Classification topologique probabiliste pour des données catégorielles

Nicoleta Rogovschi et Mohamed Nadif*

LIPADE, Université Paris Descartes
45, rue des Saints Pères, 75006 Paris, France
Prénom.Nom@parisdescartes.fr

Résumé. Cet article présente une carte auto-organisatrice probabiliste pour l'analyse et la classification topologique des données catégorielles. En considérant un modèle de mélanges parcimonieux nous introduisons une nouvelle carte auto-organisatrice (SOM) probabiliste. L'estimation des paramètres de notre modèle est réalisée à l'aide de l'algorithme EM classique. Contrairement à SOM, l'algorithme d'apprentissage proposé optimise une fonction objective. Ces performances ont été évaluées sur des données réelles et les résultats obtenus sont encourageants et prometteurs à la fois pour la classification et pour la modélisation.

1 Introduction

Data visualization is an important step in the exploratory phase of data analysis. This step is more difficult when it involves binary data and categorical variables (Andreopoulos et al., 2006; Saund, 1995). Self-organizing maps are being increasingly used as tools for visualization, as they allow projection over small areas that are generally two dimensional. The basic model proposed by (Kohonen, 2001), was only designed for numerical data, but it has been successfully applied to treating textual data, (Kaski et al., 1998). This algorithm has also been applied to binary data following transformation of the original data (Ibbou et Cottrell, 1995; Lebbah et al., 2000). Developing generative models of the Kohonen map has long been an important goal. These models vary in the form of the interactions, and they assume the hidden generators may follow in generating the observations. Some extensions and reformulations of the Kohonen model have been described in the literature. They include probabilistic self-organizing maps (Anouar et al., 1997) which define a map as a gaussian mixture and use the maximum likelihood approach to define an iterative algorithm.

In Verbeek et al. (2005), the authors propose a probabilistic generalization of Kohonen's SOM which maximizes the variational free-energy that sums data log-likelihood and Kullback-Leibler divergence between a normalized neighbourhood function and the posterior distribution on the given data for the components. We have also Soft topographic vector quantization (STVQ), which uses some measure of divergence between data items and cells to minimize a new error function (Heskes, 2001; Graepel et al., 1998). Another model, often presented as the probabilistic version of the self-organizing map, is the Generative Topographic Map (GTM) (Bishop et al., 1998; Kaban et Girolami, 2001). However, the manner in which GTM achieves the topographic organization is quite different from those used in the SOM models. In GTM mixture components are parameterized by a linear combination of nonlinear functions of the locations of the components in the latent space. The GTM was developed for continuous data. A specific GTM model was subsequently developed for binary data by adopting a variational approximation to the binomial likelihood (Graepel et al., 1998).

Also, in Kaban et al. (2004), the authors concentrate on modelling binary coded data where only the presence or absence of a variable is of interest. In contrast to other approaches, the model is linear. The model is seen as a Bernoulli analogue of the multinomial decomposition model. In Jollois et Nadif (2007), the main of the proposed method is to speed-up convergence of EM, and second to yield same results (or not so far) than traditional EM using categorical data. Others similar techniques have been developed to cluster large data sets (Kostiainen et Lampinen, 2002; Hofmann, 2001).

Here, we concentrate on modelling qualitative data using binary coding. This model involves use of the probabilistic formalism of the topological map used in (Anouar et al., 1997); therefore, it consists of estimating the parameters of the model by maximizing the likelihood of the data set. The learning algorithm that we propose is an application of the EM standard algorithm, (McLachlan et Krishman, 1997). Some variants are proposed to speed-up EM in reducing the time spent in the E-step in the case of categorical data, (Jollois et Nadif, 2007). In this paper we proposed a new method called WeCSOM (Weighted Categorical Self-Organizing Map) which combine the benefits of SOMs, K-mode (Huang, 1998) algorithm and mixture models to design a new mixture for categorical data. This approach is based on the model of the self-organizing maps and uses a parsimonious mixture models which has the advantage of being directly applicable to the categorical data without using a specific encoding a priori. The proposed learning algorithm is an application of the classical EM algorithm that allows to weight the variables considering the number of modes of each one during the learning process, thus achieving an optimized classification of the data.

The rest of this paper is organized as follows: we present the principle of probabilistic map and categorical data in section 2. Our proposed approach is presented in sections 2.1 and 2.2. In sections 3, we present different results and, finally the paper ends with a conclusion and some future works for the proposed methods.

2 Categorical data and Probabilistic self-organizing map

As with a traditional self-organizing map, we assume that the lattice $\mathcal{C}$ has a discrete topology (discrete output space) defined by an undirect graph. Usually, this graph is a regular grid in one or two dimensions. We denote the number of cells in $\mathcal{C}$ as N_{cell}. For each pair of cells (c,r) on the map, the distance $\delta(c,r)$ is defined as the length of the shortest chain linking cells r and c.

2.1 General probabilistic formalism

To define the model of topological maps based on mixture models we associate to each cell c of the map $\mathcal{C}$ a density function $\mathbf{f}_c(\mathbf{x}) = p(\mathbf{x}|\theta_c)$ whose parameters are denoted by θ. Following the bayesian formalism, presented in (Luttrel, 1994; Anouar et al., 1997), we assume that each observation $\mathbf{x}$ is generated by the following process: We start by associating to each cell $c \in \mathcal{C}$ a probability $p(\mathbf{x}|c)$ where $\mathbf{x}$ is a vector in the data space. Next, we pick a cell c^* from $\mathcal{C}$ according to the prior probability $p(c^*)$. For each cell c^*, we select an associated cell $c \in \mathcal{C}$ following the conditional probability $p(c|c^*)$. All cells $c \in \mathcal{C}$ contribute to the generation of $\mathbf{x}$ with $p(\mathbf{x}|c)$ according to the proximity to c^* described by the probability $p(c|c^*)$. Thus, a high proximity to c^* implies a high probability $p(c|c^*)$, and therefore the contribution of c to the generation of $\mathbf{x}$ is high.

Due to the "Markov" property, $p(\mathbf{x}|c,c^*) = p(\mathbf{x}|c)$, the probability distribution of the observations generated by a cell c^* of $\mathcal{C}$ is a mixture $p_{c^*}(\mathbf{x}|c^*)$ of probabilities completely defined from the map as:

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

The generative model considers the mixture of probabilities, given by :

$$p(\mathbf{x}) = \sum_{c,c^* \in C} p(c,c^*,\mathbf{x}) = \sum_{c,c^* \in C} p(\mathbf{x}|c)p(c|c^*)p(c^*) = \sum_{c^* \in \mathcal{C}} p(c^*)p_{c^*}(\mathbf{x}), \tag{1}$$

with

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

where the conditional probability $p(c|c^*)$ is assumed to be known. To introduce the self-organizing process in the mixture model learning, we assume that $p(c|c^*)$ can be defined as:

$$p(c|c^*) = \frac{K^T(\delta(c,c^*))}{\sum_{r \in \mathcal{C}} K^T(\delta(r,c^*))},$$

where K^T is a neighbourhood function depending on the parameter T (called temperature): $K^T(\delta) = K(\delta/T)$, where K is a particular kernel function which is positive and symmetric ($\lim_{|x| \to \infty} K(x) = 0$). Thus K defines for each cell c^* a neighbourhood region in $\mathcal{C}$. The parameter T allows control of the size of the neighbourhood influencing a given cell on the map. As with the Kohonen algorithm, we decrease the value of T between two values T_{max} and T_{min}.

2.2 The proposed model

In the following, let we focus on categorical data. Let be a set of N instances $\mathbf{x}_1, \ldots, \mathbf{x}_N$ described by n categorical attributes $\mathbf{x}^1, \ldots, \mathbf{x}^n$. The data matrix is noted x and defined by $\mathbf{x} = \{(x_i^j); i = 1, \ldots, N; k = 1, \ldots, n\}$. Each instance $_i$ is represented as $[x_i^1, \ldots, x_i^n]$ and for each attribute $\mathbf{x}^j$, we note c^j the number of categories. We consider a restricted latent class model [16], then the conditional distribution in $p(\mathbf{x}_i|c)$ is now given as the product of univariate single distributions

$$p(\mathbf{x}_i|c) = f_c(\mathbf{x}_i|\mathbf{w}_c,\epsilon_c) = \prod_{k=1}^{n} f_c(x_i^k|w_c^k,\epsilon_c^k),$$

where $\mathbf{w}_c = (w_c^1, \ldots, w_c^n)$ represents the vector of categories and $\epsilon_c = (\varepsilon_c^1, \ldots, \varepsilon_c^n)$ is a vector of probabilities. Taking

$$f_c(x_i^k|w_c^k,\varepsilon_c^k) = (1 - \varepsilon_c^k)^{1-d(x_i^k,w_c^k)} \left(\frac{\varepsilon_c^k}{c^k - 1}\right)^{d(x_i^k,w_c^k)},$$

where $d(a,b) = 1$ if $a = b$ and 0 otherwise, we define a parsimonious model where the parameter c consists of $(\mathbf{w}_c,\epsilon_c)$ with $\mathbf{w}_c$ is the mode of the the component and ϵ_c is a k-dimensional vector of probabilities indicating the degree of heterogeneity. The density $f_c(\mathbf{x}_i|\mathbf{w}_c,\epsilon_c)$ expresses that, for c, the attribute $\mathbf{x}^k$ takes category w_c^k with the greatest probability $(1 - \epsilon_c^k)$ and takes each other category with the same probability $\frac{\epsilon_c^k}{(c^k-1)}$. Note that, setting the clustering problem under the classification maximum likelihood approach, the authors in [16] have defined a generalization of the $k$modes criterion and proposed better fit criteria. In our situation, we can assume that the parameter ε_c^k depends only on a cell $c \in \mathcal{C}$. Then, the model mixture generator becomes:

$$p(\mathbf{x}) = \sum_{c^* \in \mathcal{C}} p(c^*) \sum_{c \in \mathcal{C}} p(c|c^*) f_c(\mathbf{x},\mathbf{w}_c,\epsilon_c). \tag{3}$$

Therefore, the parameters $\theta = \theta^{\mathcal{C}} \cup \theta^{\mathcal{C}^*}$ which define the model mixture generator (3) are constituted of the parameters ($\theta^{\mathcal{C}} = \{\theta^c,c = 1..N_{cell}\}$, where $\theta^c = (\mathbf{w}_c,\epsilon_c)$), and all the prior probabilities, also called mixing coefficients ($\theta^{\mathcal{C}^*} = \{\theta^{c^*},c^* = 1..N_{cell}\}$ where $\theta^{c^*} = p(c^*)$). The difficulty now is to define the cost function and the learning algorithm for estimating all these parameters dedicated to categorical data. Our WeCSOM algorithm was inspired by a probabilistic SOM model proposed by (Anouar et al., 1997) and represents a generalization of the model proposed by (Lebbah et al., 2007).

2.3 Cost function and optimization algorithm

The learning algorithm is based on maximizing the likelihood of the observations by applying the EM algorithm (Dempster et al., 1977). Learning is facilitated by introducing N hidden variables $\Xi = (\xi_1, \ldots ,\xi_N)$; each hidden variable $\xi = (c,c^*)$ indicates which of the cell pairs c and c^*, generate the corresponding data observation $\mathbf{x}$. We introduce the hidden variable $\xi = (c,c^*)$ in expression (3):

$$p(\mathbf{x}) = \sum_{\xi \in \mathcal{C}\times\mathcal{C}} p(\mathbf{x},\xi) = \sum_{c,c^* \in \mathcal{C}} p(c^*)p(c|c^*) f_c(\mathbf{x},\mathbf{w}_c,\varepsilon_c). \tag{4}$$

We define a binary indicator variable $\alpha_i^{(c,c^*)}$ which indicates the hidden generator that may follow in generating the observation $\mathbf{x}_i$ as: $\alpha_i^{(c,c^*)} = \begin{cases} 1 & \text{for } \xi_i = (c,c^*) \\ 0 & \text{otherwise} \end{cases}$. Using expression (4), and the binary indicator $\alpha_i^{(c,c^*)}$, we can define the classification likelihood of the observations using the hidden variables as follows:

$$L^T(,\Xi;\theta) = \prod_{i=1}^{N} \prod_{c^* \in \mathcal{C}} \prod_{c \in \mathcal{C}} \left[\theta^{c^*} p(c|c^*) f_c(\mathbf{x},\mathbf{w}_c,\epsilon_c)\right]^{\alpha_i^{(c,c^*)}}.$$

The log-likelihood becomes:

$$\ln L^T(,\Xi;\theta) = \sum_{i=1}^{N} \sum_{c,c^* \in \mathcal{C}} \alpha_i^{(c,c^*)} \left[\ln(\theta^{c^*}) + \ln\left(\frac{K^T(\delta(c^*,c))}{T_{c^*}}\right) + \ln(f_c(\mathbf{x},\mathbf{w}_c,\epsilon_c))\right],$$

where $T_{c^*} = \sum_{r \in \mathcal{C}} K^T(\delta(r,c^*))$. The application of the EM algorithm [7] for the maximization of log-likelihood requires $Q^T(\theta^t,\theta^{t-1})$ to be maximised for a fixed temperature T defined as:

$$Q^T(\theta^t,\theta^{t-1}) = E\left[\ln L^T(,\Xi;\theta^t)|,\theta^{t-1}\right],$$

where θ^t is the set of the parameters estimated at the t^{th} step of the learning algorithm. However, the E-step calculates the expectation of log-likelihood with respect to the hidden variable while maintaining the established parameter θ^{t-1}. During the M-step, after updating $Q^T(\theta^t,\theta^{t-1})$ from the previous step, we maximize the $Q^T(\theta^t,\theta^{t-1})$ with respect to θ^t, ($\theta^t = \arg\max_\theta(Q^T(\theta,\theta^{t-1}))$). The two-steps increase the function likelihood. The function $Q^T(\theta^t,\theta^{t-1})$ is defined as:

$$\begin{aligned} Q^T(\theta^t,\theta^{t-1}) &= \sum_{i=1}^{N}\sum_{c^* \in \mathcal{C}}\sum_{c \in \mathcal{C}} E(\alpha_i^{(c,c^*)}|\mathbf{x}_i,\theta^{t-1}) \\ &\times \left[\ln(\theta^{c^*}) + \ln\left(\frac{K^T(\delta(c^*,c))}{T_{c^*}}\right) + \ln(f_c(\mathbf{x},\mathbf{w}_c,\epsilon_c))\right] \end{aligned}$$

where $E(\alpha_i^{(c,c^*)}|\mathbf{x}_i,\theta^{t-1}) = p(\alpha_i^{(c,c^*)} = 1|\mathbf{x}_i,\theta^{t-1}) = p(c,c^*|\mathbf{x}_i,\theta^{t-1})$, with

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

The function $Q^T(\theta^t,\theta^{t-1})$ breaks into three terms

$$Q^T(\theta^t,\theta^{t-1}) = Q_1^T(\theta^{\mathcal{C}},\theta^{t-1}) + Q_2^T(\theta^{\mathcal{C}^*},\theta^{t-1}) + Q_3^T(\theta^{t-1}) \qquad (5)$$

where

$$Q_1^T(\theta^{\mathcal{C}},\theta^{t-1}) = \sum_{k=1}^{n}\sum_{i=1}^{N}\sum_{c \in \mathcal{C}}\sum_{c^* \in \mathcal{C}} p(c,c^*|\mathbf{x}_i,\theta^{t-1})\ln(f_c(x^k,w_c^k,\varepsilon_c^k)),$$

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

$$Q_3^T(\theta^{t-1}) = \sum_{i=1}^{N}\sum_{c^* \in \mathcal{C}}\sum_{c \in \mathcal{C}} p(c,c^*|\mathbf{x}_i,\theta^{t-1})\ln\left(\frac{K^T(\delta(c^*,c))}{T_{c^*}}\right).$$

The parameters $\theta^{\mathcal{C}}$ and $\theta^{\mathcal{C}^*}$ indicate the parameters estimated at the t^{th} step. The first term $Q_1^T(\theta^{\mathcal{C}},\theta^{t-1})$ depends on $\theta^{c,k} = (w_c^k,\varepsilon_c^k)$; the second term $Q_2^T(\theta^{\mathcal{C}^*},\theta^{t-1})$ depends on θ^{c^*}, and the third term is constant. Maximizing $Q^T(\theta^t,\theta^{t-1})$ with respect to θ^{c^*} and θ^c can be performed separately including the parameter $\mathbf{w}_c$ and ϵ_c. The maximization of $Q^T(\theta^t,\theta^{t-1})$ leads to the updates that are calculated using the parameters estimated at the $t-1^{th}$ step. The expressions are defined as follows:

$$\theta^{c^*} = p(c^*) = \frac{\sum_{\mathbf{x}_i \in \mathcal{A}} p(c^*|\mathbf{x}_i,\theta^{t-1})}{N} \qquad (6)$$

where
$p(c^*|\mathbf{x}_i,\theta^{t-1}) = \sum_{c \in \mathcal{C}} p(c,c^*|\mathbf{x}_i,\theta^{t-1})$ and $p(c|\mathbf{x}_i,\theta^{t-1}) = \sum_{c^* \in \mathcal{C}} p(c,c^*|\mathbf{x}_i,\theta^{t-1})$. Each component of $\mathbf{w}_c = (w_c^1,\ldots,w_c^k,\ldots,w_c^n)$ and $\epsilon_c = (\varepsilon_c^1,\varepsilon_c^2,\ldots,\varepsilon_c^k,\ldots,\varepsilon_c^n)$ is then computed as follows:

$$w_c^k =_{e=1,\ldots,c^k} \sum_{i=1}^{N} p(c|\mathbf{x}_i,\theta^{t-1}) d(x_i^k,w_c^k) \tag{7}$$

and

$$\varepsilon_c^k = \frac{\sum_{i=1}^{N} p(c|\mathbf{x}_i,\theta^{t-1}) d(x_i^k,w_c^k)}{\sum_{i=1}^{N} p(c|_i,\theta^{t-1})}, \tag{8}$$

The application of EM for the maximization gives rise to the iterative algorihtm of WeCSOM. The version of the WeCSOM algorithm for a fixed T parameter is presented in the following way:

Algorithm 1 Principal stages of the learning algorithm WeCSOM

1. **Initialization** (iteration t = 0) Choose the initial parameters (θ^0) and the number of iterations N_{iter}.
2. **Basic Iteration at a constant** T (iteration $t \geq 1$) Calculate all the parameters $\theta^t = \{\theta^{c^*},\mathbf{w}_c,\epsilon_c\}$ from the previous parameters θ^{t-1} associated with each cell c and c^* by applying the formulas: (6), (7) and (8).
3. **Repeat** the basic iteration until $t > N_{iter}$.

The WeCSOM learning algorithm allows us to estimate the parameters maximizing the log-likelihood function for a fixed T. As in the SOM algorithm, we decrease the value of T between two values T_{max} and T_{min}, to control the size of the neighbourhood influencing a given cell on the map. For each T value, we get a likelihood function L^T, and therefore the expression varies with T. When decreasing T, the learning algorithm of WeCSOM is defined in the Algorithm 2.

Algorithm 2 Algorithm WeCSOM varying T

1. **Initialization Phase** (iteration $t = 0$): Choose T_{max}, T_{min} and N_{iter}. Apply the principal stages of WeCSOM algorithm described above for the value of T fixed to T_{max}.
2. **Iterative step:** We assume that the previous parameter θ^t are known. Compute the new value of T by applying the following formula: $T = T_{max}\left(\frac{T_{min}}{T_{max}}\right)^{\frac{t}{N_{iter}-1}}$.
   For fixed value of the parameter T, apply the basic iteration described in the principal stages, which estimates the new parameter θ^{t+1} using the formulas (6), (7) and (8).
3. **Repeat** the Iterative step while $t \leq N_{iter}$.

We can define two steps in the operating of the algorithm:

– The first step corresponds to high T values. In this case, the influencing neighbourhood of each cell c on the map is important and corresponds to higher values of $K^T(\delta(c,r))$.

Formulas (6), (7) and (8) use a high number of observations to estimate model parameters. This step provides the topological order.

- The second step corresponds to small T values. The number of observations in formulas (6), (7) and (8) is limited. Therefore, the adaptation is very local. The parameters are accurately computed from the local density of the data.

3 Experimentations and validations

To evaluate the quality of clustering, we adopt the approach of comparing the results to a "ground truth". We use the clustering accuracy for measuring the clustering results. This is a common approach in the general area of data clustering.

This procedure is defined by (Jain et Dubes, 1988) as "validating clustering by extrinsic classification", and has been followed in many other studies (Andreopoulos et al., 2006; Khan et Kant, 2007).

Thus, to adopt this approach we need labeled data sets, where the external (extrinsic) knowledge is the class information provided by labels. Hence, if the WeCSOM finds significant clusters in the data, these will be reflected by the distribution of classes. Therefore we operate a vote step for clusters and compare them to the behavior methods from the literature. The so-called vote step consists in the following. For each cluster $c \in \mathcal{C}$:

- Count the number of observation of each class l (call it N_{cl}).
- Count the total number of observation assigned to the cell c (call it N_c).
- Compute the proportion of observations of each class (call it $S_{cl} = N_{cl}|N_c$).
- Assign to the cluster the label of the most represented class ($l^* = \arg\max_l(S_{cl})$.

A cluster c for which $S_{cl} = 1$ for some class labeled l is usually termed a "pure" cluster, and a purity measure can be expressed as the percentage of elements of the assigned class in a cluster. The experimental results are then expressed as the fraction of observations falling in clusters which are labeled with a class different from that of the observation. This quantity is expressed as a percentage and termed "purity percentage" (indicated as $Purity\%$ in the results).

To test the performance of our approach we used many publics data sets extracted from the UCI repository (Asuncion et Newman, 2007). The table 1 summarizes a short description of these data sets.

TAB. 1 – – *Description of the used datasets for the validations.*

Data set	Size	nb. of classes
Zoo	101×16	7
Congressional vote	435×16	2
Wisconsis-B-C	699×9	2
Nursery	12960×8	2
Car	1728×6	4
Post-Operative	90×8	3

To conduct experimental comparison and to verify the efficacy of our proposed model, we compare our method with the RTC (Relational Topological Clustering), (Labiod et al., 2010).

We choose this method because it is based on the same principle of the Kohonens model (conservation of the data topological order) and uses the Relational Analysis formalism by optimizing a cost function defined by analogy with Condorcet criterion. One disavantage of the RTC method is that this approach treats all the features equally. We use the same categorical data sets obtained from UCI repository (Asuncion et Newman, 2007) and used in (Labiod et al., 2010).
For each dataset we learned a map of different sizes (from 5x5 to 10x10) and we indicate in the table 2 the purity of clustering for RTC technique and WeCSOM. The results illustrate that the proposed technique increase the purity index compared to the RTC and also presents the advantage to treat directly the categorical data without using the binary coding.

We compared also the performance of our method with the result provided in (Khan et Kant, 2007) that used a version of K-modes clustering method dedicated to categorical data. Table 3 lists the classification error obtained with different methods. We compute the fraction of observations falling in clusters which are labeled with a class different from that of the observation. We can observe that our results are much better then the results provided by K-modes (Khan et Kant, 2007). Also we improve the error rate compared to BinBatch algorithm which represents the classical SOM approach dedicated to binary data using Hamming distance.

TAB. 2 – – *Comparison between RTC et WeCSOM using purity index. RTC : Relational Topological Clustering dedicated to categorical data using the Relational Analysis formalism.*

Purity: %	Size map	RTC	WeCSOM
Zoo	(5×5)	97.84	98.13
Nursery	(6×6)	78.69	81.52
Car	(10×10)	80.17	82.19
Post-Operative	(5×5)	78.21	81.34

TAB. 3 – – *Comparison of the classification performances reached by K-modes, BinBatch and WeCSOM clustering algorithms.*

Error rate: %	K-modes	BinBatch	WeCSOM
Wisconsis-B-C	13.2	3.87	2.34
Zoo	16.6	2.97	1.87
Congressional vote	13.2	5.91	5.77

4 Conclusion

This study reports the development of a computationally efficient EM approach to maximize the likelihood of the data set to estimate the parameters of a probabilistic self-organizing map model dedicated to categorical variables. This algorithm has the advantage of providing a prototype with the same coding as the input data. The extention of the proposed method to the co-clustering will be an interesting future work for dealing with large-scale problems.

Références

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., 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.

Asuncion, A. et D. Newman (2007). UCI machine learning repository. http://www.ics.uci.edu/∼mlearn/MLRepository.html.

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

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

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.

Hofmann, T. (2001). Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning 42*, 177–196.

Huang, Z. (1998). Extensions to the k-means algorithm for clustering large data sets with categorical values. In *Data Mining and Knowledge Discovery 2*.

Ibbou, S. et M. Cottrell (1995). Multiple correspondance analysis crosstabulation matrix using the kohonen algorithm. In *Verlaeysen, M. Editor proc of ESANN'95*, pp. 27–32. Dfacto Bruxelles.

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

Jollois, F.-X. et M. Nadif (2007). Speed-up for the expectation-maximization algorithm for clustering categorical data. *Journal of Global Optimization 37*(4), 513–525.

Kaban, A., E. Bingham, et T. Hirsimäki (2004). Learning to read between the lines: The aspect bernoulli model. In *Proceedings of the Fourth SIAM International Conference on Data Mining, Lake Buena Vista, Florida, USA*.

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.

Kaski, S., T. Honkela, K. Lagus, et T. Kohonen (1998). Websom–self-organizing maps of document collections. *Neurocomputing 21*, 101–117.

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.

Kostiainen, T. et J. Lampinen (2002). On the generative probability density model in the self-organizing map. *Neurocomputing 48*, 217–228.

Labiod, L., G. Nistor, et B. Younes (2010). Relational topographic clustering (rtc). *International Joint Conference on Neural Networks, IJCNN'10*.

Lebbah, M., N. Rogovschi, et Y. Bennani (2007). Besom : Bernoulli on self-organizing map. In *IJCNN*, pp. 631–636. IEEE.

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 analysis of self-organizing maps. *Neural Computing 6*, 767 – 794.

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

Saund, E. (1995). A multiple cause mixture model for unsupervised learning. *Neural Comput. 7*(1), 51–71.

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

Summary

This paper introduces a probabilistic self-organizing map for topographic clustering, analysis of categorical data. By considering a parsimonious mixture model, we present a new probabilistic Self-Organizing Map (SOM). The estimation of parameters is performed by the EM algorithm. Contrary to SOM, our proposed learning algorithm optimizes an objective function. Its performance is evaluated on real datasets.