Modèles de mélanges topologiques pour la classification de données catégorielles et mixtes

Nicoleta Rogovschi*,Mustapha Lebbah**, Younès Bennani**

*LIPADE, Université Paris Descartes
45 rue des Saints Pères
75270 Paris Cedex 06
France
**LIPN-UMR 7030 Université Paris 13 - CNRS
99, av. J-B Clément - F-93430 Villetaneuse France.
prénom.nom@lipn.univ-paris13.fr

Résumé. Cet article présente une méthode basée sur les cartes auto-organisatrices probabilistes dédiées à la classification non supervisée et la visualisation de données catégorielles et des données mixtes contenant des composantes quantitatives et binaires. Pour chacun de ces types de données, nous proposons un formalisme probabiliste dans lequel les unités de la carte topologique sont représentées par un modèle de mélanges de loi de Bernoulli, dans le cas des données binaires et par un modèle de mélanges de lois de Bernoulli et Gaussienne dans le cas des données mixtes. Dans cette étude, la carte topologique est vue comme un modèle génératif et est revisitée dans un formalisme probabiliste de modèles de mélanges. L'idée de base de ce travail repose sur le principe de la conservation de la structure initiale des données en utilisant le formalisme probabiliste. Les modèles de mélanges proposés ici vérifient ce principe et fournissent des résultats directement interprétables par rapport aux données initiales, qu'elles soient simplement binaires ou mixtes. L'apprentissage consiste alors à estimer les paramètres du modèle en maximisant la vraisemblance des données d'apprentissage. L'algorithme d'apprentissage (PrMTM :Probabilistic Mixed Topological Map) que nous proposons est basé sur l'algorithme EM (Estimation-Maximisation). Nous avons montré que l'algorithme à base de modèles de mélanges fournit différentes informations pertinentes qui peuvent être utilisées dans des applications pratiques. Nos approches ont été validées sur différentes bases de données réelles et fournissent des résultats prometteurs.

1 Introduction

L'apprentissage non supervisé consiste à construire des représentations simplifiées de données, pour mettre en évidence les relations existantes entre les caractéristiques relevées sur des données et les ressemblances ou dissemblances de ces dernières, sans avoir aucune connaissance sur les classes. On peut distinguer deux grandes familles : les méthodes probabilistes et les méthodes déterministes ou tout simplement les méthodes de quantification. Ce travail concerne le

traitement des données qualitatives et mixtes à l'aide des cartes auto-organisatrices, (Kohonen, 2001; Anouar, 1996; Thiria et al., 1997) dans le cadre du formalisme des modèles de mélanges.

Les cartes topologiques utilisent un algorithme d'auto-organisation (Self-Organizing Map, SOM) qui permet d'une part de quantifier de grandes quantités de données en regroupant les observations similaires en groupes ou clusters et d'autre part de projeter les groupes obtenus de façon non-linéaire sur une carte, permettant ainsi de visualiser la structure du jeu de données en deux dimensions, tout en respectant la topologie initiale des données. La mise en oeuvre de l'algorithme sur des données qualitatives et mixtes suppose toujours une phase de prétraitement permettant d'extraire une information numérique des observations pour la partie qualitative. Des travaux antérieurs (Lebbah et al., 2005; Lebbah, 2003) ont permis de proposer un modèle de cartes topologiques déterministes pour les données binaires et un autre modèle probabiliste dédié aux données catégorielles. Différentes approches ont été envisagées, pour différents types de données, reposant sur des formalismes probabilistes. Dans (Verbeek et al., 2005), les auteurs proposent une généralisation probabiliste des cartes auto-organisatrices qui maximise l'énergie variationnelle et qui additionne la log-vraisemblance des données et la divergence de Kullback-Leibler entre une fonction de voisinage normalisée et la distribution postérieure sur les données pour les composants. Nous avons également STVQ (Soft topographic vector quantization), qui emploie une certaine mesure de divergence entre les données élémentaires et les référents afin de minimiser une nouvelle fonction d'erreur (Heskes, 2001; Graepel et al., 1998). Il existe aussi un modèle original qui permet de créer un graphe entre les prototypes en se basant sur les modèles de mélanges et les graphes de Delaunay (Aupetit, 2005). Un autre modèle, souvent présenté comme la version probabiliste des cartes auto-organisatrices, est GTM (Generative Topographic Map) (Bishop et al., 1998; Kaban et Girolami, 2001). Cependant, la façon dont GTM atteint l'organisation topologique est très différente de celle utilisée dans les modèles des cartes topologiques traditionnelles. Dans GTM le mélange est paramétré par une combinaison linéaire de fonctions non linéaires des positions des cellules de la carte (GTM à l'origine a été conçu pour les données quantitatives). Des extensions de ce modèle à un modèle dédié aux données discrètes et binaires ont été proposées (Girolami, 2001; Priam et Nadif, 2006; Priam et al., 2008).

La difficulté principale en ce qui concerne l'application des modèles de mélanges de distributions multivariables pour le partitionnement "clustering" est qu'ils sont liés à un type de données : les distributions normales pour les données quantitatives, la distribution de Bernoulli pour les données binaires, et le modèle des classes latentes pour les variables catégorielles. Ceci est limitatif puisque la plupart des ensembles de données sont mixtes et impliquent différents types d'informations : quantitatifs, catégoriels, binaires.

L'idée de base de ce travail repose sur le principe de la conservation de la structure initiale des données en utilisant le formalisme probabiliste. Les modèles de mélanges proposés ici vérifient ce principe et fournissent des résultats directement interprétables par rapport aux données initiales, qu'elles soient simplement binaires ou mixtes.

2 Modèle de mélange topologique

Pour définir le modèle des cartes topologiques à base de modèle de mélange on associe à chaque cellule c de la carte $\mathcal{C}$ une fonction densité $\mathbf{f}_c(\mathbf{x}) = p(\mathbf{x}/\theta_c)$ dont les paramètres seront notés θ. Pour définir le mélange de densités des cartes topologiques, nous utiliserons le

formalisme bayésien introduit par Luttrel (Luttrel, 1994). Ce formalisme suppose que les observations $\mathbf{x}$ sont générées de la manière suivante : on commence par choisir une cellule c^* de $\mathcal{C}$ qui permet de déterminer un voisinage de cellule suivant la probabilité conditionnelle $p(c/c^*)$ qui est supposée connue. Ainsi l'observation $\mathbf{x}$ est générée suivant la probabilité $p(\mathbf{x}/c)$. Ce processus permet de modéliser les différentes étapes de propagation de l'information entre les différentes cellules $c \in \mathcal{C}$ et $c^* \in \mathcal{C}$.

La première étape de propagation attribue à une observation $\mathbf{x}$ une cellule c de $\mathcal{C}$ avec la probabilité $p(c/\mathbf{x})$. La seconde étape attribue à toute cellule c de $\mathcal{C}$ une cellule c^* de $\mathcal{C}$ avec la probabilité $p(c^*/c)$. Afin de simplifier le calcul des probabilités on suppose que le processus de propagation vérifie la propriété de Markov :

$$p(\mathbf{x}/c, c^*) = p(\mathbf{x}/c)$$

et

$$p(c^*/c, \mathbf{x}) = p(c^*/c)$$

Ce formalisme nous amène à définir le générateur des données $p(\mathbf{x})$ par un mélange de probabilités défini comme suit :

$$\begin{aligned} p(\mathbf{x}) &= \sum_{c,c^* \in C} p(c, c^*, \mathbf{x}) \\ &= \sum_{c,c^* \in C} p(\mathbf{x}/c)p(c/c^*)p(c^*) \\ &= \sum_{c^* \in \mathcal{C}} p(c^*)p_{c^*}(\mathbf{x}) \end{aligned} \tag{1}$$

avec

$$p_{c^*}(\mathbf{x}) = p(\mathbf{x}/c^*) = \sum_{c \in \mathcal{C}} p(c/c^*)p(\mathbf{x}/c). \tag{2}$$

Ainsi, $p(\mathbf{x})$ apparaît comme un mélange des probabilités $p_{c^*}(\mathbf{x})$. L'observation $\mathbf{x}$ s'obtient premièrement par la sélection de c^* puis de c de $\mathcal{C}$, ensuite par la sélection de $\mathbf{x}$ à l'intérieur du sous-échantillon avec la probabilité $p(\mathbf{x}/c)$.

Les coefficients du mélange sont les probabilités $p(c^*)$ *a priori* et les fonctions densités relatives à chaque élément du mélange qui sont données par $p_{c^*}(\mathbf{x})$ (eq.(2)). Ce formalisme montre qu'on peut calculer $p(\mathbf{x})$ à condition de connaître pour chaque cellule c la fonction de densité $p(\mathbf{x}/c)$ et la probabilité $p(c/c^*)$ d'activation de la cellule c connaissant c^*.

Afin d'introduire la notion de voisinage dans le formalisme probabiliste, nous supposons que chaque cellule c de la carte $\mathcal{C}$ est d'autant plus active qu'elle est proche de la cellule choisie c^*, (eq.(1), eq.(2)). Ceci nous permet de définir la probabilité $p(c/c^*)$ en fonction de la fonction de voisinage $\mathcal{K}^T(.)$:

$$p(c/c^*) = \frac{\mathcal{K}^T(\delta(c, c^*))}{T_{c^*}} \tag{3}$$

où $T_{c^*} = \sum_{r \in C} \mathcal{K}^T(\delta(r, c^*))$, est un terme normalisant pour obtenir des probabilités.

Pour définir complètement $p(\mathbf{x})$ il reste à définir les coefficients du mélange $p(c^*)$ et les paramètres de la densité $p(\mathbf{x}/c)$. Ce formalisme a déjà été utilisé dans (Anouar et al., 1997) et a permis de définir le modèle PrSOM dédié aux données continues (Anouar, 1996; Thiria et al., 1997). Ce modèle qui généralise le modèle classique des cartes topologiques introduit par Kohonen permet d'obtenir une quantification de l'espace des données, mais aussi une estimation des densités locales.

3 Algorithme EM

L'algorithme EM (Bishop, 1995; Dempster et al., 1977; McLachlan et Krishman, 1997) est un algorithme itératif qui permet de trouver un maximum local de la fonction de vraisemblance des observations, lorsque chaque observation contient une partie cachée (ou non observée). Ainsi, on suppose que chaque donnée est un couple de types $(\mathbf{x}, \mathbf{z})$ où $\mathbf{x}$ est sa partie observable et $\mathbf{z}$ sa partie cachée (non observable). Nous supposons connus d'une manière explicite la forme de la fonction densité jointe $p(\mathbf{x}, \mathbf{z}; \theta)$ où θ est l'ensemble des paramètres du modèle à estimer. On suppose que l'on dispose d'une série de données indépendantes : $(\mathbf{x}_1, \mathbf{z}_1), (\mathbf{x}_2, \mathbf{z}_2), ..., (\mathbf{x}_N, \mathbf{z}_N)$, pour lesquelles $\mathbf{x}_i$ sont les parties qu'on a réellement observées et les $\mathbf{z}_i$ sont les parties cachées (donc inconnues).

Nous souhaitons par la suite maximiser le logarithme de la vraisemblance des parties, des données réellement observées $\mathcal{A} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$ défini comme suit :

$$\ln V(\mathcal{A}; \theta) = \ln V(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N; \theta) = \sum_{i=1}^{N} \ln p(\mathbf{x}_i; \theta) \tag{4}$$

où $p(\mathbf{x}; \theta)$ est la fonction densité de la partie observée $\mathbf{x}$. En pratique $p(\mathbf{x}; \theta)$ est calculable en marginalisant la fonction densité $p(\mathbf{x}, \mathbf{z}; \theta)$ ($p(\mathbf{x}; \theta) = \int p(\mathbf{x}, \mathbf{z}; \theta) d\mathbf{z}$), ce qui donne souvent une fonction log-vraisemblance $\ln V(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N; \theta)$ qui n'est pas simple à optimiser.

L'algorithme EM proposé par Dempster et al (Dempster et al., 1977) maximise l'expression (4) en utilisant la log-vraisemblance des données entières $\ln V(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N, \mathbf{z}_1, ..., \mathbf{z}_N; \theta)$. On désigne par $\mathbf{Z} = \{\mathbf{z}_1, ..., \mathbf{z}_N\}$ l'ensemble des parties correspondantes et non observées. Chaque itération de l'algorithme EM comporte deux étapes :

- L'étape d'Estimation (Expectation step) ; dite aussi étape "E"
- L'étape de Maximisation (Maximization step) ; dite étape "M"

Ainsi à l'itération t ces deux étapes se présentent de la manière suivante :

– **Etape E (Expectation step)**
On suppose à cette étape, que la fonction densité de la partie cachée conditionnée par la partie observée $(\mathbf{z}/\mathbf{x})$ correspond à la valeur du paramètre θ^{t-1} calculée à l'itération précédente (ou égale à l'initialisation θ^0 si $t = 1$) ; cette fonction densité s'écrit donc $(p(\mathbf{z}/\mathbf{x}, \theta^{t-1}))$. On calcule alors l'espérance :

$$\begin{aligned} Q(\theta, \theta^{t-1}) &= E\left[\ln V(\mathcal{A}, \mathbf{Z}/\theta)/\mathcal{A}, \theta^{t-1}\right] \\ &= \int \ln V(\mathcal{A}, \mathbf{Z}/\theta) p(\mathbf{Z}/\mathcal{A}, \theta^{t-1}) \\ &= \int \ln V(\mathcal{A}, \mathbf{Z}/\theta) \prod_{i=1}^{N} p(\mathbf{z}_i/\mathbf{x}_i, \theta^{t-1})\, d\mathbf{z}_i \end{aligned} \tag{5}$$

Cette expression qui est parfois appelée "la vraisemblance relative" se justifie "intuitivement". En effet, étant donné qu'on ne connait pas les valeurs des variables cachées $\mathbf{z}_i$ associées aux observations $\mathbf{x}_i \in \mathcal{A}$, on calcule l'espérance de la log-vraisemblance relativement aux variables cachées.

– **Etape de maximisation (Maximization step)**
Ayant calculé $Q(\theta, \theta^{t-1})$ à l'étape E, il s'agit dans cette étape de maximiser cette expression par rapport à θ. On prend alors :

$$\theta^{'} = \arg\max_{\theta} Q(\theta, \theta^{t-1})$$

Il est démontré alors que chaque itération (E-M) fait croître la fonction log-vraisemblance (4) $(\ln V(\mathcal{A}, \theta^t) \geq \ln V(\mathcal{A}, \theta^{t-1}))$ (Dempster et al., 1977). Ainsi, l'algorithme E-M se présente de la manière suivante :

---

1. **Initialisation** : Choisir des paramètres initiaux θ^0 et N_{iter} (le nombre d'itérations).
2. **Itération de Base** $(t \geq 1)$
   – Etape **E** : Estimer l'expression $Q(\theta, \theta^{t-1})$ définie par l'expression 5.
   – Etape **M** : Maximiser $Q(\theta, \theta^{t-1})$ par rapport à θ, prendre $\theta^t = \arg\max_\theta Q(\theta, \theta^{t-1})$
3. **Répéter** l'itération de base, jusqu'à stabilisation de θ^t ou jusqu'à $t \geq N_{iter}$

Remarque : L'algorithme EM est largement utilisé en classification pour bâtir de façon itérative, à partir d'un nombre d'observations données, des modèles de mélanges paramétriques.

4 Modèle de mélanges pour des données binaires et mixtes

Nous supposons par la suite que les données $\mathbf{x}$ sont des vecteurs à d composantes composées de deux parties : une partie quantitative $\mathbf{x}_i^{r[.]} = (x_i^{r[1]}, x_i^{r[2]}, ..., x_i^{r[n]})$ $(\mathbf{x}_i^{r[.]} \in \Re^n)$ et d'une

partie qualitative codée en binaire $\mathbf{x}_i^{b[.]} = (x_i^{b[1]}, x_i^{b[2]}, ..., x_i^{b[k]}, ..., x_i^{b[m]})$ où la $k$ième composante $x_i^{b[k]}$ est une variable binaire ($x_i^{b[k]} \in \beta = \{0, 1\}$). Chaque observation $\mathbf{x}_i$ est ainsi la réalisation d'une variable aléatoire appartenant à $\mathcal{R}^n \times \beta^m$. Avec ces notations une observation particulière $\mathbf{x}_i = (\mathbf{x}_i^{r[.]}, \mathbf{x}_i^{b[.]})$ est un vecteur composé d'une partie quantitative et une binaire avec la dimension $d = n + m$. Nous rappelons que $\mathcal{A}$ représente l'ensemble des observations de taille N.

Afin de simplifier ce modèle, nous supposons que la composante quantitative est indépendante de la composante binaire. Cette hypothèse est appliquée pour la majorité des algorithmes probabilistes de partitionnement. Ainsi la probabilité conditionnelle peut être réécrite comme le produit de deux termes :

$$p(\mathbf{x}/c) = p(\mathbf{x}^{r[.]}/c) \times p(\mathbf{x}^{b[.]}/c)$$

Nous prenons aussi une hypothèse forte qui est de considérer que les n composantes quantitatives et les m binaires sont indépendantes sachant une cellule c :

$$p(\mathbf{x}^{b[.]}/c) = \prod_{k=1}^{m} p(x^{b[k]}/c)$$

$$p(\mathbf{x}^{r[.]}/c) = \prod_{k=1}^{n} p(x^{r[k]}/c)$$

Nous associons dans la suite à chaque cellule $c \in \mathcal{C}$ de la carte une probabilité conditionnelle qui décrit la génération d'observation par une cellule c avec les deux parties :

- La fonction densité de la partie quantitative qui est une gaussienne sphérique $p(\mathbf{x}^{r[.]}/c) = \mathcal{N}(\mathbf{x}^{r[.]}, \mathbf{w}^{r[.]}, \sigma_c^2 I)$, définie par "la moyenne" (vecteur référent $\mathbf{w}^{r[.]} = (w^{r[1]}, ..., w^{r[k]}, w^{r[n]})$), la matrice covariance, définie par $\sigma_c^2 I$ où σ_c est l'écart type et I la matrice identité,

$$\mathcal{N}(\mathbf{x}^{r[.]}, \mathbf{w}^{r[.]}, \sigma_c^2 I) = \frac{1}{(2\pi\sigma_c)^{\frac{n}{2}}} \exp\left[\frac{||\mathbf{w}_c^{r[.]} - \mathbf{x}_i^{r[.]}||^2}{2\sigma_c^2}\right] \tag{6}$$

- La fonction densité de la partie binaire qui est la distribution de Bernoulli $p(\mathbf{x}^{b[.]}/c) = p(\mathbf{x}^{b[.]}/\varepsilon_c, \mathbf{w}_c^{b[.]}) = f_c(\mathbf{x}^{b[.]}, \mathbf{w}_c^{b[.]}, \varepsilon_c)$ avec les paramètres $\mathbf{w}_c^{b[.]} = (w_c^{b[1]}, w_c^{b[2]}, ..., w_c^{b[k]}, ..., w_c^{b[m]}) \in \beta^m$ avec la probabilité $\varepsilon_c \in]0, \frac{1}{2}[$ associée à $\mathbf{w}_c^{b[.]} \in \{0, 1\}^m$. Le paramètre ε_c définit la probabilité d'être différent du prototype $\mathbf{w}_c^{b[.]}$. La distribution associée à chaque cellule $c \in \mathcal{C}$ est définie comme suit :

$$f_c(\mathbf{x}^{b[.]}, \mathbf{w}_c^{b[.]}, \varepsilon_c) = \varepsilon_c^{\mathcal{H}(\mathbf{x}^{b[.]}, \mathbf{w}_c^{b[.]})} (1 - \varepsilon_c)^{m - \mathcal{H}(\mathbf{x}^{b[.]}, \mathbf{w}_c^{b[.]})} \tag{7}$$

  $\mathcal{H}$ est la distance de Hamming qui permet la comparaison de deux vecteurs binaires. Cette distance mesure le nombre de composantes binaires différentes entre $\mathbf{x}_i^{b[.]}$ et $\mathbf{x}_j^{b[.]}$:

$$\mathcal{H}(\mathbf{x}_i^{b[.]}, \mathbf{x}_j^{b[.]}) = \sum_{k=1}^{m} |x_i^{b[k]} - x_j^{b[k]}|$$

Les paramètres $\theta = \theta^{\mathcal{C}} \cup \theta^{\mathcal{C}^*}$ qui définissent le modèle générateur de mélange sont constitués à la fois des paramètres de la distribution Gaussienne et Bernoulli ($\theta^{\mathcal{C}} = \{\theta^c, c = 1..K\}$, où $\theta^c = (\mathbf{w}_c = (\mathbf{w}_c^{r[.]}, \mathbf{w}_c^{b[.]}), \sigma_c^2, \varepsilon_c)$), et de la probabilité a priori aussi appellée coefficient de la mixture ($\theta^{\mathcal{C}^*} = \{\theta_{c^*}, c^* = 1..K\}$, où $\theta_{c^*} = p(c^*)$). L'objectif maintenant est de définir la fonction de coût ainsi que l'algorithme d'apprentissage associé qui permettra d'estimer ces paramètres. Cet algorithme sera noté par la suite PrMTM :Probabilistic Mixed Topological Map.

4.1 Algorithme d'apprentissage

L'algorithme d'apprentissage consiste à maximiser la vraisemblance des observations en appliquant l'algorithme EM. L'usage de l'algorithme EM s'explique par l'existence d'une variable cachée notée $\mathbf{z}$, constituée par le couple de cellule c et c^*, $\mathbf{z} = (c, c^*)$, impliquées dans la génération d'une donnée observée $\mathbf{x}$. En effet, la variable cachée $\mathbf{z} = (c, c^*)$ apparaît lorsqu'on écrit :

$$\begin{aligned} p(\mathbf{x}) &= \sum_{\mathbf{z} \in \mathcal{C} \times \mathcal{C}} p(\mathbf{x}, \mathbf{z}) \\ &= \sum_{c, c^* \in \mathcal{C}} p(\mathbf{x}/c) p(c^*) p(c/c^*) \mathcal{N}(\mathbf{x}^{r[.]}, \mathbf{w}^{r[.]}, \sigma_c^2 I) \times f_c(\mathbf{x}^{b[.]}, \mathbf{w}_c^{b[.]}, \varepsilon_c) \end{aligned} \quad (8)$$

A chaque donnée mixte réellement observée $\mathbf{x}$, correspond une variable indicatrice non observée $\mathbf{z}$ qui appartient à $\mathcal{C} \times \mathcal{C}$ et qui permet d'identifier le couple de cellules responsables de la génération de l'observation $\mathbf{x}_i$. On note par $\mathbf{Z} = \{\mathbf{z}_1, ..., \mathbf{z}_N\}$ l'ensemble des variables non observées.

$$z_i^{(c,c^*)} = \begin{cases} 1 & \text{pour } \mathbf{z}_i = (c, c^*) \\ 0 & \text{sinon} \end{cases} \quad (9)$$

Donc, nous pouvons définir la vraisemblance des données de la façon suivante :

$$V^T(\mathcal{A}, \mathbf{Z}; \theta) = \prod_{i=1}^{K} \prod_{c^* \in \mathcal{C}} \prod_{c \in \mathcal{C}} \left[p(c^*) p(c/c^*) \mathcal{N}(\mathbf{x}_i^{r[.]}, \mathbf{w}^{r[.]}, \sigma_{c_1}^2 I) \times f_c(\mathbf{x}_i^{b[.]}, \mathbf{w}_c^{b[.]}, \varepsilon_c) \right]^{z_i^{(c,c^*)}} \quad (10)$$

L'application de l'algorithme EM pour la maximisation de la vraisemblance des données observées nécessite d'une part l'estimation de

$$Q^T(\theta, \theta^t) = E\left[\ln V^T(\mathcal{A}, \mathbf{Z}; \theta) / \mathcal{A}, \theta^t \right],$$

où θ^t est l'ensemble des paramètres estimés à la $t^{ème}$ itération de l'algorithme, et θ l'ensemble des paramètres recherchés.

L'étape "E (Estimation)" calcule l'espérance de la log-vraisemblance par rapport aux variables cachées en prenant en considération les paramétres θ^{t-1}. A l'issue de l'étape "M (Maximisation)", la fonction $Q^T(\theta^t, \theta^{t-1})$ est maximisée par rapport θ^t, ($\theta^t = \arg\max_\theta(Q^T(\theta, \theta^{t-1}))$). Ces deux étapes maximisent la fonction objective $Q^T(\theta^t, \theta^{t-1})$ où $\ln V^T(\mathcal{A}, \theta^t) \geq$

$\ln V^T(\mathcal{A}, \theta^{t-1})$. Ainsi, la fonction $Q^T(\theta^t, \theta^{t-1})$ est définie de la manière suivante :

$$Q^T(\theta^t, \theta^{t-1}) = \sum_{\mathbf{x}_i \in \mathcal{A}} \sum_{c^* \in \mathcal{C}} \sum_{c \in \mathcal{C}} E(z_i^{(c,c^*)}/\mathbf{x}_i, \theta^{t-1})$$
$$\left(\ln(\theta^{c^*}) + \ln\left(\frac{\mathcal{K}^T(\delta(c^*, c))}{T_{c^*}}\right) + \ln\left(\mathcal{N}(\mathbf{x}^{r[.]}, \mathbf{w}_c^{r[.]}, \sigma_c^2 I) \times f_c(\mathbf{x}^{b[.]}, \mathbf{w}_c^{b[.]}, \varepsilon_c)\right)\right) \quad (11)$$

où $T_{c^*} = \sum_{r \in \mathcal{C}} \mathcal{K}^T(\delta(r, c^*))$ et la variable $z_i^{(c,c^*)}$ est une variable de Bernoulli.

Ainsi, l'espérance est définie comme suit :

$$E(z_i^{(c,c^*)}/\mathbf{x}_i, \theta^{t-1}) = p(z_i^{(c,c^*)} = 1/\mathbf{x}_i, \theta^{t-1}) = p(c, c^*/\mathbf{x}_i, \theta^{t-1})$$

où

$$p(c, c^*/\mathbf{x}_i, \theta^{t-1}) = \frac{p(c^*)p(c/c^*)p(\mathbf{x}/c)}{p(\mathbf{x})}$$

Ainsi la fonction $Q^T(\theta^t, \theta^{t-1})$ s'écrit :

$$\begin{aligned}
Q^T(\theta^t, \theta^{t-1}) &= \sum_{\mathbf{x}_i \in \mathcal{A}} \sum_{c \in \mathcal{C}} \sum_{c^* \in \mathcal{C}} p(c, c^*/\mathbf{x}_i, \theta^{t-1}) \ln(\mathcal{N}(\mathbf{x}^{r[.]}, \mathbf{w}_c^{r[.]}, \sigma_c^2 I)) \\
&+ \sum_{\mathbf{x}_i \in \mathcal{A}} \sum_{c \in \mathcal{C}} \sum_{c^* \in \mathcal{C}} p(c, c^*/\mathbf{x}_i, \theta^{t-1}) \ln(f_c(\mathbf{x}^{b[.]}, \mathbf{w}_c^{b[.]}, \varepsilon_c)) \\
&+ \sum_{\mathbf{x}_i \in \mathcal{A}} \sum_{c^* \in \mathcal{C}} \sum_{c \in \mathcal{C}} p(c, c^*/\mathbf{x}_i, \theta^{t-1}) \ln(\theta^{c^*}) \\
&+ \sum_{\mathbf{x}_i \in \mathcal{A}} \sum_{c^* \in \mathcal{C}} \sum_{c \in \mathcal{C}} p(c, c^*/\mathbf{x}_i, \theta^{t-1}) \ln\left(\frac{\mathcal{K}^T(\delta(c^*, c))}{T_{c^*}}\right)
\end{aligned}$$

Les variables θ^c et θ^{c^*} indiquent les paramètres estimés à l'itération t. Nous observons que la fonction objective $Q^T(\theta^t, \theta^{t-1})$ (12) est définie comme une somme de trois termes principaux. Le premier terme $Q_1^T(\theta^{\mathcal{C}}, \theta^{t-1})$ est composé de deux parties qui dépendent respectivement des paramètres $(\mathbf{w}_c^{r[.]}, \sigma_c)$ et $(\mathbf{w}_c^{b[.]}, \varepsilon_c)$; le second terme $Q_2^T(\theta^{\mathcal{C}^*}, \theta^{t-1})$ dépend du paramètre θ^{c^*}, et le troisième terme est constant. La maximisation de la fonction $Q^T(\theta^t, \theta^{t-1})$ par rapport à θ^{c^*} et θ^c est réalisée séparément. Ainsi,

$$Q^T(\theta^t, \theta^{t-1}) = Q_1^T(\theta^{\mathcal{C}}, \theta^{t-1}) + Q_2^T(\theta^{\mathcal{C}^*}, \theta^{t-1}) + Q_3^T(\theta^{t-1}) \quad (12)$$

où

$$Q_2^T(\theta^{\mathcal{C}^*}, \theta^{t-1}) = \sum_{\mathbf{x}_i \in \mathcal{A}} \sum_{c^* \in \mathcal{C}} \sum_{c \in \mathcal{C}} p(c, c^*/\mathbf{x}_i, \theta^{t-1}) \ln(\theta^{c^*})$$

et

$$Q_1^T(\theta^{\mathcal{C}}, \theta^{t-1}) = Q_1^{r[.],T}(\mathbf{w}^{r[.]}, \sigma_c, \theta^{t-1}) + Q_1^{b[.],T}(\mathbf{w}^{b[.]}, \varepsilon_c, \theta^{t-1})$$

$$Q_1^{r[.],T}(\mathbf{w}^{r[.]}, \sigma_c, \theta^{t-1}) = \sum_{\mathbf{x}_i \in \mathcal{A}} \sum_{c \in \mathcal{C}} \sum_{c^* \in \mathcal{C}} p(c, c^*/\mathbf{x}_i, \theta^{t-1}) \ln(\mathcal{N}(\mathbf{x}^{r[.]}, \mathbf{w}_c^{r[.]}, \sigma_c^2 I))$$

$$Q_1^{b[.],T}(\mathbf{w}^{b[.]}, \varepsilon_c, \theta^{t-1}) = \sum_{\mathbf{x}_i \in \mathcal{A}} \sum_{c \in \mathcal{C}} \sum_{c^* \in \mathcal{C}} p(c, c^*/\mathbf{x}_i, \theta^{t-1}) \ln(f_c(\mathbf{x}^{b[.]}, \mathbf{w}_c^{b[.]}, \varepsilon_c))$$

Maximisation de la fonction $Q^T(\theta^t, \theta^{t-1})$

Maximiser la fonction (12) revient à maximiser les deux premiers termes séparément. Notons que la maximisation du premier terme $Q_1^T(\theta^{\mathcal{C}}, \theta^{t-1})$ nécessite aussi la maximisation de deux autres termes. Finalement, maximiser la fonction (12) revient à maximiser trois termes. Le terme $Q_1^{r[.],T}(\mathbf{w}^{r[.]}, \sigma_c, \theta^{t-1})$ par rapport à $\mathbf{w}^{r[.]}$ et σ_c, la fonction $Q_1^{b[.],T}(\mathbf{w}^{b[.]}, \varepsilon_c, \theta^{t-1})$ par rapport à ε_c et θ^{t-1}, et le terme $Q_2^T(\theta^{\mathcal{C}^*}, \theta^{t-1})$ par rapport à $\theta^{\mathcal{C}^*}$.

– **Maximiser** $Q_2^T(\theta^{\mathcal{C}^*}, \theta^{t-1})$:

$$Q_2^T(\theta^{\mathcal{C}^*}, \theta^{t-1}) = \sum_{\mathbf{x}_i \in \mathcal{A}} \sum_{c^* \in \mathcal{C}} \sum_{c \in \mathcal{C}} p(c, c^*/\mathbf{x}_i, \theta^{t-1}) \ln(\theta^{c^*})$$

Afin d'estimer les paramètres $\theta^{\mathcal{C}^*}$ qui est l'ensemble des probabilités a priori, on donne une forme explicite aux probabilités a priori. Ainsi la probabilité a priori est estimée avec l'équation suivante :

$$\theta^{c^*} = p(c^*) = \frac{\sum_{\mathbf{x}_i \in \mathcal{A}} p(c^*/\mathbf{x}_i, \theta^{t-1})}{K} \tag{13}$$

– **Maximiser** $Q_1^{r[.],T}(\mathbf{w}^{r[.]}, \sigma_c, \theta^{t-1})$:

$$Q_1^{r[.],T}(\mathbf{w}^{r[.]}, \sigma_c, \theta^{t-1}) = \sum_{\mathbf{x}_i \in \mathcal{A}} \sum_{c \in \mathcal{C}} \sum_{c^* \in \mathcal{C}} p(c, c^*/\mathbf{x}_i, \theta^{t-1}) \ln(\mathcal{N}(\mathbf{x}^{r[.]}, \mathbf{w}_c^{r[.]}, \sigma_c^2 I))$$

La maximisation de cette fonction nécessite une maximisation en deux étapes. La première consiste à fixer le paramètre $\mathbf{w}_c^{r[.]}$ et à maximiser la fonction $Q_1^{r[.],T}(\mathbf{w}^{r[.]}, \varepsilon_c, \theta^{t-1})$ en dérivant la fonction par rapport à σ_c. Et la deuxième consiste à fixer l'écart σ_c et à dériver par rapport à $\mathbf{w}_c^{r[.]}$:

$$\mathbf{w}_c^{r[.]} = \frac{\sum_{\mathbf{x}_i \in \mathcal{A}} \mathbf{x}_i^{r[.]} p(c/\mathbf{x}_i)}{\sum_{\mathbf{x}_i \in \mathcal{A}} p(c/\mathbf{x}_i)} \tag{14}$$

$$\sigma_c^2 = \frac{\sum_{\mathbf{x}_i \in \mathcal{A}} ||\mathbf{w}_c^{r[.]} - \mathbf{x}_i^{r[.]}||^2 p(c/\mathbf{x}_i)}{n \sum_{\mathbf{x} \in \mathcal{A}} p(c/\mathbf{x}_i)} \tag{15}$$

– **Maximiser** $Q_1^{b[.],T}(\mathbf{w}^{b[.]}, \varepsilon_c, \theta^{t-1})$

$$Q_1^{b[.],T}(\mathbf{w}^{b[.]}, \varepsilon_c, \theta^{t-1}) = \sum_{\mathbf{x}_i \in \mathcal{A}} \sum_{c \in \mathcal{C}} \sum_{c^* \in \mathcal{C}} p(c, c^*/\mathbf{x}_i, \theta^{t-1}) \ln(f_c(\mathbf{x}^{b[.]}, \mathbf{w}_c^{b[.]}, \varepsilon_c))$$

Dans ce cas le paramètre ϵ dépend seulement de la cellule c. La maximisation de la fonction est aussi réalisée en deux étapes. Une première maximisation par rapport à $w_c^{b[k]}$. Celle-ci correspond au calcul du centre médian. Puis on effectue une maximisation par rapport à ε_c en résolvant l'équation $\frac{\partial Q_1^{b[.],T}(\mathbf{w}^{b[.]},\varepsilon_c,\theta^{t-1})}{\partial \varepsilon_c} = 0$. Ainsi les expressions qui permettent de maximiser cette partie de la fonction, connaissant les paramètres à l'instant $t-1$ sont définies par :

$$w_c^{b[k]} = \begin{cases} 0 & \text{si } \left[\sum_{\mathbf{x}_i \in \mathcal{A}} p(c/\mathbf{x}_i, \theta^{t-1})(1 - x_i^{b[k]})\right] \geq \\ & \left[\sum_{\mathbf{x}_i \in \mathcal{A}} p(c/\mathbf{x}_i, \theta^{t-1}) x_i^{b[k]}\right] \\ 1 & \text{sinon} \end{cases} \tag{16}$$

$$\varepsilon_c = \frac{\sum_{\mathbf{x}_i \in \mathcal{A}} p(c/\mathbf{x}_i, \theta^{t-1}) \mathcal{H}(\mathbf{x}_i^{b[.]}, \mathbf{w}_c^{b[.]})}{\sum_{\mathbf{x}_i \in \mathcal{A}} mp(c/\mathbf{x}_i, \theta^{t-1})} \tag{17}$$

où
$p(c^*/\mathbf{x}_i, \theta^{t-1}) = \sum_{c \in \mathcal{C}} p(c, c^*/\mathbf{x}_i, \theta^{t-1})$
et
$p(c/\mathbf{x}_i, \theta^{t-1}) = \sum_{c^* \in \mathcal{C}} p(c, c^*/\mathbf{x}_i, \theta^{t-1})$.

L'algorithme d'apprentissage PrMTM est une autre application de l'algorithme EM aux modèles des cartes auto-organisatrices probabilistes. En supposant que les probabilités conditionnelles $p(c^*/c)$ sont fixées pour une valeur donnée de T, l'algorithme est défini d'une manière itérative où les paramètres à l'itération $t+1$ sont calculés en fonction des paramètres estimés à l'itération t. L'algorithme PrMTM pour un paramètre T fixé se présente de la manière suivante :

---

1. **Initialisation** ($t = 0$)
   - détermination de l'ensemble des paramètres initiaux (θ^0) et du nombre d'itérations $iter$.
2. **Itération de base à T constant** ($t \geq 1$)
   - Calcul de l'ensemble des paramètres θ^{t+1} à partir de l'ensemble des paramètres déjà calculés θ^t en appliquant les formules (13),(14), (15),(16) et (17).
3. **Répéter** l'itération de base tant que $t < iter$.

Nous avons présenté le principe de l'algorithme d'apprentissage PrMTM permettant d'estimer les paramètres maximisant la fonction vraisemblance pour un paramètre T fixé. Le paramètre T permet de contrôler la taille du voisinage d'influence d'une cellule sur la carte, qui décroît avec le paramètre T. De la même manière et par analogie avec l'algorithme des cartes topologiques, on peut faire décroître la valeur de T entre deux valeurs T_{max} et T_{min}. Pour chaque valeur de T, on obtient une fonction de vraisemblance V^T qui varie avec T. Nous pouvons

observer deux étapes dans le fonctionnement de l'algorithme.

- La première étape correspond aux grandes valeurs de T. Dans ce cas, le voisinage d'influence d'une cellule c de la carte est grand et correspond aux valeurs "significatives" de $K^T(\delta(c,r))$. Les formules, (13),(14), (15),(16) et (17) ont tendance, dans ce cas, à faire participer un très grand nombre d'observations à l'estimation des paramètres du modèle PrMTM.

- La deuxième étape correspond aux petites valeurs de T. Le nombre d'observations intervenant dans les formules (13),(14), (15),(16) et (17) est alors restreint. L'adaptation est très locale.

Si on utilise cette décroissance de T, l'algorithme d'apprentissage de PrMTM se présente de la manière suivante :

---

1. **Phase d'initialisation** ($t = 0$)
   - Choisir T_{max}, T_{min} et N_{iter}. Effectuer l'algorithme d'apprentissage PrMTM pour la valeur de T constante égale à T_{max}.
2. **Etape itérative**
   - L'ensemble des paramètres θ^t de l'étape précédente est connu. Calculer la nouvelle valeur de T en appliquant la formule suivante :

$$T = T_{max}\left(\frac{T_{min}}{T_{max}}\right)^{\frac{t}{iter-1}}$$

   - Pour cette valeur du paramètre T, calculer l'itération de base définie dans l'algorithme indiqué précédemment pour un paramètre fixe T.
3. **Répéter** l'étape itérative tant que $t \leq N_{iter}$

Par la suite nous développerons trois versions de PrMTM (tableau 1). La première est développée sous l'hypothèse que le paramètre de probabilité ϵ dépend uniquement de la cellule ($\epsilon = \varepsilon_c$). La deuxième est le cas général où le paramètre de probabilité dépend de la cellule et de la variable $\epsilon_c = (\varepsilon_c^1, ..., \varepsilon_c^k, ..., \varepsilon_c^n)$. La troisième version estime un paramètre qui dépend seulement de la carte ($\epsilon = \varepsilon$).

PrMTM-ε_c	$\epsilon = \varepsilon_c$	$\mathbf{w}_c = (w_c^1, ..., w_c^k, ..., w_c^n)$
PrMTM-ϵ_c	$\epsilon_c = (\varepsilon_c^1, ..., \varepsilon_c^k, ..., \varepsilon_c^n)$	$\mathbf{w}_c = (w_c^1, ..., w_c^k, ..., w_c^n)$
PrMTM-ε	$\epsilon = \varepsilon$	$\mathbf{w}_c = (w_c^1, ..., w_c^k, ..., w_c^n)$

TAB. 1 – *Les paramètres des trois versions PrMTM :Probabilistic Mixed Topological Map*

4.2 PrMTM et l'algorithme traditionnel des cartes topologiques

Il existe dans la littérature un modèle déterministe à base de cartes topologiques dédié aux données binaires et mixtes utilisant une fonction de coût similaire à l'algorithme traditionnel de Kohonen (MTM : Mixed Topological Map), (Kohonen, 2001; Lebbah et al., 2005). Etant donné que pour des vecteurs à composantes binaires la distance Euclidienne n'est que la distance de Hamming $\mathcal{H}$, alors la distance Euclidienne pour les données mixtes peut-être réécrite comme suit :

$$||\mathbf{x} - \mathbf{w}_c||^2 = ||\mathbf{x}^{r[.]} - \mathbf{w}_c^{r[.]}||^2 + \mathcal{H}(\mathbf{x}^{b[.]}, \mathbf{w}_c^{b[.]}).$$

Utilisant cette expression, la fonction de coût de l'algorithme classique de Kohonen peut être exprimée comme suit :

$$\begin{aligned} \mathcal{G}(\phi, \mathcal{W}) &= \sum_{\mathbf{x}_i \in \mathcal{A}} \sum_{r \in C} \mathcal{K}^T(\delta(r, \phi(\mathbf{x}_i)))||\mathbf{x}_i^{r[.]} - \mathbf{w}_r^{r[.]}||^2 \\ &\quad + \sum_{\mathbf{x}_i \in \mathcal{A}} \sum_{r \in C} \mathcal{K}^T(\delta(r, \phi(\mathbf{x}_i)))\mathcal{H}(\mathbf{x}_i^{b[.]}, \mathbf{w}_r^{b[.]}) \end{aligned} \tag{18}$$

où ϕ affecte chaque observation $\mathbf{x}$ à une seule cellule de $\mathcal{C}$. Le premier terme correspond à la fonction de coût classique utilisée par l'algorithme batch de Kohonen, et le deuxième terme représente la fonction de coût utilisée dans le modèle Binbatch dédié uniquement aux données binaires (Lebbah et al., 2000). La fonction de coût (18), est minimisée en utilisant un processus itératif à deux étapes.

1. **L'étape d'affectation** qui utilise la fonction suivante :

$$\forall \mathbf{x},\ \phi(\mathbf{x}) = \arg\min_c \left(||\mathbf{x}^{r[.]} - \mathbf{w}_c^{r[.]}||^2 + \mathcal{H}(\mathbf{x}^{b[.]}, \mathbf{w}_c^{b[.]}) \right)$$

2. **L'étape de quantification**. Il est facile d'observer que la minimisation de ces deux termes séparément nous permet de définir :
   - La partie quantitative $\mathbf{w}_c^{r[.]}$ du vecteur référent $\mathbf{w}_c$ qui est calculée comme suit :

$$\mathbf{w}_c^{r[.]} = \frac{\sum_{\mathbf{x}_i \in \mathcal{A}} \mathcal{K}(\delta(c, \phi(\mathbf{x}_i)))\mathbf{x}_i^{r[.]}}{\sum_{\mathbf{x}_i \in \mathcal{A}} \mathcal{K}(\delta(c, \phi(\mathbf{x}_i)))},$$

   - La partie binaire $\mathbf{w}_c^{b[.]}$ du vecteur référent $\mathbf{w}_c$ qui est définie comme le centre médian de la partie binaire des observations $\mathbf{x}_i^{b[.]} \in \mathcal{A}$ pondérées par $\mathcal{K}(\delta(c, \phi(\mathbf{x}_i)))$. Chaque composante du vecteur $\mathbf{w}_c^{b[.]} = (w_c^{b[1]}, ..., w_c^{b[k]}, ..., w_c^{b[m]})$ est calculée de la manière suivante :

$$w_c^{b[k]} = \begin{cases} 0 & \text{si } \left[\sum_{\mathbf{x}_i \in \mathcal{A}} \mathcal{K}(\delta(c, \phi(\mathbf{x}_i)))(1 - x_i^{b[k]})\right] \geq \\ & \left[\sum_{\mathbf{x}_i \in \mathcal{A}} \mathcal{K}(\delta(c, \phi(\mathbf{x}_i)))x_i^{b[k]}\right] \\ 1 & \text{sinon} \end{cases},$$

L'analyse de ces formules indique que la mise à jour pour le centre médian $\mathbf{w}_c^{b[.]}$ et la partie réelle $\mathbf{w}_c^{r[.]}$ de notre modèle PrMTM coïncide avec le modèle BinBatch (Lebbah et al., 2000) et le modèle batch de Kohonen pour lesquels chaque observation $\mathbf{x}$ est pondérée proportionnellement par la fonction de voisinage ou une probabilité centrée sur le prototype gagnant.

Dans les deux modèles (PrMTM et MTM) nous minimisons l'inertie des observations $\mathbf{x}$ dans l'espace réel et binaire ($\Re^n \times \beta^m$). Dans le modèle probabiliste PrMTM, la définition du gagnant est différente de celle du modèle déterministe MTM. L'affectation est effectuée à la fin de l'algorithme d'apprentissage. Notons également qu'il existe un lien fort entre le modèle déterministe des cartes topologiques dédiées aux données mixtes (MTM : Mixed Topological Map), et le modèle de mélange PrMTM. A chaque pas d'apprentissage nous calculons la probabilité a posteriori $p(c/\mathbf{x})$ qui est utilisée pour pondérer l'observation $\mathbf{x}$. Dans le modèle MTM, l'affectation est simplifiée en minimisant $||\mathbf{x}^{r[.]} - \mathbf{w}_c^{r[.]}||^2 + \mathcal{H}(\mathbf{x}^{b[.]}, \mathbf{w}_c^{b[.]})$.

Ainsi, si nous supposons que ε et σ représentent le même paramètre pour toutes les cellules et pour toutes les variables (binaires et quantitatives), et si les probabilités a priori $p(c^*) = \frac{1}{K}$ sont égales, alors une distance faible implique une probabilité $p(\mathbf{x}/c) = p(\mathbf{x}^{r[.]}/c) \times p(\mathbf{x}^{b[.]}/c)$ élevée. Nous déduisons donc, que les probabilités a posteriori $p(c/\mathbf{x})$ et $p(c^*/\mathbf{x})$ deviennent élevées elles aussi. Celles-ci sont écrites de la manière suivante :

$$p(c/\mathbf{x}) = \frac{p(\mathbf{x}^{r[.]}/c) \times p(\mathbf{x}^{b[.]}/c) \sum_{c^* \in C} \mathcal{K}^T(\delta(c, c^*))}{KT_{c^*} p(\mathbf{x})},$$

$$p(c^*/\mathbf{x}) = \frac{\sum_{c \in C} \mathcal{K}^T(\delta(c, c^*)) p(\mathbf{x}^{r[.]}/c) \times p(\mathbf{x}^{b[.]}/c)}{KT_{c^*} p(\mathbf{x})},$$

Dans ces conditions, la maximisation de la fonction de coût $Q^T(\theta^t, \theta^{t-1})$ est la même que la maximisation de la fonction de coût $Q_1^T(\theta^{\mathcal{C}}, \theta^{t-1})$ qui est décomposée en deux termes (12) et qui dépendent seulement de $\theta^c = (\mathbf{w}_c, \varepsilon, \sigma)$, où la probabilité ε et l'écart σ dépendent seulement de la carte. Par conséquent, maximiser $Q_1^T(\theta^{\mathcal{C}}, \theta^{t-1})$ par rapport à ε et σ nécessite le calcul de la dérivée simple du $Q_1^{b[.],T}(\mathbf{w}^{b[.]}, \varepsilon, \theta^{t-1})$ par rapport à ε et $Q_1^{r[.],T}(\mathbf{w}^{r[.]}, \sigma_c, \theta^{t-1})$ par rapport à σ. Par contre, maximiser $Q_1^T(\theta^{\mathcal{C}}, \theta^{t-1})$ par rapport à $\mathbf{w}_c$ nécessite la minimisation de la fonction de coût simplifiée $G(\mathbf{w})$ qui est définie par :

$$\begin{aligned} G(\mathbf{w}) &= \sum_{c \in \mathcal{C}} \sum_{\mathbf{x}_i \in \mathcal{A}} p(c/\mathbf{x}_i, \theta^{t-1}) \mathcal{H}(\mathbf{x}_i^{b[.]}, \mathbf{w}_c^{b[.]}) \\ &\quad + \sum_{c \in \mathcal{C}} \sum_{\mathbf{x}_i \in \mathcal{A}} p(c/\mathbf{x}_i, \theta^{t-1}) ||\mathbf{x}_i^{r[.]} - \mathbf{w}_c^{r[.]}||^2 \end{aligned} \tag{19}$$

On en déduit ainsi que le modèle MTM a tendance à rendre toutes les cellules équivalentes. La détermination de la cellule responsable de la génération c^* avec une fonction d'affectation, nous permet de déduire que la fonction de coût $\mathcal{G}(\phi, \mathbf{w})$ (18) n'est qu'une simplification de la fonction de coût $G(\mathbf{w})$ (19). Il est clair que le modèle PrMTM probabiliste fournit plus d'information que le modèle déterministe MTM.

5 Expérimentations

Pour évaluer la qualité de la classification, on adopte une approche d'évaluation qui utilise des étiquettes externes. Ainsi, on utilise la pureté de la classification pour mesurer les résultats de la classification. C'est une approche fréquente dans le domaine de la classification des données. En général, les résultats de la classification sont évalués en se basant sur des connaissances externes sur la façon dont les classes doivent être structurées. Cela peut impliquer de calculer des critères comme : la séparation, la densité, la connectivité, etc. La seule manière de juger l'utilité du résultat du clustering est la validation indirecte, par laquelle les classes sont appliquées à la solution d'un problème et l'exactitude est évaluée par rapport à l'objectif des connaissances externes. Cette procédure est définie par (Jain et Dubes, 1988) comme "validation du clustering par des variables externes", et a été utilisée dans plusieurs travaux (Khan et Kant, 2007; Andreopoulos et al., 2006). Nous pensons que cette approche est raisonnable, si on ne veut pas juger les résultats de la classification par certains indices de validité de la classe, qui ne sont qu'un choix parmi certaines propriétés préférées des classes (par exemple : compact, ou bien séparés, ou connectés). Par conséquent, pour adopter cette approche on a besoin des jeux de données étiquetées, où la connaissance externe représente l'information sur la classe fournit par les étiquettes. Ainsi, si l'algorithme PrMTM trouve des classes pertinentes dans les données, cela sera reflété par la distribution des classes. Donc on utilise un vote majoritaire pour les clusters et on les compare au comportement des méthodes dans la littérature. La technique du vote majoritaire contient les étapes suivantes. Pour chaque cluster $c \in \mathcal{C}$:

- On compte le nombre d'observations de chaque classe l (appelé N_{cl}).
- On compte le nombre total d'observations affectées à la cellule c (appelé N_c).
- On calcule la proportion d'observation de chaque classe (appelée $S_{cl} = N_{cl}/N_c$).
- On attribue au cluster l'étiquette de la classe la plus représentée ($l = \arg\max_l(S_{cl})$.

Une cellule c pour laquelle $S_{cl} = 1$ pour certaine classe étiquettée l est d'habitude appelée un cluster "pure", et une mesure de pureté peut être exprimée comme le pourcentge d'éléments de la classe affectée au cluster. Les résultats expérimentaux sont ensuite exprimés comme une fraction des observations qui interviennent dans les clusters et qui sont étiquetées avec une classe différente de celle de l'observation. Cette quantité est exprimée sous forme de pourcentage et est appelée "erreur de classification" (indiquée comme $Err\%$ dans les résultats). Concernant les dimensions des cartes topologiques, nous avons utilisé la dimension fournie d'une manière heuristique à l'aide de la "SOM toolbox".

5.1 Application aux bases binaires et catégorielles

5.1.1 La base Zoo

C'est une base de données extraite du répertoire UCI (Blake et Merz, 1998). On utilise ce simple jeu de données pour montrer les bonnes performances de notre algorithme PrMTM. Ce jeu de données contient l'information sur 101 animaux décrits par 16 variables qualitatives : dont 15 variables sont binaires et une est catégorielle avec 6 modalités. Chaque animal est étiquetté de 1 à 7 conformément à sa classe (son espèce). Utilisant le codage disjonctif pour les variables qualitatives avec 6 valeurs possibles, définit dans le tableau 2, on obtient une matrice binaire 101×21 ($individus \times variables$).

Afin de montrer l'apport du modèle appliqué aux données binaires, nous avons appris le modèle globale sur une carte de dimensions 5×5 cellules. L'algorithme d'apprentissage nous fournit dans ce cas les paramètres de la distribution de Bernoulli pour chaque cellule, $\mathbf{w}_c = (w_c^1, w_c^2, ..., w_c^k, ..., w_c^{21}) \in \beta^n$ et un vecteur de probabilité $\epsilon_c = (\varepsilon_c^1, \varepsilon_c^2, ..., \varepsilon_c^k, ..., \varepsilon_c^{21})$.

Modalité	Codage binaire
1	1 0 0 0 0 0
2	0 1 0 0 0 0
3	0 0 1 0 0 0
4	0 0 0 1 0 0
5	0 0 0 0 1 0
6	0 0 0 0 0 1

TAB. 2 – *Variable catégorielle avec 6 valeurs possibles. Chaque modalité est codée de la même manière utilisant le codage disjonctif complet.*

A la fin de la phase d'apprentissage, chaque observation, qui correspond à un animal, est affectée à la cellule avec la plus grande probabilité a posteriori $p(c/\mathbf{x})$. Pour visualiser la cohérence de la carte avec les étiquettes des animaux, la figure 1 nous montre le numéro de la classe qui correspond à chaque cellule après l'application du vote majoritaire pour chaque cellule. On observe que les cellules qui ont la même classe sont proches l'une de l'autre sur la carte. On voit aussi que les insectes qui ont l'étiquette "6" sont affectés vers la partie droite en bas de la carte. On peut faire la même analyse pour le reste des "groupes".

La figure 2 nous montre l'évolution de la fonction objective $Q^T(\theta^t, \theta^{t-1})$ sans calculer le terme constant $Q_3^T(\theta^{t-1})$ et variant T de $T_{max} = 2$ à $T_{min} = 0.5$. On visualise aussi les termes $Q_1^T(\theta^{\mathcal{C}}, \theta^{t-1})$ et $Q_2^T(\theta^{\mathcal{C}^*}, \theta^{t-1})$. La maximisation de la fonction objective est évidente.

5.1.2 La base de données des chiffres manuscrits

Cette expérience concerne une base de données composées de chiffres manuscrits ("0"−"9") extraits à partir d'une collection de cartes des services hollandais (Blake et Merz, 1998). On a 200 exemples pour chaque caractère, ainsi on a au total 2000 exemples. Chaque exemple est une imagette binaire (pixel "noir" ou "blanc") de dimension 15×16. L'ensemble de données forme une matrice binaire de dimension 2000×240. Chaque variable qualitative est un pixel à deux valeurs possibles "On=1" et "Off=0". L'algorithme PrMTM$-\epsilon$ associe à chaque cellule uniquement une distribution de Bernoulli avec un vecteur référent $\mathbf{w}_c = (w_c^1, w_c^2, ..., w_c^k, ..., w_c^{240}) \in \beta^n$ et un vecteur de probabilités $\epsilon_c = (\varepsilon_c^1, \varepsilon_c^2, ..., \varepsilon_c^k, ..., \varepsilon_c^{240})$.

La figure 3 nous montre les paramètres estimés par notre modèle PrMTM. La figure 3.(a) nous montre le vecteur prototype $\mathbf{w}_c \in \beta^n$ qui est représenté comme une imagette de dimension 15×16. La figure 3.(b) nous montre le vecteur paramètre ϵ_c comme une imagette de la même dimension 15×16. Ainsi, chaque pixel définit la probabilité ε_c^k d'être différent de la variable binaire w_c^k associée au vecteur prototype binaire $\mathbf{w}_c$ représenté dans la figure 3.(a). La nuance

antelope buffalo deer elephant giraffe hare mole opossum oryx vole (1)	dolphin porpoise (1)	bass catfish chub dogfish herring pike piranha stingray tuna (4)	carp haddock seahorse sole (4)	clam seawasp (7)
aardvark bear boar cheetah leopard lion lynx mink mongoose polecat puma pussycat raccoon wolf (1)	frog newt toad tuatara (5)	pitviper slowworm (3)	slug worm (7)	crab crayfish lobster octopus starfish (7)
calf cavy goat hamster pony reindeer (1)	scorpion (7)	kiwi ostrich penguin rhea vulture (2)	girl seal sealion (1)	platypus seasnake tortoise (3)
fruitbat squirrel vampire (1)	duck flamingo swan (2)	chicken dove lark parakeet pheasant sparrow wren (2)	flea termite (6)	gnat (6)
gorilla wallaby (1)	crow gull hawk skimmer skua (2)	honeybee wasp (6)	housefly moth (6)	Ladybird (6)

FIG. 1 – 5×5 *La carte PrMTM−ϵ. On montre les animaux associés à la plus grande probabilité. Le numéro entre parenthèses indique l'étiquette de la classe pour chaque cellule après l'application de la règle du vote majoritaire*

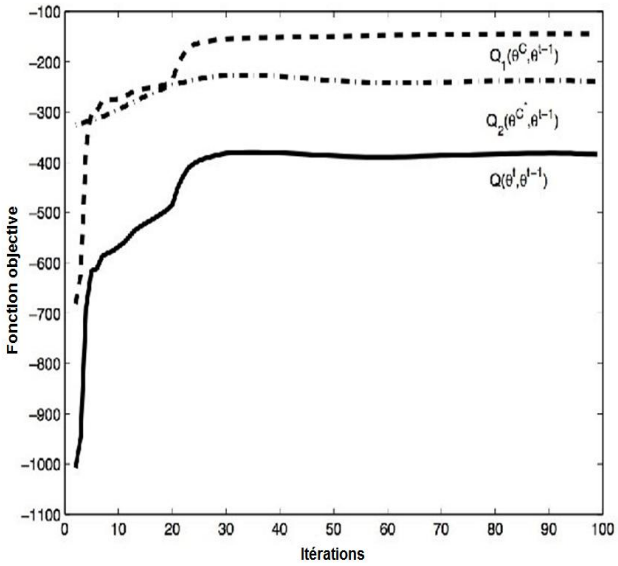


FIG. 2 – *L'évolution de la fonction de coût $Q^T(\theta^t, \theta^{t-1})$ au cours de l'apprentissage pour le jeu de données Zoo. La ligne en pointillée représente la fonction $Q_1^T(\theta^{\mathcal{C}}, \theta^{t-1})$. La ligne tiret-point représente la fonction $Q_2^T(\theta^{\mathcal{C}^*}, \theta^{t-1})$*

de gris de chaque pixel est proportionnelle à la probabilité d'être différent du prototype estimé $\mathbf{w}_c$. On constate que les composantes ou les probabilités ε_c^k, qui correspondent au contour de l'image, sont élevées. En observant les deux figures à la fois, on observe une organisation topologique des prototypes assez claire sur toute la carte.

Pour étudier l'influence du choix de la loi de Bernoulli, nous avons réalisé un apprentissage de l'algorithme PrMTM-ε_c qui suppose que le paramètre ε_c dépend uniquement d'une cellule $c \in \mathcal{C}$. Au lieu d'estimer le vecteur de probabilités ϵ_c, le modèle PrMTM calcule une probabilité scalaire ε_c pour chaque cellule. La figure 4 nous montre les paramètres estimés par le modèle PrMTM-ε. La figure 4.(a) montre le vecteur prototype $\mathbf{w}_c \in \beta^n$ qui est représenté comme une imagette de dimension 15×16. On observe une organisation topologique des prototypes assez nette. La figure 4.(b) représente le paramètre ε_c estimé pour chaque cellule $c \in \mathcal{C}$. Ainsi, la nuance de gris de chaque cellule est proportionnelle à la probabilité ε_c d'être différente du vecteur prototype binaire $\mathbf{w}_c$ représenté dans la figure 4.(a).

Evidemment, on constate que le modèle général PrMTM$-\epsilon$ présenté dans la figure 3 fournit plus d'informations que le modèle réduit PrMTM-ε_c (figure 4). Le modèle PrMTM-ε_c est intéressant quand on manipule de grandes bases de données.

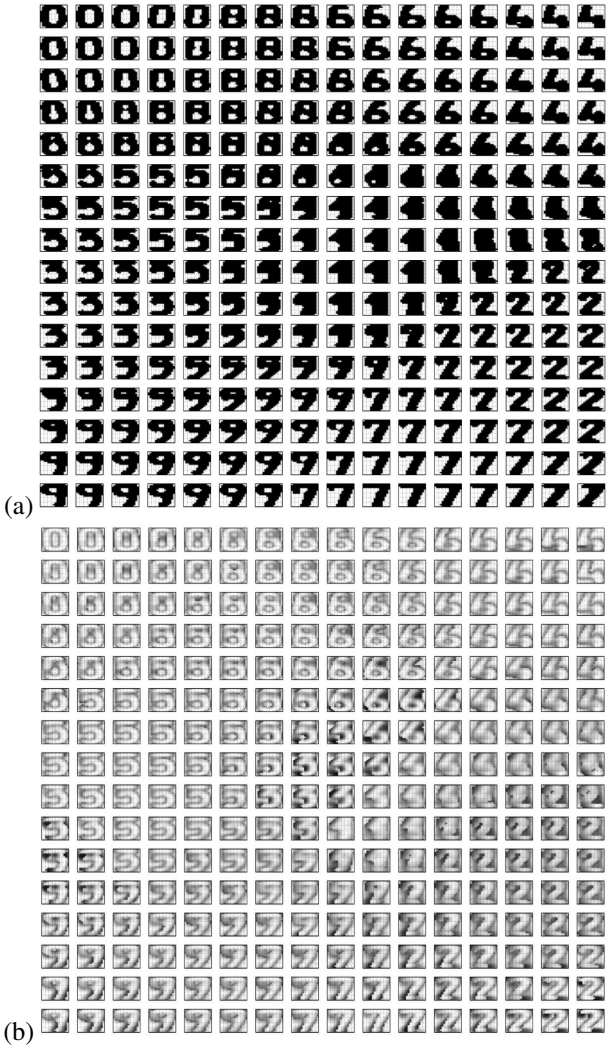
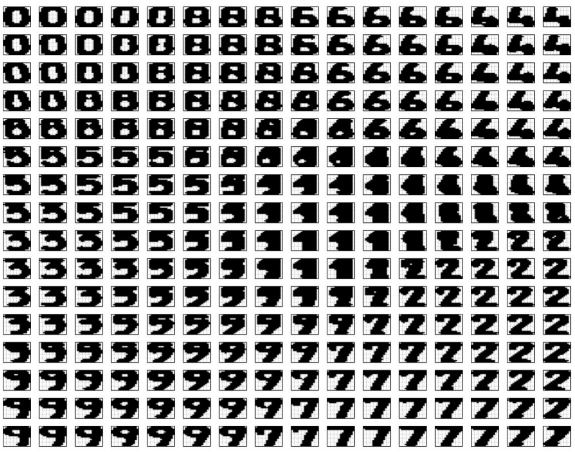
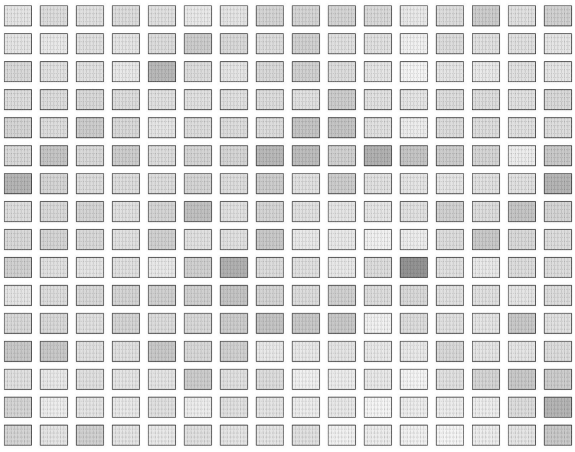


FIG. 3 – *La carte PrMTM−ϵ de dimension* 16×16. **(a)**. *Chaque image est le vecteur prototype* $\mathbf{w}_c = (w_c^1, w_c^2, ...w_c^k, ..., w_c^{240})$ *qui est un vecteur binaire. Les images des prototypes sont bien organisées. Le pixel blanc indique "0" et le pixel noir indique "1".* **(b)**.*Chaque image représente un vecteur de probabilité* $\epsilon_c = (\varepsilon_c^1, \varepsilon_c^2, ..., \varepsilon_c^k, ..., \varepsilon_c^{240})$.



(a)



(b)

FIG. 4 – *La carte de PrMTM-ε de dimension* 16×16 *. Dans la figure (a) chaque image représentée est le vecteur prototype* $\mathbf{w}_c = (w_c^1, w_c^2, ...w_c^k, ..., w_c^{240})$ *qui est un vecteur binaire. Les prototypes des images sont bien organisés. Le pixel blanc indique "0" et le pixel noir indique "1". Dans la figure (b) chaque image représentée est la probabilité* ε_c*. Le niveau de gris de chaque cellule est proportionnel à la probabilité d'être différent du prototype estimé* $\mathbf{w}_c$ *(figure(b))*

5.1.3 Discussions

Avec le modèle que l'on propose, on peut obtenir différents niveaux de pertinences. En fonction de l'hypothèse sur la probabilité ε on peut avoir trois cas :

1. La probabilité ε dépend de la cellule c et de la variable j : $\epsilon_c = (\varepsilon_c^1, \varepsilon_c^2, ..., \varepsilon_c^j, ..., \varepsilon_c^n)$ figure 5(a)**(cas général)**
2. La probabilité ϵ dépend d'une cellule seulement $\epsilon = \varepsilon_c$, figure 5(b)
3. La probabilité ε dépend seulement de la carte $\mathcal{C}$ (une seule valeur pour toute la carte), figure 5(c)

Nous observons que les valeurs des probabilités permettent de détecter les variables pertinentes. On arrive visuellement à reconnaître les chiffres. Par conséquent, il est possible d'utiliser ces valeurs pour caractériser les groupes. Dans le deuxième cas, il n'est pas possible de caractériser la forme des chiffres, mais nous pouvons caractériser les groupes. Ainsi, il est possible de sélectionner des groupes d'individus et de réaliser un échantillonnage optimisé.

Dans le troisième cas, une seule valeur réelle est estimée pour toute la carte. Afin de montrer l'intérêt de ce type de modèle nous avons appris plusieurs cartes avec des initialisations différentes et par la suite nous avons calculé la pureté associée à chacune de ces cartes. La figure 5(c) indique la variation de la pureté selon la probabilité ε estimée par le modèle. Plus la probabilité est faible plus le modèle est bon. Nous observons que les modèles estimant des probabilités supérieures à 0.5 fournissent des puretés faibles. A l'opposé, ceux estimant des probabilités inférieures à 0.5 correspondent à des puretés fortes. Il convient donc de choisir comme meilleur modèle celui qui est associé à la valeur de probabilité la plus faible. Ce type d'application montre qu'il est possible d'utiliser ce principe pour faire de la sélection de modèles.

5.1.4 Autres bases de données

Nous avons comparé notre modèle PrMTM avec la version classique des cartes topologiques binaires. Nous avons utilisé deux autres bases :

La maladie du cancer du sein (Wisconsin Breast Cancer Data)
Ce jeu de données contient 699 observations avec 9 variables. Chaque observation est étiquetée bénigne (458 ou 65.5%) ou maligne (241 ou 34.5%). Dans notre cas, toutes les variables sont considérées catégorielles avec valeurs entre $1, 2, ..., 10$ et codées en binaire (Annexe A).

La base de données "Vote" (Congressional Vote Data)
C'est un jeu de données qui contient 435 observations, dont 168 appartiennent à une classe et 267 appartiennent à la deuxième classe.

Le tableau 3 indique le taux de l'erreur de classification obtenu avec les deux méthodes. Nous pouvons observer que les résultats utilisant le modèle BinBatch et PrMTM$-\epsilon$ sont généralement similaires, bien qu'ils soient meilleurs avec PrMTM$-\epsilon$. Comme nous l'avons signalé

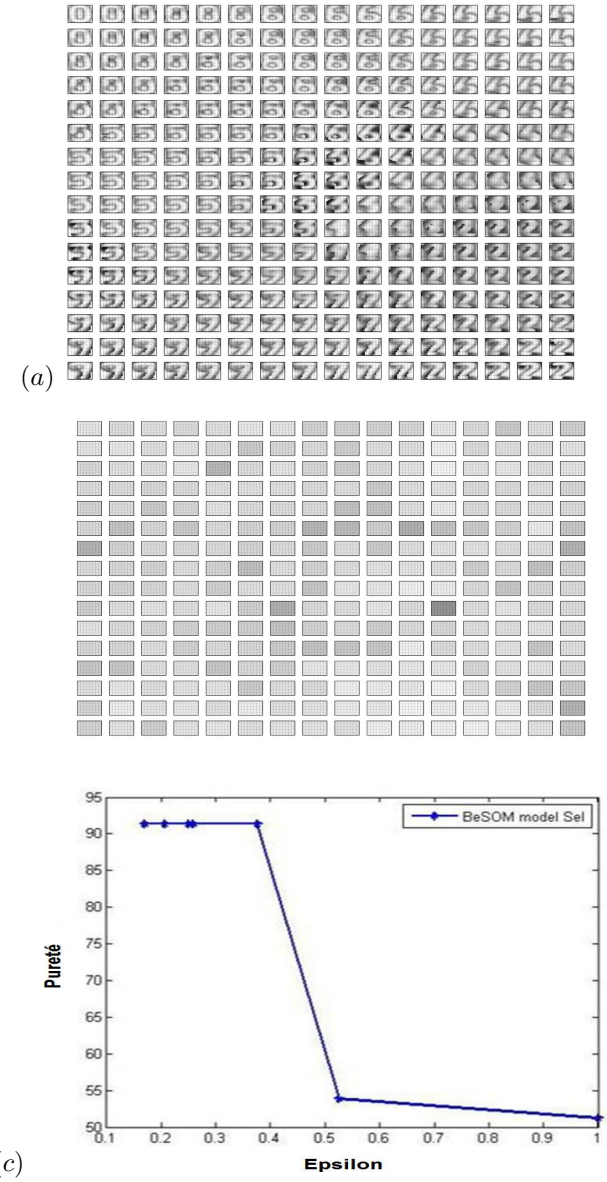


FIG. 5 – *Dans la figure (a) chaque image représente un vecteur de probabilité. Dans la figure (b) on voit la probabilité par cellule, sur la figure (c) on voit la probabilité pour toute la carte.*

dans les sections précédentes le modèle PrMTM fournit plus d'informations que le modèle déterministe BinBatch (qui représente les cartes classiques utilisant la distance de Hamming).

Le jeu de données / Err	BinBatch	**PrMTM**$-\epsilon$
Wisconsin-B-C	3.87%	**2.43%**
Zoo	2.97%	**1.98%**
Congressional vote	5.91%	**5.77%**

TAB. 3 – *La comparaison des perfomances de classification de PrMTM*$-\epsilon$*, et BinBatch.*

5.2 Application aux données mixtes

5.2.1 La base de données "Cleve"

Dans cette partie, nous illustrons la convergence de l'algorithme PrMTM sur la base médicale *cleve*. Cette base a été fournie par le Dr. Detrano en 1988. Elle décrit la maladie du coeur traitée dans la clinique de Cleveland. L'ensemble des données contient 303 patients où chacun est décrit par 6 variables quantitatives et 7 qualitatives. Les patients sont classés en deux classes, représentant la présence de la maladie ou son absence. En utilisant un codage disjonctif binaire on obtient 17 variables binaires.

L'apprentissage d'une carte de dimension 13×7, fournit pour chaque cellule des vecteurs référents avec deux parties $(\mathbf{w}_c^{r[.]}, \mathbf{w}_c^{b[.]})$ associées respectivement à l'écart-type σ_c pour la partie quantitative, et la probabilité ε_c d'être différent du prototype pour la partie binaire. Notre modèle PrMTM nous a permis de disposer des mêmes visualisations que les cartes topologiques traditionnelles. Nous rappelons que le modèle PrMTM utilise une affectation probabiliste pour projeter les observations sur la carte. La figure 6 illustre la variation de la partie quantitative correspondant à toutes les observations $\mathbf{x}^{r[.]}$ captées par chaque cellule. On observe une auto-organisation de la partie quantitative malgré la présence de la partie binaire lors de la phase d'apprentissage.

La figure 7 affiche la probabilité ε_c d'être différent de la partie binaire $\mathbf{w}^{b[.]}$ indiquée par la partie binaire (figure 8). La nuance de gris de chaque prototype est proportionnelle à la probabilité d'être différent du prototype binaire estimé. Cette information est très importante pour l'interprétation des résultats. La probabilité ε indique la confiance qu'on peut faire aux variables binaires et par conséquent aux variables qualitatives. Lorsque la probabilité est très forte, l'expert peut décider d'ignorer l'interprétation des variables binaires correspondantes.

5.2.2 Autres données réelles

Dans cette section nous montrons les apports de notre modèle relativement à l'algorithme déterministe de classification topologique MTM. Pour la suite on utilise deux autres bases obtenues du répertoire UCI (Blake et Merz, 1998). Le tableau 4, fournit une courte description des bases de données utilisées.

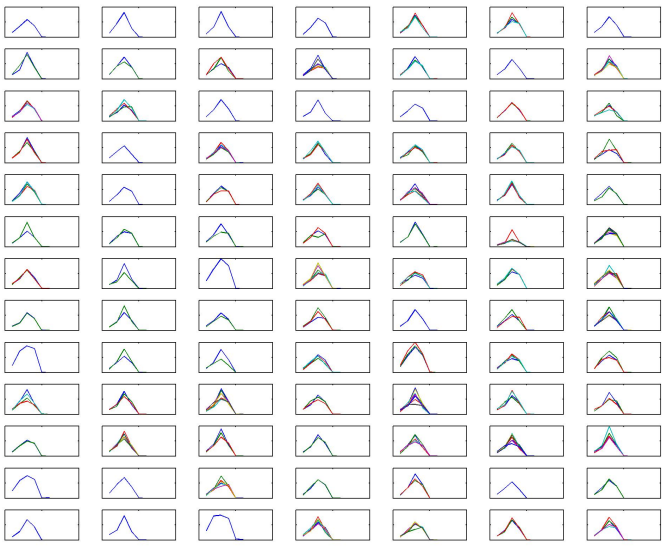


FIG. 6 – *La carte PrMTM pour la base de données Cleve. Chaque cellule indique la partie quantitative des observations captées par chaque cellule.*

Base de données	dim. qualit	dim. quantitative	# obs	# classes
Cleve	7	6	303	2
Credit	6	9	666	2
Thyroid	12	7	3163	2

TAB. 4 – *Jeux de données utilisées pour l'évaluation. # obs : nombre d'observation ; # classes : nombre de classes.*

Nous avons comparé notre modèle PrMTM avec l'algorithme classique déterministe MTM. Dans ces expérimentations, la comparaison des différents résultats est mesurée à l'aide du taux de pureté en utilisant l'étiquette connue de chaque observation. La comparaison est réalisée en calculant la moyenne des puretés sur 50 expériences. Le tableau 5 montre les performances atteintes par notre modèle PrMTM et part le modèle MTM. Nous observons une amélioration des puretés sur toutes les bases.

En examinant le tableau 5, nous observons par exemple, pour la base *Cleve* une amélioration de la pureté de 86.78% à 87.25% en réduisant l'écart-type de 2.62 à 1.95. Pour la base de données *Credit* nous observons des résultats équivalents, mais en réduisant l'écart-type de 2.46 à 1.96. En ce qui concerne la base de données *Thyroid* on observe une amélioration de pureté avec une légère augmentation de l'écart-type.

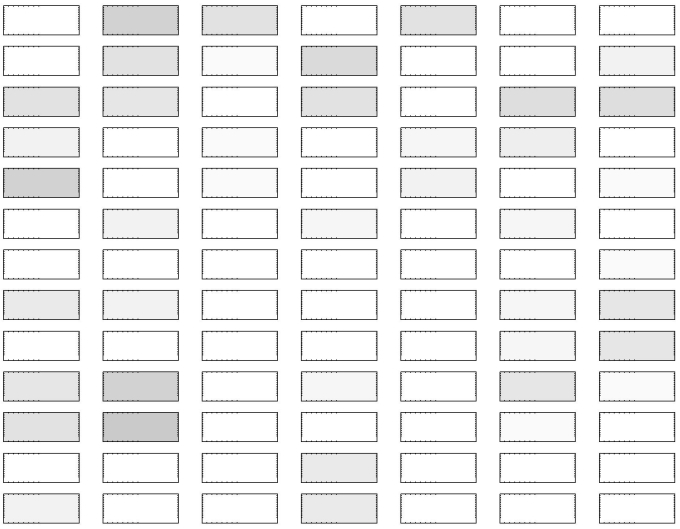


FIG. 7 – *La carte PrMTM pour les données Cleve de dimension* 13×7. *La carte indique la probabilité* ε_c *(valeur par cellule). Le niveau de gris de chaque cellule est proportionnel à la probabilité d'être différent du prototype binaire estimé* $\mathbf{w}_c^{b[.]}$.

Pureté : %	MTM	PrMTM
Cleve (13×7)	86.78 ± 2.62	87.25 ± 1.95
Credit (13×10)	81.60 ± 2.46	81.93 ± 1.96
Thyroid (21×14)	95.85 ± 1.06	96.22 ± 1.09

TAB. 5 – *Comparaison entre MTM et PrMTM utilisant l'indice de pureté (moyenne*$\pm$ *écart-type sur 50 expérimentations). MTM : Carte topologique classique dédiées aux données mixtes. PrMTM : carte topologique utilisant la loi Gaussienne et de Bernoulli*

Pour valider notre approche, nous avons utilisé la validation croisée. Par conséquent, nous avons divisé les bases en 3 sous-ensembles (un tiers de la base est utilisé pour la validation et deux tiers pour l'apprentissage). Ainsi, 15 expériences sont réalisées pour chaque base. Le tableau 6 indique la moyenne calculée avec 15 expériences. Pour le même objectif de comparaison, nous avons enrichi le modèle PrMTM avec l'estimation d'un vecteur de probabilités par cellule au lieu d'une valeur par cellule. Ce modèle fait référence au modèle modèle PrMTM-ϵ_c dédié aux données binaires où la probabilité dépend à la fois de la cellule et de la variable, $\epsilon_c = (\varepsilon_c^1, ..., \varepsilon_c^k, ..., \varepsilon_c^n)$. Nous appelons par la suite ce modèle PrMTM-ϵ_c.

Nous observons clairement que le modèle PrMTM fournit de meilleurs résultats que le modèle

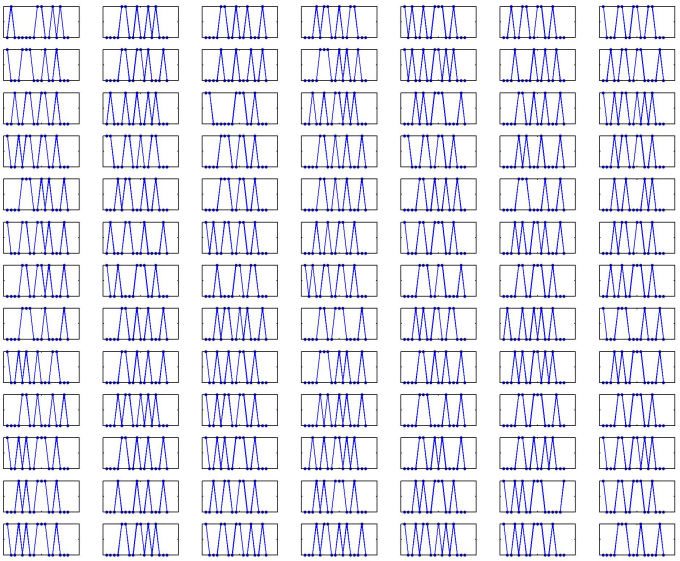


FIG. 8 – *La carte PrMTM pour les données Cleve de dimension* 13×7. *La carte indique les profils de la partie binaire ("1/0"qualitative).*

Taux de classification : %	MTM	PrMTM-ϵ_c	PrMTM
Cleve (13×7)	72.78	75.26	74.93
Credit (13×10)	62.76	64.58	63.34
Thyroid (21×14)	82.75	85.41	84.69

TAB. 6 – *Comparaison entre MTM, PrMTM et PrMTM-*ϵ_c *utilisant la technique de la validation croisée. On indique le taux de classification pour chaque base. MTM : Carte topologique classique dédiées aux données mixtes. PrMTM : carte topologique utilisant la loi Gaussienne et de Bernoulli (où la probabilité est calculée pour chaque cellule de la carte). PrMTM-*ϵ_c *: carte topologique utilisant la loi Gaussienne et de Bernoulli (où le vecteur probabilité* ϵ_c *dépend de la cellule* c *et de la variable,* $\epsilon_c = (\varepsilon_c^1, ..., \varepsilon_c^k, ..., \varepsilon_c^n)$*)*

déterministe. Ceci est rassurant puisque le modèle PrMTM est plus riche en information. Il est clair aussi que lorsqu'on estime un vecteur de probabilité pour la partie binaire "PrMTM-ϵ_c", les performances augmentent. Comme nous l'avons signalé dans les sections précédentes, le modèle PrMTM fournit plus d'informations que le modèle déterministe MTM. Pour le modèle proposé, la probabilité ε_c et l'écart-type σ_c peuvent être utilisés pour la sélection des modèles. Supposant que la probabilité et l'écart-type dépendent des prototypes et des variables, on estime pour les deux distributions, le vecteur d'écart-type $\sigma_c = (\sigma_c^1, \sigma_c^2, ..., \sigma_c^n)$ et le vecteur de probabilités $\epsilon_c = (\varepsilon_c^1, \varepsilon_c^2, ..., \varepsilon_c^m)$. Ainsi ce modèle peut-être utilisé pour la sélection non-

supervisée des variables. Il n'est pas évident de comparer notre modèle avec un algorithme qui n'utilise pas une architecture similaire. Nous rappelons que le modèle PrMTM, basé sur l'architecture SOM, fournit plus de clusters que les modèles hiérarchiques ou le K-means dédié aux données mixtes (Ahmad et Dey, 2007).

6 Conclusion

Les travaux présentés dans cet article nous ont permis d'analyser les propriétés mathématiques de l'algorithme des cartes auto-organisatrices dans un cadre probabiliste. Tenir compte de la distribution des données implique l'intégration de plus d'informations dans le modèle des cartes topologiques selon le type de données et donc l'amélioration des performances. Nous avons proposé un nouveau modèle d'apprentissage qui tient compte de la distribution locale des données. Le modèle, PrMTM (Probabilistic Mixed Topological Map) dédié aux données mixtes, utilise le formalisme probabiliste des cartes topologiques et associe simultanément à chaque cellule de la carte la distribution Gaussienne et la distribution de Bernoulli. Dans le cas de la distribution de Bernoulli, chaque cellule est caractérisée par un prototype avec le même codage binaire de l'espace d'entrée et par la probabilité d'être différent de ce prototype. Cette approche PrMTM dédiée aux données catégorielles et mixtes (quantitatives et binaires) utilise l'algorithme EM pour maximiser la vraisemblance de données afin d'estimer les paramètres d'un modèle de mélange des lois de Bernoulli et Gausiennes. Cet algorithme a l'avantage de fournir des prototypes qui sont de même nature que les données initiales. Nous avons montré que l'algorithme PrMTM nous fournit différentes informations qui peuvent être utilisées dans des applications pratiques. La complexité de notre algorithme d'apprentissage est de $O(NK^2)$. Il est K fois plus lent que l'algorithme MTM et peut-être plus lent pour de grandes bases de données. Toutefois, en limitant le voisinage au cours de l'apprentissage, nous pouvons obtenir un faible coût de calcul. Une autre façon de réduire le coût de calcul est d'introduire une étape d'affectation au lieu d'attribuer les observations à la fin de l'apprentissage (Celeux et Govaert, 1992).

Plusieurs perspectives de développement peuvent être envisagées comme éventuellement l'extension du modèle PrMTM à la classification croisée en utilisant le formalisme probabiliste des cartes topologiques. Nous pouvons envisager un critère entre groupes d'observations et groupes de variables qui découle de la fonction de vraisemblance.

Références

Ahmad, A. et L. Dey (2007). A k-mean clustering algorithm for mixed numeric and categorical data. *Data Knowl. Eng. 63*(2), 503–527.

Andreopoulos, B., A. An, et X. Wang (2006). Bi-level clustering of mixed categorical and numerical biomedical data. *International Journal of Data Mining and Bioinformatics 1*(1), 19 – 56.

Anouar, F. (1996). Modélisation probabiliste des auto-organisées : Application en classification et en régression. thèse de doctorat soutenue au conservatoires national des arts et métiers.

Anouar, F., F. Badran, et S. Thiria (1997). Self-organizing map, a probabilistic approach. In *Proceedings of WSOM'97-Workshop on Self-Organizing Maps, Espoo, Finland June 4-6*, pp. 339–344.

Aupetit, M. (2005). Learning topology with the generative gaussian graph and the em algorithm. In *NIPS*, pp. 592–598.

Bishop, C. M. (1995). Neural networks for pattern recognition. *Clarrendon Press, Oxford*.

Bishop, C. M., M. Svensén, et C. K. I.Williams (1998). Gtm : The generative topographic mapping. *Neural Comput 10*(1), 215–234.

Blake, C. et C. Merz (1998). Uci repository of machine learning databases. technical report.

Celeux, G. et G. Govaert (1992). A classification em algorithm for clustering and two stochastic versions. *Comput. Stat. Data Anal. 14*(3), 315–332.

Dempster, A., N. Laird, et D. Rubin (1977). Maximum likelihood from incomplete data via the em algorithm. *Roy. Statist. Soc 39*(1), 1–38.

Girolami, M. (2001). The topographic organisation and visualisation of binary data using mutivariate-bernoulli latent variable models. *I.E.E.E Transactions on Neural Networks 12*(6), 1367–1374.

Graepel, T., M. Burger, et K. Obermayer (1998). Self-organizing maps : generalizations and new optimization techniques. *Neurocomputing 21*, 173–190.

Heskes, T. (2001). Self-organizing maps, vector quantization, and mixture modeling. *IEEE Trans. Neural Networks 12*, 1299–1305.

Jain, A. K. et R. C. Dubes (1988). *Algorithms for clustering data*. Upper Saddle River, NJ, USA : Prentice-Hall, Inc.

Kaban, A. et M. Girolami (2001). A combined latent class and trait model for the analysis and visualization of discrete data. *IEEE Trans. Pattern Anal. Mach. Intell 23*, 859–872.

Khan, S. S. et S. Kant (2007). Computation of initial modes for k-modes clustering algorithm using evidence accumulation. In *IJCAI*, pp. 2784–2789.

Kohonen, T. (2001). *Self-organizing Maps*. Springer Berlin.

Lebbah, M. (2003). *Carte topologique pour données qualitatives : application à la reconnaissance automatique de la densité du trafic routier*. Thése de doctorat en Informatique, Université de Versailles Saint-Quentin en Yvelines.

Lebbah, M., A. Chazottes, F. Badran, et S. Thiria (2005). Mixed topological map. In *ESANN*, pp. 357–362.

Lebbah, M., S. Thiria, et F. Badran (2000). Topological map for binary data. In *Proceedings European Symposium on Artificial Neural Networks-ESANN 2000, Bruges, April 26-27-28*, pp. 267–272.

Luttrel, S. P. (1994). A bayesian ananlysis of self-organizing maps. *Neural Computing 6*, 767 – 794.

McLachlan, G. et T. Krishman (1997). *The EM algorithm and Extensions*. Wiley, New York.

Priam, R. et M. Nadif (2006). Carte auto-organisatrice probabiliste sur données binaires. In *EGC*, pp. 445–456.

Priam, R., M. Nadif, et G. Govaert (2008). Binary block gtm : Carte auto-organisatrice probabiliste pour les grands tableaux binaires. In *Extraction et gestion des connaissances (EGC'2008), Actes des 8èmes journées Extraction et Gestion des Connaissances*, Revue des Nouvelles Technologies de l'Information, Sophia-Antipolis, France,, pp. 265–272. Cépaduès-Éditions.

Thiria, S., Y. Lechevallier, O. Gascuel, et S. Canu (1997). Statistique et méthodes neuronales.

Verbeek, J., N. Vlassis, et B. Kröse (2005). Self-organizing mixture models. *Neurocomputing 63*, 99–123.

Summary

This paper introduces a method based on probabilistic self-organizing maps dedicated for topographic clustering, analysis and visualization of categorical and mixed data. For each kind of data we propose a probabilistic formalism in which cells are represented by a mixture model of Bernoulli distribution in the case of binary data and by a mixture model of Bernoulli and Gaussian laws in the case of mixed data. Each cell is characterized by a prototype with the same binary coding as used in the data space and the probability of being different from this prototype. The learning algorithm, PrMTM, that we propose is an application of the EM standard algorithm. We illustrate the power of this method with several binary and mixed data sets taken from a public data set repository. The results show a good quality of the topological ordering and homogenous clustering.