The Many Faces of a Kohonen Map

A Case Study: SOM-based Clustering for On-Line Fraud Behavior Classification

V. Lemaire and F. Clérot

Telecommunication and Neural Techniques Group
France Telecom Research and Development
FTR&D/DTL/TIC
2 Avenue Pierre Marzin
22307 Lannion cedex FRANCE
{vincent.lemaire,fabrice.clerot}@francetelecom.com

Abstract: The Self-Organizing Map (SOM) is an excellent tool for exploratory data analysis. 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 article we present a novel methodology using SOM for exploratory analysis, dimensionality reduction and/or variable selection for a classification problem. The methodology is applied to a real case study and the results are compared with other techniques.

Keywords: Self-Organizing Map, Exploratory Analysis, Dimensionality Reduction, Variable Selection

1 Introduction

The Self-Organizing Map (SOM) [5] is an excellent tool 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 article). This ordered grid can be used as a convenient visualization surface for showing different features of the data.

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 [9, 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 an analysis deals with the cases and constitutes only a first step. In this article we follow the pioneering work of Juha Vesanto [10] on the use of Kohonen maps for data mining and we propose a second step, which involves a very similar techniques, but deals with the analysis of the variables: each input variable can be described by its projection upon the map of the cases. A visual inspection can be performed 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 manual examination of the variables becomes impossible when the number of input variables is large and we propose an automatic examination: the projections of each variable on the map of the cases 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 automatically group together variables which have similar behaviors.

The organization of the article is as follows: the next section deals with the real case studies and in section 3 we present our methodology for exploratory analysis with SOM. In section 4 we present our methodology for dimensionality reduction and variable selection with SOM. Section 5 describes experimental conditions and comparative results between 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 post-paid phone card. Post-paid cards are characterized by:

- card number (12 digits written on the card)
- card identifier (4 digits only known by the owner of the card)
- used in public phones (only need to enter the identifier)
- used in any fixed phone (enter the 16 digits for identification)

Here the “fraud” term includes all cases which may lead to a fraudulent non-payment by the caller. The purpose is to prevent non-payments by warning the owners of phone card that the current use of their card is unusual.

The original database contains 15330 individuals described with 226 input variables of various kinds:

- sociological variables
- a series of indicators of traffics;
- variables of descriptive statistics.

Using a large number of these variables in the modeling phase achieves good fraudulent/non-fraudulent classification performances but such models cannot be applied on-line because of computing and data extraction time constraints. It is thus necessary to reduce the number of variables while maintaining good performance.

We split the data into 3 sets: a training set, a validation set and a test set which contain respectively 70%, 15% and 15% of the cases. Whatever the method evaluated below, the test set is never used to build the classifier. 92 % of the examples in the database belong to the class “not fraudulent” and 8 % belong to the class “fraudulent”.

3 Methodology

3.1 A Two-Step Two-Level Approach

The methodology used in this article is depicted in the Figure 1. The primary benefits of each two-level approach are the reduction of the computational cost [9, 6] and an easier interpretation of the map structure. The second benefit of this methodology is the simultaneous visualization of clusters of cases and clusters of variables for exploratory analysis purposes. Finally, dimensionality reduction and/or variable selection can be implemented.

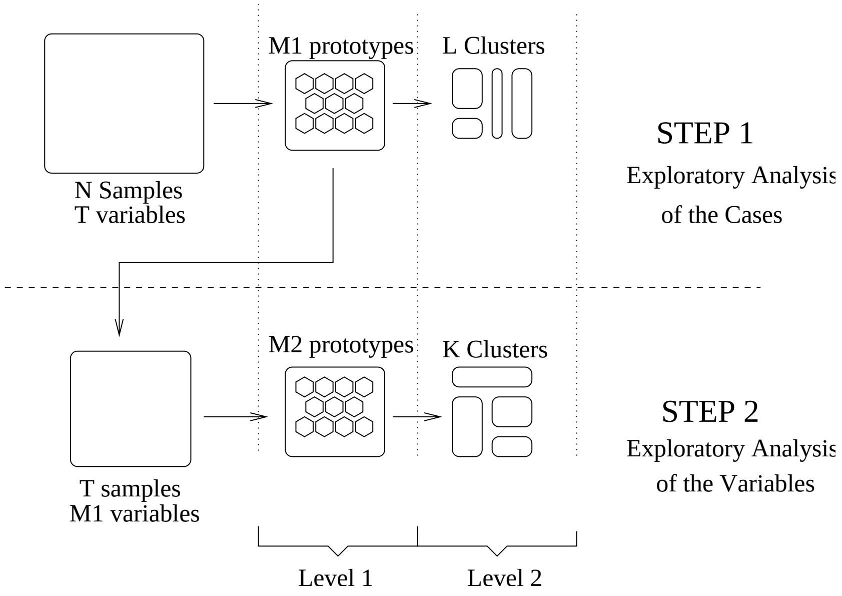

Fig. 1. The Two-Step Two-Level Approach.

All the SOM in this article 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 reconstruction error for a given size is estimated as an average on 10 attempts).

3.2 Top View: Exploratory Analysis of the Cases

The first step of the method is to build a SOM of the cases[1]. The best map size, for the case study was determined to be 12x12. We used the training set and the validation set to achieve a final training of the SOM of the cases.

[1] All the experimentations on SOM have been done with the SOM Toolbox package for Matlab © [11]

This map allows to track down the characteristic behaviors of the cases: a standard clustering algorithm can be run on top of the map, revealing groups of cases with similar behaviors (see Figure 2). This clustering is done onto the prototypes of the SOM themselves, not on the prototypes weighted by the number of cases belonging to each prototype. Extensive tests have not shown any significant difference between k-means and hierarchical agglomerative clustering with the Ward criterium for this clustering of the map.

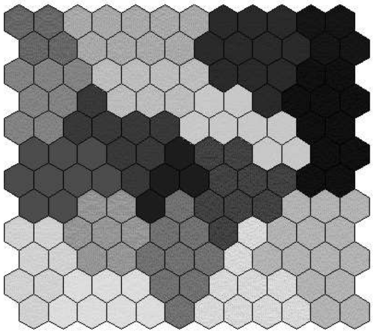

Fig. 2. Groups of cases with similar behaviors found using a hierarchical clustering.

Projecting the class information (fraudulent use or not in our case study; this information is not used 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.

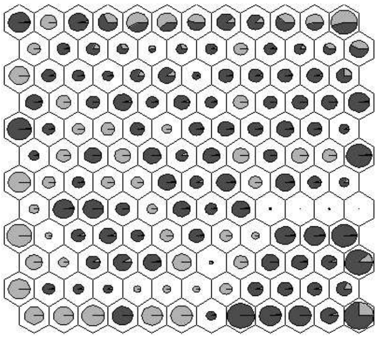

Fig. 3. The two populations for each prototype. The size of the pie indicates the number of cases belonging to each prototype. The lighter the color, the less fraudulent the behavior. For each pie the light grey proportion indicates the proportion of fraudulent behavior. We can project other auxiliary data in a similar manner.

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. Upon visual inspection (see Figure 4), one can determine how a variable behaves on the map and relate this behavior to the clusters of cases.

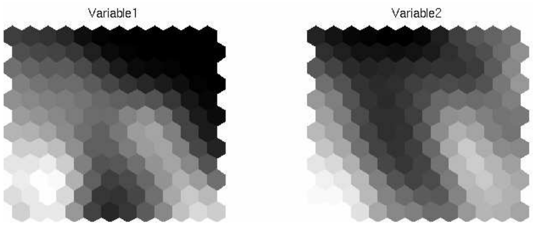

Fig. 4. The projections of the first two variables on the map of the cases: the darker the color, the stronger the value for the corresponding prototype.

It is also possible to visually analyze the relationships between different variables. This visual method consists in the visualization of each projection on the map of the cases and to group together variables with similar projections (see Figure 5). However this visual inspection of the variables becomes impossible when the number of input variables grows.

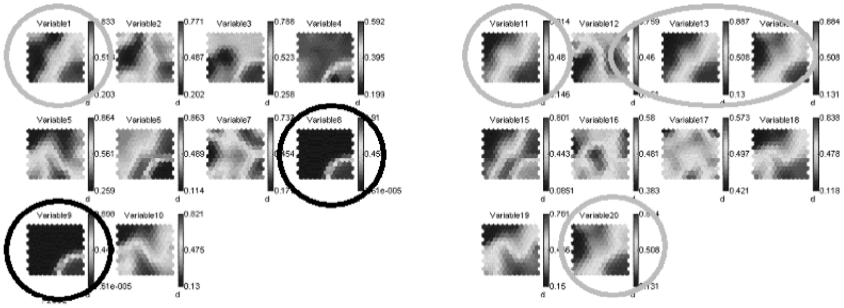

Fig. 5. Each subfigure above shows the projection of 10 variables. Visual inspection allows to find some strongly correlated variables (two "obvious" groups in this example) but is of limited efficiency when the number of variables is large.

Nevertheless, an automatic clustering of the variables according to their projection can be achieved: the projections of each variable on the map of the cases are 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 automatically group together variables which have similar behaviors.

3.3 Side View: Exploratory Analysis of the Variables

In the 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 cases, hence by a vector having as many dimensions as the number of neurons of the map of the cases. 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 cases and is therefore quite small. We use a 5-fold cross-validation [12] 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 codebooks of the SOM of the cases to perform a final training of the SOM of the variables.

This map allows to explore the relationships between variables and to study the correlation between variables; we also run a standard clustering algorithm on top on this map to create groups of variables with similar behaviors. Again, this clustering is done onto the prototypes of the SOM themselves, not on the prototypes weighted by the number of variables belonging to each prototypes. The clusters found on the map of the variables can be visualized as for the map of cases (see Figure 6).

Figure 6 summarizes the results of this analysis of the variables: subfigure (a) shows the map of the variables and its clustering; subfigures (b) and (c) show the projections of the variables for two clusters of variables. The similarity of the projection for variables belonging to the same cluster is obvious and it can be seen that different clusters indeed correspond to different behaviors of the variables.

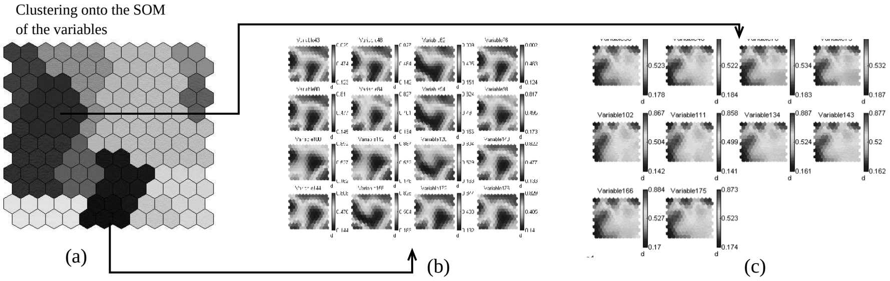

Fig. 6. Groups of variables with similar characteristics using K-means clustering

We used K-means for the clustering onto the SOM of the variables. Here again 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 $K^* = 11$.

3.4 Top View vs. Side View and Exploratory Data Analysis

Figure 7 sums up the complete process described above.

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 behaviors

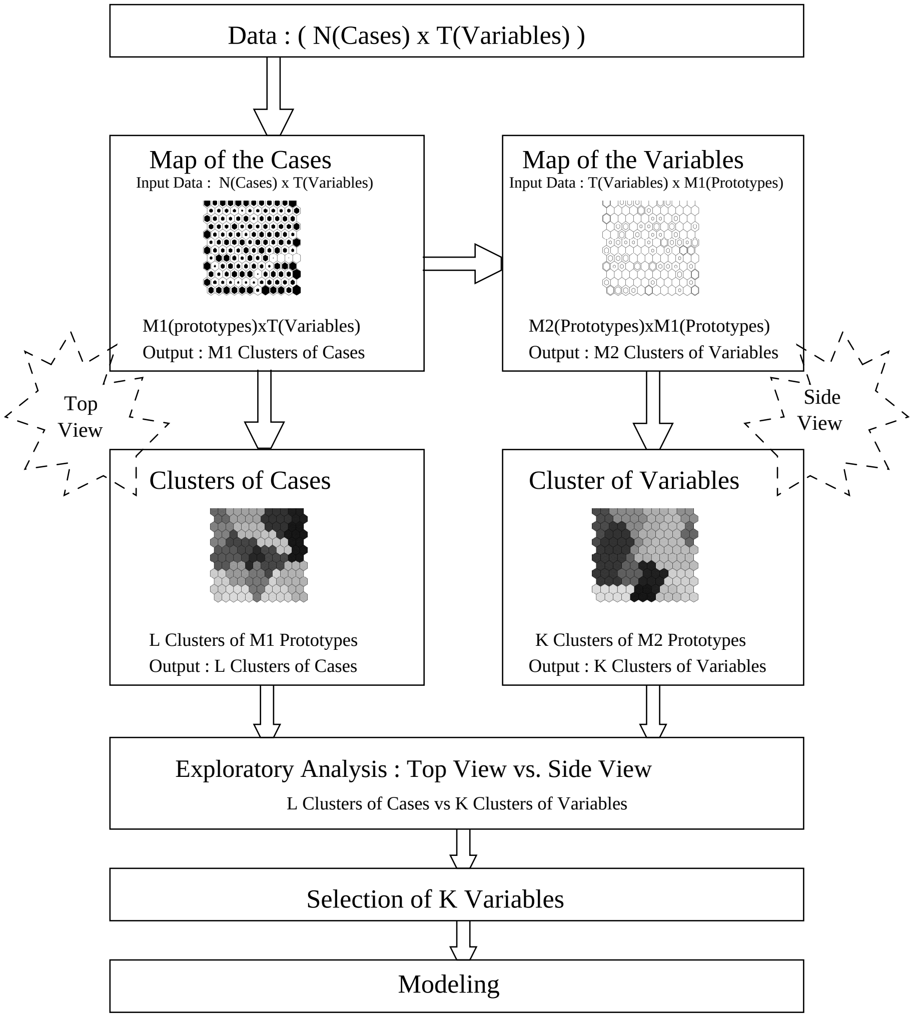
Data : (N(Cases) x T(Variables))
Map of the Cases
Input Data : N(Cases) x T(Variables)
M1(prototypes)xT(Variables)
Output : M1 Clusters of Cases
Map of the Variables
Input Data : T(Variables) x M1(Prototypes)
M2(Prototypes)xM1(Prototypes)
Output : M2 Clusters of Variables
Top View
Side View
Clusters of Cases
L Clusters of M1 Prototypes
Output : L Clusters of Cases
Cluster of Variables
K Clusters of M2 Prototypes
Output : K Clusters of Variables
Exploratory Analysis : Top View vs. Side View
L Clusters of Cases vs K Clusters of Variables
Selection of K Variables
Modeling

Fig. 7. Top View vs. Side View

relative to groups of variables and reciprocally, 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 cases.

The map of the variables allows an easier interpretation of the clusters of cases by re-ordering the variables according to their cluster. We see in Figure 8(a) the clusters on the map of the cases; in Figure 8(b) the mean value of the variables for the cases belonging to the cluster (A) without re-ordering; in Figure 8(c) the mean value of the variables for the cases belonging to the cluster (A) re-ordered according to their cluster. Figure 8(c) immediately shows how the different clusters of variables contribute to the formation of the cluster of cases A. Such accurate visual analysis is impossible with the raw ordering of the variables (Figure 8(b)).

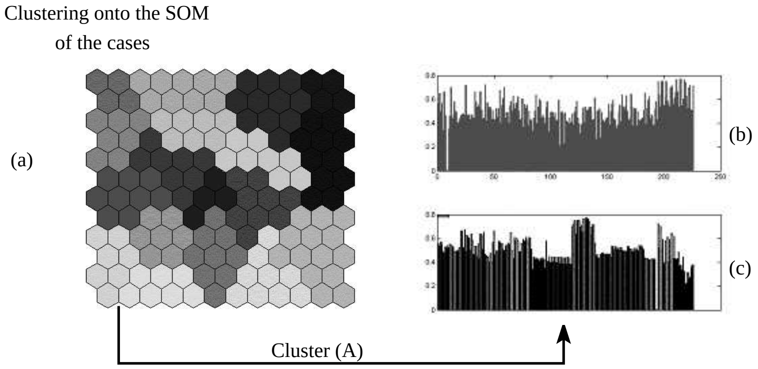

Fig. 8. Re-ordering the variables according to their cluster allow an easier interpretation of the cluster of the cases.

Figure 9 illustrates the complete exploratory analysis process which can be done using the method described above. The clustering of the SOM of the cases identifies 12 clusters of cases (a) The projection of the class information allows to visualize the fraudulent behaviors (b). The cluster (A) of the SOM of the cases (in the south-west corner of the map) exhibits fraudulent behavior (b).

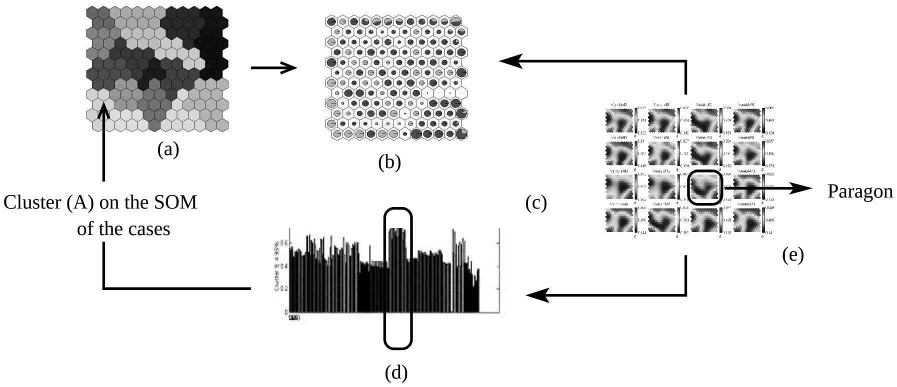

Fig. 9. Example of exploratory analysis.

The clustering obtained on the SOM of the variables allows the re-ordering of the mean value of the variables for the cases belonging to each cluster (c). For the cluster (A) of the SOM of the cases we see a cluster of stronger variables (d). This group of variable is presented in (e): all the variables are strongly correlated. The grouping of these variables in this cluster is naturally interpreted: these variables represent information about card phone (the amount of communication via a card phone under five temporal observation windows) and indicate that a specific card phone usage pattern is strongly correlated with a fraudulent behavior.

4 Dimensionality Reduction vs. Variable Selection

In this article, “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 modeling problem. In both cases, this is a combinatorial optimization problem.

The direct approach (the “wrapper” method) re-trains and re-evaluates a given model for many different feature/variable sets. The “filter” method approximation optimises 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].

As we have seen that each cluster of variables gathers variables with very close profiles, we can exploit this clustering for variable selection in a very natural way: we choose one representative variable per cluster, as the “paragon” of the cluster, i.e. the variable which minimizes the sum of the distances to the other variables of the cluster.

We choose to implement dimensionality reduction by building one feature per cluster as a sum of the variables of the cluster (variables are mean-centered and reduced to unit variance before summing). Both techniques are illustrated in Figure 10.

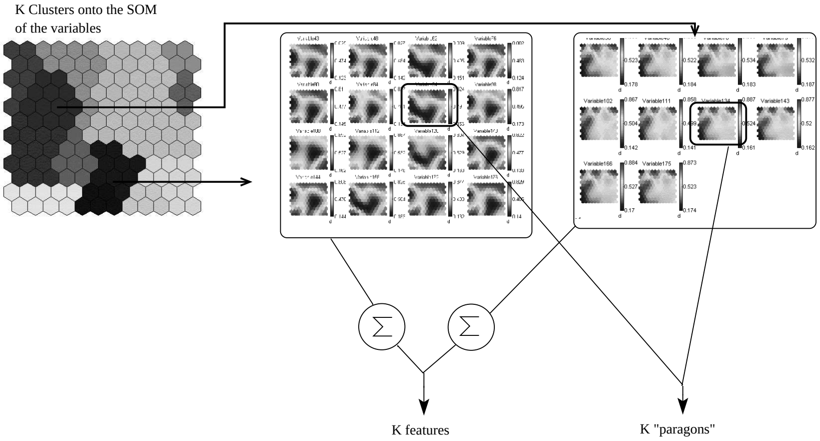

Fig. 10. SOM-based dimensionality reduction and variable selection

Both methods reduce the number of input variables to the number K^* of clusters found on the map of the variables. Modeling after variable selection 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.

5 Methodology: Comparison and Results

Other machine learning techniques also allow to realize variable selection such as 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.

5.1 Experimental Conditions

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 the number of clusters of variables found in section 3.3.

Multi-layer Perceptrons

Each neural network, in this article, 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 achieved 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 of neural network is the average estimated on 20 attempts). The P^* values are respectively 12, 10, 6 and 10.

Decision Tree

We used a commercial version of the algorithm C4.5 [8]. 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 $K^* = 11$.

Bayesian Network

The Bayesian network (BN) found is similar to the "Naïve 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 naïve Bayes classifier gives very good results (see [4]). The BN uses 37 variables[2], which is more than three times more than $K^* = 11$.

5.2 Results

The various classification performances are given below in the form of lift curves. The methodology described in the article gives excellent results (see Figure 11)

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. The SOM-based dimensionality reduction method has a performance 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 only relies on a few input variables.

[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.

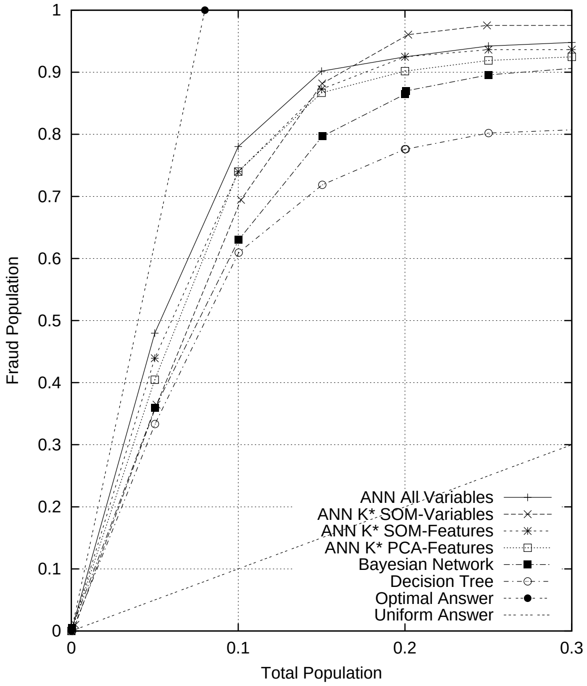

Fig. 11. Detection rate (%) of the fraudulent users obtained with different learning methods (ANN: Artificial Neural Network), given as a lift curve.

6 Conclusion

In this article 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 both in terms of visualization/interpretation ability and classification performance.

We have successfully applied the methodology described in this article on a variety of problems. It allows:

- to track down characteristic behavior of cases;
- to visualize synthetically various behaviors and their relationships;
- to group together variables which contribute in a similar manner to the constitution of the clustering of cases;
- to analyze the contribution of every group of variables to every group of cases;
- to realize a selection of variables;
- to make all interpretations with the initial variables.

Another example of the application of this methodology can be found in [3]. The authors show how SOM can be used to build successive layers of abstraction starting from low-level traffic data to achieve an interpretable clustering of customers and how the unique visualisation ability of SOM makes the analysis quite natural and easy to interpret.

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. F. Clérot and F. Fessant (2003). From IP port numbers to ADSL customer segmentation: Knowledge aggregation and representation using kohonen maps. In *Datamining 2003*, Rio Janeiro, December.
4. David J. Hand and Keming Yu. Idiot's bayes - not so stupid after all. *International Statistical Review*, 69(3):385–398, 2001.
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.R. Quinlan. *C4.5: Programs for Machine Learning.* Morgan Kaufmann, 1993.
9. Juha Vesanto. Som-based data visualization methods. *Inteligent Data Analysis*, 3(2):111–126, 1999.
10. Juha Vesanto. *Data Exploration Process Based on the Self-Organizing Map.* PhD thesis, Helsinki University of Technology, 2002.
11. 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/`.
12. S.M. Weiss and C.A. Kulikowski. *Computer Systems That Learn.* Morgan Kaufmann, 1991.