IDIAP RESEARCH REPORT

SOM-based Clustering for On-line Fraud Behavior Classification: a Case Study

Vincent Lemaire [a]   Fabrice Clérot [a]

IDIAP–RR 02-30

July 2002

to appear in
Fuzzy Systems and Knowledge Discovery (FSKD) 2002

Dalle Molle Institute for Perceptual Artificial Intelligence • P.O.Box 592 • Martigny • Valais • Switzerland

phone +41 − 27 − 721 77 11
fax +41 − 27 − 721 77 12
e-mail secretariat@idiap.ch
internet http://www.idiap.ch

[a] Telecommunication and Neural Techniques Group - France Telecom Research and Development - FTR&D/DTL/TIC - 2 Avenue Pierre Marzin - 22307 Lannion cedex FRANCE - *email: vincent.lemaire@francetelecom.com email: fabrice.clerot@francetelecom.com*

IDIAP Research Report 02-30

SOM-based Clustering for On-line Fraud Behavior Classification: a Case Study

Vincent Lemaire Fabrice Clérot

July 2002

to appear in
Fuzzy Systems and Knowledge Discovery (FSKD) 2002

Abstract. The Self-Organizing Map (SOM) is an excellent tool in exploratory phase of classification problems. It projects the input space on prototypes of a low-dimensional regular grid which can be efficiently used to visualize and explore the properties of the data. In this paper we present a novel methodology using SOM for exploratory analysis, dimensionality reduction and/or variable selection. The methodology is applied to a real case study and the results are compared with other techniques.

1 Introduction

The Self-Organizing Map (SOM) [5] is especially suitable for data survey because it has prominent visualization properties. It creates a set of prototype vectors representing the data set and carries out a topology preserving projection of the prototypes from the d-dimensional input space onto a low-dimensional grid (two dimensions in this paper). This ordered grid can be used as a convenient visualization surface for showing different features of the SOM (and thus of the data), for example the cluster structure [10].

When the number of SOM units is large, similar units have to be grouped together (clustered) so as to ease the quantitative analysis of the map. Different approaches to clustering of a SOM have been proposed [11, 6] such as hierarchical agglomeration clustering or partitive clustering using k-means. This SOM-based exploratory analysis is therefore a two-stage procedure:

1. a large set of prototypes (much larger than the expected number of clusters) is formed using a large SOM;
2. these prototypes are combined to form the final clusters.

Such analysis deals with the individuals and constitutes only a first step. In this paper we propose a second step which is very similar but deals with the analysis of the variables: each input variable can be described by its projection upon the map of the individuals. A visual inspection can be done to see where (i.e. for which prototype(s) of the SOM) each variable is strong (compared to the other prototypes). It is also possible to compare the projections of different variables. This analysis of the variables is however impossible when the number of input variables is large and we propose to make it automatic: the projections of each variable on the map of the individuals is taken as a representative vector of the variable and we train a second SOM with these vectors; this second map (map of the variables) can then be clustered, allowing to group automatically together variables which have similar behaviors.

The organization of the paper is as follows: in the next section we present the real case study and the database. In section 3 we present our methodology for exploratory analysis and dimensionality reduction with SOM. Section 4 describes experimental conditions and comparative results on our methodology and others machine learning techniques. A short conclusion follows.

2 Case Study

The case study is the on-line detection of the fraudulent use of a phone card. Using a large number of variables in the modeling phase allows to obtain good performances but such models cannot be applied on-line because of computing and data extraction time constraints; therefore, it is necessary to reduce the number of variables while keeping good performances.

The original database contains 15330 individuals for which we have 226 inputs variables. We split it into 3 set: a training set, a validation set and a test set which contain respectively 70%, 15% and 15% of the individuals. Whatever the method evaluated below, the test set was never used to build the classifier.

The database contains 92 % examples which belong to the class "not fraudulent" and 8 % which belong to class "fraudulent".

3 Methodology

3.1 A Two-Step Two-Level Approach

The methodology used in this paper is depicted in the Figure 1. The primary benefit of each two-level approach is the reduction of the computational cost [10, 6] and an easier interpretation of the map structure. The second benefit of this methodology is the simultaneous visualization of clusters of individuals and clusters of variables for exploratory analysis purposes. Finally, dimensionality reduction and/or variable selection can be implemented.

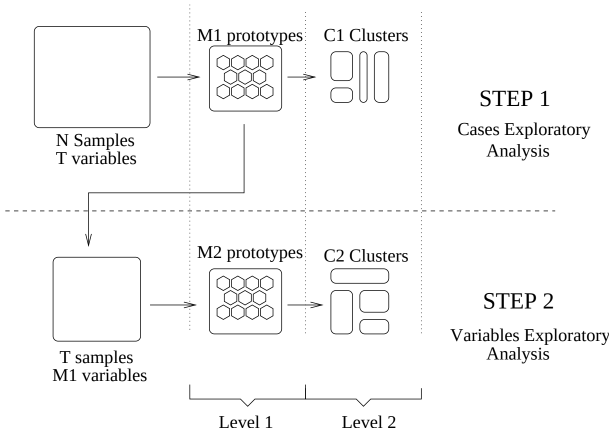

Figure 1: The Two-Step Two-Level Approach.

All the SOM in this paper are square maps with hexagonal neighborhoods and are initialized with Principal Component Analysis (PCA). We use a validation set or a cross-validation to measure the error reconstruction and select the map size above which the reconstruction error does not decrease significantly (the result reconstruction error for a given size is an average on 10 attempts).

3.2 Exploratory Analysis of the Cases

The first step of the method is to build a SOM of the cases[1] [12].

It should be emphasized that the goal of this SOM-based exploratory analysis is the automatic dimension reduction and/or variable selection. This phase serves as a pre-processing to the modeling experiments described below. Therefore, we shall only briefly comment the results of the first step.

The best map size was determined to be 12x12. We used the training set and the validation set to do a final training of the SOM of the cases.

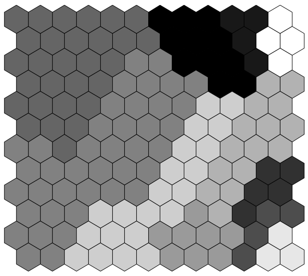

Figure 2: Groups of cases with similar behaviors found using a hierarchical clustering.

This map allows to track down the characteristic behaviors of the cases: a standard clustering algorithm (k-means or hierarchical clustering) can be run on top of the map, revealing groups of cases with similar behaviors (see Figure 2). Projecting the class information (fraudulent use or not; this information is not use for the construction of the map) on the map allows to investigate the distinctive profiles of the classes in terms of all the input variables (see Figure 3).

[1]All the experimentations on SOM have been done with the SOM Toolbox package for Matlab ©

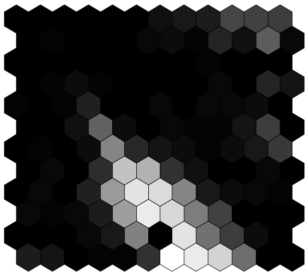

Figure 3: The two classes population for each prototype. The lighter color, the more fraudulent the behavior.

This constitutes the first step. We then proceed to the second step: each input variable is described by its projection upon the map of the cases. A visual inspection (see for example Figure 4) can be done to see where (for which prototype of the SOM) each variable is strong (compare to the other prototypes). Thus it is possible to relate each variable to each cluster of the SOM. It is also possible to analyze visually the relationships between different variables. The second step described below makes this analysis more automatic.

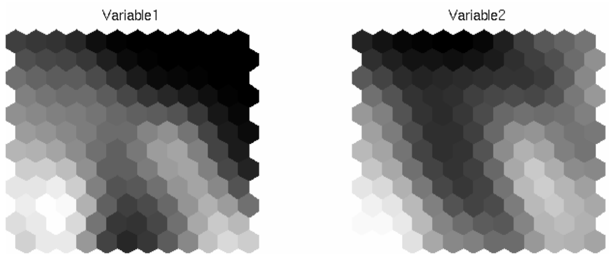

Figure 4: The projections of the first two variables on the map of the cases: the darker the color, the stronger the value for the corresponding prototype.

3.3 Exploratory Analysis of the Variables

In a second step we build a second SOM to explore the variables as follows: each input variable is described by its projection upon the map of the individuals, hence by a vector having as many dimensions as the number of neurons of the map of the individuals. These variables descriptors are used to train the second map, the map of the variables.

For this SOM we cannot use a validation set since the database is the codebook of the SOM of the individuals and is therefore quite small. We use a 5-fold cross-validation [13] method to find the best size of the SOM of the variables. The selected size is 12 x 12. Knowing the best size of SOM of the variables, we used all the codebook of the SOM of the individuals to do a final training of the SOM of the variables.

This map allows to explore the relationships between variables; in particular, we also run a standard clustering algorithm on top on this map to create groups of variables with similar behaviors.

We used a K-means [8, 4] for the clustering onto the SOM of the variables. Here still we cannot use a validation set to determine the optimal K value and we used a 5-fold cross-validation. We chose the value of K^* above which the error reconstruction does not decrease significantly (the result for a given size is an average on 20 attempts). The selected value is $K^* = 11$. The clusters found on the map of the variables could be visualized as for the Figure 2 and the Figure 5 shows the projections of the variables of one such cluster upon the map of the cases (the similarity of the projections is obvious).

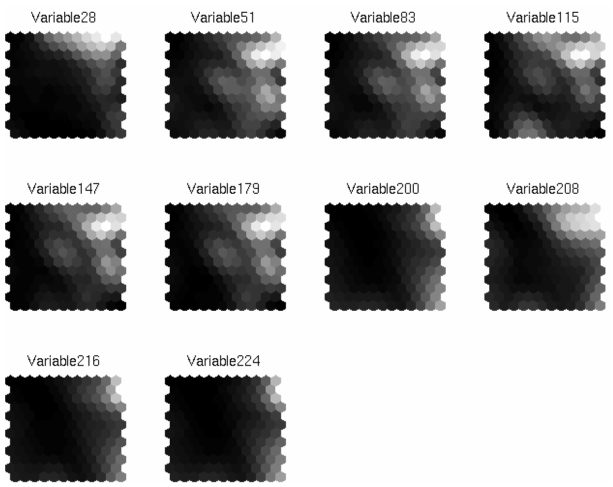

Figure 5: Projection of the variables belonging to the fifth cluster upon the map of the individuals.

Knowing the best value of K, we used all the codebook of the SOM of the variables for 200 trainings of the K^*-means; we kept the best solution to choose the paragons of the clusters, that is the variables which minimize the sum of the distances to the other variables of their cluster.

At this point, we end up with two clusterings, a clustering of cases and a clustering of variables, which are consistent together: groups of cases have similar behavior relative to groups of variables and vice-versa, a situation reminiscent of the duality property of PCA. This allows a much easier exploratory analysis and can also be used for dimensionality reduction and/or variable selection since variables of the same group contribute in the same way to the description of the individuals.

3.4 Dimensionality Reduction vs. Variable Selection

In this paper, "dimensionality reduction" refers to techniques which aim at finding a sub-manifold spanned by combinations of the original variables ("features"), while "variable selection" refers to techniques which aim at excluding variables irrelevant to the modelling problem. In both cases, this is a combinatorial optimization problem. The direct approach (the "wrapper" method) retrains and re-evaluates a given model for many different feature/variable sets. An approximation (the "filter" method) instead optimizes simple criteria which tend to improve performance. The two simplest optimization methods are forward selection (keep adding the best feature/variable) and backward elimination (keep removing the worst feature/variable) [2, 7].

With the SOMs obtained above, dimensionality reduction can be implemented by defining a feature for each cluster as the normalized sum of the variables of the cluster. Variable selection can be implemented by choosing one variable per cluster of variables (the "paragon" of the cluster, that is the variable which minimizes the sum of the distances to the other variables of the cluster).

Both methods therefore reduce the number of input variables to the number K^* of clusters found on the map of the variables. Modeling after variable selection only relies on fewer input variables, therefore relying on less information, while modeling after dimensionality reduction relies on fewer features which may gather all the relevant information in the input variables but are often impossible to interpret in an intelligible way.

4 Methodology: Comparison and Results

Other machine learning techniques also allow to realize variable selection such as the decision trees, Bayesian networks or dimensionality reduction methods such as PCA. We shall compare the methodology described above to such techniques and this section details the experimental conditions for this comparison.

We shall report a comparison of the results obtained on our case-study:

- on the one hand we shall compare the performance of models which use dimensionality reduction: a neural network trained with all the input variables, a neural network which uses the K^* variables found with dimensionality reduction method described below, and a PCA where we kept the first K^* eigenvectors.
- on the other hand we shall compare the performance of models which use a variable selection: a neural network which uses the K^* variables found with the variable selection method proposed below, a Bayesian network, and a decision tree.

4.1 Experimental Conditions

4.1.1 Principal Component Analysis

The principal components are random variables of maximal variance constructed from linear combinations of the input features. Equivalently, they are the projections onto the principal component axes, which are lines that minimize the average squared distance to each point in the data set [1]. The Principal Component Analysis (PCA) has been constructed on the training set and projected using the first $K^* = 11$ eigenvectors on the validation set and the test set. This may not be the optimal number of eigenvectors but, for comparison purposes, the number of eigenvectors kept has to correspond to should be the number of clusters of variables found in the section 3.3.

4.1.2 Multi-layer Perceptrons

Each neural network, in this paper, is a multilayer perceptron, with standard sigmoidal functions, $K^* = 11$ input neurons, one hidden layer with P neurons and one output. We used the stochastic version on the squared error cost function. The training process is stopped when the cost does not decrease significantly as compared to the previous iteration on the validation set. The learning rate is $\beta = 0.001$.

The optimal number P^* of hidden units was selected for the final cost, between 2 and 50 for each case: the neural network trained with all the input variables, the neural network which uses the $K^* = 11$ variables found with the dimensionality reduction method described above, the neural network where we kept the first $K^* = 11$ eigenvectors found with the PCA and the neural network which uses the $K^* = 11$ variables found with the variable selection method proposed above (the result for a given size is an average on 20 attempts). The P^* values are respectively 12, 10, 6 and 10.

4.1.3 Decision Tree

We used a commercial version of the algorithm C4.5 [9]. The principal training parameters and the pruning conditions are:

- the splitting on the predictor variables continues until all terminal nodes in the classification tree are "pure" (i.e, have no misclassification) or have no more than the minimum of cases computed from the specified fraction of cases (here 100) for the predicted class for the node;
- the Gini measure that reaches a value of zero when only one class is present at a node (with priors estimated from class sizes and equal misclassification costs).

With these parameters the number of variables used by the decision tree is 17, that is more than $K^* = 11$.

4.1.4 Bayesian Network

The Bayesian network (BN) found is similar to the 'Naive Bayes' which assumes that the components of the measurement vector, i.e. the features, are conditionally independent given the class. Like additive regression, this assumption allows each component to be modeled separately. Such an assumption is very restrictive but on this real problem a naive Bayes classifier gives very good results (see [3]). The BN uses 37 variables[2], that is more than three times more than $K^* = 11$.

4.2 Results

The various classification performances are given below in the form of lift curves.

The methodology described in the paper gives excellent results (see Figure 6): regarding the variable selection method, the performances of the neural network trained with the selected variables are better than the performances of the Decision Tree and the Bayesian Network. As compared to the neural network trained with all the input variables (226 variables), the neural network trained with the selected variables only ($K^* = 11$ variables) shows a marginal degradation of the performance for very small segments of the population and even has a slightly better behavior for segments larger than 20% of the population; regarding the SOM-based dimensionality reduction method, its performance is similar to the PCA-based method.

These comparisons show that, on this real application, it is possible to obtain excellent performances with the methodology described above and in particular with the variable selection method, hence allowing a much simpler interpretation of the model as it relies on a few input variables only.

5 Conclusion

In this paper we have presented a SOM-based methodology for exploratory analysis, dimensionality reduction and/or variable selection. This methodology has been shown to give excellent results on a real case study when compared with other methods either in terms of visualization ability and classification performance.

These results are extremely encouraging. Future work should address several points; among others:

- if there is no visualization requirement, could such methodology retain its efficiency when both SOMs are replaced by standard k-means?
- does this methodology allow better results than PCA dimension reduction on other real problems?

Acknowledgments

Vincent Lemaire works for France Telecom Research and Development but was invited at IDIAP (Dalle Molle Institute for Perceptual Artificial Intelligence, www.idiap.ch) for six months in the framework of a collaboration. A part of this work has been done while he was at IDIAP. This author would like to thank Samy Bengio for discussions.

[2]The BN was built by Prof. Munteanu and coworkers, ESIEA, 38 rue D. Calmette Guérin 53000 Laval France, in the framework of the contract "Bayes-Com" with France Telecom. The research report is not available.

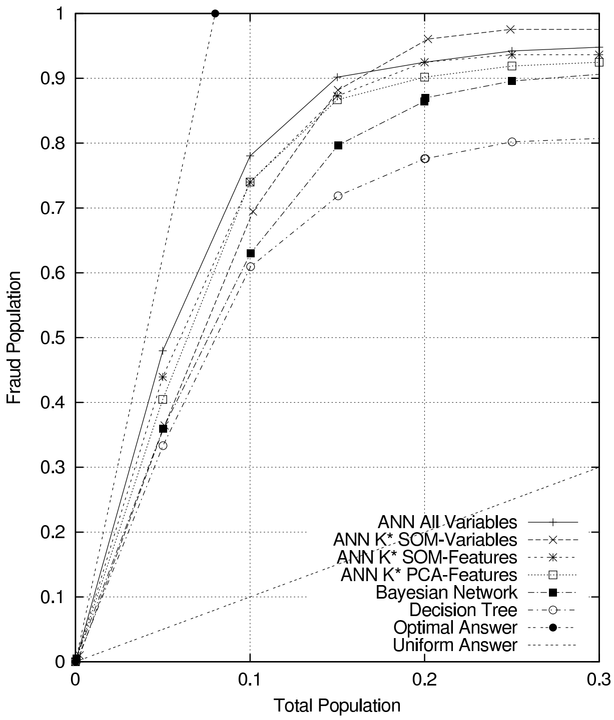

Figure 6: Detection rate (%) of the fraudulent users obtained with different learning methods (ANN: Artificial Neural Network), given as a lift curve.

References

[1] Christopher M. Bishop. *Neural Network for Pattern Recognition.* Oxford University Press, 1996.

[2] A. Blum and P. Langley. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, pages 245–271, 1997.

[3] David J. Hand and Keming Yu. Idiot's bayes - not so stupid after all. *International Statistical Review*, 69(3):385–398, 2001.

[4] A.K. Jain and R.C. Dubes. Algorithms for clustering data. *Prentice Hall*, pages 96–101, 1988.

[5] Teuvo Kohonen. Self-organizing maps. In *Springer Series in Information Sciences*, volume 30. Springer, Berlin, Heildelberg, 1995.

[6] J. Lampinen and E. Oja. Clustering properties of hierarchical self-organizing maps. *Journal of Mathematical Imaging and Vision*, 2(3):261–272, 1992.

[7] P. Langley. Selection of relevant features in machine learning. In AAAI Press, editor, *AAAI Fall Symposium on Relevance*, New Orleans, 1994.

[8] J. Moody and Darken C.J. Fast learning in networks of locally-tuned processing units. *Neural Computation*, 1(2):281–294, 1989.

[9] J.R. Quilan. *C4.5: Programs for Machine Learning.* Morgan Kaufmann, 1993.

[10] Juha Vesanto. Som-based data visualization methods. *Inteligent Data Analysis*, 3(2):111–126, 1999.

[11] Juha Vesanto and Esa Alhoniemi. Clustering of the self-organizing map. *IEEE Transactions on Neural Networks*, 11(3):586–600, 2000.

[12] Juha Vesanto, Johan Himberg, Esa Alhoniemi, and Juha Parhankangas. Som toolbox for matlab 5. Report A57, Helsinki University of Technology, Neural Networks Research Centre, Espoo, Finland, April 2000. `http://www.cis.hut.fi/projects/somtoolbox/`.

[13] S.M. Weiss and C.A. Kulikowski. *Computer Systems That Learn*. Morgan Kaufmann, 1991.