μ-SOM : Weighting features during clustering

Sébastien Guérif, Younès Bennani
LIPN - CNRS - University of Paris 13
Villetaneuse. France
{sebastien.guerif, younes.bennani}@lipn.univ-paris13.fr

Éric Janvier
Numsight Consulting France
Boulogne Billancourt. France
e.janvier@numsight.com

Abstract - *Real life datasets used in marketing studies contain a lot of redundant features which may prevent data-mining techniques such as self-organizing maps from discovering relevant clusters. An extension of the batch Kohonen's algorithm is proposed in this paper to avoid the large amount of work which is required by data preprocessing if redundancy isn't treated explicitly by the training method. The proposed approach integrates a weighting of variables built on a simultaneous clustering of both observations and variables and avoids the side effects of redundancy. An application to market segmentation is then briefly described to validate the learning algorithm introduced; identified clusters of products and motivations are used to simplify the analysis of the consumer segmentation by giving the user a first rough description of the different groups.*

Key words - Data-mining, Market segmentation, Redundant features, Self-Organizing Map, Weighting

1 Introduction

In real life application, data-mining techniques are applied to datasets which contain numerous redundant features. On the one hand, strong correlations between variables may be useful to deal with missing values [2] or to detect outliers. On the other hand, clustering algorithms built on Euclidean distance may be prevented from discovering correct clusters if data are not preprocessed. Intuitively, redundancy gives more importance to some information which are represented by many features and may occult others that are less present. In the worst case, some irrelevant informations would be expressed by many dimensions and some relevant knowledge by very few variables; this extreme situation may lead to a less interesting clustering of the data. To address this problem, different ways are proposed, in which three categories can be distinguished: selection of variables, extraction of features or weighting of features [1].

Some methods for unsupervised selection of variables using similarity of features have been proposed in [7, 8]. It is well known that Euclidean distance can be approximated when few dimensions compared with the data dimension are missing, but then eliminating some features makes it harder to treat correctly missing values. Principal component analysis (PCA) [6, 9] and factor analysis [13] address efficiently this problem by reducing the attribute space from a large number of variables to a smaller number of orthogonal factors which preserve the maximum of variance. However, they require an important effort from the user to interpret and understand the new representation of one's data. Moreover, these techniques are built on the correlation matrix computation which requires the whole data to be known, and the computation of its eigenvalues and associated eigenvector which may suffer from numerical instabilities. The Mahalanobis distance has been introduced to take care of correlations between dimensions but suffers from the same numerical instabilities as PCA or factor analysis, because it requires the computation of the correlation matrix inverse.

The proposed approach is built on a simultaneous clustering of both observations and variables using self-organizing maps [4] which are well known for their ability to make good representation of data in large dimension. A weighting mechanism which decrease the weight of redundant features has been integrated to the learning algorithm.

The remainder of this paper is organized as follows. Section 2 presents the new algorithm designed to reduce redundancy side effects during the construction of self-organizing maps. Section 3 discusses obtained results and application of our approach to market segmentation while section 4 concludes the paper.

2 μ-SOM: weighting features during clustering

2.1 Outlines and algorithm of μ-SOM

Two self-organizing maps are constructed simultaneously, the first one represents observations and the second one the features' profile. [11] suggests to first realize a clustering of observations and then a clustering of component planes to detect correlations between variables, it is the starting point of our approach. The first basic idea used here is that components planes are a good representation of features, robust to outliers. The second basic idea is that the total weight of variables could be shared between the different dimensions according to the distribution of their best matching units over the map.

The high-level algorithm 1 gives outlines of the μ-SOM learning. The map of observations $SOM^{(data)}$ is made up of $m^{(data)}$ units noted $U^{(data)} = \{1, \ldots, m^{(data)}\}$. Analogously, the map of features $SOM^{(attr)}$ comprises $m^{(attr)}$ units noted $U^{(attr)} = \{1, \ldots, m^{(attr)}\}$. The unit $i \in U^{(data)}$ (resp. $j \in U^{(attr)}$) has $\omega_i(t) \in \mathbb{R}^n$ (resp. $\omega_i(t) \in \mathbb{R}^{m^{(data)}}$) as profile at iteration t. Some details of the μ-SOM learning algorithm have to be defined:

- The distance used to find the best matching unit of an observation at the t^{th} iteration is the following weighted Euclidean distance $d^{(data)}(x,y) = \sqrt{\sum_{i=1}^{n} \mu_i(t)(x_i - y_i)^2}$, where $\mu_i(t) \in \mathbb{R}_+$ are such that $\sum_{i=1}^{n} \mu_i(t) = 1$.

- The profile of features $i \in \{1, \ldots, n\}$ at iteration t is given by the corresponding component plane, that is $fp_i(t) = \left(\omega_{ji}^{(data)}(t)\right)_{j \in U^{(data)}}$, which are normalized to unit range.

- $\alpha : \{0, \ldots, T_{Max}\} \to [0,1]$, where T_{Max} is the number of iterations, increases from 0 to 1 and is used to avoid oscillations of weights during the learning process. A linear function such $\alpha(t) = \frac{t}{(T_{Max}-1)}$ is appropriated.

Algorithm 1 μ-SOM learning

```
Initialize $\mu_i(0) = \frac{1}{n}$, for $i = 1, \ldots, n$
Initialize $\omega_i^{(data)}(0) \in \mathbb{R}^n$, for $i \in U^{(data)} = \{1, \ldots, m^{(data)}\}$
Rough training of $SOM^{(data)}$
Extract profile of attributes $fp_i(t)$ from $SOM^{(data)}$
Initialize $\omega_i^{(attr)}(0) \in \mathbb{R}^{m^{(data)}}$, for $i \in U^{(attr)} = \{1, \ldots, m^{(attr)}\}$
Rough training of $SOM^{(attr)}$
Compute new weights $\mu_i^{new}(0)$
Update weights $\mu_i(1) \leftarrow \alpha(0).\mu_i(0) + (1 - \alpha(0)).\mu_i^{new}(0)$
Initialize $t \leftarrow 1$
while ($t < T_{max}$) do
  Fine training epoch on $SOM^{(data)}$
  Extract profile of attributes from $SOM^{(data)}$
  Fine training epoch on $SOM^{(attr)}$
  Compute new weights $\mu_i^{new}(t)$
  Update weights $\mu_i(t) \leftarrow \alpha(t).\mu_i(t) + (1 - \alpha(t)).\mu_i^{new}(t)$
  $t \leftarrow t + 1$
end while
```

The map of observations is first roughly trained to organize neurons according to topological ordering. Then profiles of features are extracted and used to roughtly train the map of variables. Finally, fine tuning epoches of both maps are alternated and weighting is computed after each update of the map of features.

2.2 Details of the weighting mechanism

The basic idea of the integrated weighting mechanism is to share total weight between a set of features $F = \{1, \ldots, n\}$ according to their similarity. It proceeds as follows :

1. Each unit $i \in U^{(attr)}$ receives a potential weight to share between the different features that is computed using Geary local spatial auto-correlation index [3, 5]:

$$G_i(t) = \frac{\frac{1}{2.L_i(t)} \sum_{j \in U^{(attr)}} c_{ij}(t) . \|\omega_i(t) - \omega_j(t)\|^2}{\frac{1}{m^{(attr)}-1} \sum_{j \in U^{(attr)}} \|\omega_i(t) - \omega_j(t)\|^2}$$

   where $L_i(t) = \sum_{j \in U^{(attr)}} c_{ij}(t)$. $c_{ij}(t) \in \{1, 0\}$ indicates whether units i and j are neighbors or not. Typically, $c_{ij}(t) = \left(d^{(attr)}(i,j) < 1\right)$, where $d^{(attr)}(i,j)$ is the distance between units $i \in U^{(attr)}$ and $j \in U^{(attr)}$ on the map of features.

2. Then, each variables $i \in F$ asks each units $j \in U^{(attr)}$ in the neighborhood of its best matching units $\tilde{i}$ for a part of its potential weight : $part_i^{(j)}(t) = \exp\left(-\frac{1}{2}\left(\frac{d^{(attr)}(\tilde{i},j)}{\sigma(t)}\right)^2\right)$

3. Finally, the potential weight of each units is shared between features according to the requested part: $\mu_i^{new}(t) = \frac{1}{\sum_{j \in U^{(attr)}} G_j(t)} \sum_{j \in U^{(attr)}} G_j . \left(\frac{part_i^{(j)}(t)}{\sum_{k \in F} part_k^{(j)}(t)}\right)$

The Geary local spatial auto-correlation index has been chosen for its ability to measure the similarity of a unit and its neighbors compared to the global variance of unit's prototype. Indeed, areas of the map which represent highly similar features have a lower potential weight than areas with high distortion. It has been noticed that units on the border of the map are slightly penalized because they have less neighbors than the other, leading to a lower local variance is for units in the middle of the map.
It must be pointed out that the set of features F can be replaced by any of its subsets; actually the proposed approach is ready to deal with missing values.

2.3 Cluster analysis

When using self-organizing maps, more or less as many clusters as units on the map are obtained so it is impracticable to analyze each one separately. A clustering of unit prototypes permits to reduce the number of clusters. Hierarchical Ascending Classification (HAC) or k-means are often used to perform this task. We have chosen to apply the method proposed in [12] to cluster our maps. Several k-means clustering are computed for varying number of centers and then the Davies-Bouldin index is used to choose the best one.
Thus, a first rough description of identified clusters of observations can be made using features groups. In the same way, class of observations should be used to roughly describe clusters of attributes. we proposed to proceed as follow:

1. For each cluster i of observations, compute the mean $\overline{x_{ij}}$ of each dimension j.
2. Then, normalize to unit range each mean per dimension $pos_{ij} = \frac{\overline{x_{ij}} - \min_i\{\overline{x_{ij}}\}}{\max_i\{\overline{x_{ij}}\} - \min_i\{\overline{x_{ij}}\}}$.
3. For each cluster i, compute the mean $\overline{pos_i} = mean_{j \in F}\left(pos_{ij}\right)$ and standard deviation $\sigma_{pos_i} = std_{j \in F}\left(pos_{ij}\right)$ of the normalized means pos_{ij}.
4. For each cluster i, select all dimensions j such $pos_i j \geq \overline{pos_i} + \sigma_{pos_i}$
5. Representation ratios of each classes of features is a useful rough description of the cluster i.

Rough descriptions given by representation ratios are useful to give the user a first idea of relationships between observations and features clusters and facilitate a cross analysis of revealed groups.

3 Application and results

3.1 Results

Our approach has been evaluated using various dataset and obtained results on the *waveform* and the *isolet* datasets from the UCI Machine Learning Repository [10] are presented here. Cross validation has been used to compare the quality of maps obtained using μSOM to those built with the batch version of Kohonen's algorithm. Each dataset has been divides in five parts; four subsets has been used by the training algorithm and the last one to evaluate the quality of the map. Three indexes has been used to evaluate the quality of topological maps:

- mean quantification error (Qerr)

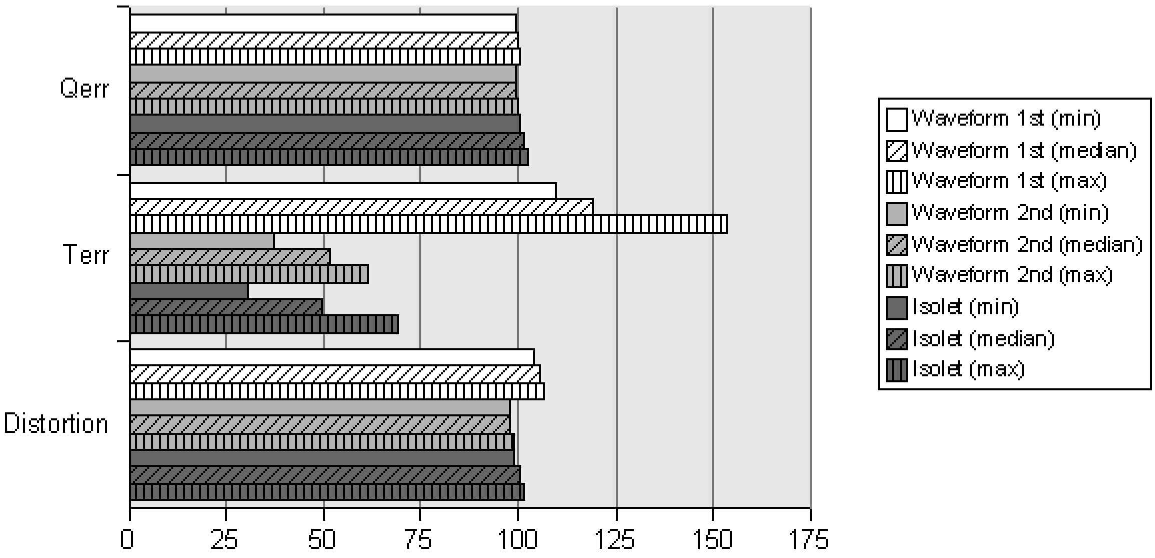

Figure 1: Relative quality of μSOM (index 100 for SOM)

- topological error rate (Terr)
- distortion measures

In our first experiments on the *waveform* dataset (waveform 1st), the number of neurons on the map of features was greater than the number of variables. The resulting map was unusable to identify correct correlations between features. Then, the number of units has been decreased (waveform 2nd). Observed differences on quantification error and distortion measure between topological maps obtained using μSOM and the standard algorithm are not significant. Nevertheless, it should be noticed that the topological error rate has been greatly improved on both dataset.

3.2 Application to marketing

The aim of market studies is to understand the behavior of consumers and identify groups which share the same interests. Data are generally collected by a sample survey of consumers and contains typically several hundred of observations described by several tens of variables. A segmentation of both observations and variables allows us to identify group of consumers, categories of products and relationship between them.
Our dataset contains some 230 answers from 1006 consumers. The application of μ-SOM algorithm and the clustering of the obtained map have permitted to identify 17 categories of products and 14 groups of consumers. The segmentation of products has been analyzed first and then rough descriptions of groups of consumers have been computed. They are very helpful in practice because they give a first idea of what a cluster contains and gives a pertinent axis of analysis.

The figure 2 presents both the distributions of consumers over the map and the different identified groups. Then the whole classes of features are presented figure 5 and a zoom on

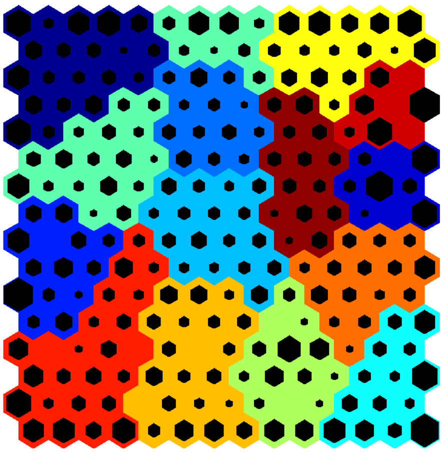

Figure 2: Distribution and classes of observations over the map

two diferent areas is proposed figure 3 and 4. Finally, figure 6 shows the distribution of features' weights.

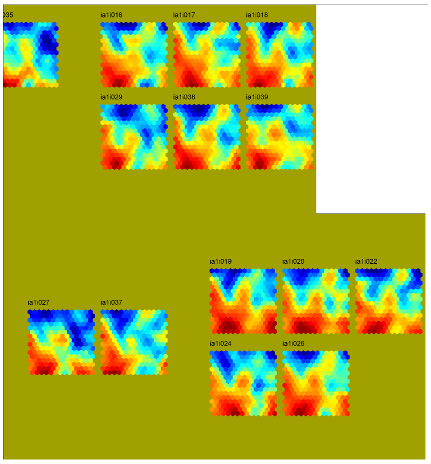

Figure 3: Upper right corner of the map of features.

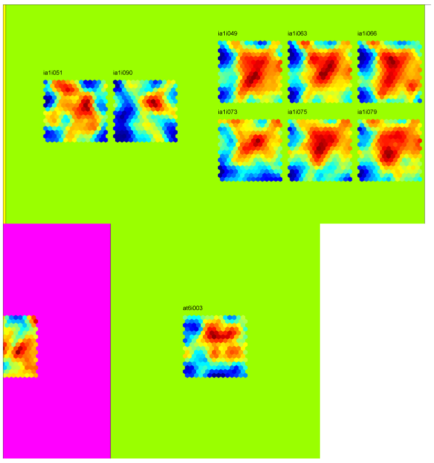

Figure 4: Middle right area of the map of features.

4 Conclusions and further research

A novel learning algorithm for Self-Organizing Map is presented in this paper. It leads to better quality maps than the batch version of the Kohonen's batch algorithm. Actually, it has been successfully applied on market studies datasets and appears to be useful for both avoiding a large amount of work needed to preprocess data and providing rough descriptions of clusters which could be used as starting point for the analysis. Experiments are under way to evaluate the ability of the proposed algorithm to deal with missing values and noisy data. Future work includes adaptation of this method to the on-line version of Kohonen's algorithm and improvement of the quality of the distance used with features profile.

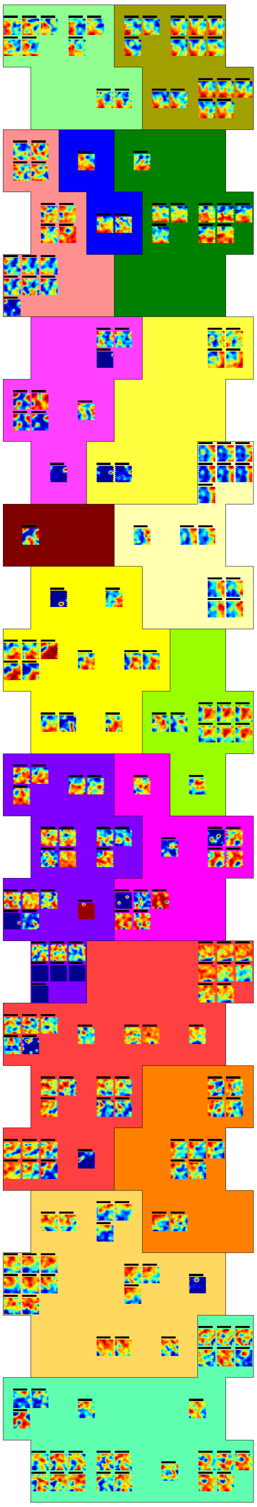

Figure 5: Distribution of features and categories. Component planes of th map of observations are represented at the position of their best matching units. This visualization is useful to analysis features correlations.

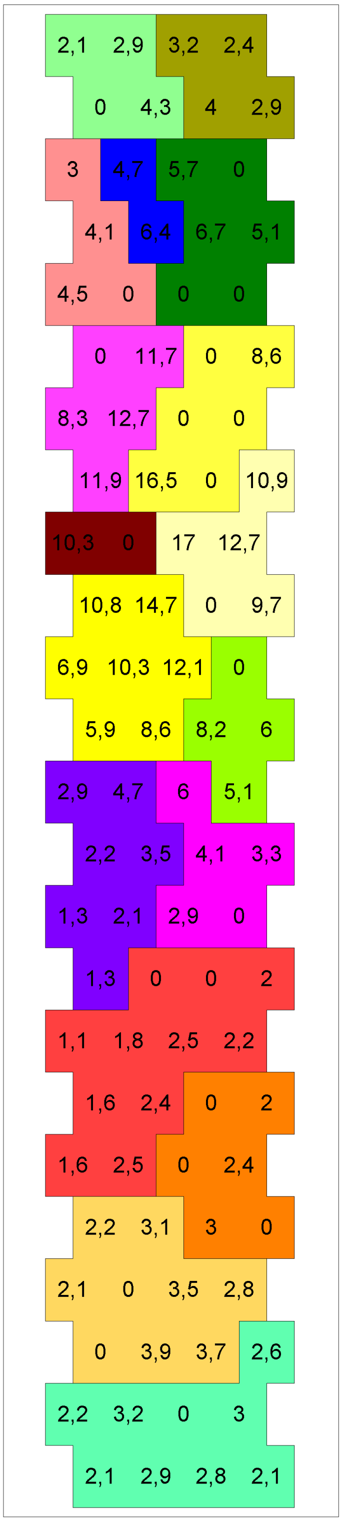

Figure 6: Distribution of weight ($\times 10^{-3}$) of features. Each features is given a weight according to it best matching unit.

Acknowledgement

We would like to thank Mark Kerslake from Numsight Consulting France for our discussion about the relevance of revealed classes of both products and consumers, his review and english correction.

References

[1] Y. Bennani (1999), Adaptive weighting of pattern features during learning, *International Joint Conference on Neural Networks, IJCNN'99*, **vol. 5**, p. 3008-3013.

[2] M. Cottrell, S. Ibbou et P. Letrémy (2003), Traitement des données manquantes au moyen de l'algorithme de Kohonen, *Actes de la dixième conférence ACSEG, Nantes.*

[3] R. C. Geary (1954), The contiguity ratio and statistical mapping, *The Incorporated Statistician*, p. 115-145.

[4] T. Kohonen (2001), *Self-Organizing Maps 3rd edition*, Heidelberg, Springer.

[5] L. Lebart (1969), Analyse statistique de la contiguité, *Publications de l'ISUP*, p. 81-112.

[6] L. Lebart, A. Morineau et M. Piron (2000), *Statistique exploratoire multidimensionnelle 3e édition*, Dunod.

[7] P. Mitra, C.A. Murthy and Sankar K. Pal (2002), Unsupervised Feature Selection Using Feature Similarity, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **vol. 24-3**, p. 301-312.

[8] Sankar K. Pal, Rajat K. De and J. Basak (2000), Unsupervised Feature Evaluation: A Neuro-Fuzzy Approach, *IEEE Transactions on Neural Networks*, **vol. 11-2**, p. 366-376.

[9] G. Saporta (1990), *Probabilités, analyse de données et statistiques*, Paris, Editions Technip.

[10] UCI Machine Learning Repository, *http://www.ics.uci.edu/ mlearn/MLRepository.html.*

[11] J. Vesanto and J. Ahola (1999), Hunting for Correlations in Data Using the Self-Organizing Map, *In Proceeding of the International ICSC Congress on Computational Intelligence Methods and Applications (CIMA'99)*, ICSC Academic Press, p. 279-285.

[12] J. Vesanto and E. Alhoniemi (2000), Clustering of the Self-Organizing Map, *In IEEE Transactions on Neural Networks*, **vol. 11-3** p. 586-600.

[13] N. Wu and J. Zhang (2005), Factor-analysis based anomaly detection and clustering, *Decision Support Systems*, **to appear**.