{x}_{n-1}). \tag{3.1}$$

Ce concept de dépendance markovienne peut être étendu de bien des manières. Les champs de Markov sont une extension de la notion de dépendance markovienne pour des processus stochastiques dont l'indice appartient à un espace multidimensionel et plus seulement à un sous-ensemble de $\mathbb{R}$. Deux cas sont à distinguer :

- les champs de Markov dont l'indice varie de façon continue ;
- les champs de Markov dont l'indice est discret.

Les premiers trouvent leur domaine d'application en physique théorique et les seconds servent entre autre de modèles pour les statistiques ayant un caractère spatial. Seul le second cas sera examiné dans ce document.

Si l'indice n'appartient pas à un sous ensemble de $\mathbb{R}$ mais à un sous ensemble de $\mathbb{R}^d$, les notions de passé et de futur par rapport à un indice t ne tiennent plus, et il faut recourir au concept plus général de voisinage.

Définition 3.3 *(Geman et Geman 1984) Soit* $S = \{s_1, s_2, ..., s_N\}$ *un ensemble d'indices (dans un contexte de modélisation spatiale, l'indice représente les coordonnées d'un site).* $G = \{G_s, s \in S\}$ *un ensemble de parties de* S *est un système de voisinage pour* S *si et seulement si,* $\forall r, s \in S$,

1. $s \notin G_s$,
2. $s \in G_r \Leftrightarrow r \in G_s$.

Notons que $\{S, G\}$ est un graphe.

Un autre concept utile lié à la notion de système de voisinage est celui de clique :

Définition 3.4 *Soit G un système de voisinage sur un ensemble de sites S, une clique c est un sous-ensemble de S tel que tous les éléments de c soient voisins les uns des autres au sens de G.*

L'ensemble des indices (des sites) d'un graphe de voisinage forme un réseau. On distingue les réseaux à mailles régulières et les réseaux à mailles irrégulières. Les premiers sont utilisés pour modéliser la distribution d'une population (végétale, animale...) échantillonnée de manière très régulière lors d'une expérience. Les seconds sont utilisés pour décrire la répartition naturelle d'une population.

Exemple 3.6 Toutes les communes composant un département sont caractérisées par des nombres liés à leur activité agricole. L'activité agricole ne semble pas indépendante de la localisation d'une commune et il semble judicieux de modéliser la répartition spatiale de cette activité par un champ de Markov. L'ensemble S des indices peut être choisi comme les coordonnées du centre de la commune et deux communes sont considérées comme voisines si elles partagent une frontière commune. Les mailles de ce réseau sont irrégulières.

△

Lorsque les relations de voisinage ne sont pas définies de manière explicite et si les sites ne sont pas répartis régulièrement, il faut définir précisément la notion de voisinage avant de pouvoir recourir à une modélisation markovienne. Une solution possible consiste à dessiner une tesselation de Voronoï, et dire que deux sites sont voisins si leurs polygones de Voronoï respectifs partagent un côté commun.

Définition 3.5 *Soit S un ensemble d'indices muni d'un système de voisinage G et $\boldsymbol{X} = \{\boldsymbol{X}_s, s \in S\}$ une famille de variables aléatoires prenant leurs valeurs sur Ω. $\boldsymbol{X}$ est un champ de Markov par rapport à G si :*

1. $P(\boldsymbol{X} = \boldsymbol{x}) > O, \forall \boldsymbol{x} \in \Omega$;
2. $P(\boldsymbol{X}_s = \boldsymbol{x}_s | \boldsymbol{X}_r, r \neq s) = P(\boldsymbol{X}_s = \boldsymbol{x}_s | \boldsymbol{X}_r, r \in G_s), \forall s \in S.$

Cette définition affirme que l'état d'un site ne dépend que des voisins immédiats de ce site, mais elle n'est pas directement utilisable en pratique pour définir un champ de Markov sans la connaissance du théorème d'Hammersley-Clifford démontré en 1971 :

Théorème 3.1 *Soit $\boldsymbol{X} = \{\boldsymbol{X}_s, s \in S\}$ un champ de Markov sur un réseau S de n sites, muni d'un système de voisinage. La distribution de probabilité du champ $\boldsymbol{X}$ est une distribution de Gibbs:*

$$\pi(\boldsymbol{x}) = P(\boldsymbol{X} = \boldsymbol{x}) = \frac{1}{Z(T)} e^{-E(\boldsymbol{x})/kT},$$

où la fonction d'énergie E est de la forme:

$$E(\boldsymbol{x}) = \sum_{1\leq i\leq n} x_i G_i(x_i) + \sum \sum_{1\leq i<j\leq n} x_i x_j G_{i,j}(x_i, x_j) + \cdots + x_1 x_2 \cdots x_n G_{1,2,\cdots,n}(x_1 x_2 \cdots x_n),$$

telle que pour tout $1 \leq i < j \cdots < s \leq n$, la fonction $G_{i,j,\cdots,s}$ peut être non nulle si et seulement si les sites $i, j, \cdots, s$ forment une clique.

Un modèle binaire

Comme le fait observer Besag (1974), dans le cas où les variables de chaque site sont binaires, les fonctions G peuvent être remplacées par de simple paramètres sans perte de généralité. Si on se limite aux cliques de un et deux sites, la fonction d'énergie considérée aura la forme suivante:

$$E(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i x_i + \sum_{i=1}^{N} \sum_{j\in G_i, i<j} \beta_{i,j} x_i x_j \tag{3.2}$$

et la probabilité conditionnelle d'avoir $X_i = x_i$ sachant la réalisation de tous les voisins sera simplement:

$$P(X_i = x_i | X_j, j \neq i) = \frac{\exp\left(x_i(\alpha_i + \sum \beta_{i,j} x_j)\right)}{1 + \exp\left(\alpha_i + \sum \beta_{i,j} x_j\right)} \tag{3.3}$$

Exemple 3.7 (Billoire 1992) Le modèle de champ de Markov le plus connu trouve son origine en mécanique statistique. Il s'agit du modèle d'Ising inventé en 1925 pour expliquer certaines propriétés des ferromagnétiques. Les variables X_s (qui représentent la valeur du 'spin' d'un atome) peuvent prendre deux valeurs $+1$ ou -1, et sont associées aux sites d'un réseau hypercubique S muni d'un système de voisinage. À l'équilibre, la probabilité que le système soit dans une configuration $\mathbf{x}$ est une distribution de Gibbs de fonction d'énergie :

$$E(\mathbf{x}) = \alpha \sum_{s\in S} x_s + \beta \sum_{r,s\in S/r \text{ et } s \text{ voisins}} x_s x_r, \tag{3.4}$$

avec α et β des paramètres mesurant respectivement le champ magnétique extérieur et les forces de liaison. Lorsque $\alpha = 0$ (pas de champ extérieur) et que la température est grande toutes les configurations deviennent équiprobables et lorsque la température est basse deux configurations dominent : celle où tous les spins valent $+1$ et celle où tous les spins valent -1. A basse température le système reste piégé dans l'un des deux états et met très longtemps à en sortir. Ceci explique le phénomène d'aimantation rémanente.

△

Le modèle de Strauss (1977)

Le modèle de Strauss peut être considéré comme une généralisation du modèle d'Ising, dans le cas où les variables prennent des valeurs discrètes. Dans le cas isotrope, la distribution de Gibbs est définie par la fonction d'énergie :

$$E(\mathbf{x}) = \beta \sum_{r,s \in S/r \ et \ s \ voisins} I\!I_{\{\mathbf{X}_s = \mathbf{X}_r\}}. \tag{3.5}$$

Cette fonction d'énergie compte le nombre de paires de sites voisins qui ont la même valeur. Elle est maximum si les variables de tous les sites prennent une même valeur.

Des modèles gaussiens

Dans de nombreux cas, il est raisonnable de modéliser la distribution jointe des sites (ou plutôt d'une certaine variable en chaque site) par une loi normale multidimensionnelle. Dans cette optique, deux approches sont possibles :

- l'approche simultanée dite SAR (Simultaneous Autoregression) ;
- l'approche conditionnelle dite CAR (Conditional Autoregression).

La première solution définit le processus par N équations autorégressives simultanées :

$$X_i = \mu_i + \sum \beta_{i,j}(X_j - \mu_j) + \epsilon_i, \tag{3.6}$$

où $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$ (Besag 1974). Cette définition correspond à la distribution de probabilité suivante :

$$P(\mathbf{X} = \mathbf{x}) = (2\pi\sigma^2)^{\frac{-1}{2}N} det(\mathbf{B}) exp\{\frac{-1}{2\sigma^2}(\mathbf{x} - \boldsymbol{\mu})^T \mathbf{B}^T \mathbf{B}(\mathbf{x} - \boldsymbol{\mu})\}, \tag{3.7}$$

où $\boldsymbol{\mu}$ est un vecteur de taille N contenant les moyennes μ_i de tous les sites, et $\mathbf{B}$ une matrice $N \times N$ inversible qui contient des 1 sur la diagonale et les terme $-\beta_{i,j}$ partout ailleurs.

La deuxième approche définit le modèle de manière conditionnelle,

$$I\!E(\mathbf{X}_i | X_j = x_j, j \neq i) = \mu_i + \sum \beta_{i,j}(x_j - \mu_j).$$

et

$$var(\mathbf{X}_i | X_j = x_j, j \neq i) = \sigma^2.$$

Dans ce cas la densité du champ gaussien s'écrit :

$$P(\mathbf{X} = \mathbf{x}) = (2\pi\sigma^2)^{\frac{-1}{2}N} det(\mathbf{B})^{\frac{1}{2}} exp\{\frac{-1}{2\sigma^2}(\mathbf{x} - \boldsymbol{\mu})^T \mathbf{B}(\mathbf{x} - \boldsymbol{\mu})\}. \tag{3.8}$$

3.3 Approche statistique de la segmentation non supervisée

Dans de nombreux domaines scientifiques, les images sont une source d'information et donnent matière à une analyse. Une image est une grille, le plus souvent rectangulaire, dont chaque case est un pixel. Chaque pixel est caractérisé par une mesure, ou un vecteur de mesures. Typiquement la mesure est une niveau de gris entre 0 et 256, ou encore, dans le cas des images en couleurs, un vecteur de dimension 3 dont chaque composante mesure l'intensité d'une des couleurs primaires.

Des analyses très diverses sont réalisables, mais dans ce document nous détaillerons simplement les techniques statistiques d'analyse d'images qui relèvent de la classification automatique : la segmentation non supervisée d'images. Dans ce contexte le but est de résumer, de simplifier l'information contenue dans l'image, c'est-à-dire affecter chaque pixel à une certaine classe, le nombre de classes étant réduit par rapport au nombre de valeurs que peut prendre le pixel.

Exemple 3.8 Au cours de l'étude d'une céramique qui contient des grains de carbure de silicium, on veut déterminer le pourcentage de carbure de silicium contenu dans le matériau. Une méthode possible est de prendre une photo d'une surface de ce matériau puis déterminer quelle surface est couverte par les grains de carbure de silicium. Le rapport entre cette surface et la surface totale donne une approximation du pourcentage cherché. La photo est numérisée et l'image résultante est codée en 256 niveaux de gris. La première étape de l'analyse consiste donc à segmenter l'image en deux classes, c'est-à-dire distinguer les pixels qui représentent le carbure de silicium du reste.

△

Dans un cadre statistique, plusieurs approches sont possibles pour aborder le problème de la segmentation. Masson et Pieczinsky (1993) proposent la classification suivante :

- Les méthodes globales considèrent une image comme la réalisation d'une variable aléatoire. Les hypothèses posées concernent l'image dans son ensemble. Notons que souvent une seule image sert de support à l'analyse et que dans ce cas précis, l'échantillon considéré est de taille 1.
- Les méthodes locales peuvent être de deux types :
  - Contextuelles : Les individus considérés sont de petits groupes de pixels, nommés contextes. Typiquement un contexte pourra être composé d'un pixel, et de ses deux voisins horizontaux et ses deux voisins verticaux.
  - Aveugles : Les individus constituant l'échantillon sont les pixels. Dans ce cas, aucune information spatiale n'est prise en compte et les méthodes utilisées sont les méthodes classiques de la classification automatique (algorithme EM,...).

3.3.1 Méthodes globales et approche bayésienne

La modélisation statistique globale des images pour la segmentation suppose l'existence de deux champs aléatoires. L'image observée, $\mathbf{x}$, est la réalisation d'un premier champ aléatoire $\mathbf{X} = \{\mathbf{X}_s, s \in S\}$ et l'image segmentée, $\mathbf{c}$, est la réalisation d'un second champ $\mathbf{C} = \{\mathbf{C}_s, s \in S\}$ (avec S, l'ensemble des pixels). Les variables aléatoires $\mathbf{X}_s$ prennent leur valeur dans $\mathbb{R}^d$ et les $\mathbf{C}_s$ dans un ensemble fini $\Omega = \{\omega_1, \cdots, \omega_K\}$ avec K le nombre de classes. La modèle considère que $\mathbf{X}$ est une observation bruitée de $\mathbf{C}$. Ainsi une relation existe entre les deux champs :

$$\mathbf{X} = R(\mathbf{C}, \mathbf{N}), \tag{3.9}$$

où $\mathbf{N}$ est le bruit. La segmentation d'image pose alors le problème le suivant : une ou plusieurs réalisations de $\mathbf{X}$ étant disponibles, comment trouver un estimateur $\hat{\mathbf{c}}$ de $\mathbf{c} = R(\mathbf{x}, \mathbf{N})$.

Ce problème peut être résolu en utilisant une approche bayésienne. Dans ce cadre, une distribution a priori $P(\mathbf{C})$ sur l'image segmentée, ainsi qu'une distribution sur les données, $P(\mathbf{X}|\mathbf{C})$, sont postulées. Notons que la distribution de probabilité $P(\mathbf{X}|\mathbf{C})$ est déterminée par la relation qui existe entre les champs $\mathbf{X}$ et $\mathbf{C}$ (Equation 3.9). La distribution a posteriori peut être exprimée par le théorème de Bayes :

$$P(\mathbf{C} = \mathbf{c}|\mathbf{X} = \mathbf{x}) = \frac{P(\mathbf{C} = \mathbf{c})P(\mathbf{X} = \mathbf{x}|\mathbf{C} = \mathbf{c})}{P(\mathbf{X} = \mathbf{x})}$$

La stratégie bayésienne consiste alors à minimiser le coût a posteriori :

$$\hat{\mathbf{c}} = arg \min_{\mathbf{c}} \rho(\mathbf{c}|\mathbf{x})$$

avec

$$\rho(\mathbf{c}|\mathbf{x}) = \mathbb{E}[L(\mathbf{c}, \mathbf{Z})|\mathbf{x}] = \sum_{\mathbf{z}} L(\mathbf{c}, \mathbf{z}) P(\mathbf{C} = \mathbf{z}|\mathbf{X} = \mathbf{x}),$$

où $L(\mathbf{c}, \mathbf{z})$ est le coût de dire que l'image segmentée est $\mathbf{c}$ lorsque l'image segmentée est en fait $\mathbf{z}$. Deux fonctions de coût sont couramment utilisées en analyse d'image :

- $L(\mathbf{c}, \mathbf{z}) = \mathbb{I}_{\{\mathbf{c} \neq \mathbf{z}\}}$, c'est le coût "0–1" qui vaut 0 pour la bonne décision et 1 pour une mauvaise décision. Dans ce cas l'estimateur de $\mathbf{c}$ est le maximum a posteriori (MAP) :
$$\hat{\mathbf{c}} = arg \max_{\mathbf{c}} P(\mathbf{C} = \mathbf{c}|\mathbf{x})$$

- $L(\mathbf{c}, \mathbf{z}) = \sum_{s \in S} \mathbb{I}_{\{\mathbf{c}_s \neq \mathbf{z}_s\}}$, qui considère non plus l'image dans son intégralité mais le nombre de pixels bien classés. Dans ce cas l'estimateur est celui qui maximise les probabilités marginales a posteriori (MPM) (Marroquin *et al.* 1987):
$$\hat{\mathbf{c}_s} = arg \max_{\mathbf{c}_s} P(\mathbf{C}_s = \mathbf{c}_s|\mathbf{x}), \ \forall s \in S.$$

L'approche bayésienne de la segmentation non supervisée pose les problèmes suivants :

1. Quel modèle relatif aux données observées adopter ($P(\mathbf{X}|\mathbf{C})$), et comment modéliser les connaissances a priori sur la structure de l'image segmentée ($P(\mathbf{C})$)?
2. Comment estimer les paramètres du modèle choisi?
3. Comment trouver l'estimateur de l'image segmentée qui minimise la fonction de coût choisie?

Les méthodes globales de segmentation résolvent souvent le problème du choix de modèle en utilisant les champs de Markov utilisés en statistique spatiale.

3.3.2 Modèles Markoviens en analyse d'image

Si le modèle de l'image est un champ de Markov, alors la distribution des données, les probabilités a priori et a posteriori sont des distributions de Gibbs définies de la façon suivante :

$$\begin{aligned} P(\mathbf{x}|\mathbf{c}) &\propto \exp -U^r(\mathbf{x},\mathbf{c},\Phi) \\ P(\mathbf{c}) &\propto \exp -U^a(\mathbf{c},\beta) \\ P(\mathbf{c}|\mathbf{x}) &\propto \exp\{-U^a(\mathbf{c},\beta) - U^r(\mathbf{x},\mathbf{c},\Phi)\} \end{aligned}$$

où Φ et β sont les paramètres des distributions. L'énergie U^a est relative aux informations a priori à propos de l'image segmentée $\mathbf{c}$. Plus $\mathbf{c}$ respecte les informations a priori et plus U^a est petite. L'énergie U^r est dite énergie de rappel aux données, c'est par son intermédiaire que la relation entre les champs $\mathbf{X}$ et $\mathbf{C}$ est modélisée.

Exemple 3.9 Reprenons l'exemple 3.8 de la segmentation d'une image en niveau de gris en deux classes : C_s prend ses valeurs dans $\{-1, 1\}$. Deux pixel voisins ont plus de chance d'appartenir à la même classe que deux pixels quelconques. Cette information a priori peut être modélisée par le modèle d'Ising qui est une distribution de Gibbs de fonction d'énergie :

$$U^a(\mathbf{c}) = -\beta \sum_{r,s\in S/r \text{ et } s \text{ voisins}} c_s c_r. \tag{3.10}$$

Si l'on considère que l'image observée est une dégradation de l'image segmentée telle qu'en chaque pixel x_s le bruit est gaussien de moyenne c_s et de variance $1/\Phi$, l'énergie de rappel aux données est:

$$U^r(\mathbf{x},\mathbf{c},\beta) = -\Phi \sum_{s\in S}(x_s - c_s)^2 \tag{3.11}$$

△

Les informations sur les relations entre les champs $\mathbf{X}$ et $\mathbf{C}$ et les a priori sur la forme du champ $\mathbf{C}$ peuvent être associée à des énergies. L'énergie de la distribution a posteriori est dans ce cas la somme de toutes ces énergies :

$$U = \sum U_i = U^a + U^r \tag{3.12}$$

Si la traduction des connaissances *a priori* dans une formulation markovienne semble assez aisée, l'estimation des paramètres du modèle et la segmentation de l'image constituent des problèmes délicats.

3.3.3 Estimation des paramètres et reconstitution d'image

Dans un contexte markovien, l'estimation des paramètres du modèle nécessite des réalisations d'image segmentée issues de ce modèle, et la segmentation nécessite la connaissances des paramètres du modèle. Pour résoudre ce problème, de nombreux algorithmes de segmentation non supervisée basés sur les champs de Markov sont des algorithmes itératifs qui utilisent un principe similaire à celui de l'algorithme EM :

1. choix initial de (β^0, Φ^0),
2. à l'itération m :
   - simulation d'une ou plusieurs images segmentées en utilisant le modèle de paramètres (β^m, Φ^m),
   - estimation de $(\beta^{m+1}, \Phi^{m+1})$ en utilisant l'image observée et une ou plusieurs images segmentées obtenues au cours des itérations précédentes.

Nous donnerons quelques exemples de tels algorithmes dans la suite.

Segmentation d'image connaissant les paramètres du modèle

Rappelons que la distribution a posteriori $P(\mathbf{c}|\mathbf{x})$ est un champ de Markov de paramètres (β, Φ). Lorsque ces paramètres sont connus, il est possible de simuler des réalisations de cette distribution de Gibbs en utilisant un échantillonneur de Gibbs de la forme :

1. choix initial d'une image segmentée $\mathbf{c}^0$,
2. à l'itération m :
   - un pixel i est choisi,
   - ${\mathbf{c}_i}^{m+1}$ est tiré au hasard suivant la loi $P(\mathbf{c}_i|\mathbf{x}; \mathbf{c}_j^m, j \neq i)$.

Cet échantillonneur de Gibbs produit une suite d'images $\mathbf{c}^0, \cdots, \mathbf{c}^m$. Quand m est grand, on peut considérer que $\mathbf{c}^m$ est une réalisation de $P(\mathbf{c}|\mathbf{x})$. De plus on a la propriété suivante :

$$\lim_{m\to\infty} \frac{1}{m}[f(\mathbf{c}^0) + \cdots + f(\mathbf{c}^m)] = I\!E[f(\mathbf{C})],$$

avec f une fonction mesurable quelconque et $\mathbf{C}$ une variable aléatoire de loi $P(\mathbf{c}|\mathbf{x})$.

Si un échantillonneur de Gibbs permet d'obtenir des simulations suivant la loi souhaitée, il ne permet pas de déterminer directement l'image segmentée qui minimise le critère du MAP ou du MPM.

Geman et Geman (1984) proposent un échantillonneur de Gibbs modifié qui fait tendre la suite des images $\mathbf{c}^m$ vers le MAP. L'idée consiste à utiliser le principe du recuit simulé en introduisant un paramètre de température T dans la distribution $P(\mathbf{c}|\mathbf{x})$, qui sécrit

$$P(\mathbf{c}|\mathbf{x}) = \frac{\exp(\frac{1}{T} \cdot U_{\Phi,\beta}(\mathbf{c}, \mathbf{x}))}{Z(T)}.$$

A chaque étape la température décroît vers zéro. La convergence de l'algorithme est démontrée si la température décroît assez lentement. Cette décroissance lente a le désavantage de demander un très grand nombre d'itérations avant d'obtenir un estimateur du MAP satisfaisant.

Pour pallier cette lenteur, Besag (1986) propose un algorithme déterministe qui correspond à l'algorithme de Geman en prenant une température nulle dès le départ. Chaque itération de cet algorithme, baptisé ICM (iterative Conditional Mode) modifie la classe d'un pixel de la façon suivante :

$$\mathbf{c}_i{}^{m+1} = arg \max_{\mathbf{c}_i} P(\mathbf{c}_i|\mathbf{x}; \mathbf{c}_j^m, j \neq i).$$

L'algorithme ICM a l'avantage de converger en moins de 10 examens de toute l'image et de faire croître $P(\mathbf{c}^m|\mathbf{x})$ à chaque itération. Le principal inconvénient de l'algorithme est sa forte dépendance par rapport aux conditions initiales.

Une autre approche consiste à considérer le critère du MPM. L'estimation des probabilités marginales peut être réalisée grâce à un échantillonneur de Gibbs (Marroquin et Girosi 1993). En chaque pixel i, la fréquence empirique, m_{ik}, de la classe k est mesurée et on classe chaque pixel comme suit :

$$c_{ik} = \begin{cases} 1 \ si \ k = arg \max_\ell m_{i\ell}; \\ \\ 0 \ sinon. \end{cases}$$

Estimation des paramètres à partir d'une image segmentée

Lorsque l'on dispose d'une image bruitée et de l'image segmentée correspondante, plusieurs solutions existent pour estimer les paramètres (Φ, β). Pour rendre les calculs

possibles, on suppose que

$$P_\Phi(\mathbf{x}|\mathbf{c}) = \prod_{i=1}^{N} P(\mathbf{x}_i|\mathbf{c}_i),$$

ce qui revient à considérer que le bruit est spatialement non corrélé et que les observations sont indépendantes conditionnellement à la connaissance des classes. Sous ces conditions, l'estimation de Φ connaissant $\mathbf{c}$ est un problème simple. L'estimation de β semble par contre une tâche plus délicate.

Notons qu'il est, en pratique, impossible de calculer la vraisemblance d'un paramètre β donné. En effet, la vraisemblance s'écrit

$$\ell(\beta; \mathbf{c}) = \frac{\exp -U^a(\mathbf{c}, \beta)}{Z(\beta)}$$

où $Z(\beta)$ est incalculable car c'est une somme sur toutes les images segmentées possible. Pour contourner ce problème, Besag (1974) propose de trouver un estimateur qui optimise un critère calculable, la pseudo vraisemblance :

$$\beta^* = arg \max_\beta \prod_{i \in code\ell} P_\beta(\mathbf{c}_i|V(\mathbf{c}_i))$$

où $code\ell$ est un ensemble de pixels conditionnellement indépendants et $V(\mathbf{c}_i)$ est le voisinage du pixel i. Cette méthode a l'inconvénient de n'utiliser qu'une partie des données. Une extension assez naturelle consiste à utiliser tous les pixels même s'ils ne sont pas indépendants. Soit :

$$\beta^* = arg \max_\beta \prod_{i=1}^{N} P_\beta(\mathbf{c}_i|V(\mathbf{c}_i)).$$

D'après Lakshmanan et Derin (1989), ce critère donnerait des résultats plus fiables. Notons que même si la pseudo vraisemblance est facilement calculable pour une valeur donnée de β, l'obtention d'un β^* nécessite souvent l'utilisation d'algorithmes d'optimisation numérique.

Younes (1988) suggère une idée originale pour trouver un estimateur du maximum de vraisemblance de β. Soit $\mathbf{c}_o$ l'image segmentée disponible. Une condition nécessaire d'optimalité est

$$\nabla_\beta P_\beta(\mathbf{c}_o) = 0,$$

pour $\beta = \hat{\beta}_{MV}$. Cette équation peut se mettre sous la forme :

$$U^{a\prime}(\mathbf{c}) = I\!E[U^{a\prime}(\mathbf{C})]$$

où $U^{a\prime}(\mathbf{c})$ est le gradient de l'énergie U^a par rapport au vecteur β. Une montée de gradient stochastique peut alors être mis en œuvre pour résoudre cette dernière équation :

1. choix initial du vecteur β^0,

2. à l'itération m :

   - exécution d'une étape d'un échantillonneur de Gibbs qui simule $P_{\beta^m}(\mathbf{c})$; une nouvelle image $\mathbf{c}^m$ est obtenue,
   - calcul de
   $$\beta^{m+1} = \beta^m + \frac{\lambda}{m+1}[U^{a\prime}(\mathbf{c}^{m+1}) - U^{a\prime}(\mathbf{c}_o)]$$
   où λ est une constante.

La convergence de cet algorithme est démontrée, mais il est bien évident que le maximum atteint n'est que local.

Segmentation non supervisée

Les algorithmes de segmentation non supervisée utilisent donc en alternance des méthode de simulations de $\mathbf{c}$, utilisant des paramètres connus, et des techniques d'estimation utilisant les images segmentées obtenues. Ces algorithmes sont trop nombreux pour que nous les détaillions tous (Chalmond 1989, Besag 1986, Geman et Geman 1984, Lakshmanan et Derin 1989, Pieczinsky et Cahen 1994). Nous donnerons donc un seul exemple proche des algorithmes proposés dans ce chapitre et qui illustrera les comparaisons numériques.

Chalmond (1989) propose un algorithme baptisé EM Gibbsien qui est destiné à trouver les paramètres d'un modèle Markovien qui maximisent la pseudo vraisemblance et donne une image segmentée sur le principe du MPM. La pseudo vraisemblance s'écrit :

$$\mathcal{P}_\Theta(\mathbf{c}, \mathbf{x}) = P_\Phi(\mathbf{x}|\mathbf{c}) \cdot \prod_{i=1}^{N} P_\beta(\mathbf{c}_i|V(\mathbf{c}_i))$$

où $\Theta = (\Phi, \beta)$. S'inspirant de l'algorithme EM, l'algorithme prend la forme suivante :

1. choix initial du vecteur θ^0,

2. à l'itération $(m+1)$:

   - **Etape E :**
     - simulation d'une nouvelle série d'images $\mathbf{c}^1, \mathbf{c}^2, \cdots, \mathbf{c}^{s_o}, \cdots, \mathbf{c}^{s_m}$ suivant la loi $P_{\Theta^m}(\mathbf{c}_i|\mathbf{x}; \mathbf{c}_j, j \neq i)$ (s_0 est le nombre d'itérations requis pour que la suite $\{\mathbf{c}^m\}$ soit en régime stationnaire),
     - estimation des $u_{ik} = P_{\Theta^m}(c_i = k|\mathbf{x})$ (probabilité que $\mathbf{x}_i$ appartienne à la classe k conditionnellement à l'image observée) :
     $$u_{ik} = \frac{1}{s_m - s_0} \sum_{s=s_0}^{s_m} I\!I_{\{c_{ik}=1\}};$$

- **Etape M :** Calcul de
  - $\Phi^{m+1} = arg\max_{\Phi} I\!E[\log P_{\Phi}(\mathbf{c}|\mathbf{x})|\mathbf{x}, \Phi^{m+1}]$. Dans le cas d'un mélange de K gaussiennes :

$$\boldsymbol{\mu}_k^{m+1} = \frac{\sum_{i=1}^{n} u_{ik}}{n_k}; \tag{3.13}$$

$$\boldsymbol{\Sigma}_k^{m+1} = \sum_{k=1}^{K}\sum_{i=1}^{n} \frac{u_{ik}(\mathbf{x}_i - \boldsymbol{\mu}_k^{m+1})(\mathbf{x}_i - \boldsymbol{\mu}_k^{m+1})^t}{n_k}; \tag{3.14}$$

    où $n_k = \sum_{i=1}^{n} u_{ik}$.
  - Au lieu de calculer

$$\beta^{m+1} = arg\max_{\beta} I\!E[\log \prod_{i=1}^{N} P_{\beta}(\mathbf{c}_i|V(\mathbf{c}_i))|\mathbf{x}, \Phi^{m+1}],$$

    Chalmond calcule directement les probabilités $P_{\beta}(c_{ik} = 1|\mathbf{c}_j, j \neq i)$. D'après la distribution *a priori* $P_{\beta}(\mathbf{c})$ choisit par l'auteur, il constate que la probabilité $P_{\beta}(c_{ik} = 1|\mathbf{c}_j, j \neq i)$ prend un nombre fini de valeurs qui dépendent de la classe k du site i ainsi que de la configuration du voisinage entourant ce site. En notant $P(k|j)$ la valeur de la probabilité d'avoir le site i appartenant à la classe k, conditionnellement au voisinage j, Chalmond calcule les

$$\hat{P}(k|j) = arg\max_{P(k|j)} I\!E[log \prod_{i=1}^{N} P_{\beta}(\mathbf{c}_i|V(\mathbf{c}_i))|\mathbf{x}, \Phi^{m+1}].$$

    Notons que ces valeurs sont utilisées dans l'itération E suivante pour déterminer les $P_{\beta}(c_{ik} = 1|\mathbf{c}_j, j \neq i)$ qui servent à l'échantillonneur de Gibbs car

$$P(c_{ik} = 1|\mathbf{x}; \mathbf{c}_j,\ j \neq i) \propto P_{\Phi}(\mathbf{x}_i|\Phi) \cdot P_{\beta}(c_{ik} = 1|\mathbf{c}_j, j \neq i).$$

3.4 Classification Spatiale et Algorithme EM

En classification automatique, les modèles de mélanges gaussiens permettent de proposer des méthode de classification efficaces et variées. Dans ce cadre, les données à classer sont considérées comme un échantillon $(\mathbf{x}_1, ..., \mathbf{x}_N)$ de variables aléatoires à valeurs dans $I\!R^d$, indépendantes, et identiquement distribuées suivant une densité mélange de gaussiennes $f(\mathbf{x}_i|\Phi)$. Dans ce cas, la vraisemblance peut s'écrire sous la forme d'un critère de classification floue (Hathaway 1986) :

$$D(\mathbf{c}, \Phi) = \sum_{i=1}^{N}\sum_{k=1}^{K} c_{ik} \log p_k f_k(\mathbf{x}_i|\theta_k) - \sum_{i=1}^{N}\sum_{k=1}^{K} c_{ik} \log c_{ik}, \tag{3.15}$$

avec

$$c_{ik} = \frac{p_k f_k(\mathbf{x}_i|\theta_k)}{f(\mathbf{x}_i)}. \tag{3.16}$$

Rappelons que la matrice c possède alors toutes les caractéristiques d'une matrice de classification floue :

$$\mathbf{c} = \{c_{ik} : 0 \leq c_{ik} \leq 1, \sum_{k=1}^{K} c_{ik} = 1, \sum_{i=1}^{N} c_{ik} > 0 (1 \leq i \leq n, 1 \leq k \leq K)\}.$$

Dans la suite de cette section, nous proposons d'adapter ce critère aux données spatiales.

3.4.1 Recherche d'une partition floue

Pour prendre en compte le côté spatial des données, nous proposons de chercher une partition floue qui optimise le critère d'Hathaway pénalisé par un terme favorisant les classes géographiquement homogènes.

La définition du terme régularisant présuppose l'existence d'une structure de voisinage, résumée par la matrice $\mathbf{V}$:

$$v_{ij} = \begin{cases} 1 & \text{si } \mathbf{x}_i \text{ et } \mathbf{x}_j \text{ sont voisins ;} \\ 0 & \text{sinon.} \end{cases}$$

Les coefficients de la matrice de voisinage peuvent être calculés de diverses manières : par un graphe de voisinage, par une fonction de la distance géographique entre deux sites...

Exemple 3.10 (Besag 1974)
Soient N individus localisés spatialement. Si une tesselation de Voronoï (Voir Figure 3.5) est réalisée, on pourra par exemple définir la matrice de voisinage de la façon suivante :

$$v_{ij} = \begin{cases} 0 & \text{si les polygones de sites } i \text{ et } j \text{ n'ont pas de bord commun;} \\ 1 & \text{sinon.} \end{cases}$$

△

Nous proposons le terme régularisant (dans un sens large) suivant :

$$G(\mathbf{c}) = \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{k=1}^{K} c_{ik} \cdot c_{kj} \cdot v_{ij} \tag{3.17}$$

où K est le nombre de classes et c_{ik} le coefficient flou d'appartenance de l'individu $\mathbf{x}_i$ à la classe K.

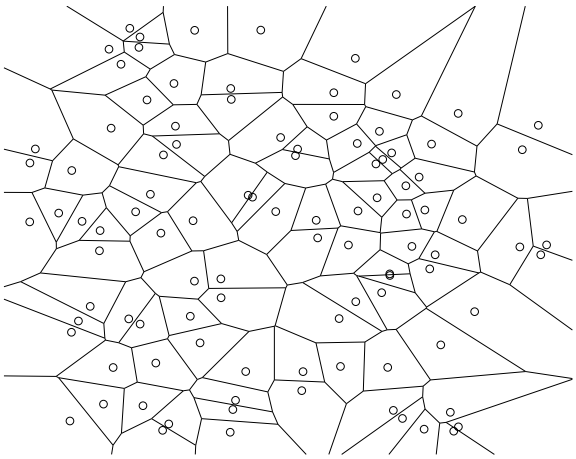


FIG. 3.5 - : Définition des relation de voisinage par pavage de Voronoï

Dans la suite nous supposerons que $\mathbf{V}$ est une matrice symétrique de diagonale nulle, ce qui nous permettra de faire un lien avec les champs de Markov et de simplifier considérablement les calculs.

Si l'on note $\mathbf{c}_i = [c_{i1} \cdots c_{iK}]^t$ le vecteur de classification du i^{e}individu, le critère peut être reformulé comme suit :

$$G(\mathbf{c}) = \sum_{i<j} v_{ij} \cdot \mathbf{c}_i^t \cdot \mathbf{c}_j. \tag{3.18}$$

Plus les classes contiennent d'individus géographiquement voisins et plus le terme G est important.

Exemple 3.11 Soit $\mathbf{c}$ une matrice de classification dure ($c_{ik} = 1$ si et seulement si $\mathbf{x}_i \in P_k$, et $c_{ik} = 0$ sinon) et une matrice de voisinage $\mathbf{V}$. Dans ce cas le terme G représente le nombre de couples de sites voisins qui appartiennent à la même classe.

△

Le nouveau critère que nous considérons s'exprime alors comme la somme pondérée du critère D et du terme de pénalisation géographique:

$$U(\mathbf{c}, \Phi) = D(\mathbf{c}, \Phi) + \beta \cdot G(\mathbf{c}) \tag{3.19}$$

avec β un coefficient fixé qui contrôle l'importance du rôle de l'information spatiale.

Nous suggérons l'utilisation d'un algorithme d'optimisation alternée pour optimiser ce critère. Ayant la même structure que l'algorithme EM, une itération de l'algorithme proposé maximise alternativement $U(\mathbf{c}, \Phi)$ par rapport à la matrice de

classification, $\mathbf{c}$, puis par rapport aux paramètres Φ du mélange. Nous avons baptisé cet algorithme *Neighborhood EM algorithm* (NEM) :

1. **Initialisation :**
   - Φ, les paramètres du mélange sont initialisés ;
   - la matrice de voisinage $\mathbf{V}$ doit être définie.
2. **Itérations :** L'algorithme s'arrête si les matrices de classification restent inchangées deux itérations de suite. Chaque itération comprend deux étapes :
   - **Estimation** d'une nouvelle matrice $\mathbf{c}$ qui maximise $U(\mathbf{c}, \Phi^q)$:

$$\mathbf{c}^{q+1} = arg\max_{\mathbf{c}} U(\mathbf{c}, \Phi^q) \tag{3.20}$$

Les conditions nécessaires d'optimalité amènent les équations suivantes:

$$\begin{cases} \frac{\partial U}{\partial c_{ik}}|_{c_{ik}=c_{ik}^{q+1}} = \log\left(p_k f_k(\mathbf{x}_i|\theta_k)\right) - 1 - \log c_{ik}^{q+1} + \lambda_i + \beta \sum_{j=1}^{N} c_{jk}^{q+1} v_{ij} = 0; \\ \sum_{k=1}^{K} c_{ik}^{q+1} = 1, \end{cases}$$

ce qui donne,

$$\begin{cases} c_{ik}^{q+1} = \exp\left\{\log(p_k f_k(\mathbf{x}_i|\theta_k)) - 1 + \lambda_i + \beta \sum_{j=1}^{N} c_{jk}^{q+1} v_{ij}\right\}; \\ \sum_{k=1}^{K} \exp\left\{\log(p_k f_k(\mathbf{x}_i|\theta_k)) - 1 + \lambda_i + \beta \sum_{j=1}^{N} c_{jk}^{q+1} v_{ij}\right\} = 1. \end{cases}$$

Finalement, on obtient $N \times K$ équations du type :

$$c_{ik}^{q+1} = \frac{p_k f_k(\mathbf{x}_i|\theta_k) \cdot \exp\{\beta \sum_{j=1}^{N} c_{jk}^{q+1} v_{ij}\}}{\sum_{\ell=1}^{K} p_\ell f_\ell(\mathbf{x}_i|\theta_\ell) \cdot \exp\{\beta \sum_{j=1}^{N} c_{j\ell}^{q+1} v_{ij}\}}. \tag{3.21}$$

La matrice de classification $\mathbf{c}^{q+1}$, est donc la solution du système :

$$\mathbf{c}^{q+1} = F(\mathbf{c}^{q+1}).$$

Pour résoudre ce système d'équations non linéaires nous proposons une méthode de type point fixe, avec :

$$\begin{cases} \mathbf{c}^{q+1;0} = \mathbf{c}^q; \\ \mathbf{c}^{q+1;m+1} = F(\mathbf{c}^{q+1;m}). \end{cases}$$

D'un point de vue pratique quelques itérations suffisent à calculer une nouvelle matrice de classification raisonnable qui peut être utilisée pour l'étape suivante.

– **Maximisation** du critère par rapport au vecteur de paramètres Φ. Cette étape est identique à l'étape M de l'algorithme EM :

$$\Phi^{q+1} = arg \max_{\Phi} U(\mathbf{c}^{q+1}, \Phi) \tag{3.22}$$

Notons que cette étape est identique à l'étape M de l'algorithme EM, car le terme de pénalisation est indépendant de la matrice de classification, l'on a :

$$\Phi^{q+1} = arg \max_{\Phi} D(\mathbf{c}^{q+1}, \Phi). \tag{3.23}$$

Théorème 3.2 *Si $\beta < \frac{1}{V_{max}}$, où $V_{max} = \max_i \sum_j v_{ij}$ est le nombre maximum de voisins d'un individu, alors la fonction F de $]0,1[^{N\times K}$ dans $]0,1[^{N\times K}$ est une contraction et admet un unique point fixe, c^+, tel que :*

$$\boldsymbol{c}^+ = arg \max_{\boldsymbol{c}} U(\boldsymbol{c}, \Phi).$$

Démonstration : Pour des raisons pratiques, la démonstration adopte les convention suivantes :

– $U_\Phi(\mathbf{c})$ dénote la fonction $U(\mathbf{c}, \Phi)$ pour Φ fixé,

– $\mathbf{c}$ dénote la matrice de classification mise sous forme vectorielle. C'est un vecteur colonne à $N \times K$ éléments :

$$\mathbf{c} = (c_{11}, c_{12}, \cdots c_{1K}, c_{21}, \cdots, c_{NK})^t.$$

Chaque élément possède un indice double. Ainsi c_{ik} est le $i \times k$ième élément du vecteur.

– F est la fonction de $]0,1[^{N\times K}$ dans $]0,1[^{N\times K}$ de l'étape d'estimation de NEM :

$$F(\mathbf{c}) = (F_{11}(\mathbf{c}), F_{12}(\mathbf{c}), \cdots, F_{1K}(\mathbf{c}), F_{21}(\mathbf{c}), \cdots, F_{NK}(\mathbf{c}))^t.$$

Montrons que $U_\Phi(\boldsymbol{c})$ admet un unique maximum : La fonction U_Φ est continue par rapport à $\mathbf{c}$ et sa matrice hessienne $H(\mathbf{c})$, est définie quelque soit $\mathbf{c} \in]0,1[^{N\times K}$:

$$\frac{\partial^2 U_\Phi}{\partial c_{ik} \cdot \partial c_{j\ell}} = \begin{cases} \frac{-1}{c_{ik}} \; si \; k = \ell \; et \; i = j; \\ \\ \beta \; si \; k = \ell et \; v_{ij} = 1; \\ \\ 0 \; sinon. \end{cases}$$

La matrice hessienne est donc une matrice symétrique dont les éléments diagonaux valent $\frac{-1}{c_{ik}}$, et les autres tantôt 0, tantôt β. D'après le théorème de Gerschgörin-Hadamard, les valeurs propres de $H(\mathbf{c})$ appartiennent à l'union des $N \times K$ disques D_{ik}, du plan complexe, où D_{ik} est défini par :

$$|z - H_{ik;ik}| \leq \sum_{(j,\ell)=(1,1)}^{N\times K} |H_{ik;j\ell}|.$$

Chaque valeur propre, λ_{ik}, est donc située dans un disque de centre $(\frac{-1}{c_{ik}}, 0)$ et de rayon $\beta \cdot \sum_j^N v_{ij}$. Chaque centre est forcément en dessous de $(-1, 0)$. Si on prend $\beta < \frac{1}{V_{max}}$, le rayon de chaque disque est plus petit que l'unité et l'on est sur que toutes les valeurs propres sont alors négatives et que la matrice hessienne est définie strictement négative. Dans ce cas, la fonction $U_\Phi(\mathbf{c})$ est concave et admet un unique maximum, c^+.

Montrons que la suite $\{\boldsymbol{c}^m\}_{m=1,2\ldots}$ *générée par* $\boldsymbol{c}^{m+1} = F(\boldsymbol{c}^m)$ *converge vers* $\boldsymbol{c}^+$*:* Comme $\mathbf{c}^+$ est l'unique maximum maximum de $U_\Phi(\mathbf{c})$, les équations

$$\nabla_{\mathbf{c}} U_\Phi(\mathbf{c}) = 0,$$

sont satisfaites pour $\mathbf{c} = \mathbf{c}^+$, ce qui est équivalent à :

$$\mathbf{c}^+ = F(\mathbf{c}^+).$$

Les dérivées partielles de F existent et sont continues. D'après le théorème du point fixe, la suite converge si une norme $\|.\|$ de la matrice jacobienne de l'opérateur F, notée F', est strictement inférieure à 1. Notons $F'(\mathbf{c}^m)$ la matrice jacobienne de F en $\mathbf{c}^m$:

$$F'(\mathbf{c}^m) = \begin{pmatrix} \frac{\partial F_{11}}{\partial c_{11}^m} & \frac{\partial F_{11}}{\partial c_{12}^m} & \cdots & \frac{\partial F_{11}}{\partial c_{NK}^m} \\ \frac{\partial F_{12}}{\partial c_{11}^m} & \frac{\partial F_{12}}{\partial c_{12}^m} & \cdots & \frac{\partial F_{12}}{\partial c_{NK}^m} \\ \vdots & \vdots & \cdots & \vdots \\ \frac{\partial F_{NK}}{\partial c_{11}^m} & \frac{\partial F_{NK}}{\partial c_{12}^m} & \cdots & \frac{\partial F_{NK}}{\partial c_{NK}^m} \end{pmatrix}.$$

On a :

$$\frac{\partial F_{ik}}{\partial c_{j\ell}^m} = \begin{cases} 0 \; si \; k = \ell \; et \; i = j; \\ \beta \cdot c_{ik}^{m+1} - \beta \cdot (c_{ik}^{m+1})^2 \; si \; k = \ell \; et \; v_{ij} = 1; \\ -\beta \cdot c_{ik}^{m+1} \cdot c_{j\ell}^{m+1} \; si \; k \neq \ell \; et \; v_{ij} = 1; \\ 0 \; sinon. \end{cases}$$

Calculons la somme des valeurs absolues des éléments d'une ligne de la matrice $F'(\mathbf{c}^m)$:

$$\begin{aligned} \sum_{j\ell} |\frac{\partial F_{ik}}{\partial c_{j\ell}^m}| &= \sum_{j=1}^N v_{ij} \cdot |\beta \cdot c_{ik}^{m+1}(1 - c_{ik}^{m+1})| + \sum_{j=1}^N \sum_{\ell \neq k} v_{ij} \cdot |\beta \cdot c_{ik}^{m+1} \cdot c_{j\ell}^{m+1}|, \\ &< V_{max} \cdot \beta \cdot c_{ik}^{m+1} \{|1 - c_{ik}^{m+1}| + 1\} \\ &< V_{max} \cdot \beta \cdot c_{ik}^{m+1} \{2 - c_{ik}^{m+1}\} \end{aligned}$$

Si $0 < \beta < \frac{1}{Vmax}$, alors le polynôme $V_{max} \cdot \beta \cdot c_{ik}^{m+1}\{2 - c_{ik}^{m+1}\}$ est toujours strictement compris entre 0 et 1, quelle que soit la valeur de c_{ik}^{m+1}. On en déduit donc que

$$\|F'(\mathbf{c}^m)\|_\infty = \max_{ik} \sum_{j\ell} |\frac{\partial F_{ik}}{\partial c_{j\ell}^m}| < 1$$

et que la suite $\{\mathbf{c}^m\}_{m=1,2\ldots}$ converge vers $\mathbf{c}^+$ si $0 < \beta < \frac{1}{Vmax}$.

Théorème 3.3 *Soit* $\boldsymbol{c}$ *une matrice de classification. Il existe un unique vecteur* Φ^+, *tel que :*

$$\Phi^+ = arg \max_{\Phi} U(\Phi, \boldsymbol{c}).$$

Démonstration : soit $U_{\mathbf{c}}(\Phi)$, la fonction $U(\mathbf{c}, \Phi)$ pour $\mathbf{c}$ fixé. La matrice hessienne $H(\Phi)$ est définie négative pour tout vecteur Φ appartenant à l'ensemble de tous les vecteurs paramètres possibles. Il existe donc un unique maximum Φ^+ de $U_{\mathbf{c}}(\Phi)$, et les équations

$$\nabla_\Phi U_{\mathbf{c}}(\Phi) = 0,$$

sont satisfaites pour $\Phi = \Phi^+$. Ces équations permettent de déterminer Φ^+ directement. On en déduit ainsi directement qu'il existe un unique Φ^+, tel que :

$$\Phi^+ = arg \max_{\Phi} U(\Phi, \mathbf{c}).$$

Notons que Φ^+ peut parfois être une solution singulière (au moins une matrice $\mathbf{\Sigma}_k$ définie négative) (Duda et Hart 1973).

Théorème 3.4 *Soient pour* $\{\boldsymbol{c}^0, \Phi^0 = arg \max_\Phi U(\Phi, \boldsymbol{c}^0)\}$, *et la suite* $\{\boldsymbol{c}^q, \Phi^q\}_{q=0,1,2\ldots}$ *générée par l'algorithme NEM. Le critère* $U(\Phi, \boldsymbol{c})$ *croît de façon monotone sur* $\{\boldsymbol{c}^q, \Phi^q\}_{q=0,1,2\ldots}$ *et tend vers une limite* U^* *si* U *est bornée.*

Démonstration : d'après le théorème 3.2, on a :

$$U(\Phi^q, \mathbf{c}^q) \leq U(\Phi^q, \mathbf{c}^{q+1}).$$

Le théorème 3.3, nous donne :

$$U(\Phi^q, \mathbf{c}^{q+1}) \leq U(\Phi^{q+1}, \mathbf{c}^{q+1}).$$

Il est alors évident que :

$$U(\Phi^q, \mathbf{c}^q) \leq U(\Phi^{q+1}, \mathbf{c}^{q+1}).$$

Les valeurs successives de U forment une suite monotone croissante, qui converge vers une limite U^* si le critère est borné.

3.4.2 Recherche d'une partition dure

Plusieurs solutions d'adaptations sont envisageables pour obtenir une matrice de classification dure en utilisant l'algorithme NEM :

- A la convergence la matrice de classification floue obtenue peut être modifiée en matrice de classification dure en affectant chaque individu à la classe la plus probable a posteriori (principe du MAP), en se basant sur les c_{ik} obtenus.
- Comme dans l'algorithme CEM (Celeux et Govaert 1992), on peut ajouter une étape intermédiaire déterministe de classification entre les étape E et M. Chaque individu est affecté suivant le principe du MAP à une classe et la matrice de classification caractéristique de cette nouvelle partition sera alors utilisée pour l'étape M.
- Comme dans l'algorithme SEM , on peut introduire une étape stochastique intermédiaire de classification entre les étape E et M (Celeux et Diebolt 1986).

Ces alternatives posent de nouveaux problèmes d'optimalité des paramètres trouvés. Détaillons la "version CEM" de l'algorithme NEM. Nous noterons cette version NCEM. Dans le cas d'une partition dure le critère considéré sera :

$$U_d(\mathbf{c}, \Phi) = \sum_{i=1}^{N} \sum_{k=1}^{K} c_{ik} \log\{p_k f_k(\mathbf{x}_i|\theta_k)\} + \beta \cdot G(\mathbf{c}) \tag{3.24}$$

L'algorithme CEM devient alors :

1. **Initialisation :** idem NEM
2. **Itérations :** L'algorithme s'arrête si les matrice de classification reste inchangées deux itérations de suite. Chaque itération comprend deux étapes :
   - **Estimation** d'une nouvelle matrice de classification floue, $\mathbf{c}^{q+\frac{1}{2}}$:

$$c_{ik}^{q+\frac{1}{2}} = \frac{p_k f_k(\mathbf{x}_i|\theta_k) \cdot \exp\{\beta \sum_{j=1}^{N} c_{jk}^q v_{ij}\}}{\sum_{\ell=1}^{K} p_\ell f_\ell(\mathbf{x}_i|\theta_\ell) \cdot \exp\{\beta \sum_{j=1}^{N} c_{j\ell}^q v_{ij}\}}. \tag{3.25}$$

   - **Classification** Transformation de la matrice de classification floue $\mathbf{c}^{q+\frac{1}{2}}$ en matrice de classification dure $\mathbf{c}^{q+1}$:

$$c_{ik}{}^{q+1} = \begin{cases} 1, \ si \ k = arg\max_\ell(c_{i\ell}{}^{q+\frac{1}{2}}); \\ \\ 0, \ sinon. \end{cases}$$

   - **Maximisation** du critère U_d par rapport au vecteur de paramètres Φ :

$$\Phi^{q+1} = arg\max_{\Phi} U_d(\mathbf{c}^{q+1}, \Phi) \tag{3.26}$$

Comme dans le cas de NEM, il est possible de montrer que sous certaines conditions chaque itération fait croître le critère optimisé. Ceci sera démontré dans la section suivante.

3.4.3 Interprétation bayésienne

Partition dure

Dans le cas où l'on cherche une partition dure le critère U_d donne lieu à une interprétation bayésienne et NCEM peut être alors rapproché de certaines méthodes de segmentation statistique d'image qui utilisent le paradigme bayésien.

En effet, nous constatons que maximiser le critère $U_d(\mathbf{c}, \Phi)$ revient à maximiser

$$\begin{aligned} \exp\{U_d(\mathbf{c},\Phi)\} &= \exp\{\sum_{i=1}^{N}\sum_{k=1}^{K} c_{ik}\log\{p_k f_k(\mathbf{x}_i|\theta_k)\}\}\cdot\exp\{\beta\cdot G(\mathbf{c})\}, \\ &= \prod_{i=1}^{N} p_{c_i} f_{c_i}(\mathbf{x}_i|\theta_{c_i})\cdot\exp\{\beta\cdot G(\mathbf{c})\}. \end{aligned}$$

Si les proportions $p_1, \cdots, p_K$ et β sont connus alors :

$$\exp\{U_d(\mathbf{c},\Phi)\} \propto \prod_{i=1}^{N} f_{c_i}(\mathbf{x}_i|\theta_{c_i})\cdot\frac{\exp\{\beta\cdot G(\mathbf{c})\}}{Z(\beta)}, \tag{3.27}$$

où $Z(\beta) = \sum_{\mathbf{c}'\in\mathcal{C}_{dure}}\exp\{\beta\cdot G(\mathbf{c}')\}$ est une fonction qui dépend uniquement de β.

Remarquons alors que les deux fonctions,

$$P_\Phi(\mathbf{x}|\mathbf{c}) = \prod_{i=1}^{N} f_{c_i}(\mathbf{x}_i|\theta_{c_i}),$$

et

$$P(\mathbf{c}) = \frac{\exp\{\beta\cdot G(\mathbf{c})\}}{Z(\beta)},$$

sont des distributions de probabilité.

La première, $P_\Phi(\mathbf{x}|\mathbf{c})$, peut être interprétée comme la distribution des observations conditionnellement à la connaissance de la partition. Dans ce cas les observations sont supposées indépendantes conditionnellement à la connaissance de la partition. Ceci revient à supposer que le bruit est gaussien et spatialement non corrélé. Cette hypothèse est courante en segmentation statistique, bien qu'elle ne corresponde pas toujours à la réalité physique. La forme de cette distribution implique aussi que

$$P(\mathbf{x}_i|\mathbf{c}) = P(\mathbf{x}_i|\mathbf{c}_i).$$

La seconde distribution,

$$\begin{aligned} P(\mathbf{C}=\mathbf{c}) &= \frac{1}{Z} exp\{\beta G(\mathbf{c})\} \\ &= \frac{1}{Z} exp\{\beta(\#\text{paires de sites voisins de même classe})\} \end{aligned}$$

est une distribution de Gibbs qui correspond au modèle de Strauss (1977). Cette distribution définit un champ de Markov, et peut être interprétée comme une distribution *a priori* sur la partition.

Ainsi le problème de la maximisation de U_d est équivalent à la recherche d'un estimateur du MAP car

$$\begin{aligned} \exp U_d(\mathbf{c},\Phi) &\propto P(\mathbf{c}) \cdot P_\Phi(\mathbf{x}|\mathbf{c}), \\ &\propto P_\Phi(\mathbf{c}|\mathbf{x}), \end{aligned}$$

et on peut écrire, que NCEM cherche :

$$(\mathbf{c}^*,\Phi^*) = arg \max_{(\mathbf{c},\Phi)} P_\Phi(\mathbf{c}|\mathbf{x}).$$

Cette interprétation bayésienne postule l'existence de deux champs aléatoires: $\mathbf{X} = \{\mathbf{X}_s, s \in S\}$, qui est le champ aléatoire dont on a observé une réalisation et $\mathbf{C} = \{\mathbf{C}_s, s \in S\}$, qui est le champ aléatoire correspondant à la matrice de partition (S est l'ensemble des sites). Les $\mathbf{X}_s$ sont des variables aléatoires qui prennent leurs valeurs dans $I\!R^d$ et les $\mathbf{C}_s = (\mathbf{C}_{s1}, \cdots, \mathbf{C}_{sK})$ sont des variables qui prennent leurs valeurs dans un sous ensemble $\{0,1\}^K$ défini par les contraintes $\sum_{k=1}^K C_{sk} = 1$. Les champs $\mathbf{X}$ et $\mathbf{C}$ sont distribués suivant une distribution de Gibbs et la relation entre les deux champs est de la forme :

$$\mathbf{x}_i = \sum_{k=1}^{K} c_{ik}(\boldsymbol{\mu}_k + \mathbf{n}_k), \tag{3.28}$$

où $\boldsymbol{\mu}_k$ est le vecteur moyenne de la classe k et $\mathbf{n}_k$ est une variable aléatoire indépendante des $\mathbf{x}_i$, qui suit une loi multinormale de moyenne nulle et de matrice de variance covariance $\boldsymbol{\Sigma}_k$.

Analysons maintenant l'algorithme NCEM à l'aide du modèle mis en évidence :

- L'étape E calcule :

$$c_{ik}{}^{q+\frac{1}{2}} \propto f_k(\mathbf{x}_i|\theta_k) \cdot P(c_{ik} = 1|\mathbf{c}_j{}^q,\ v_{ij} = 1) \tag{3.29}$$

  où $P(c_{ik} = 1|\mathbf{c}_j{}^q,\ v_{ij} = 1)$ représente la probabilité que le site i soit de classe k conditionnellement à la connaissance de la classe des voisins du site i. On peut montrer que, pour un site i donné :

$$P(c_{ik} = 1|\mathbf{x}; \mathbf{c}_j{}^q,\ j \neq i) \propto f_k(\mathbf{x}_i|\theta_k) \cdot P(c_{ik} = 1|\mathbf{c}_j{}^q,\ v_{ij} = 1) \tag{3.30}$$

- L'étape C revient à affecter chaque individu $\mathbf{x}_i$ à la classe la plus probable a posteriori :

$$c_{ik}{}^{q+1} = \begin{cases} 1\ si\ k = arg\max_\ell P(c_{i\ell} = 1|\mathbf{x}; \mathbf{c}_j{}^q, j \neq i); \\ \\ 0\ sinon. \end{cases}$$

- L'étape M consiste à trouver :

$$\Phi^{q+1} = arg \max_{\Phi} P_\Phi(\mathbf{c}^{q+1}|\mathbf{x}). \tag{3.31}$$

Cette présentation met en évidence le fait que les itérations E et C correspondent à une itération de l'algorithme ICM (Besag 1986), où la nouvelle image segmentée est estimée en tous les pixels simultanément.

Dans son article de 1986, Besag propose plusieurs versions de ICM. Une itération de ICM peut ainsi :

- classer les pixels dans toute l'image de manière simultanée (comme ceci est décrit pour l'étape EC) ;
- classer les pixels bloc par bloc ;
- classer les pixels un par un.

Remarquons que dans les deux derniers cas, ICM est une sorte d'échantillonneur de Gibbs déterministe.

Théorème 3.5 *Soient* $\{\boldsymbol{c}^0, \Phi^0 = arg\max_\Phi U_d(\Phi, \boldsymbol{c}^0)\}$, *et la suite* $\{\boldsymbol{c}^q, \Phi^q\}_{q=0,1,2\ldots}$ *générée par l'algorithme NCEM, où les étapes EC se font pixel par pixel. Le critère* $U(\Phi, \boldsymbol{c})$ *croît de façon monotone sur* $\{\boldsymbol{c}^q, \Phi^q\}_{q=0,1,2\ldots}$ *et tend vers une limite* U_d^* *si* U_d *est bornée. De plus, si les estimateurs* Φ^q *sont toujours définis et le critère* U_d *borné, la suite* $\{\boldsymbol{c}^q, \Phi^q\}_{q=0,1,2\ldots}$ *converge vers une valeur* $\{\boldsymbol{c}^*, \Phi^*\}$.

Démonstration : *Montrons que* $U_d(\boldsymbol{c}^q, \Phi^q) \leq U_d(\boldsymbol{c}^{q+1}, \Phi^q)$ *:* Lors de l'étape EC, les pixels sont examinés selon une certaine séquence. Lorsque le pixel i est examiné, tout le reste de l'image segmentée est connue. Remarquons que dans le reste de l'image segmentée, la classe de certains pixels a été estimée à l'étape q alors que la classe des autres pixels provient des calculs de l'étape $q+1$. Nous notons $\mathbf{c}^{q+1;m}$ cette image segmentée intermédiaire avant le classement du pixel i, et $\mathbf{c}^{q+1;m+1}$ après le classement du pixel i. Par construction, nous avons :

$$P_{\Phi^q}(\mathbf{c}_i{}^{q+1;m+1}|\mathbf{x}; \mathbf{c}_j{}^{q+1;m}, j \neq i) = arg\max_{\mathbf{c}_i} P_{\Phi^q}(\mathbf{c}_i|\mathbf{x}; \mathbf{c}_j{}^{q+1;m}, j \neq i), \tag{3.32}$$

ce qui implique que :

$$\begin{aligned} P_{\Phi^q}(\mathbf{c}^{q+1;m+1}|\mathbf{x}) &= P_{\Phi^q}(\mathbf{c}_i{}^{q+1;m+1}|\mathbf{x}; \mathbf{c}_j{}^{q+1;m}, j \neq i) \cdot P(\mathbf{c}_j{}^{q+1;m}, j \neq i|\mathbf{x}); \\ &\geq P_{\Phi^q}(\mathbf{c}_i{}^{q+1;m}|\mathbf{x}; \mathbf{c}_j{}^{q+1;m}, j \neq i) \cdot P(\mathbf{c}_j{}^{q+1;m}, j \neq i|\mathbf{x}); \\ &= P_{\Phi^q}(\mathbf{c}^{q+1;m}|\mathbf{x}). \end{aligned}$$

Il est alors évident que

$$\begin{aligned} P_{\Phi^q}(\mathbf{c}^{q+1;N}|\mathbf{x}) &\geq P_{\Phi^q}(\mathbf{c}^{q+1;0}|\mathbf{x}) \\ P_{\Phi^q}(\mathbf{c}^{q+1}|\mathbf{x}) &\geq P_{\Phi^q}(\mathbf{c}^q|\mathbf{x}) \\ U_d(\mathbf{c}^{q+1}, \Phi^q) &\geq U_d(\mathbf{c}^q, \Phi^q) \end{aligned}$$

Par construction, on a $U_d(\boldsymbol{c}^{q+1}, \Phi^q) \leq U_d(\boldsymbol{c}^{q+1}, \Phi^{q+1})$. Le critère $U_d(\Phi, \mathbf{c})$ croît donc de façon monotone sur $\{\mathbf{c}^q, \Phi^q\}_{q=0,1,2\ldots}$.

Comme il existe un nombre fini d'images segmentées de N pixels en K classes, et que les paramètres Φ^q sont déterminés de manière unique à partir de $\mathbf{c}^q$, la suite $U_d(\mathbf{c}^q, \Phi^q)_{q=0,1,2\ldots}$ prend un nombre fini de valeurs et converge vers un valeur stationnaire. Ainsi, pour q assez grand,

$$U_d(\mathbf{c}^q, \Phi^q) = U_d(\mathbf{c}^{q+1}, \Phi^q) = U_d(\mathbf{c}^{q+1}, \Phi^{q+1}).$$

De la première égalité, et si les estimateurs du maximum de vraisemblance sont bien définis, on déduit que $\Phi^{q+1} = \Phi^q$. De la deuxième égalité et par construction, il s'ensuit que $\mathbf{c}^{q+1} = \mathbf{c}^q$.

Notons que l'algorithme NCEM était déjà proposé par Besag en 1986 pour faire de la segmentation non supervisée.

Partition floue

Dans le cas où l'on recherche une partition floue, l'interprétation bayésienne du critère U semble plus délicate. En effet, nous constatons que maximiser le critère $U(\mathbf{c}, \Phi)$ revient à maximiser

$$\exp\{U(\mathbf{c}, \Phi)\} = \exp\{D(\mathbf{c}, \Phi)\} \cdot \exp\{\beta \cdot G(\mathbf{c})\}. \tag{3.33}$$

On peut toujours écrire que :

$$\exp\{\beta \cdot G(\mathbf{c})\} \propto P(\mathbf{c}) \tag{3.34}$$

où $P(\mathbf{c})$ est une distribution de Gibbs de fonction d'énergie $-\beta \cdot G(\mathbf{c})$ qui définit un champ de Markov. Pour se ramener à une interprétation bayésienne semblable à celle du cas "recherche de partition dure", il faudrait démontrer que

$$\begin{aligned}\exp\{D(\mathbf{c}, \Phi)\} &= \frac{1}{\exp\{\sum_{i=1}^{N}\sum_{k=1}^{K} c_{ik} \log c_{ik}\}} \cdot \prod_{i=1}^{N}\prod_{k=1}^{K} (\exp\log p_k f_k(\mathbf{x}_i|\theta_k))^{c_{ik}} \\ &= \prod_{i=1}^{N}\prod_{k=1}^{K} \frac{(p_k f_k(\mathbf{x}_i|\theta_k))^{c_{ik}}}{\exp\{c_{ik}\log c_{ik}\}},\end{aligned}$$

est proportionnel à une distribution de probabilité $P_\Phi(\mathbf{x}|\mathbf{c})$. C'est à dire que

$$\exp\{D(\mathbf{c}, \Phi)\} = \lambda \cdot P_\Phi(\mathbf{x}|\mathbf{c}) \tag{3.35}$$

avec λ une fonction à valeurs dans $\mathbb{R}$ qui ne dépend pas de Φ. En effet si $\lambda = g(\Phi)$, il est peu vraisemblable que maximiser $\exp\{D(\mathbf{c}, \Phi)\}$ par rapport à Φ soit équivalent à maximiser $\frac{\exp\{D(\mathbf{c},\Phi)\}}{g(\Phi)}$.

Dans le cas où l'interprétation bayésienne se révélerait valide pour le modèle flou, l'approche NEM proposerait ainsi une alternative aux méthodes de segmentation floue qui sont encore rares en segmentation statistique non supervisée (Kent et Mardia 1991, Caillol *et al.* 1993).

3.4.4 Estimation du facteur de pénalisation

Dans les sections précédentes, nous avons supposé le facteur de pénalisation β connu. La nature des résultats produits par les algorithmes NEM et NCEM dépend fortement de ce paramètre. Plus la valeur de β est élevée plus la segmentation résultante sera spatialement homogène (Figure 3.6).

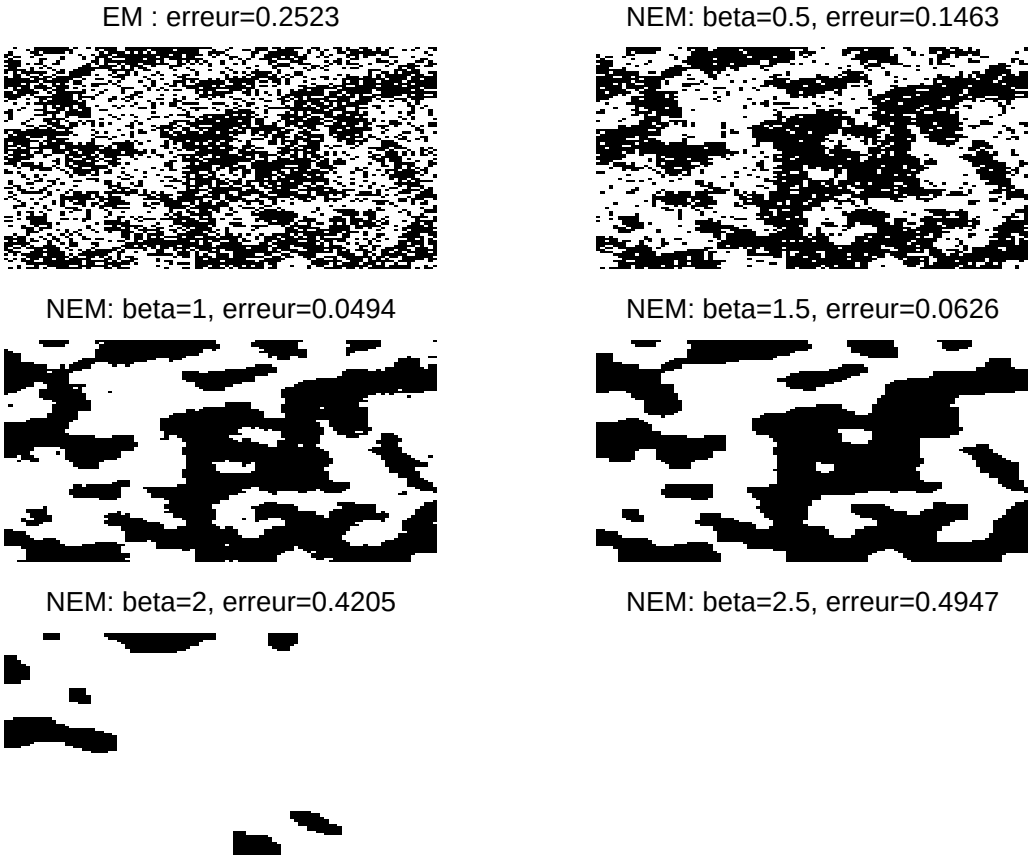


FIG. 3.6 - : Segmentation d'image par NEM avec différentes valeurs de β

Rappelons aussi que le facteur de pénalisation joue un rôle dans la convergence de l'algorithme NEM. Si β est trop grand la convergence de l'algorithme NEM n'est plus garantie. En pratique on observe que plus la valeur de β est importante plus la convergence est lente et lorsque β est trop important le critère ne se stabilise pas.

Les considérations précédentes montrent que le choix de β est d'importance. Mais comment choisir ce paramètre? Plusieurs approches sont possibles :

- Le paramètre peut être connu, la valeur du paramètre venant d'une expérience passée, d'une connaissance a priori ou bien d'une connaissance du processus physique qui a produit les données soumises à l'analyse.

- Le paramètre peut être estimé par essais successifs : La segmentation des données est réalisée pour de nombreuses valeurs du paramètre et l'on choisit celui qui donne le meilleur résultat selon l'avis d'un expert.
- Le paramètre peut être estimé par une procédure statistique au cours des itérations de l'algorithme tout comme les paramètres liés au bruit. Cette dernière approche possède l'inconvénient de rendre les démonstrations de convergence obsolètes.

Traduction de connaissances a priori

Dans certains cas, quelques arguments subjectifs permettent d'estimer des paramètres du modèle. En effet le champ de Markov du modèle de Strauss est défini par des probabilités conditionnelles. Ces probabilités peuvent faire l'objet d'une estimation subjective. Chaque jugement subjectif produit une valeur de β. Il reste ensuite à combiner toutes ces valeurs, en prenant par exemple, leur moyenne.

Exemple 3.12 Considérons le modèle de Strauss pour une image binaire (il y a alors équivalence avec le modèle d'Ising). $\mathbf{c}$ est une image binaire et la probabilité qu'un pixel s soit de la classe A ($\mathbf{c}_s = [1\ 0]$) ou de la classe B ($\mathbf{c}_s = [0\ 1]$) connaissant le reste de l'image est donné par:

$$P(c_{sk} = 1 | \mathbf{c}_r, r \neq s) = \frac{\exp\left(\beta \sum_{r=1}^{N} v_{rs} c_{rk}\right)}{\sum_{\ell=1}^{2} \exp\left(\beta \sum_{r=1}^{N} v_{rs} c_{r\ell}\right)} \tag{3.36}$$

Dans le cas où quatre voisins sont pris en compte, cette expression peut prendre 5 valeurs différentes, suivant que le pixel s appartienne à la classe A ou à la classe B et la nature du voisinage de ce pixel :

	Pixel $s \in A$	Pixel $s \in B$
0 voisin de classe A	$\frac{1}{1+\exp(4\cdot\beta)}$	$\frac{\exp(4\cdot\beta)}{1+\exp(4\cdot\beta)}$
1 voisin de classe A	$\frac{\exp(\beta)}{\exp(\beta)+\exp(3\cdot\beta)}$	$\frac{\exp(3\cdot\beta)}{\exp(\beta)+\exp(3\cdot\beta)}$
2 voisins de classe A	$\frac{\exp(2\cdot\beta)}{2\cdot\exp(2\cdot\beta)}$	$\frac{\exp(2\cdot\beta)}{2\cdot\exp(2\cdot\beta)}$
3 voisins de classe A	$\frac{\exp(3\cdot\beta)}{\exp(\beta)+\exp(3\cdot\beta)}$	$\frac{\exp(\beta)}{\exp(\beta)+\exp(3\cdot\beta)}$
4 voisins de classe A	$\frac{\exp(4\cdot\beta)}{1+\exp(4\cdot\beta)}$	$\frac{1}{1+\exp(4\cdot\beta)}$

TAB. 3.1 - : Probabilités conditionnelles pour un modèle de Strauss binaire

Si l'on observe que le pixel s est entouré de quatre pixel de classe A, notre opinion est que s a une forte probabilité d'être de A:

$$\begin{array}{ccc} & A & \\ A & s & A \\ & A & \end{array}$$

En quantifiant ce jugement, on pose,

$$P(\mathbf{c}_s = [1\ 0] | \mathbf{c}_r = [1\ 0], r \neq s) = \frac{\exp(4 \cdot \beta)}{1 + \exp(4 \cdot \beta)} = 0.9 \tag{3.37}$$

on trouve $\beta = 0.55$.

Ce calcul peut être réalisé pour toutes les configuration possibles. Chaque probabilité subjective nous donne une valeur de β. La moyenne de toutes ces valeurs peut par exemple servir d'estimation de β.

△

Pseudo vraisemblance et estimation de β

Lors de l'étape M de NCEM, l'on dispose d'une image segmentée, que nous notons $\mathbf{c}$. Cette image peut servir de base à l'estimation du paramètre de pénalisation. Une solution possible consiste à considérer la pseudo vraisemblance de β :

$$\mathcal{P}_\beta(\mathbf{c}) = \prod_{i=1}^{N} P(c_i | V(c_i)).$$

Pour trouver la valeur de β qui maximise ce critère, il existe de nombreux algorithmes d'optimisation numérique : descente de gradient ...

Dans Chalmond (1989), une solution originale est proposée : comme nous l'avons déjà remarqué dans la section précédente, la probabilité conditionnelle $P(c_i|V(c_i))$ prend un nombre fini de valeurs suivant le nombre de classes considérées par le modèle. Si l'on note $P(k|\ell)$ la probabilité d'avoir un pixel de classe k entouré par un voisinage de type ℓ, la pseudo vraisemblance peut s'écrire sous la forme :

$$\mathcal{P}_\beta(\mathbf{c}) = \prod_{k,\ell} P(k|\ell)^{n_{k\ell}(\mathbf{c})},$$

avec $n_{k\ell}(\mathbf{c})$ le nombre d'occurrences de la configuration, k entouré de ℓ, dans l'image $\mathbf{c}$. Au lieu de considérer le paramètre β comme inconnu, on peut s'attacher à trouver les valeurs des $P(k|\ell)$ qui maximisent la pseudo vraisemblance.

Maximiser le critère $\mathcal{P}_\beta(\mathbf{c})$ par rapport aux $P(k|\ell)$ sous les contraintes

$$\sum_k P(k|\ell) = 1, \ \forall \ell$$

nous donne les estimateurs suivants :

$$\hat{P}(k|\ell) = \frac{n_{k\ell}(\mathbf{c})}{\sum_{j=1}^{K} n_{j\ell}(\mathbf{c})}.$$

Ces estimateurs permettent de poser un système d'équations, où chaque équation est de la forme :

$$\log\{\frac{P(k_i|\ell)}{P(k_1|\ell)}\} = a_i \cdot \beta.$$

où a_i est un scalaire. Ce système d'équations est bien sur surdéterminé et l'on peut trouver une estimation de β par les moindres carrés. Un problème qui peut se produire, est qu'une configuration $P(k|\ell)$ ne soit pas représentée dans l'image $\mathbf{c}$. Dans ce cas le système n'admet pas de solutions.

3.5 Simulations numériques

Pour évaluer les performances des algorithmes NEM et NCEM, nous les comparons dans cette section à deux autres algorithmes de segmentation statistique :

- EM, qui est un algorithme de segmentation locale et ne considère pas le coté spatial des données. Cette comparaison vise à montrer l'intérêt de prendre en compte les relations de voisinage inter-individus dans une classification de données spatiales.

- Gibbsian EM (Chalmond 1989), qui est un algorithme de segmentation globale. Cette comparaison permet de situer NEM et NCEM parmi les algorithmes de segmentation de l'approche markovienne.

3.5.1 Description des tests

Pour comparer les 4 algorithmes sur une base commune, des hypothèses concernant la loi des observations et les distributions (pour GEM, NEM et NCEM) ont été adoptées :

- La loi des observations (ou encore le modèle du bruit) est un mélange de 2 gaussiennes de même matrice de variance covariance, $\mathbf{\Sigma} = \sigma^2 \cdot I$. Les proportions du mélange sont égales à $\frac{1}{2}$. Dans le cas de NCEM et GEM, cela revient à supposer que le bruit est gaussien et indépendant conditionnellement à la connaissance de la partition.

- La distribution *a priori* est une distribution de Strauss de paramètre β fixé, qui définit un champ de Markov isotrope. Notons que dans l'article original de Chalmond, GEM est présenté avec une procédure d'estimation automatique des paramètres de la distribution *a priori*, alors que dans les comparaisons de cette section, la valeur de β dans GEM est imposée.

Pour faciliter l'interprétation des résultats, le jeu de données a été choisi le plus simple possible : il s'agit d'une image binaire synthétique de 100 par 100 pixels (Figure 3.7), générée par un échantillonneur de Gibbs, qui approche une réalisation d'une distribution de Strauss isotrope de paramètre $\beta = 2$. Les pixels de la classe 1 ont un niveau de gris μ_1 et ceux de la classe 2 un niveau μ_2. A partir de cette image

FIG. 3.7 - : Image binaire réalisation d'une distribution de Strauss de paramètre $\beta = 2$

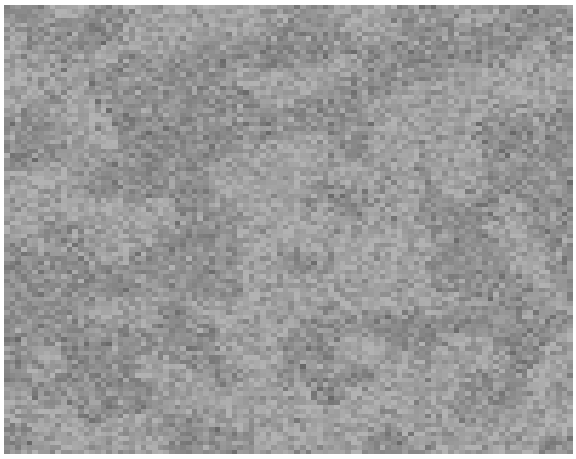


FIG. 3.8 - : Image binaire dégradée par un bruit additif gaussien spatialement indépendant de moyenne nul et d'écart type $\sigma = 0.75$

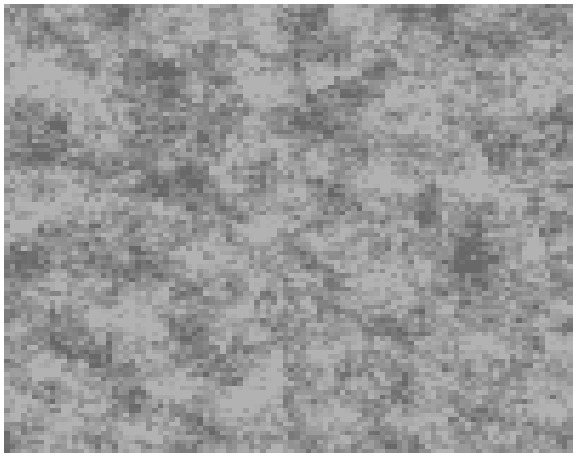


FIG. 3.9 - : Image binaire dégradée par un bruit additif gaussien corrélé de moyenne nul et d'écart type $\sigma = 0.75$

originale, six images bruitées ont été produites. Deux types de bruit ont été utilisés pour dégrader l'image :

- un bruit spatialement indépendant de moyenne nulle et de variance σ^2. Ce type de bruitage correspond aux hypothèses de travail des quatre algorithmes (Figure 3.8).
- un bruit spatialement corrélé (Figure 3.9), réalisation du processus gaussien CAR (voir section 3.2.2) de moyenne nulle et de matrice de variance covariance

$$\mathbf{\Sigma} = \sigma^2 \cdot (I - 0.25 \cdot \mathbf{V})$$

avec $\mathbf{V}$ la matrice de voisinage de l'image (voir section 3.4.1).

Pour chaque type de bruit, l'image a été bruitée avec $\sigma = 0.5,\ 0.75,\ 1$. Six images dégradées constituent l'ensemble des jeux de données tests.

Désirant tester l'influence du paramètre β sur la qualité de la reconstitution des images bruitées, 3 valeurs différentes du paramètre de pénalisation spatiale ont été essayées : la valeur utilisée pour simuler l'image binaire ($\beta = 2$), une valeur inférieure ($\beta = 1$) et une valeur supérieure $\beta = 3$. Le taux de pixel bien classés sert de critère pour juger de la qualité de la segmentation réalisée par les différents algorithmes.

3.5.2 Interprétation des résultats

En considérant les quatre tableaux de résultats (Tableaux 3.2 ,3.3 ,3.4, 3.5), et les figures 3.10, 3.11, on peut faire quelques observations générales :

- Les images dégradées par un bruit non corrélé sont reconstituées avec une plus grande précision par tous les algorithmes, que les images dégradées avec un

TAB. 3.2 - : Taux de pixels mal classés, par 4 algorithmes de segmentation utilisés pour segmenter une image synthétique dégradée par du bruit de variance σ. Les algorithmes NCEM, NEM et GEM utilisent une valeur de β fixée à 0.5

		$\sigma = 0.5$	$\sigma = 0.75$	$\sigma = 1$
Bruit non corrélé	EM	0.1605	0.2523	0.3065
	NCEM	0.0857	0.1786	0.2405
	NEM	0.0766	0.1462	0.1999
	GEM	0.0836	0.1578	0.2158
Bruit corrélé	EM	0.2484	0.3058	0.3502
	NCEM	0.2199	0.2849	0.3287
	NEM	0.2097	0.2824	0.3275
	GEM	0.2168	0.2899	0.3314

TAB. 3.3 - : Taux de pixels mal classés, par 4 algorithmes de segmentation utilisés pour segmenter une image synthétique dégradée par du bruit de variance σ. Les algorithmes NCEM, NEM et GEM utilisent une valeur de β fixée à 1

		$\sigma = 0.5$	$\sigma = 0.75$	$\sigma = 1$
Bruit non corrélé	EM	0.1605	0.2523	0.3065
	NCEM	0.033	0.0799	0.4829
	NEM	0.028	0.0507	0.0712
	GEM	0.0353	0.0764	0.1016
Bruit corrélé	EM	0.2484	0.3059	0.3502
	NCEM	0.1946	0.2824	0.3271
	NEM	0.1754	0.2683	0.3204
	GEM	0.1913	0.2722	0.3209

TAB. 3.4 - : Taux de pixels mal classés, par 4 algorithmes de segmentation utilisés pour segmenter une image synthétique dégradée par du bruit de variance σ. Les algorithmes NCEM, NEM et GEM utilisent une valeur de β fixée à 2

		$\sigma = 0.5$	$\sigma = 0.75$	$\sigma = 1$
Bruit non corrélé	EM	0.1605	0.2523	0.3065
	NCEM	0.0269	0.4967	0.4945
	NEM	0.0237	0.4947	0.4947
	GEM	0.0285	0.1118	0.2993
Bruit corrélé	EM	0.2486	0.3059	0.3502
	NCEM	0.4504	0.3277	0.4688
	NEM	0.2445	0.3704	0.4947
	GEM	0.1803	0.2618	0.3203

TAB. 3.5 - : Taux de pixels mal classés, par 4 algorithmes de segmentation utilisés pour segmenter une image synthétique dégradée par du bruit de variance σ. Les algorithmes NCEM, NEM et GEM utilisent une valeur de β fixée à 3

		$\sigma = 0.5$	$\sigma = 0.75$	$\sigma = 1$
Bruit non corrélé	EM	0.1605	0.2523	0.3066
	NCEM	0.487	0.4947	0.4949
	NEM	0.4947	0.4947	0.4947
	GEM	0.0441	0.4782	0.4595
Bruit corrélé	EM	0.2486	0.3058	0.3502
	NCEM	0.467	0.4943	0.4832
	NEM	0.4145	0.4527	0.4947
	GEM	0.2157	0.2842	0.3472

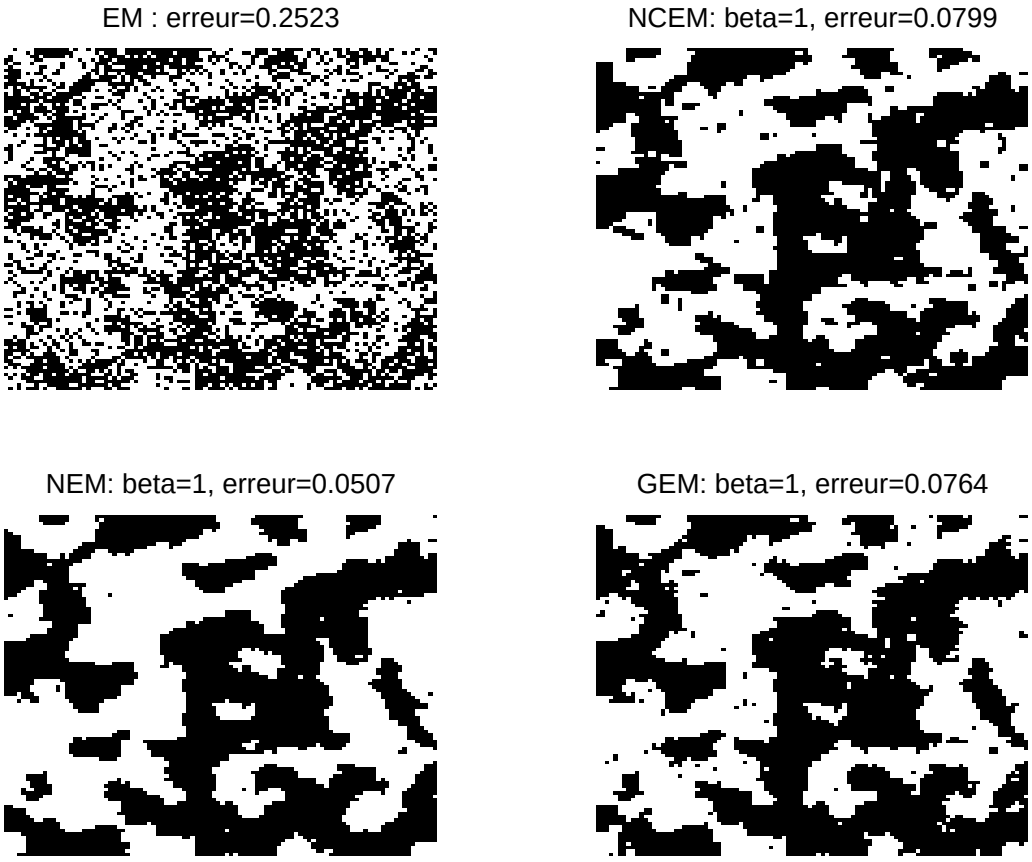


FIG. 3.10 - : Segmentation de l'image dégradée par du bruit indépendant en prenant $\beta = 1$

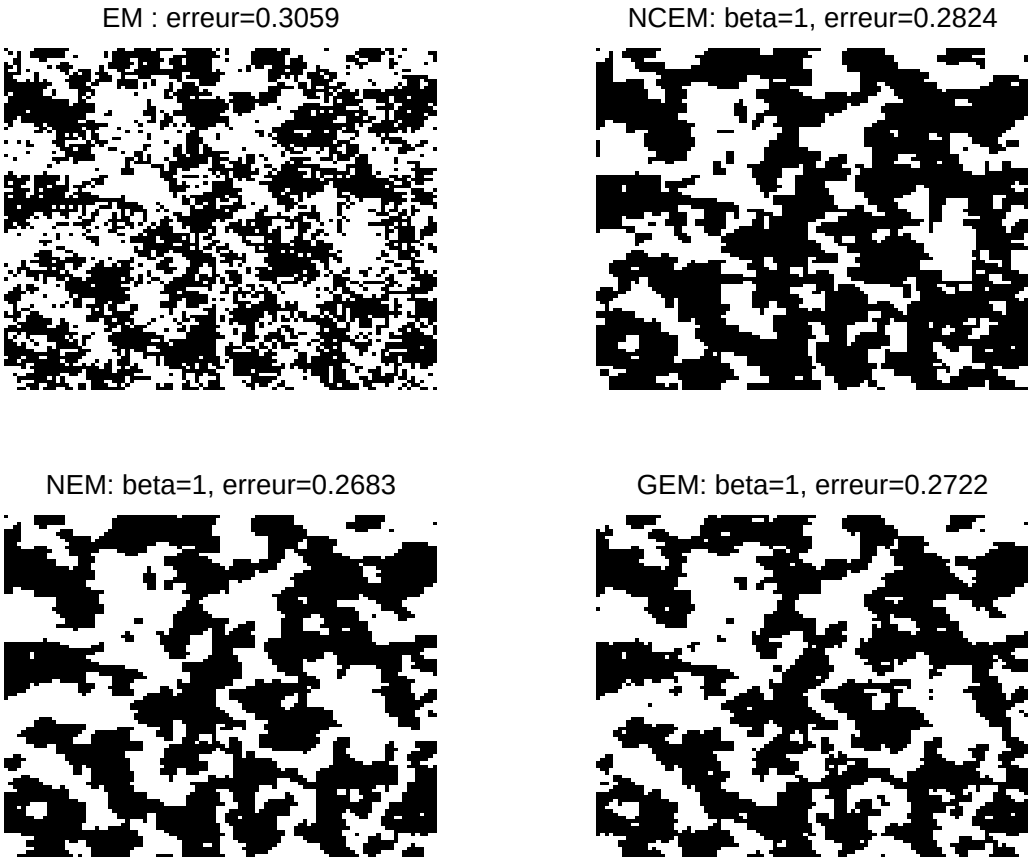


FIG. 3.11 - : Segmentation de l'image dégradée par du bruit corrélé en prenant $\beta = 1$

bruit spatialement corrélé. Cette constatation semble logique, car les quatre algorithmes font l'hypothèse d'indépendance du bruit.

- L'algorithme GEM est moins sensible que les autres à la valeur du paramètre β.

- Sur les six images tests, les meilleurs taux de reconstitution sont tous obtenus par l'algorithme NEM avec une valeur de $\beta = 1$.

Si l'on considère les résultats obtenus pour $\beta = 0.5$ et $\beta = 1$, on peut noter que :

- l'ordre de classement des performances des algorithmes est toujours :

$$EM < NCEM < GEM < NEM.$$

  Le fait que l'algorithme EM donne toujours les résultats les plus mauvais semble justifié, car c'est le seul qui ne prend pas en compte l'aspect spatial des données.

  Les algorithmes GEM et NCEM utilisent exactement, dans ces simulations, le même modèle statistique. Si GEM donne toujours de meilleurs résultats que NCEM, l'explication n'est donc pas à chercher dans le modèle sous-jacent. Remarquons que GEM est un algorithme stochastique qui cherche l'estimateur du MPM alors que NCEM est un algorithme déterministe qui cherche l'estimateur du MAP. L'aspect stochastique de GEM, rend cet algorithme moins sensible au problème d'initialisation, mais cela explique-t-il les meilleurs résultats ? En effet, chaque test est réalisé avec dix initialisations au hasard, et les résultats produits par NCEM sont donc peu dépendants des conditions initiales. Ainsi il semblerait que la différence entre ces algorithmes tient surtout au type d'estimateur recherché. Il serait donc intéressant d'adapter GEM à la recherche du MAP pour déterminer si cela influence beaucoup ses performances.

  Dans le cas de cette image synthétique, il semble que l'aspect flou de NEM permettent d'améliorer le taux de pixels mal classés de façon sensible. Nous n'avons pas d'explication de ce phénomène.

Lorsque la valeur de β est supérieure ou égale à deux :

- Les algorithmes NEM et NCEM donnent des résultats inexploitables (taux d'erreurs d'environs 50%). Ces taux d'erreurs très importants proviennent du fait que les algorithmes "vident" complètement une des deux classes et produisent donc des images segmentées d'une seule couleur. Il est tentant d'expliquer ce comportement par le fait que NEM ne converge pas pour des valeurs de β trop grandes. Mais si l'on se rappelle que chaque itération de NCEM fait croître le critère, on peut déduire qu'une image d'une seule couleur, pour ces valeurs de β, produit un meilleur critère U_d qu'une image binaire. Le problème n'est donc pas un problème de convergence, mais vient du fait que chercher l'estimateur du MAP, n'est pas du tout équivalent à chercher l'image qui possède le meilleur taux de pixels bien classés (estimateur du MPM).

Pieczinsky et Cahen (1994) note que l'algorithme GEM produits des résultats similaires aux algorithmes de segmentation statistique non supervisés les plus connus. Ainsi, on peut raisonnablement penser que les algorithmes NEM et NCEM offrent des alternatives compétitives en segmentation d'images, et de manière plus générale en classification automatique de données spatiales. D'un point de vue pratique il semble assez avantageux d'utiliser l'algorithme NEM avec des valeurs de β petites, plutôt que l'algorithme GEM. En effet, sur les simulations de cette section les résultats produits par NEM sont légèrement meilleurs, et de plus, le temps de calcul nécessité par GEM est très important. Ceci vient du fait que chaque itération de Gibbsian EM fait intervenir un échantillonneur de Gibbs très coûteux. En ce qui concerne les différences observées entre NCEM et NEM, on constate une différence maximum de 5% entre les taux de pixels mal classés. A chaque fois NEM donne de meilleurs performances que NCEM, mais notons que NCEM converge généralement en beaucoup moins d'itérations que NEM.

Chapitre 4

Applications

Dans ce chapitre trois applications, liées aux algorithmes présentés dans les chapitres précédents, sont développées. Comme cela a déjà été souligné dans le deuxième chapitre, la cartographie associative peut être utilisée pour la classification automatique, ou bien pour la représentation. Dans la première section l'algorithme TPEM est appliqué à un problème de classification liée à des données pétrolières. La représentation bidimensionnelle d'un tableau de distances horaires illustre dans une deuxième section les performances de ce même algorithme de cartographie associative dans le cadre de la réduction de dimension. Ce chapitre se termine par la démonstration des capacités des algorithmes de classification spatiale, introduits au chapitre trois, dans le contexte de la segmentation d'image.

4.1 Données géologiques

Les données considérées dans cette section sont des mesures réalisées sur différentes roches (Tableaux 4.1 et 4.2). Dix types de roches sont décrites par huit mesures différentes et pour chaque roche une dizaine de vecteurs de mesures est disponible. Dans la terminologie de l'analyse de données, le jeu de données est un tableau individus/variables de 150 individus, 8 variables quantitatives et une variable qualitative indicatrice de la classe (la nature de la roche).

Le problème que nous nous sommes posé est un problème de classification : supposons que la lithologie relative à ces séries de mesures ne soit pas connue (absence d'information de classe), comment retrouver la structure de classe ? Nous avons utilisé pour cela une approche probabiliste paramétrique et un algorithme de classification automatique présenté au chapitre 2 (Topology Preserving EM). Les performances optimales de cet algorithme sont déterminées dans une première partie par un classifieur bayésien basé sur les mêmes modèles probabilistes, et utilisant le jeu de données

TAB. 4.1 - : Données géologiques

Densite	Neutron	Sonique	Gamma	Ray	Pe	Rt	Rxo	Litholo
2.35	0.305	95	100	2.5	0	0	0.5	Argile
2.15	0.375	87	75	3.1	-0.1	-0.1	0.55	Argile
2.625	0.305	100	100	3	0.15	0.25	0.1	Argile
2.45	0.35	90	70	2.8	0.35	0.4	1.25	Argile
2.5	0.45	87	95	3	0.35	0.35	0.05	Argile
2.14	0.335	97	120	2.77	-0.1	0.22	0.15	Argile
2.625	0.325	92	85	2.8	0.2	0.25	0.5	Argile
2.625	0.295	85	100	2.9	0.2	0.25	0.15	Argile
2.45	0.4	91	140	2.7	0.1	0.15	0	Argile
2.55	0.325	85	100	2.3	0.39	0.4	0.5	Argile
2.5	0.36	92	102	2.87	0	0.27	0.15	Argile
2.125	0.325	97	135	3.3	0	0.1	0.5	Argile
2.125	0.35	90	85	2.95	0	0.1	0.25	Argile
2.45	0.475	89	75	3.1	0	-0.05	0.25	Argile
2.55	0.375	94	93	2.1	0.05	0.05	0.05	Argile
2.59	0.29	87	85	3.45	-0.05	0.15	0.2	Argile
2.51	0.12	64	25	4.75	1	1.9	0	Calcaire
2.32	0.225	75	30	4.95	0.3	0.9	-0.05	Calcaire
2.625	0.054	56	30	4.45	2	2.8	0.15	Calcaire
2.715	0.01	45	27	4.7	2.9	3	0	Calcaire
2.625	0.075	55	35	5	2.05	2.55	-0.15	Calcaire
2.585	0.065	58	65	5.1	1.8	2.5	0.25	Calcaire
2.335	0.25	82	35	4.65	0.1	1	0	Calcaire
2.32	0.25	84	40	4.75	0.2	1	0.05	Calcaire
2.52	0.135	67	25	4.7	0.7	1.6	0.05	Calcaire
2.5	0.135	69	30	4.6	0.6	1.6	0	Calcaire
2.45	0.135	68	35	4.8	0.9	1.5	0.02	Calcaire
2.475	0.145	70.5	45	4.48	0.35	1.65	-0.05	Calcaire
2.325	0.235	80.5	55	4.2	0.1	1.25	0.05	Calcaire
2.355	0.22	76.5	40	4.1	0.05	1.25	-0.05	Calcaire
2.295	0.24	84	40	4.5	0.3	1	0	Calcaire
2.305	0.26	85	35	4.55	0.2	0.95	0.05	Calcaire
2.465	0.205	72	15	3.85	0.35	1.25	-0.05	Calc-Dolo
2.62	0.11	56	40	4.1	1.3	2.5	0	Calc-Dolo
2.74	0.03	50	30	3.95	3	3.2	0	Calc-Dolo
2.71	0.05	50	45	4	2.5	2.9	0	Calc-Dolo
2.63	0.08	54	15	4.2	1.3	2.6	-0.05	Calc-Dolo
2.685	0.11	56	30	4	1.35	2.4	0	Calc-Dolo
2.595	0.135	60	40	3.9	0.7	1.7	0	Calc-Dolo
2.595	0.155	63	55	4	0.5	1.4	-0.04	Calc-Dolo
2.59	0.142	60	50	3.85	0.7	1.7	0.03	Calc-Dolo
2.59	0.16	63	45	3.9	0.6	1.4	-0.05	Calc-Dolo
2.54	0.21	69	55	3.95	0.6	1.4	-0.04	Calc-Dolo
2.49	0.21	70	45	3.85	0.7	1.2	-0.02	Calc-Dolo
2.45	0.25	76	55	4	0.6	1.3	-0.02	Calc-Dolo
2.4	0.25	77	65	3.85	0.5	1.4	-0.02	Calc-Dolo
2.27	0.3	84	25	3.7	0.05	0.8	0.02	Calc-Dolo
2.25	0.32	85	35	3.6	0	0.7	0	Calc-Dolo
2.55	0.215	67	25	3.25	0.7	1.7	0	Dolomie
2.8	0.1	50	35	3.3	2.7	3.1	-0.05	Dolomie
2.53	0.24	68	55	3.85	0.3	1.1	0.04	Dolomie
2.53	0.27	70	40	3.95	0.4	1.3	-0.03	Dolomie
2.525	0.24	70	65	3	0.3	1.2	0	Dolomie
2.54	0.26	72	20	3.1	0.3	0.95	0.1	Dolomie
2.83	0.05	45	45	3.35	2.7	3	-0.02	Dolomie
2.8	0.05	46	55	3.45	2.45	2.7	-0.01	Dolomie
2.83	0.05	45	45	3.15	3	3	0	Dolomie
2.8	0.045	45	55	3.25	2.7	2.7	0.01	Dolomie
2.53	0.24	68	55	3.05	0.9	1.3	0.05	Dolomie
2.67	0.15	55	25	3.45	1.15	2	0	Dolomie
2.65	0.16	59	35	3.35	1.1	2.2	-0.04	Dolomie
2.67	0.15	54.5	20	3.1	0.95	2	-0.01	Dolomie
2.65	0.16	59	37	3.15	1.05	2.2	-0.05	Dolomie
2.505	0.265	76	65	3.05	0.3	1.4	-0.01	Dolomie
2.445	0.11	73	35	2.24	0.65	1.2	-0.05	Gres
2.18	0.25	96	50	2	-0.08	1.15	0.25	Gres
2.04	0.35	110	35	1.825	-0.45	0.54	-0.02	Gres
2.12	0.345	107	20	1.975	-0.375	0.52	0.05	Gres
2.23	0.21	99	50	1.6	0.05	1	0.2	Gres
2.2	0.23	104	50	1.7	-0.05	1.25	1	Gres
2.62	-0.005	60.5	12	1.81	2.5	3.25	0	Gres
2.615	0	59.5	15	1.83	2.5	3.25	0.05	Gres
2.555	0.04	64.5	25	1.82	1.9	2.7	0	Gres
2.56	0.04	62	10	1.8	1.75	2	0.02	Gres
2.625	0	60.5	12	1.77	2.5	3.25	-0.02	Gres

TAB. 4.2 - : Données géologiques

Densite	Neutron	Sonique	Gamma	Ray	Pe	Rt	Rxo	Litholo
2.12	0.27	98	50	2.1	0.3	0.8	-0.15	Gres
2.12	0.3	98	60	1.8	0.2	0.45	-0.15	Gres
2.12	0.27	101	100	2.36	0.3	0.8	0.35	Gres
2.425	0.115	72.5	15	1.8	0.53	1.7	-0.5	Gres
2.4	0.11	73	55	2.3	0.675	1.25	0.5	Gres
2.31	0.185	81	30	1.7	0.35	1.15	-0.5	Gres
2.315	0.185	81	45	2.5	0.35	1.1	-0.25	Gres
2.44	0.09	70	15	2.1	0.8	1.45	0.01	Gres
2.4	0.135	73	20	2.275	0.4	1.49	-0.5	Gres
2.35	0.14	76	45	1.9	0.3	1.1	-0.75	Gres
2.25	0.2	87	10	1.8	0.2	0.9	-0.02	Gres
2.325	0.195	43.5	55	4	0.1	1.25	0.1	Gres-Calc
2.515	0.085	70	25	4.4	1	1.9	0	Gres-Calc
2.4	0.145	72.5	45	4.2	0.35	1.6	-0.04	Gres-Calc
2.325	0.195	83.5	55	4	0.1	1.25	0.1	Gres-Calc
2.265	0.235	87.5	40	4.4	-0.1	1.25	-0.01	Gres-Calc
2.425	0.14	75	35	3.9	0.45	1.75	0.1	Gres-Calc
2.375	0.16	77	40	4.4	0.35	1.32	-0.1	Gres-Calc
2.355	0.18	79	20	4.1	0.25	1.15	-0.15	Gres-Calc
2.315	0.2	83	50	4.35	0	1.2	0	Gres-Calc
2.24	0.245	90	55	4	0.15	1	-0.08	Gres-Calc
2.3	0.21	82	20	3.9	-0.1	1.35	-0.05	Gres-Calc
2.57	0.045	57	30	3.8	2	2.8	0.15	Gres-Calc
2.66	0	54	25	3.9	2.6	3	0	Gres-Calc
2.63	0.02	56	45	4	2.5	3.2	-0.1	Gres-Calc
2.615	0.03	58	20	4.1	2.3	2.9	-0.01	Gres-Calc
2.505	0.09	64	25	4	1	1.9	0.01	Gres-Calc
2.515	0.085	62	5	4.4	1.4	2.2	-0.05	Gres-Calc
2.1	-0.025	69	35	4.6	1.5	2.9	0	Halite
2.05	0.01	68	10	4.65	2.5	2.5	0.05	Halite
2.03	-0.025	67	0	4.4	2	3.2	0	Halite
2.03	0	66	25	4.45	1.8	3.2	-0.15	Halite
2.03	0.025	68	20	4.55	1.8	3.2	0.01	Halite
2.06	-0.025	69	20	4.65	2.5	3.4	-0.01	Halite
2.06	0	66	15	4.6	1.8	3.9	0.15	Halite
2.06	0.025	68	30	4.7	1.5	2.8	-0.15	Halite
2.1	0	69	20	4.35	1.8	2.5	0	Halite
2.1	0.025	68	10	4.45	1.8	2.7	-0.1	Halite
2.925	0.025	52.5	15	4.77	2.2	1.8	0	Anhydrite
2.9	-0.025	54	10	4.88	2.5	3	0.02	Anhydrite
2.925	0.025	52.5	15	4.77	2	1.75	0	Anhydrite
2.9	-0.025	54	15	4.88	2.2	2	0.01	Anhydrite
2.9	0	53	20	5.05	2.5	3	-0.01	Anhydrite
2.95	-0.025	50	0	4.7	2.4	3.5	0.01	Anhydrite
2.95	0.025	51	10	4.9	1.9	2.6	0.02	Anhydrite
2.875	0.025	53	20	4.8	2.6	2.7	0.015	Anhydrite
3	0.025	49	10	5	1.75	2	-0.03	Anhydrite
2.98	-0.02	55	10	4.9	2	1.75	0.01	Anhydrite
3.025	-0.005	47	30	4.8	2.3	2.45	0	Anhydrite
3.05	0.015	49	10	5.1	1.75	2	-0.03	Anhydrite
2.91	0	50	12	5	5	4	-0.02	Anhydrite
2.95	0.02	48	13	5.05	4.9	3.9	-0.01	Anhydrite
3.05	-0.02	52.5	17	4.95	4.5	3.5	0.015	Anhydrite
2.27	0.5	52	30	4.2	2.2	2.1	0.1	Gypse
2.36	0.53	54	35	4.3	1.7	2	0	Gypse
2.27	0.5	52	30	4.2	2.2	2.3	0.1	Gypse
2.26	0.545	53	40	3.88	1.9	1.95	-0.05	Gypse
2.315	0.505	52	30	4.1	1.8	1.95	0.1	Gypse
2.37	0.51	52	35	4.3	1.7	1.85	0.04	Gypse
2.42	0.52	51	35	3.9	1.8	2	0.15	Gypse
2.4	0.5	50	40	4	2	1.9	0.1	Gypse
2.355	0.49	51	40	4.35	1.9	2.05	-0.04	Gypse
2.32	0.52	54	40	3.98	1.85	2	0.15	Gypse
2.25	0.55	52	0	4.2	2.5	3	0	Gypse
2.25	0.6	53	18	3.88	1.8	2.5	0.25	Gypse
2.3	0.53	52	20	4.1	1.8	2.35	-0.25	Gypse
2.3	0.58	54	18	3.99	2.6	3.3	0	Gypse
2.35	0.53	52	20	4.3	1.7	2.8	0.01	Gypse
2.36	0.58	51	25	3.77	2.1	3	0.25	Gypse
2.4	0.55	51	25	3.9	1.8	2.5	-0.25	Gypse
2.45	0.57	50	40	4.1	1.8	3	0	Gypse
1.75	0.6	110	35	1.3	2.2	2.6	0	Charbon
1.95	0.38	120	56	1.1	1.3	1.6	-0.05	Charbon
1.29	0.45	117	25	1	2.4	2.5	0.05	Charbon
1.49	0.75	120	50	0.85	2.4	2.6	0	Charbon
1.84	0.55	120	30	1.2	1.9	1.9	0.05	Charbon
1.93	0.5	100	35	1	1.8	1.9	0.15	Charbon
1.76	0.56	97	45	1.4	1.5	1.7	0.25	Charbon
1.83	0.755	94	52	1.45	1.85	1.8	0.3	Charbon
1.65	0.57	105	42	1.35	2.23	2.301	0.1	Charbon

TAB. 4.3 - : Statistiques relatives au jeu de données

Variable	Densité	Neutron	Sonique	Gamma	Ray	Pe	Rt	Rxo
Minimum	1.29	0.025	43.5	0	0.85	−0.45	−0.1	−0.75
Maximum	3.05	0.755	120	140	5.1	5	4	1.25
Moyenne	2.43	0.22	70.58	39.39	3.57	1.24	1.83	0.04
Variance	0.092	0.036	347.26	676.5	1.24	1.14	0.93	0.043

TAB. 4.4 - : Effectifs des différentes classes

Classe	Argile	Calcaire	Calc-Dolo	Dolomie	Gres	Gres-Calc	Halite	Anhydrite	Gypse	Charbon
Effectif	16	16	16	16	22	17	10	15	18	9

disponible comme ensemble d'apprentissage.

4.1.1 Apprentissage supervisé par classifieur de Bayes

Un classifieur de Bayes (Duda et Hart 1973) considère que les individus (vecteurs de $\mathbb{R}^d$) de la classe ω_k ont une densité de probabilité $f(\mathbf{x}|\omega_k)$. Chacune des K classes $\omega_1, \cdots, \omega_K$ possède une probabilité a priori $\pi_1, \cdots, \pi_K$. La probabilité a posteriori qu'un individu $\mathbf{x}$ appartienne à la classe ω_k peut être exprimée par le théorème de Bayes :

$$Pr(\omega_k|\mathbf{x}) = \frac{\pi_k f(\mathbf{x}|\omega_k)}{\sum_{\ell=1}^{K} \pi_\ell f(\mathbf{x}|\omega_\ell)}.$$

Supposons que les densités $f(\mathbf{x}|\omega_1), \cdots, f(\mathbf{x}|\omega_K)$ et les probabilités $\pi_1, \cdots, \pi_K$ soient connues, sur quel critère affecter un individu $\mathbf{x}$ à une classe ω_{k^*} ? Dans un cadre bayésien, la décision prise est celle qui minimise le coût a posteriori :

$$\omega_{k^*} = arg \min_{\delta} \rho(\delta|\mathbf{x})$$

avec

$$\rho(\delta|\mathbf{x}) = \mathbb{E}[L(\delta, \Omega)|\mathbf{x}] = \sum_{k=1}^{K} L(\delta, \omega_k) Pr(\omega_k|\mathbf{x})$$

où $L(\delta, \omega_k)$ est le coût de classer $\mathbf{x}$ dans la classe δ lorsque $\mathbf{x}$ appartient en fait à la classe ω_k. Si l'on considère un coût "0–1" (1 pour une mauvaise décision et 0 pour la bonne décision), l'individu $\mathbf{x}$ sera affecté à la classe qui possède la probabilité a posteriori maximum (Principe du MAP).

Pour pouvoir appliquer cette procédure de classement, la première chose à faire est d'identifier les densités $f(\mathbf{x}|\omega_1), \cdots, f(\mathbf{x}|\omega_K)$ et les probabilités $\pi_1, \cdots, \pi_K$.

Dans notre cas nous avons supposé que chaque densité $f(\mathbf{x}|\omega_k)$ était une loi gaussienne multivariée. Les paramètres à estimer sont alors le vecteur moyenne $\boldsymbol{\mu}_k$ et la matrice de variance $\boldsymbol{\Sigma}_k$, pour chaque classe. Pour estimer ces paramètres, les 155 individus ont été utilisés (ensemble d'apprentissage).

Remarquons l'ensemble d'apprentissage fournit tout au plus 20 individus pour estimer les paramètres d'une densité. Les matrices de covariances étant des matrices 8 par 8, il semble déraisonnable d'estimer 64 paramètres à partir de 20 observations (voir le Tableau 4.4). Ainsi, nous avons poser des hypothèses simplificatrices pour réduire le nombre de paramètres à estimer. Deux modèles ont été considérés :

- Le modèle linéaire : chaque classe est caractérisée par une densité gaussienne de matrice de variance $\mathbf{\Sigma}_k = \lambda I$ où I est la matrice identé. Chaque densité possède donc la même matrice de variance. Cette hypothèse revient à considérer des surfaces séparatrices linéaires (hyperplans) entre les classes. Nous noterons ce modèle **Bayes** $[\lambda I]$.

- Le modèle quadratique avec paramètre de volume : chaque classe est caractérisée par une densité gaussienne de matrice de variance $\mathbf{\Sigma}_k = \lambda_k I$ où le paramètre λ_k peut être interprété géométriquement comme le volume de la classe k. Cette hypothèse revient à considérer des surfaces séparatrices quadratiques et ne rajoute qu'un seul paramètre par classe à estimer. Nous noterons ce modèle **Bayes** $[\lambda_k I]$.

L'erreur sur l'ensemble d'apprentissage est le pourcentage d'individus bien classés par le classifieur Bayésien. Nous avons estimé l'erreur en généralisation par une procédure de *validation croisée* et par *Bootstrap* (Tableau 4.5). On constate que l'utilisation du modèle plus complexe $[\lambda_k I]$ réduit l'erreur sur l'ensemble d'apprentissage de 5% par rapport au modèle linéaire, et que les deux modèles produisent des erreurs en généralisation qui sont du même ordre.

TAB. 4.5 - : Erreur sur l'ensemble d'apprentissage et en généralisation (pourcent).

Modèle	Apprentissage	Généralisation (val. croisée)	Généralisation (Bootstrap)
Bayes $[\lambda I]$	24.52	29.03	27.11
Bayes $[\lambda_k I]$	20.65	29.68	26.27

Une démarche, qui semble intéressante dans ce cadre, consiste à regarder de plus près, quel sont les individus mal classés. Ainsi, en calculant le produit matriciel de la matrice de classification réelle et de la matrice de classification obtenue avec le classifieur de Bayes, on obtient une matrice carrée de dimension $K \times K$, où l'élément (i, j) indique le nombre d'individus de la classe i qui sont classés dans la classe j par le classifieur de Bayes (tableau 4.6). Cela permet de constater que :

- les individus des classes Gypse, Charbon, Anhydrite, Halite et Argile sont toujours classés correctement ;

- les individus de la classe Calcaire sont confondus avec ceux des classes Gres-Calc et Anhydrite ;

TAB. 4.6 - : Erreurs de classement réalisées par le classifieur de Bayes $[\lambda \cdot I]$

		Classes vraies									
		Argile	Calcaire	Calc-Dolo	Dolomie	Gres	Gres-Calc	Halite	Anhydrite	Gypse	0Charbon
Bayes	Argile	16	0	0	0	2	0	0	0	0	0
	Calcaire	0	12	0	0	0	4	0	0	0	0
	Calc-Dolo	0	0	8	7	0	1	0	0	0	0
	Dolomie	0	0	3	6	5	1	0	0	0	0
	Gres	0	0	1	0	15	0	0	0	0	0
	Gres-Calc	0	1	2	0	0	8	0	0	0	0
	Halite	0	0	0	0	0	0	10	0	0	0
	Anhydrite	0	3	2	3	0	3	0	15	0	0
	Gypse	0	0	0	0	0	0	0	0	18	0
	0Charbon	0	0	0	0	0	0	0	0	0	9

- les individus de la classe Dolomie sont souvent confondus avec ceux de la classe Calc-Dolo ;
- les individus de la classe Gres-Calc sont principalement confondus avec ceux des Classes Calcaire et Anhydrite ;
- les individus de la classe Gres sont confondus avec ceux de la classes Dolomie.

4.1.2 Classification automatique

Si nous supposons que les classes réelles des individus ne sont pas disponibles, mais que le nombre de classes est connu, l'utilisation des méthodes de classification automatique constitue une analyse possible.

Dans un contexte probabiliste de classification automatique, les individus sont considérés comme un échantillon $(\mathbf{x}_1, ..., \mathbf{x}_N)$ d'une variable aléatoire à valeurs dans $\mathbb{R}^d$ de densité:

$$f(\mathbf{x}_i|\Phi) = \sum_{k=1}^{K} p_k f_k(\mathbf{x}_i|\theta_k), \tag{4.1}$$

où les p_k sont les proportions du mélange ($0 < p_k < 1$, pour $k = 1, .., K$ et $\sum_k p_k = 1$) et $f_k(\mathbf{x}|\theta_k)$ est une loi complètement déterminée par la connaissance du vecteur θ_k.

Si les paramètres du mélange sont connus, une partition des individus peut alors être obtenue en affectant chaque individu $\mathbf{x}$ à la composante du mélange la plus probable a posteriori. C'est le principe du maximum a posteriori mentionné dans la section précédente.

L'approche de la classification que nous considérons ici comporte deux étapes :

- estimation des paramètres du mélange,
- classification par le principe du MAP.

Pour cette tâche de classification automatique nous avons utilisé l'algorithme Stochastic Topology Preserving EM (cf. Chapitre 2) qui est un algorithme de classification dérivé de l'algorithme EM (Dempster *et al.* 1977). Cet algorithme possède un avantage par rapport à l'algorithme EM : il partitionne les individus en K classes et

TAB. 4.7 - : Pourcentage d'individus bien classés.

Algorithme	Erreur
STPEM $[\lambda I]$	41.24
STPEM $[\lambda_k I]$	36.77

TAB. 4.8 - : Relation d'ordre entre les classes

Ordre	1	2	3	4	5	6	7	8	9	10
Classe	Dolomie	Calc-Dolo	Anhydrite	Halite	Gres	Calcaire	Gres	Gres-Calc	Argile	Charbon

fournit aussi des informations sur la similarité des classes les unes par rapport aux autres comme l'algorithme de Kohonen (Kohonen 1984).

Une relation linéaire entre les classes a été postulée (relation d'ordre). Nous avons considéré un mélange de lois gaussiennes et utilisé les mêmes hypothèses simplificatrices que dans la section précédente. Ainsi nous avons deux modèles.

Les performances des deux algorithmes ont été estimées à l'aide du pourcentage d'individus bien classés. Remarquons que la meilleure performance possible dans ce cadre est déterminée par un classifieur de Bayes qui connaîtrait l'appartenance aux classes de chaque individu. Ainsi les erreurs d'apprentissage de la section précédente nous donne la borne supérieure des performances de notre algorithme d'apprentissage non supervisé.

Le tableau 4.8 a été obtenu en interprétant les classes fournies par l'algorithme à l'aide de la connaissance des classes réelles. Ce tableau interpréte donc la relation d'ordre postulée a priori.

Le modèle $[\lambda_k I]$ produit un taux d'individu mal classés inférieur d'environ 5% à celui exhibé par le modèle $[\lambda I]$. Cette différence est du même ordre que celle obtenue sur l'ensemble d'apprentissage par le classifieur bayésien utilisé avec les mêmes modèles. On peut remarquer aussi que l'ordre supposé *a priori* entre les différentes classes (Tableau 4.8) permet de retrouver certaines des observations réalisées à partir du tableau 4.6 : par exemple les classes Calc-Dolo et Dolomie sont voisines. Cette relation de voisinage signifie que les individus des deux classes sont similaires et donc susceptible d'être confondus. Ainsi, en pratique, la proximité de deux classes dans l'espace de sortie se traduit par un risque de confusion accru entre les individus de ces deux classes.

4.1.3 Remarques

Notons que l'utilisation du modèle sphérique avec volumes différents apporte une amélioration par rapport aux résultats obtenus dans un cadre supervisé ou non supervisé avec un modèle sphérique postulant des volumes identiques. L'estimation des

volumes nécessite peu de données et peu de calcul. C'est donc un type de paramétrisation qui semble intéressant (au moins pour ce problème).

L'algorithme TPEM ne produit pas de meilleur résultat que l'algorithme CEM, mais permet d'obtenir quelques renseignements supplémentaires sur la validité de la classification obtenue.

Une amélioration sensible pourrait être obtenue en intégrant le fait que les données sont localisées et donc que deux mesures géographiquement proches ont des chances de caractériser une lithologie similaire. Il serait peut être intéressant dans ce cadre de mélanger les approches développées en cartographie associative et en classification spatiale.

4.2 Représentation d'un tableau de distances horaires

Les données considérées dans cette section sont dérivées des distances horaires SNCF entre 21 villes françaises (Figure 4.1). Ces distances ont été collectées par 3 étudiants du DEA "contrôle des systèmes" de l'UTC en 1994. Sur plusieurs trajets possibles entre deux villes, le temps le plus court a été retenu. L'ensemble de ces mesures constituent une matrice de dissimilarité non symétrique (Tableau 4.9).

Le problème qui nous a intéressé, consiste à représenter ces 21 villes dans le plan en respectant "au mieux" les distances horaires spécifiées par la SNCF. Ces représentations sont donc des cartes de France déformées, qui donnent une idée des liaisons ferroviaires françaises. Quatre méthodes différentes ont été testées pour obtenir ces cartes ferroviaires :

- une analyse factorielle,
- une projection de Sammon,
- une projection de Kruskal,
- une projection de Kohonen réalisé avec l'algorithme TPEM.

Ces différents tests permettent d'aider à situer les méthodes de cartographie associative parmi d'autre méthodes plus classiques de représentation.

4.2.1 Analyse des données

Les quatre méthodes utilisées dans cette section opèrent sur des matrices de dissimilarité symétriques. La première étape de l'analyse consiste donc à rendre la matrice des dissimilarités initiales symétrique. Cette opération est réalisée en retenant, comme élément constituant la nouvelle matrice symétrique, les temps de parcours les plus

TAB. 4.9 - : Tableau des durées minimales des trajets SNCF entre 21 villes françaises (durées obtenues aux guichets automatiques SNCF)

	Avignon	Bordeaux	Clermont-F	Dijon	Geneve	Grenoble	Lille	Limoges	Lyon	Marseille	Metz	Montpellier	Mulhouse	Nancy	Nantes	Nice	Paris	Perpignan	Rennes	Strasbourg	Toulouse
Avignon	0.0	6.15	5.39	3.43	3.46	2.7	5.27	7.49	1.46	0.52	7.9	0.48	6.28	6.16	7.27	3.42	3.45	2.34	6.55	7.32	3.28
Bordeaux	5.12	0.0	5.57	7.32	7.49	7.17	5.27	2.14	7.42	5.19	7.22	3.57	9.2	7.44	3.55	8.0	2.57	4.38	5.45	7.59	2.6
Clermont-F	5.2	6.1	0.0	3.43	5.13	4.42	5.25	3.34	2.40	5.46	7.58	5.22	7.4	7.49	5.36	7.46	3.19	7.13	7.35	8.9	6.2
Dijon	3.45	6.2	4.40	0.0	3.1	3.22	4.15	6.2	1.33	4.30	3.2	4.55	2.38	2.17	5.7	7.48	1.40	7.7	5.28	3.49	7.49
Geneve	3.55	8.22	5.19	2.46	0.0	1.55	5.30	8.6	1.45	4.55	6.2	5.4	3.16	6.38	6.38	8.10	3.27	6.28	6.30	4.36	7.32
Grenoble	2.4	10.20	4.28	3.25	2.1	0.0	4.55	8.1	1.11	3.31	7.1	3.27	6.32	6.15	6.10	6.13	2.58	4.27	6.15	5.0	5.30
Lille	5.47	4.59	5.49	3.22	5.44	5.4	0.0	4.57	3.53	7.46	3.46	6.36	6.21	4.35	4.28	10.31	1.1	10.19	4.22	5.7	7.7
Limoges	5.35	2.16	3.45	6.39	9.50	8.5	5.16	0.0	5.27	9.29	7.13	6.46	10.39	7.22	4.42	11.57	3.1	5.0	8.35	10.27	3.19
Lyon	1.46	7.32	2.37	1.42	1.48	1.11	3.52	5.13	0.0	2.43	4.40	2.55	4.0	4.10	4.34	5.7	2.0	4.30	4.49	5.4	5.25
Marseille	0.51	5.28	6.7	4.45	4.55	3.30	7.47	8.22	2.49	0.0	7.49	1.30	7.30	6.59	8.29	2.22	4.45	3.19	14.29	8.35	3.25
Metz	8.19	7.25	8.10	2.52	6.55	6.57	3.45	6.53	4.55	8.4	0.0	8.10	2.17	0.45	7.1	10.39	2.43	9.59	6.40	1.19	10.10
Montpellier	0.48	3.55	5.0	4.34	5.32	3.4	6.27	6.1	2.43	1.19	8.20	0.0	8.9	7.35	10.10	4.1	4.45	1.21	7.50	9.2	2.36
Mulhouse	6.16	9.2	8.2	2.34	3.25	5.42	6.19	8.52	3.49	7.19	2.7	7.45	0.0	2.22	8.25	10.22	4.16	9.6	7.42	0.49	10.18
Nancy	6.28	6.42	6.16	2.12	6.53	6.12	4.30	6.35	4.13	7.7	0.33	9.39	2.40	0.0	6.34	10.0	2.45	9.18	6.21	1.10	9.29
Nantes	6.46	3.51	5.30	5.1	10.39	6.19	4.26	5.0	4.44	8.0	6.10	8.6	7.36	6.19	0.0	10.40	2.1	9.22	1.31	7.55	6.10
Nice	3.37	7.53	9.19	7.57	8.33	6.42	9.34	10.25	4.58	2.15	10.45	3.55	10.49	9.52	10.49	0.0	7.13	6.24	11.5	11.56	6.53
Paris	3.49	2.57	3.21	1.37	3.27	2.55	1.0	3.1	2.10	4.45	2.43	4.45	4.10	2.43	2.6	6.58	0.0	6.34	2.4	3.55	5.9
Perpignan	2.38	4.30	10.2	7.30	6.28	4.27	10.11	4.55	4.34	3.22	11.39	1.36	6.46	10.55	9.10	7.4	6.44	0.0	11.20	11.45	2.7
Rennes	6.50	5.30	7.5	4.59	6.50	6.15	4.25	6.30	4.36	11.55	6.24	7.52	8.9	5.59	1.19	10.57	2.4	12.7	0.0	7.46	8.25
Strasbourg	7.24	8.11	9.6	3.55	4.19	6.48	5.7	11.18	4.46	8.33	1.18	8.39	0.51	1.11	8.5	11.6	4.4	10.10	7.54	0.0	11.1
Toulouse	3.46	2.8	6.15	7.46	7.38	5.36	7.34	3.17	5.48	3.14	9.37	2.40	11.4	10.7	6.40	5.55	5.5	2.0	8.39	11.2	0.0

FIG. 4.1 - : Situation géographique de 21 villes de France

courts; si D_{brute} est la matrice des données initiales, les éléments de la nouvelle matrice symétrique D sont définis comme

$$D_{brute}(i,j) = D_{brute}(j,i) = min\{D(i,j), D(j,i)\}.$$

L'analyse factorielle suppose implicitement que la matrice D est une matrice de distances, c'est-à-dire que l'inégalité triangulaire est vérifiée. Remarquons que l'inégalité triangulaire est souvent non vérifiée par les distances horaires de la SNCF : il faut par exemple 1h01 pour aller de Lille à Paris, 2h00 pour faire Paris-Lyon, mais la SNCF indique 3h54 pour le trajet Lille Lyon. En considérant (abusivement) que D est une matrice de distances, on peut calculer la matrice des produits scalaires, W. Les vecteurs propres et valeurs propres de cette matrice W sont ensuite déterminés et permettent d'obtenir aisément les composantes principales. En triant les valeurs propres dans l'ordre décroissant et en calculant les pourcentages d'inertie cumulée (pour les valeurs propres négatives on utilise la valeur absolue), on obtient le tableau 4.10

Indice	Valeur propre	Pourcentage d'inertie	Pourcentage cumulé
1	9.75	31.88	31.88
2	5.74	18.75	50.62
3	2.69	8.81	59.43
4	1.94	6.35	65.78
5	1.28	4.17	69.95
6	0.98	3.20	73.75
7	0.65	2.11	75.26
8	0.54	1.75	77.02
9	0.17	0.56	77.58
10	0.05	0.15	77.73
11	0.02	0.06	77.79
12	0	0	77.79
13–20	-6.79	22.21	100

TAB. 4.10 - : Valeurs propres de la matrice $\frac{W}{21}$

Observons que de nombreuses valeurs propres sont négatives et que leur pourcentage d'inertie totale est de 22.21%. Cela signifie que les dissimilarités initiales sont loin d'être des distances. La projection sur les différents plans factoriels doit théoriquement être de mauvaise qualité. En effet si l'on considère les deux premiers plans (Figures 4.2 et 4.3), on peut constater de nombreuses incohérences. Relevons par exemple que sur les deux plans factoriels Toulouse se situe à proximité de Paris, alors que plus de cinq heures sont nécessaires pour relier les deux villes.

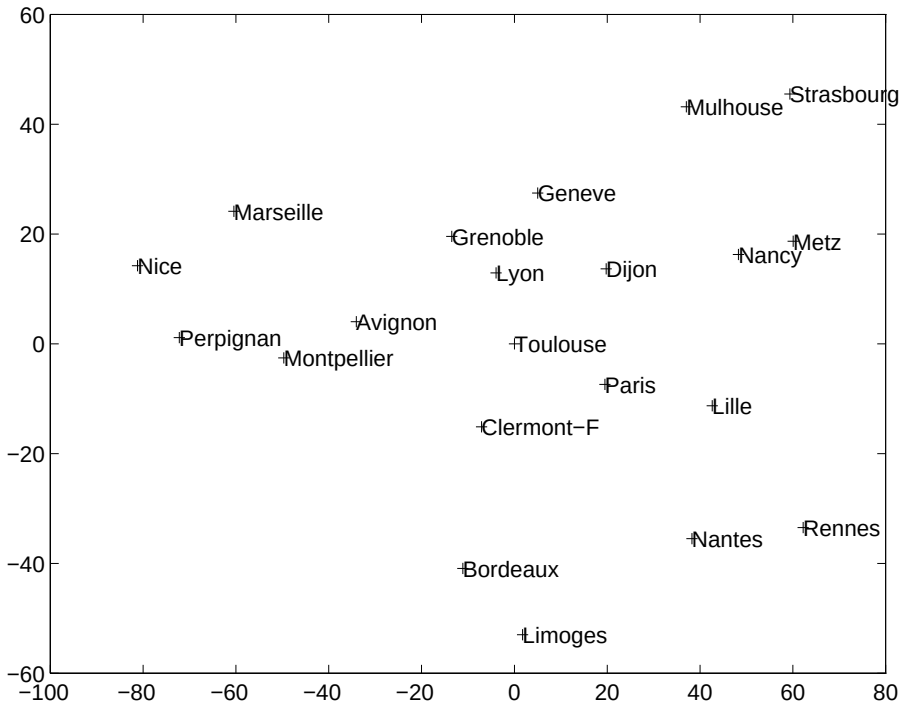


FIG. 4.2 - : Analyse factorielle du tableau de dissimilarité : axes 1 et 2

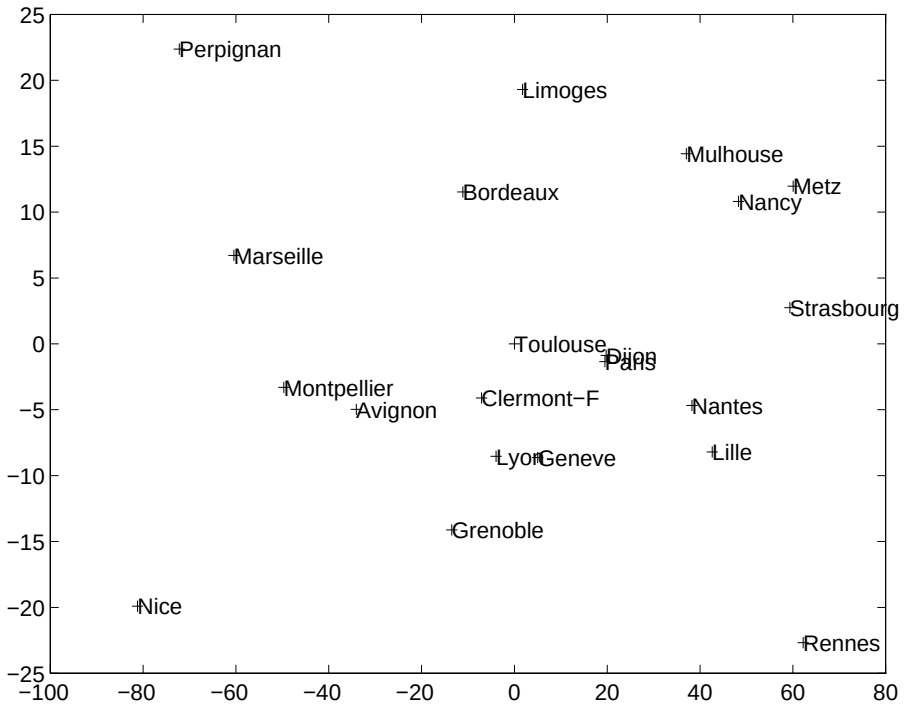


FIG. 4.3 - : Analyse factorielle du tableau de dissimilarité : axes 1 et 3

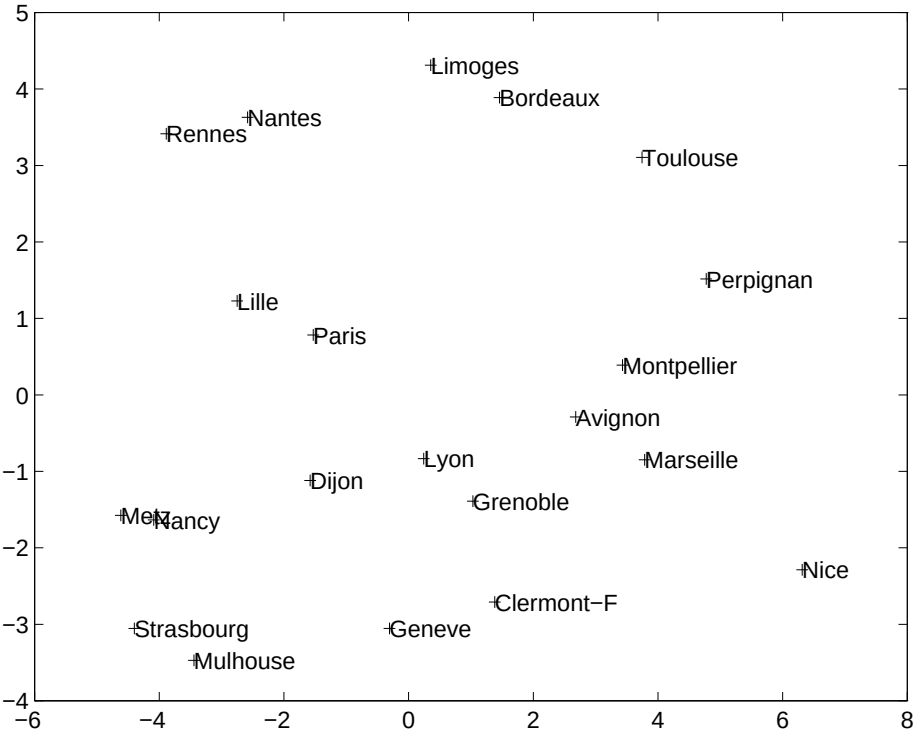


FIG. 4.4 - : Projection de Sammon des 21 villes

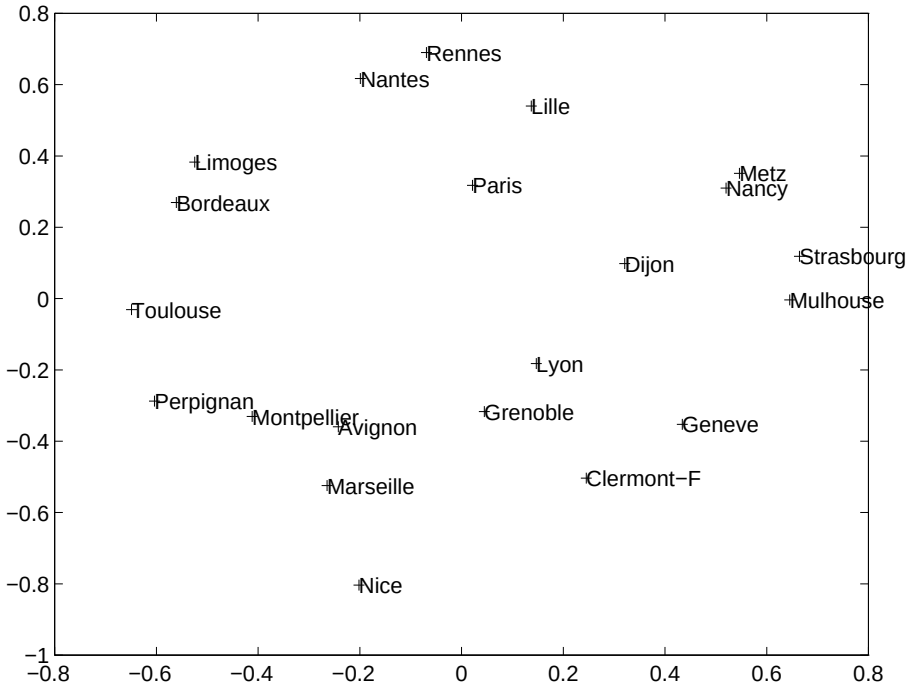


FIG. 4.5 - : Projection de Kruskal des 21 villes

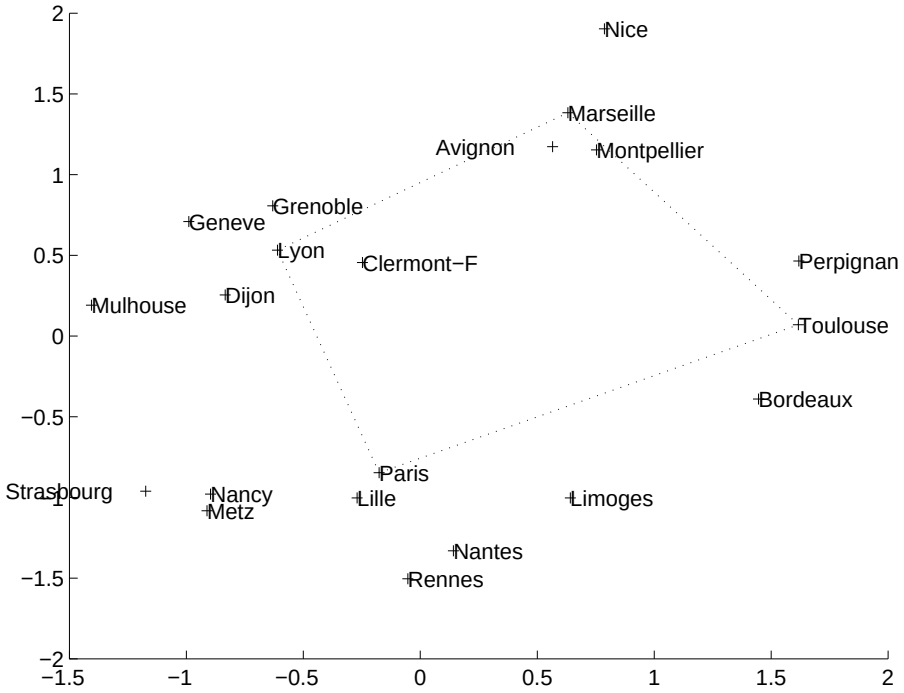


FIG. 4.6 - : Projection hybride de Kohonen–Sammon

La projection de Sammon (Sammon 1969) est une méthode de positionnement multidimensionel ("multidimensional scaling") métrique. La technique consiste à placer les villes dans le plan de manière à ce que les distances entre les points du plan soient aussi proches que possible des dissimilarités initiales au sens d'un certain critère. Une descente de gradient à pas adaptatif a été utilisée pour minimiser le critère. Le résultat produit (Figure 4.4) place Paris au centre de la représentation et les autres villes sont à peu près placées sur un cercle autour de la capitale. Les relations locales semblent bien préservées. Cette représentation met en évidence la structure centralisée du réseau ferroviaire français.

La projection de Kruskal (Kruskal 1964) est aussi une méthode de positionnement multidimensionel. Cette méthode est non métrique, c'est-à-dire qu'elle tente de préserver dans le plan les relations d'ordre entres les dissimilarités initiales. La représentation générée par l'algorithme de Kruskal (Figure 4.5) est très proche de celle obtenue par la projection de Sammon.

L'algorithme des cartes de Kohonen traite seulement des tableaux individus variables. Transformer la matrice de dissimilarité initiale en un tableau individus variables peut être réalisé par une analyse factorielle, mais nous avons déjà constaté que l'analyse factorielle engendre une perte d'information dans le cas qui nous concerne. Ainsi nous avons utilisé la version modifiée de l'algorithme TPEM, qui traite directement des matrices de dissimilarités (cf. Chapitre 2). Une carte qui possède autant de neurones qu'il y a de ville pourrait être utilisée pour obtenir une représentation. En prenant un carte bidimensionnelle de quatre fois cinq unités, chacune des 21 villes serait représentée par une unité différente et la France ferroviaire deviendrait une grille dont chaque noeud serait une ville. Ce type de représentation paraissant rigide, nous avons choisi d'utiliser une petite carte combinée avec une des méthode de projection introduite dans le chapitre deux. La carte retenue comporte quatre unité qui forme une grille deux par deux dans l'espace de sortie. La projection hybride Kohonen-Sammon a été retenue car bien adaptée à la représentation de petites cartes.

Les résultats (Figure 4.6) obtenus avec la projection hybride, montre une image qui ressemble plus à ce que l'on peut obtenir par les projections de Sammon et Kruskal, que par une analyse factorielle. La spécificité de la représentation est la dissection des 21 villes en quatre groupes distincts. Ces quatre groupes correspondent à quatre régions géographiques de France. On peut remarquer que Lyon et Toulouse appartiennent à deux groupes qui sont spatialement opposés sur la carte, ce qui correspond bien à une réalité du réseau SNCF. Remarquons aussi que la circulation entre les villes d'un même groupe est généralement aisée. Ainsi, l'algorithme TPEM a classé les villes en quatre groupes, chaque groupe semble posséder une signification par rapport au réseau SNCF, et de plus les quatre groupes sont situés de manière sensée les uns par rapport aux autres.

4.2.2 Remarques

Dans cette section, un tableau de distance horaire SNCF a été représenté dans le plan par quatre méthodes différentes. L'analyse factorielle a mis en évidence le fait que les dissimilarités du tableau était mal approximées par des distances. Les projections de Sammon et Kruskal montrent que le réseau ferroviaire français possède une structure centralisée (autour de Paris). Une carte de Kohonen projette les individus dans un espace de faible dimension, et produit une classification. La projection hybride de Kohonen–Sammon utilise avantageusement ces deux caractéristiques pour représenter les 21 villes : elle met l'accent sur la bonne connection ferroviaire à l'intérieur de différentes régions et montre les relations qui existent entre ces différentes régions.

4.3 Cultures cellulaires

Le problème : dans le laboratoire de biologie de l'université de technologie de Compiègne, des chercheurs s'intéressent aux cultures cellulaires, et testent comment des cellules vivantes se développent dans des milieux différents. Dans une optique de comparaison des milieux de cultures, les biologistes réalisent l'expérience suivante :

1. Un échantillon de cellules vivantes, un morceau de tissu gingival par exemple, est déposé sur un milieu nutritif. Le tissu organique constitué des cellules se nomme explant et le milieu nutritif est couramment appelé substrat.

2. Après quelques jours les cellules initiales se sont multipliées, et un voile formé de nouvelles cellules est visible sur le substrat.

3. Le biologiste mesure la taille de ce voile, ce qui lui fournit une indication sur la vitesse de reproduction des cellules sur le substrat de l'expérience.

La détermination manuelle de la taille du voile est une tache non évidente, qu'il serait souhaitable d'automatiser par un traitement d'image adéquat. Plusieurs alternatives sont envisageables pour résoudre ce problème par des méthodes d'analyse d'image. On peut penser, par exemple, au suivi de contour pour détecter la séparation entre les trois substances. Une autre solution consiste à segmenter l'image en trois classes en ésperant que chacune des classes correspondra effectivement :

- à l'explant,
- au voile,
- ou au substrat.

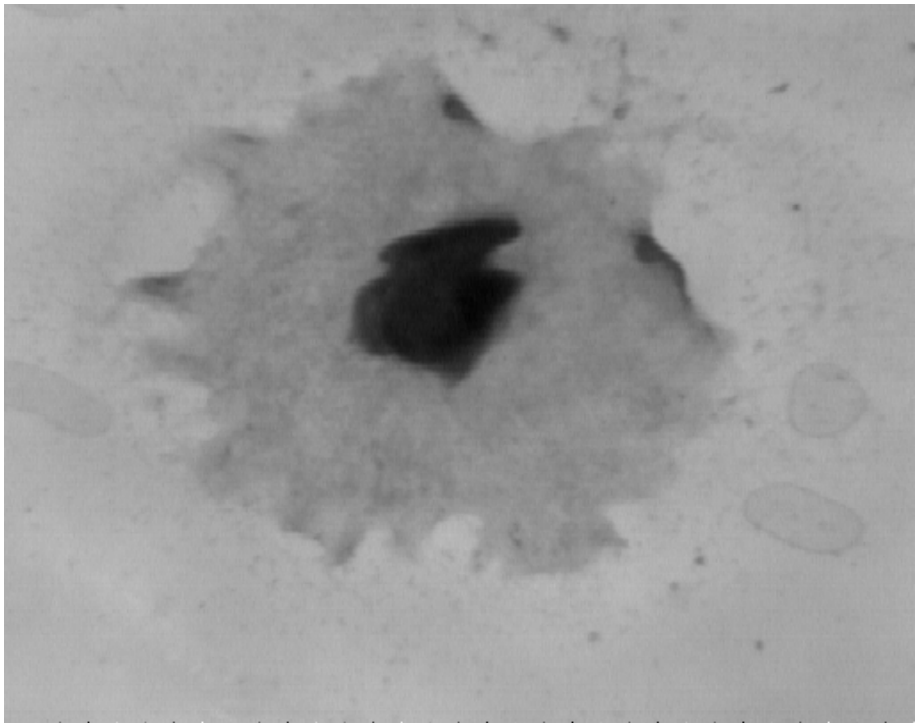


FIG. 4.7 - : Image de taille 512×512 d'une culture cellulaire d'un tissu gingivale

Les données : considérons l'image 4.7, qui est la photographie d'une culture cellulaire après quelques jours de croissance. L'oeil perçoit nettement les trois tissus différents et l'on pourrait facilement tracer à la main les deux contours séparant les trois classes. La question qui se pose est la suivante : est-ce-qu'une procédure de segmentation non supervisée permet de séparer les trois tissus de manière à pouvoir ensuite estimer automatiquement les différentes surfaces ?

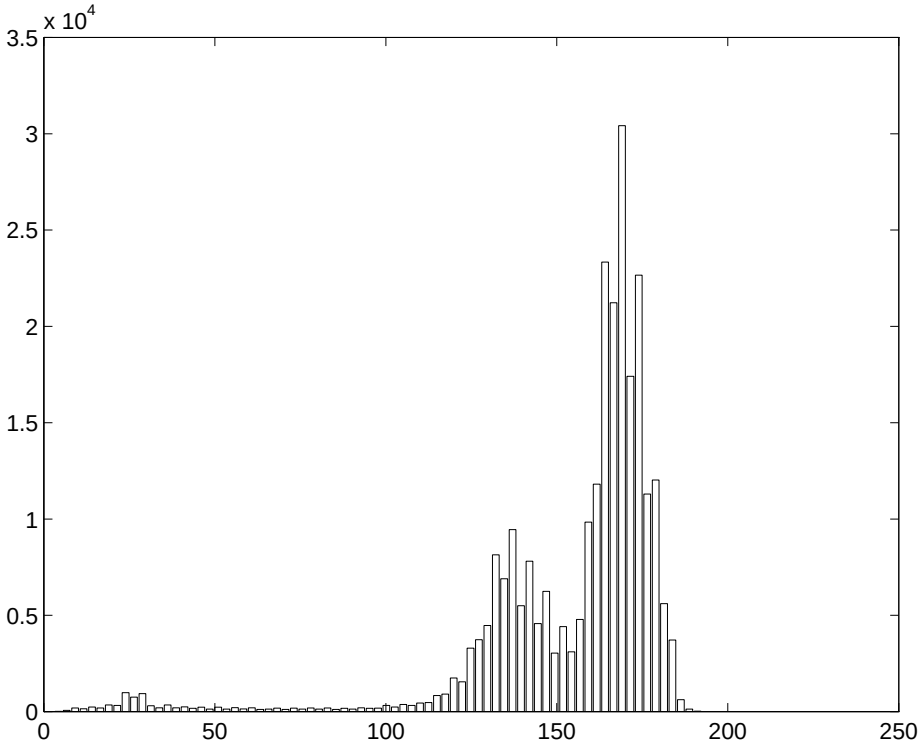


FIG. 4.8 - : Histogramme décrivant la répartition des niveaux de gris dans l'image

4.3.1 Segmentation des images

Étude de l'histogramme : en considérant l'histogramme (Figure 4.8), il semble que la distribution des niveaux de gris soit raisonnablement bien approximée par un mélange de trois gaussiennes. Si l'on note p_1, p_2 et p_3 les proportions du mélange en prenant les gaussiennes de gauche à droite, on peut remarquer que

$$p_1 < p_2 < p_3.$$

Les variances des différentes composantes du mélange paraissent sensiblement égales.

Avant d'envisager un traitement automatique, on peut se demander si chaque gaussienne est représentative d'un type de tissu. Pour répondre à cette question, il

suffit de segmenter "manuellement" l'image en repérant sur l'histogramme les niveaux de gris "frontières" entre les trois gaussiennes. En choisissant 75 et 150 comme points charnières, on obtient la figure 4.9. Cette dernière figure sépare de manière assez nette les trois classes détectées à l'oeil, et l'on peut en déduire que les trois gaussiennes visibles sur l'histogramme correspondent grossièrement à l'explant, au voile et au substrat (en prenant les gaussiennes de gauche à droite). En choisissant 75 et 160 comme points charnières, on peut observer qu'on obtient une segmentation totalement inexploitable pour la détection des trois tissus (Figure 4.10). Ainsi, une petite erreur dans l'estimation des limites des classes produit des résultats catastrophiques.

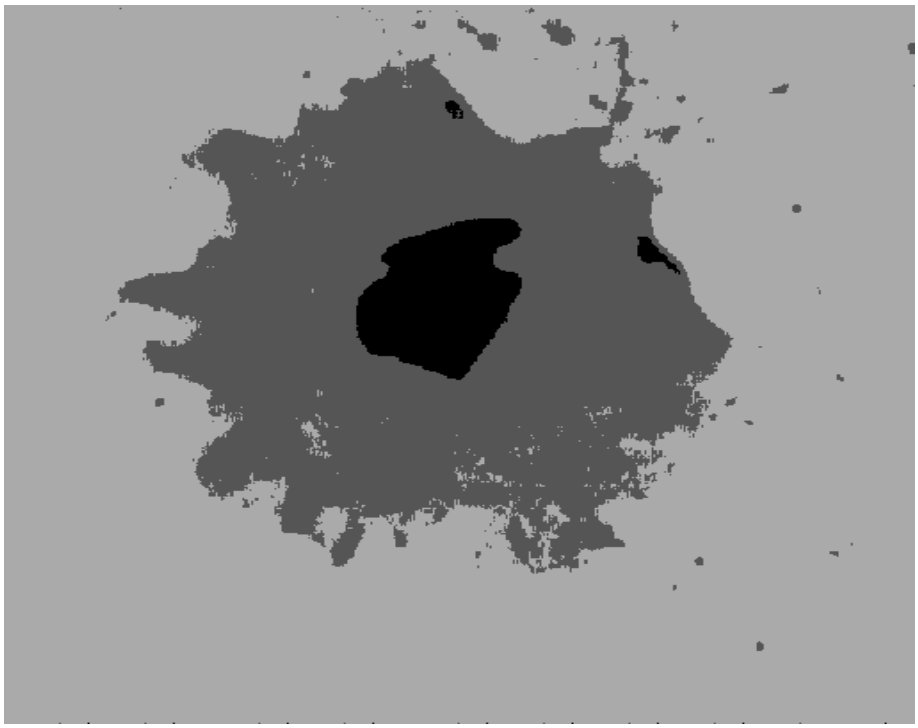


FIG. 4.9 - : Segmentation supervisée linéaire en trois classes : bonne séparation des tissus

Segmentation par l'algorithme EM : comme algorithme de segmentation automatique permettant de détecter automatiquement les niveaux de gris charnières, l'algorithme EM semble *a priori* bien adapté. Pourtant, avec différentes initialisations au hasard, on obtient invariablement le résultat exhibé par la figure 4.11 qui ne met en évidence que deux classes et ne détecte pas l'explant. Cette incapacité à trouver trois classes peut s'expliquer par un défaut bien connu de l'algorithme EM : ce dernier converge vers un maximum local. Le résultat fourni par EM dépend fortement des conditions initiales. On peut donc supposer que dans cet exemple, que l'image seg-

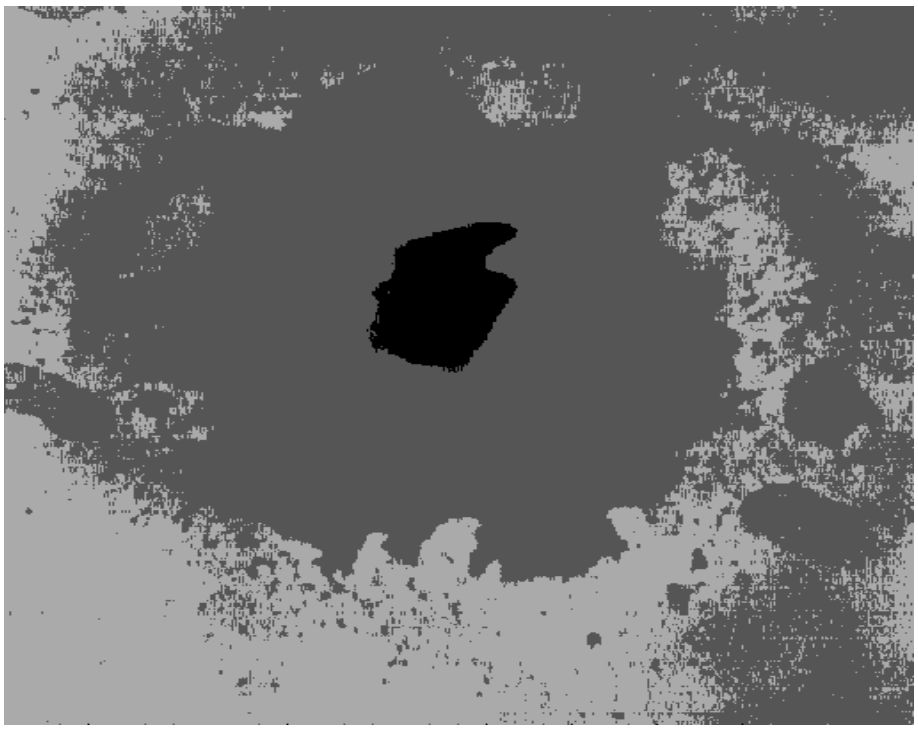


FIG. 4.10 - : Segmentation supervisée linéaire en trois classes: mauvaise séparation des tissus

mentée en deux classes est un maximum local de la vraisemblance qui est atteint en partant de nombreuses configurations initiales différentes. Vu l'apparence de l'histogramme, on peut penser qu'il existe théoriquement des initialisations, qui aboutissent à un résultat proche de celui obtenu par segmentation manuelle. Pour étayer cette hypothèse de travail, on peut initialiser l'algorithme EM en partant d'une configuration initiale proche du résultat souhaitée et observer si la segmentation obtenue se est similaire à celle trouvée "manuellement". Ainsi, la figure 4.12 est obtenue avec l'algorithme EM à partir de l'algorithme EM initialisé avec une matrice de classification spécifiant que :

- la première classe contient les pixels dont le niveau de gris est inférieur à 50,
- la deuxième ceux dont le niveau de gris est compris entre 50 et 150,
- et la dernière ceux avec un niveau de gris supérieur à 150.

On peut constater que la partition résultant de cette initialisation est effectivement très proche de la partition manuelle, ce qui confirme que l'algorithme EM est capable de fournir une partition intéressante pour notre problème, mais reste souvent bloqué sur des maxima locaux sans intérêt.

Segmentation par NEM et NCEM : l'utilisation de NEM et NCEM pour résoudre ce problème de segmentation possède deux avantages :

- d'une part, ces algorithmes donnent régulièrement une partition en trois classes qui correspondent aux trois tissus que l'on cherche à détecter,
- d'autre part, les partitions obtenues sont spatialement plus homogènes.

En d'autres termes les algorithmes NEM et NCEM donnent des résultats plus réguliers et permettent d'obtenir des images segmentées moins bruitées. Ces remarques s'appliquent, malheureusement, seulement pour certaines valeurs du paramètre de pénalisation. Comme le montre la figure 4.13, le paramètre β influence fortement le résultat produit par l'algorithme NEM. On constate que pour une valeur de β comprise entre 0.5 et 1.5 l'image segmentée obtenue doit permettre d'estimer les surfaces recouvertes par les trois tissus. Par contre, si le paramètre de pénalisation spatiale est supérieur à 1.5, les partitions générées sont inexploitables.

Sur cet exemple, l'algorithme NCEM donne des résultats très similaires à l'algorithme NEM (Figure 4.14). On peut même constater, que pour $\beta = 0$ (dans ce cas NCEM est équivalent à CEM), les partitions produites sont intéressantes. La seule différence notable entre les deux algorithmes est leur vitesse d'exécution. En effet NCEM, comme nous l'avions déjà constaté dans les simulations du chapitre 3, converge toujours en moins d'itérations.

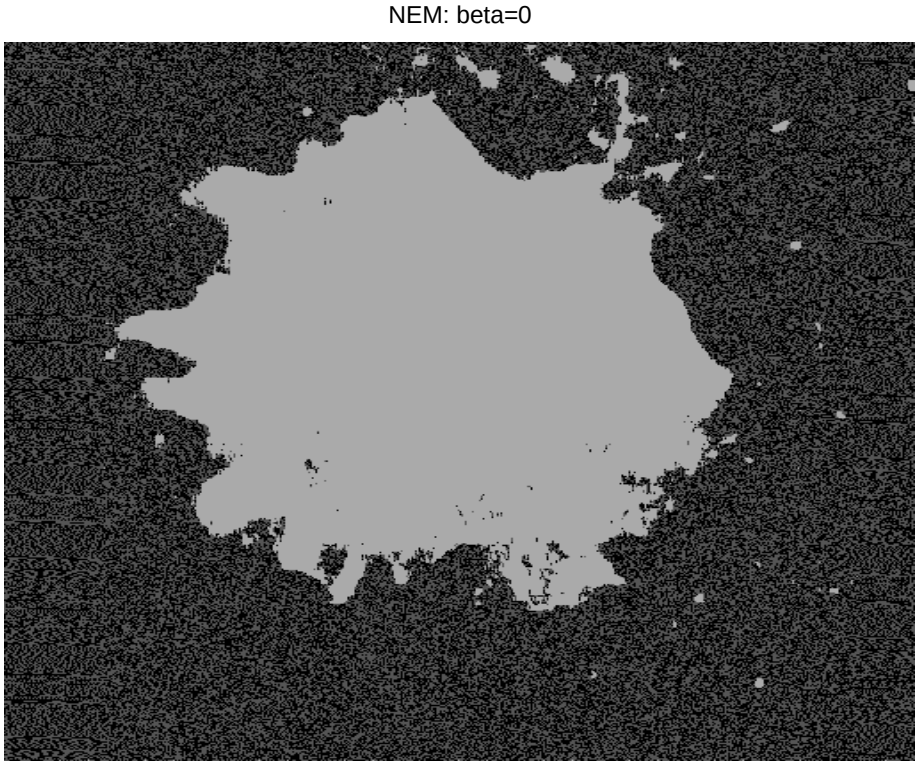


FIG. 4.11 - : Segmentation non supervisée en trois classes par l'algorithme EM initialisé aléatoirement

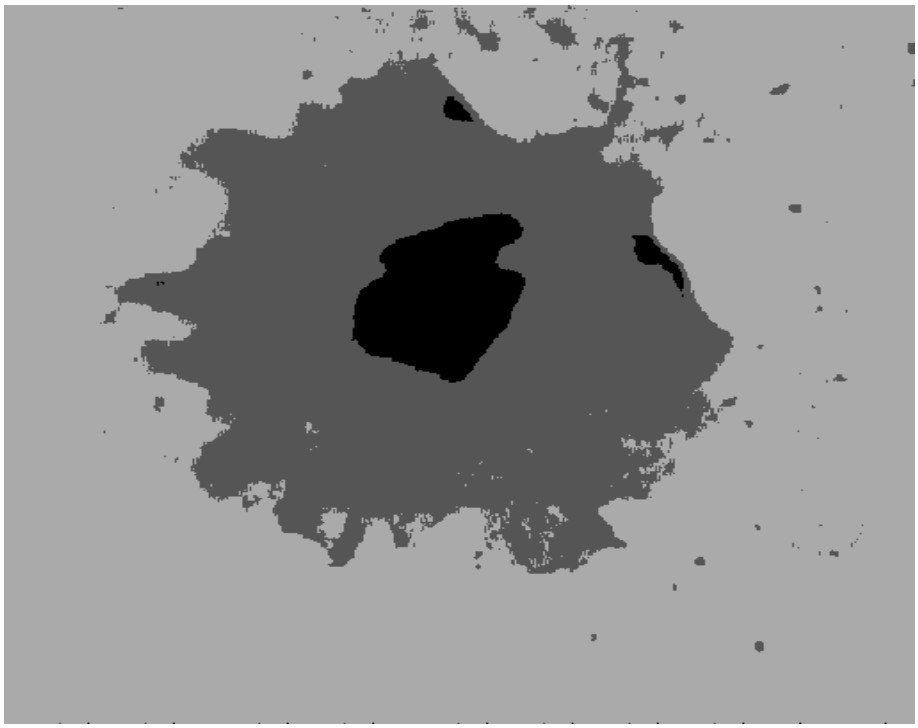


FIG. 4.12 - : Segmentation non supervisée en trois classes par l'algorithme EM initialisé à partir d'un partage de l'histogramme

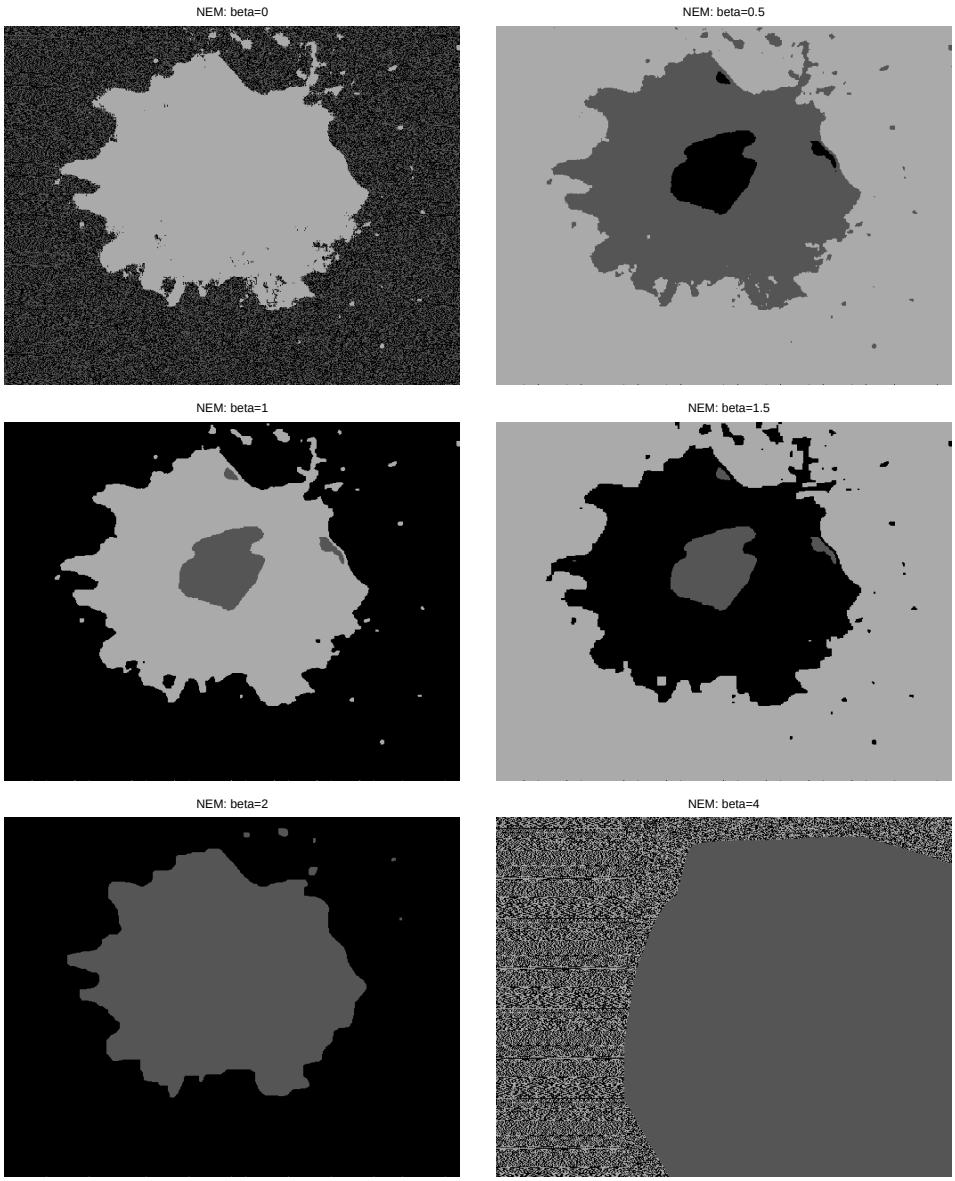


FIG. 4.13 - : Segmentation de l'image par NEM avec différentes valeurs de β

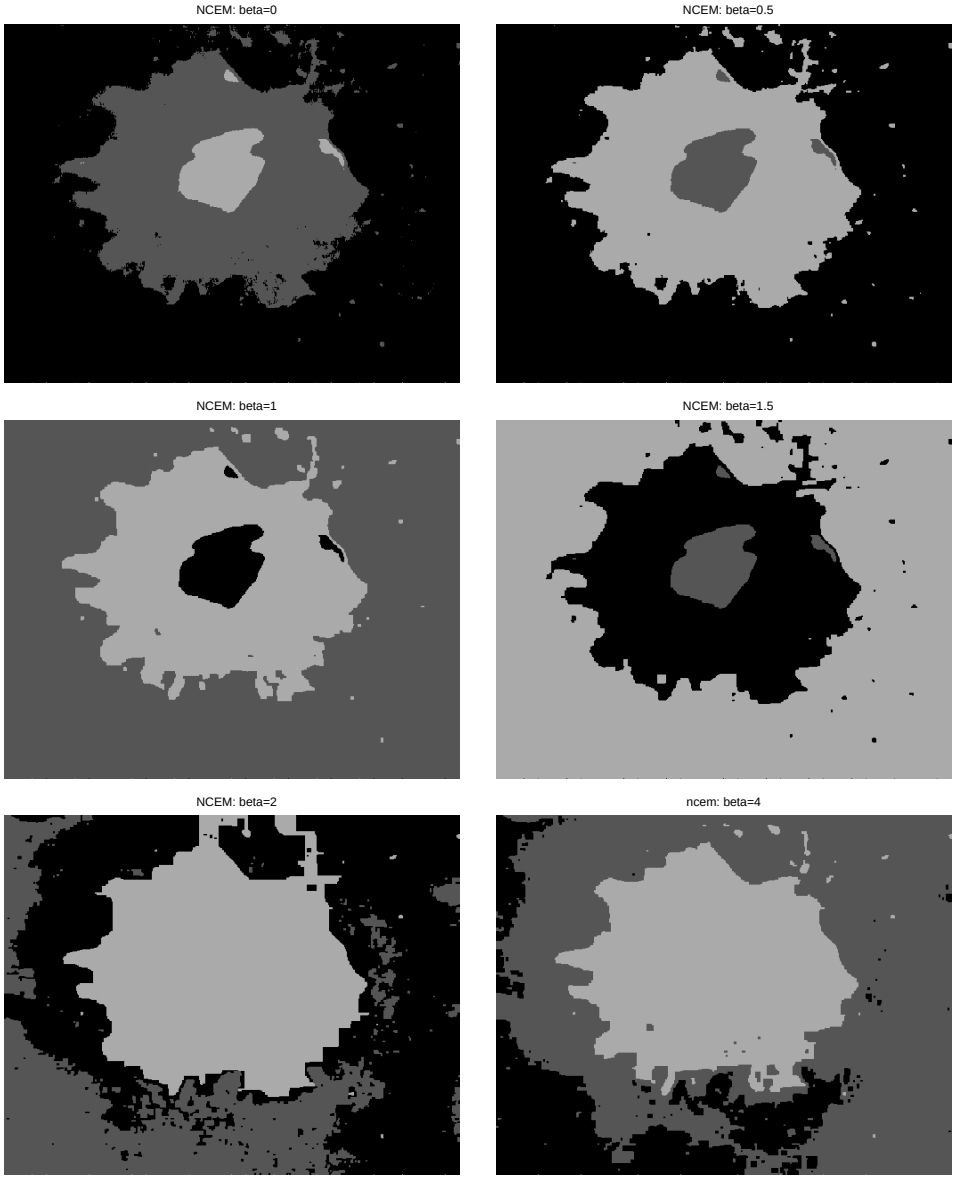


FIG. 4.14 - : Segmentation de l'image par NCEM avec différentes valeurs de β

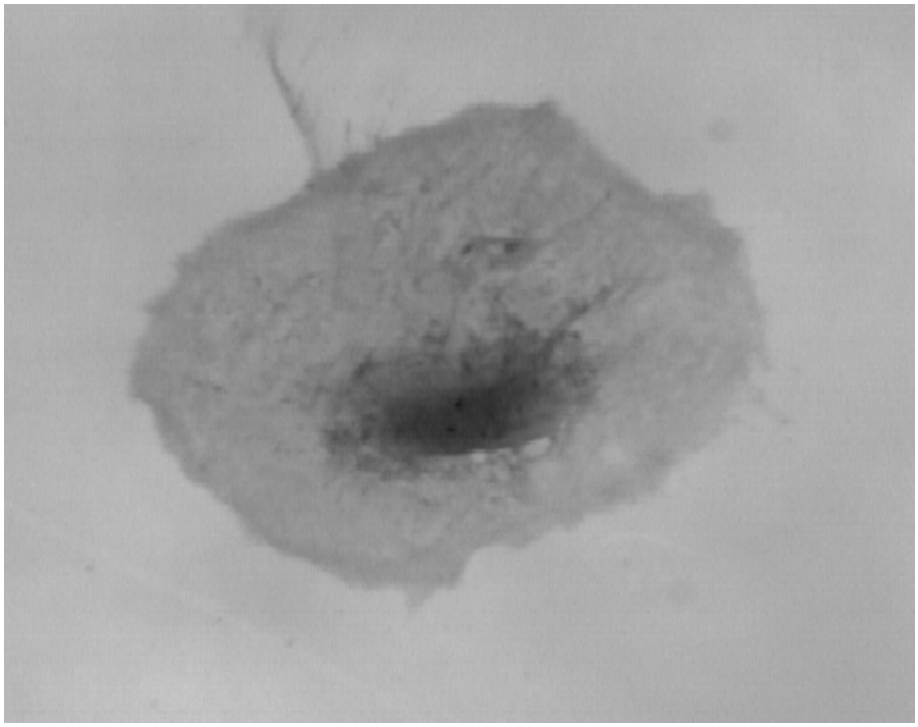


FIG. 4.15 - : Culture cellulaire d'un tissu gingivale

Étude d'une seconde image : dans un souci de validation des observations réalisée sur l'image précédente, nous avons testé l'algorithme NEM avec différentes valeurs de β sur une seconde image de culture cellulaire. La figure 4.16 montre que l'algorithme NEM se comporte conformément aux observations déjà réalisées sur l'image précédente :

- l'algorithme EM (NEM avec $\beta = 0$) initialisé aléatoirement ne permet pas de détecter les trois classes,
- l'algorithme NEM avec $\beta = 1$ donne des résultats satisfaisants,
- lorsque β est trop grand, l'algorithme NEM produit des résultats inexploitables.

4.3.2 Conclusion

Dans cette section, des méthodes de segmentation d'image ont été testées pour mettre en évidence la présence de trois tissus dans une culture cellulaire.

Deux images différentes ont servi de base aux tests et il serait naturellement nécessaire d'essayer la même approche sur un plus grand nombre d'images.

Il semble que NEM et NCEM soient utilisables pour résoudre le problème d'estimation des surfaces recouvertes par les tissus biologiques, en prenant une valeur de β égale à 1.

Notons qu'il serait aussi trés intéressant de tester le comportement de l'algorithme GEM sur ces images réelles.

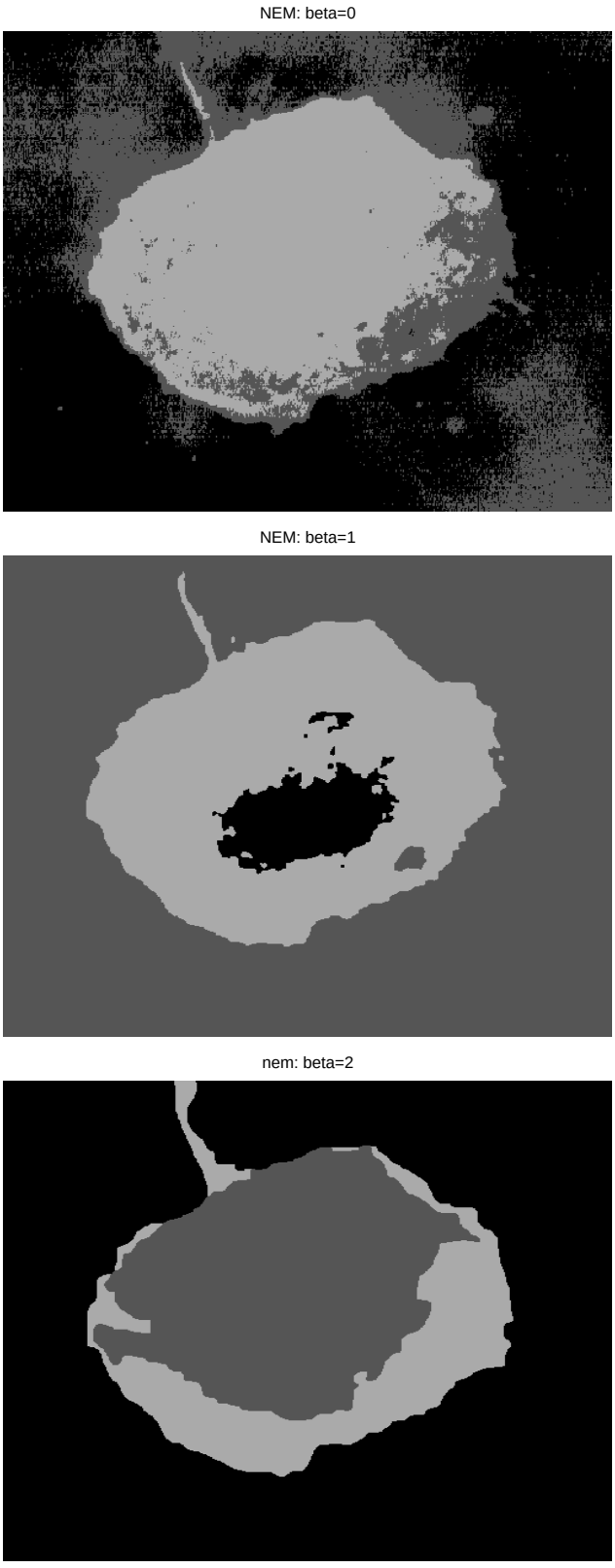


FIG. 4.16 - : Segmentation d'une image biologique par NEM avec différentes valeurs de β

Conclusion

Dans ce mémoire, des solutions nouvelles aux problèmes

- de la cartographie associative,
- de la classification automatique de données spatiales,

ont été proposées. Ces deux problémes sont issus de domaines scientifiques différents. Sous le nom de cartographie associative sont regroupées un ensemble de méthodes, initialement d'inspiration biologique, qui constituent une partie importante des techniques d'apprentissage non supervisées développées par les connexionistes. La classification automatique de données spatiales constitue une problématique qui se pose aussi bien en écologie, géologie, sociologie, qu'en traitement d'image. De nombreuses techniques inventées par des chercheurs d'horizons différents ont été développées, en parallèle, pour traiter les données spatiales.

Un des points communs de ces deux problèmes réside dans le fait qu'ils peuvent être envisagés dans le cadre unificateur de la classification automatique. Ce mémoire a abordé ces problèmes sous l'angle particulier de l'approche probabiliste en classification automatique.

Le cadre général de l'approche par modèle de mélange gaussien de la classification automatique a été présenté dans le premier chapitre. Dans ce contexte, le processus de classification repose sur trois concepts :

- un modèle statistique, qui est le modèle de mélange ;
- un critère, fonction des paramètres du modèle ;
- un algorithme pour optimiser le critère.

Les critères, qui ont été considérés, sont la vraisemblance et la vraisemblance classifiante. Nous nous sommes surtout intéressés aux différentes versions de l'algorithme EM qui permettent de maximiser ces critères et de trouver une partition. Dans ce mémoire, deux démarches différentes sont à l'origine de tous les algorithmes proposés :

- l'algorithme EM a directement été modifié pour prendre en compte les spécificités des problèmes posés,

- de nouveaux critères ont été proposés et certaines versions de l'algorithme EM adaptées pour optimiser ces critères.

Les diverses adaptations de l'algorithme EM ont surtout tiré leur inspiration du court article de Hathaway (1986), qui replace l'algorithme EM dans un cadre d'optimisation numérique classique, et des articles de Celeux et Diebolt (1986) et Celeux et Govaert (1992), qui proposent des variations de l'algorithme EM.

Des algorithmes de cartographie associative originaux ont été présentés dans le second chapitre. Ces algorithmes ont été le résultat de la rencontre des méthodes de classification automatique basées sur les modèles de mélange, et de la problématique de la cartographie associative. La cartographie associative a été abordée comme un problème de classification automatique avec des contraintes de voisinage entre les classes. Deux solutions distinctes ont été proposées pour intégrer ces contraintes dans des procédures de classification automatique probabiliste :

- des versions modifiées de l'algorithme CEM ont été présentées. Ces versions imposent les contraintes de voisinage lors d'une étape intermédiaire (algorithmes TPEM et STPEM) ;

- les contraintes de voisinage ont été exprimées sous la forme d'un terme de pénalisation qu'il est possible d'ajouter à la vraisemblance ou bien à la vraisemblance classifiante. Nous avons transformé les algorithmes EM et CEM de manière à optimiser les critères pénalisés (algorithme CEMP).

Les algorithmes de cartographie associative sont surtout employés pour faire de la représentation. D'après les exemples utilisés dans le deuxième chapitre, il nous semble qu'il est plus avantageux d'utiliser ce genre d'approche pour représenter des données étiquetées que pour détecter un structure de classe. Dans le cadre d'une utilisation en classification automatique, les algorithmes de cartographie associative offrent la possibilité de prendre en compte certains *a priori* lorsqu'ils existent. Les trois algorithmes de cartographie associatives, TPEM, STPEM et CEMP, donnent des résultats semblables à ceux de l'algorithmes de Kohonen. Les algorithmes TPEM et STPEM peuvent être considérés comme des versions de CEM qui incluent une phase d'initialisation. Cette phase d'initialisation semble réduire la dépendance aux conditions initiales, par rapport à l'algorithme CEM. Lorsque les algorithmes TPEM et STPEM sont utilisables (ensemble de données qui ne change pas au cours du temps), ils possèdent quelques avantages par rapport à l'algorithme de Kohonen :

- rapidité ;

- existence d'un critère.

D'autre part ces versions modifiés de l'algorithme EM, permettent d'utiliser la paramétrisation des matrices de variance covariance (cf chapitre 1). L'intérêt de cette paramétrisation, qui possède certains avantages en classification (Ambroise et Govaert 1995, Celeux et Govaert 1995), reste à explorer pour la représentation.

La classification de données spatiales a aussi été considérée comme un problème de classification avec contraintes de voisinage. Dans ce contexte les contraintes portent sur les individus et non plus sur les classes : on veut intégrer dans le processus de classification le fait que deux individus voisins ont *a priori* plus de chance d'appartenir à la même classe que deux individus qui ne sont pas voisins. Comme dans le cas de la cartographie associative, nous avons exprimé les contraintes de voisinages sous la forme d'un terme de pénalisation et avons modifié les algorithmes EM et CEM pour optimiser de nouveaux critères pénalisés (algorithmes NEM et NCEM). Ces algorithmes peuvent être utilisés pour trouver une partition d'un ensemble d'individus localisés géographiquement, ce qui englobe la problématique de la segmentation d'image. Un parallèle entre les méthodes développées dans ce mémoire et les techniques markoviennes de segmentation bayésienne non supervisée d'image a été établi. Notons que l'algorithme NEM produit une partition floue, ce qui est assez rare en segmentation statistique d'image et pourrait être avantageux pour certaines applications. Les simulations numériques ont mis en évidence que les algorithmes NEM et NCEM se comportent bien par rapport aux algorithmes classiques utilisés en segmentation statistique, et surtout possèdent l'avantage de produire une solution rapidement.

Les perspectives de recherches prolongeant le travail exposé sont variées.

Une direction de recherche consisterait à tester l'intérêt du mélange de la cartographie associative et de la classification de données spatiales. En effet la forme des algorithmes proposée dans ce mémoire autorise à mixer les deux approches.

Notons aussi que les algorithmes de classification spatiale ont été définis pour un terme de pénalisation particulier, mais qu'il serait possible d'étendre la même approche à de nombreux termes pénalisants. Dans cette optique, il serait intéressant de déterminer quelles propriétés doit posséder le terme de pénalité pour que l'approche proposée dans ce mémoire reste valide.

Remarquons enfin, que toutes les approches envisagées dans ce mémoire étaient relatives à des approches non supervisées. Dans les applications réelles, il est parfois possible de disposer d'un certain nombre d'individus étiquetés (dont on connaît la classe d'origine), et la démarche classique consiste alors à aborder le problème comme un problème de classement. Une démarche originale et facile à mettre en oeuvre avec des algorithmes du type EM, consiste à prendre en compte toutes les données, étiquetées et non étiquetées, en même temps. Ce type d'approche, entre le supervisé et le non supervisé est particulièrement approprié lorsque les données étiquetées sont rares (Celeux 1992) et mériterait d'être explorée plus avant avec les nouvelles versions de EM proposées dans ce mémoire.

Bibliographie

Akaike, H. (1978). 'A new look at the bayes procedure'. *Biometrika* **65**, 53–59.

Amari, S. (1980). 'Topographic organization of nerve fields'. *Bulletin of mathematical biology* **42**, 339–364.

Ambroise, C. et G. Govaert (1995). Self-organization for gaussian parsimonious clustering. In 'Proceeding of ICANN1995'. Vol. 1. pp. 425–430.

Ambroise, C. et G. Govaert (1996). Analyzing dissimilarity matrices using kohonen maps. In 'Proceeding of IFCS96'. Vol. 1. pp. 425–430.

Ambroise, C. et G. Govaert (à paraître). 'Constrained clustering and kohonen self-organizing maps'. *Journal of Classification*.

Ambroise, C. et T. Trautmann (1994). Additional fusion methods. Rapport de recherche CNRS/EMS/TN/004/08-94. EMS ESPRIT P-6757.

Ambroise, C., M. Dang et G. Govaert (To appear). Clustering of spatial data by the em algorithm. In 'Proceeding of geoENV 1996'.

Angeniol, B., G. De la Croix-Vaubois et L. J.Y. (1988). 'Self-organizing feature map and the tsp'. *Neural Networks* **1**, 289–293.

Banfield, J. et A. Raftery (1993). 'Model-based gaussian and non gaussian clustering'. *Biometrics* **49**, 803–821.

Bauer, H. et K. Pawelzik (1992). 'Quantifying the neighborhood preservation of the self-organizing feature map'. *IEEE Transactions on neural networks* **3**, 570–579.

Bayes, T. (1763). 'An essay towards solving a problem in the doctrine of chances'. *Phil. Trans. Roy. Soc.* **53**, 370–418.

Bensmail, H., G. Celeux, A. Raftery et C. Robert (1995). Inference in model-based cluster analysis. Rapport de recherche. Université de Washington (Seattle).

Benveniste, A., M. Metivier et P. Priouret (1987). *Algoritmes adaptatifs et approximations stochastiques*. Masson.

Benzécri, J. (1973). *L'analyse des données*. Dunod.

Berry, B. (1966). Essay on commodity flows and the spatial structure of the indian economy. Research paper 111. University of Chicago, departement of geography.

Bertrandias, J. (1970). *Mathématique pour l'informatique : Analyse fonctionnelle*. Armand Colin.

Besag, J. (1974). 'Spatial interaction and the statistical analysis of lattice systems'. *Journal of the Royal Statistical Society* **35**, 192–236.

Besag, J. (1986). 'Spatial analysis of dirty pictures'. *Journal of the Royal Statistical Society* **48**, 259–302.

Bezdek, J. (1974). 'Numerical taxonomy with fuzzy sets'. *Journal of mathematical biology* **1**, 57–71.

Billoire, A. (1992). Simulations de monte carlo en physique théorique. In N. Bouleau and D. Talay (Eds.). 'Probabilités numériques'. INRIA. Paris. pp. 145–162.

Bock, H. (1989). Probabilistic aspects in cluster analysis. In O. Opitz (Ed.). 'Proc. 13th Conference of the Gesellschaft fur Klassification'. pp. 12–44.

Bouton, C. et G. Pages (1992). Auto organisation de l'algorithme de kohonen en dimension 1. In 'Congrès satellite du congrès européen de maths Paris'.

Caillol, H., A. Hillion et W. Pieczinsky (1993). 'Fuzzy random fields and unsupervised image segmentation'. *IEEE transactions on geoscience and remote sensing* **31**(4), 801–810.

Celeux, G. (1992). Modèles probabilistes en classification. In J. Droesbeke, B. Fichet and P. Tassi (Eds.). 'Modèles pour l'analyse de données multidimensionnelles'. Economica. pp. 165–211.

Celeux, G. et G. Govaert (1992). 'A classification em algorithm for clustering and two stochastic versions'. *Computational statistics and data analysis* **14**, 315–332.

Celeux, G. et G. Govaert (1994). Fuzzy clustering and mixture models. In 'Proceedings of CompStat'.

Celeux, G. et G. Govaert (1995). 'Gaussian parsimonious clustering models'. *Pattern Recognition* **28**, 781–793.

Celeux, G. et J. Diebolt (1985). 'The SEM algorithm: a probabilistic teacher algorithm derived from the EM algorithm for the mixture problem'. *Comput. Stat. Quater.* **2**(1), 73–92.

Celeux, G. et J. Diebolt (1986). Comportement asymptotique d'un algorithme d'apprentissage probabiliste pour les mélanges de lois de probabilité. Rapport de recherche 563. INRIA.

Celeux, G. et J. Diebolt (1990). 'Une version de type recuit simulé de l'algorithme em'. *C. R. Acad. Sci., Serie 1* **310**, 119–124.

Chalmond, B. (1989). 'An iterative gibbsian technique for reconstruction of m-ary images'. *Pattern Recognition* **22**(6), 747–761.

Cliff, A. et J. Ord (1981). *Spatial Processes*. Pion.

Cottrell, M. et J. Fort (1987). 'Etude d'un processus d'auto-organisation'. *Annales de l'institut Henri Poincaré* **23**, 1–20.

Dang, M. (1994). Classification automatique et modèle de mélange: Détermination d'un modèle optimal. Rapport de dea. Université de Technologie de Compiègne.

Davalo, E. et P. Naim (1992). *Des réseaux de neurones*. Eyrolles.

Day, N. (1969). 'Estimating the components of a mixture of normal distributions'. *Biometrika* **56**, 463–474.

Dempster, A., N. Laird et D. Rubin (1977). 'Maximum likelihood from incomplete data via the em algorithm'. *Journal of the Royal Statistical Society* **39**, 1–38.

Diday, E. (1971). 'La méthode des nuées dynamiques'. *Revue de Statistiques appliquées* **19**(2), 19–34.

Duda et Hart (1973). *Pattern Recognition and Scene Analysis*. John Whiley and son.

Durbin, R. et D. Willshaw (1987). 'An analogue approach to the tsp using an elastic net approach'. *Nature* **326**, 689–91.

Durbin, R. et G. Mitchinson (1990). 'A dimension reduction framework for understanding cortical maps'. *Nature* **343**, 644–647.

Edwards, A. et L. Cavalli-Sforza (1965). 'A method for cluster analysis'. *Biometrics* **21**, 362–376.

Erwin, E., K. Obermayer et K. Schulten (1992). 'Self-organizing maps: ordering, convergence properties and energy functions'. *Biological Cybernetics* **67**, 47–55.

Finch, A. et J. Austin (1994). A neural network for dimension reduction and its application to image segmentation. In 'Proceeding of ICANN1994'. Vol. 2. pp. 1141–1144.

Fisher, R. (1912). 'On absolute criterion for fitting frequency curves'. *Messeng. Math.* **41**, 155–160.

Fisher, W. (1958). 'On grouping for maximum homogeneity'. *Journal of american statistical association* **53**, 789–798.

Fort, J. (1988). 'Solving a combinatorial problem via self-organizing process : an application of the kohonen algorithm to the tsp'. *Biological Cybernetics* **64**, 33–40.

Fritzke, B. (1993). Kohonen features maps and growing cell structures : A performance comparison. In J. C. C.L. Giles, S.J. Harson (Ed.). 'Advances in neural information processing systems'. Vol. 5. Kaufmann. pp. 123–130.

Geman, S. et D. Geman (1984). Stochastic relaxation, gibbs distributions, and the baysian restoration of images. In 'IEE Transactions on Pattern Analysis and Machine Intelligence'. Vol. PAMI-6. pp. 721–741.

Good, I. et R. Gaskins (1971). 'Nonparametric roughness penalties for probability densities'. *Biometrika* **58**, 255–277.

Gordon, A. (1980). *Classification: Methods for the Exploratory Analysis of Multivariate Data.* Chapman and Hall.

Grossberg, S. (1976*a*). 'Adaptative pattern classification and universal recoding: I. parallel development and coding of neural feature detectors'. *Biological Cybernetics* **23**, 121–134.

Grossberg, S. (1976*b*). 'Adaptative pattern classification and universal recoding: Ii. feedback, expectation, olfaction, illusions'. *Biological Cybernetics* **23**, 187–202.

Hartigan, J. (1982). Classification. In S. Kotz and N. Johnson (Eds.). 'Encyclopedia of Statistical Sciences'. Vol. 2. Wiley Interscience. New York. pp. 1–10.

Hathaway, R. (1986). 'Another interpretation of the em algorithm for mixture distributions'. *Journal of Statistics & Probability Letters* **4**, 53–56.

Hertz, J., A. Krogh et R. Palmer (1991). *Introduction to the theory of neural computation.* Addison-Wesley.

Jain, A. et F. Farrokhnia (1991). 'Unsupervised texture segmentation using gabor filters'. *Pattern Recognition* **24**(12), 1167–1186.

KENT, J. ET K. MARDIA (1991). 'Spatial classification using fuzzy membership'. *IEER Trans. Patt. Anal. Machine Intell.* **10**(5), 135–143.

KLEIN, R. ET R. DUBES (1989). 'Experiments in projection and clustering by simulated annealing'. *Pattern Recognition* **22**, 213–220.

KOHONEN, T. (1982). 'Self organized formation of topological correct feature maps'. *Biological Cybernetics* **43**, 59–69.

KOHONEN, T. (1984). *Self organization and associative memory 2nd ed.* Springer Verlag.

KOHONEN, T. (1988*a*). 'An introduction to neural computing'. *Neural Networks* **1/1**, 3–16.

KOHONEN, T. (1988*b*). 'Statistical pattern recognition with neural networks'. *Neural Networks.*

KOHONEN, T. (1991). Self-organizing maps: optimisation approaches. In T.Kohonen, K. Makisara, O. Simula and J. Kangas (Eds.). 'Artificial neural networks'. Vol. 2. North Holland.

KOHONEN, T. (1993). Things you haven't heard about the self-organizing map. In 'Proceedings of 1993 IEEE International Conference on Neural Networks'. pp. 1147–1156.

KRUSKAL, J. (1964). 'Non metric multidimensional scaling: A numerical method'. *Psychometrika* **29**(2), 115–129.

LAKSHMANAN, S. ET H. DERIN (1989). 'Simultaneous parameter estimation and segmentation of gibbs random fields using simulated annealing'. *IEEE transactions on pattern analysis machine intelligence* **11**(8), 799–813.

LEBART, L. (1978). 'Programme d'agrégation avec contraintes (c.a.h. contiguité)'. *Cahier de l'analyse des données* **3**, 275–287.

LEGENDRE, P. (1987). 'Constrained clustering'. *Developpments in numerical ecology. NATO ASI Series* **G 14**, 289–307.

LLOYD, S. (1957). Least squares quantization in pcm's. Rapport technique. Bell Telephone Laboratories Paper, Murray Hill.

LOWE, C. ET G. TIPPING (1995). A novel neural network technique for exploratory data analysis. In 'Proceeding of ICANN1995'. Vol. 1. pp. 434–440.

LO, Z. ET B. BAVARIAN (1991). 'On the rate of convergence in topology preserving neural networks'. *Biological Cybernetics* **63**, 55–63.

Luttrell, S. (1990). 'Derivation of a class of training algorithms'. *IEEE transactions on Neural Networks.*

Macqueen, J. (1967). Some methods for classification and analysis of multivariate observations. In 'Statistics and Probability: 5th Berkeley Symposium'. Vol. 1. University of California Press. pp. 281–297.

Marroquin, J. et F. Girosi (1993). Least squares quantization in pcm's. A.I. Memo 1390. Massachusetts institute of technology artificial intelligence laboratory.

Marroquin, J., S. Mitter et T. Poggio (1987). 'Probabilistic solution of illed posed problems in computational vision'. *Journal of the american statistical association* **82**, 76–89.

Masson, P. et W. Pieczinsky (1993). 'Sem algorithm and unsupervised statistical segmentation of satellite images'. *IEEE transactions on geoscience and remote sensing* **31**(3), 618–633.

Oliver, M. et R. Webster (1989). 'A geostatistical basis for spatial weighting in multivariate classification'. *Mathematical Geology* **21**, 15–35.

Openshaw, S. (1977). 'A geographical solution to scale and aggregation problems in region-building, partitionning and spatial modelling'. *Transactions of the institute of british geographers* **2**, 459–472.

Ord, A. (1982). Spatial processes. In S. Kotz and N. Johnson (Eds.). 'Encyclopedia of Statistical Sciences'. Vol. 7. Wiley Interscience. New York. pp. 575–580.

Pearson, K. (1894). 'Contribution to the mathematical theory of evolution'. *Philos. Trans A* **185**, 71–110.

Pieczinsky, W. et J. Cahen (1994). 'Champs de markov flous cachés et segmentation d'images'. *Revue de statistique appliquée* **42**(3), 13–31.

Redner, R. et H. Walker (1984). 'Mixture densities, maximum likelihhod and the em algorithm'. *SIAM Review* **26**(2), 195–239.

Reif, F. (1972). *'Berkeley: Cours de Physique.* Vol. 5. Armand Colin.

Ripley, B. (1981). *Spatial Statistics.* John Whiley and son.

Ripley, B. (1982). Spatial data analysis. In S. Kotz and N. Johnson (Eds.). 'Encyclopedia of Statistical Sciences'. Vol. 7. Wiley Interscience. New York. pp. 570–573.

Ritter, H. et K. Schulten (1988). Kohonen's self organizing map: exploring their computational capabilities. In 'Proceedings of ICNN'. Vol. 1. pp. 109–116.

Ritter, H. et K. Shulten (1986). 'On the sensory state of kohonen's self organizing sensory mapping'. *Biological Cybernetics* **54**, 99–106.

Robert, C. (1992). *L'analyse statistique bayésienne*. Economica.

Robert, C. (1996). Mixture of distribution : Inference and estimation. Rapport de recherche. Université de Rouen, CREST, INSEE.

Rose, K., E. Gurewitz et G. Fox (1990). 'A deterministic annealing approach to clustering'. *Pattern Recognition Letters* **11**, 589–594.

Ruspini, E. (1969). 'A new approach to clustering'. *Information and control* **15**, 22–32.

Sammon, J. (1969). 'A non linear mapping for data structure analysis'. *IEEE Transactions on Computers* **18**(5), 401–409.

Schroeder, A. (1976). 'Analyse d'un mélange de distributions de probabilités se même type'. *Revue de statistique appliquée* **24**(1), 39–62.

Scott, A. et M. Symons (1971). 'Clustering method based on likelihood ratio criteria'. *Biometrics* **27**, 387–397.

Selim, S. et M. Ismail (1984). 'K-means type algorithms: a generalized convergence theorem and characterization of local optimality'. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Siedlecki, W., K. Siedlecka et J. Skaanski (1988). 'An overview of mapping techniques'. *Pattern Recognition* **21**(5), 411–429.

Silverman, B. (1982). 'On the estimation of a p.d.f. by the maximum penalized likelihood method'. *Ann. Statist.* **10**, 795–810.

Simic, P. (1990). 'Statistical mechanics as the underlying theory of elastic and neural optimization'. *NETWORK: Comp. Neural. Syst.* **1**, 338–353.

Sokal, R. et P. Sneath (1963). *'Principles of numerical taxonomy*. Freeman.

Strauss, D. (1977). 'Clustering on coloured lattices'. *Journal of appl. probab.* **14**, 135–143.

Sutcliffe, J. (1994). On the logical necessity and priority of a monothetic conception of class, and on the consequent inadequacy of polythetic accounts of category and categorization. In Diday (Ed.). 'New approaches in Classification and data analysis'. Springer-Verlag. Berlin. pp. 55–63.

Symons, M. (1981). 'Clustering criteria and multivariate normal mixtures'. *Biometrics* **37**, 35–43.

Tanner, M. et W. Wong (1987). 'The calculation of a posterior distributions by data augmentation'. *Journal of the American Statistical Association* **82**, 528–550.

Tolat, V. (1990). 'An analysis of kohonen's self organizing maps using a system of energy functions'. *Biological Cybernetics* **64**, 155–64.

Tomasini, L. (1993). Apprentissage d'une représentation statistique et topologique d'un environnement. PhD thesis. L'Ecole nationale supérieure de l'aéronautique et de l'espace.

Trautmann, T. (1995). Développement d'un modèle de cartes topologiques auto-organisatrices à architecture dynamique et application au diagnostic. PhD thesis. Université de Technologie de Compiègne.

Ultsch, A. (1990). Kohonen's self organizing maps for exploratory data analysis. In 'ICNN 90'. pp. 305–308.

Ultsch, A. (1992). Self organizing neural networks for visualisation and classification. In 'Proc. Conf. Soc. for information and classification Dortmund'. pp. 1–6.

Ultsch, A. (1993). Self organized feature maps for monitoring and knowledge aquisition of a chemical process. In S. Gielen and B. Kappen (Eds.). 'ICANN 93'. Springer verlag. pp. 864–867.

Von der Malsburg, C. (1973). 'Self organization sensitive cells in the striate cortex'. *Kybernetic* **14**, 85–100.

Wolfe, J. (1970). 'Pattern clustering by multivariate mixture analysis'. *Multivariate Behavioral Research* **5**, 329–350.

Xu, L. et M. Jordan (1995). On convergence properties of the em algorithm for gaussian mixtures. Research paper AI memo 1520. MIT.

Younes, L. (1988). Problèmes d'estimation paramétrique pour des champs de Gibbs markoviens. Application au traitement d'images. PhD thesis. Université de Paris Sud.

Yuille, A. (1990). 'Generalized deformable models, statistical physics and matching problems'. *Neural Computation* **2**, 1–24.

Yuille, A., P. Storloz et J. Utans (1993). 'Statistical physics, mixtures of distributions and the em algorithm'. *Neural Computation* **6**, 334–340.

Zadeh, L. (1965). 'Fuzzy sets'. *Information and control* **8**, 338–353.

Zhao, Z. (1992). Weight distance display of kohonen maps. In 'Proceedings of NeuroNîmes'. pp. 611–620.

Zrehen, S. et F. Blayo (1992). A geometric organization measure for kohonen maps. In 'Proceedings of NeuroNîmes'. pp. 611–620.