Discovering Complex Regularities by Adaptive Self Organizing Classification

A. Faro, D. Giordano, and F. Maiorana

Abstract - Data mining uses a variety of techniques each of which is useful for some particular task. It is important to have a deep understanding of each technique and be able to perform sophisticated analysis. In this article we describe a tool built to simulate a variation of the Kohonen network to perform unsupervised clustering and support the entire data mining process up to results visualization. A graphical representation helps the user to find out a strategy to optmize classification by adding, moving or delete a neuron in order to change the number of classes. The tool is also able to automatically suggest a strategy for number of classes optimization.
The tool is used to classify macroeconomic data that report the most developed countries' import and export. It is possible to classify the countries based on their economic behaviour and use an ad hoc tool to characterize the commercial behaviour of a country in a selected class from the analysis of positive and negative features that contribute to classes formation.

Keywords - Unsupervised classification, Kohonen networks, macroeconomics, Visual data mining, cluster interpretation.

I. Introduction

DATA mining is a growing research field that deals with information extraction from a large amount of data. In this process we have argued elsewhere [1] that the use of "ad hoc" built tools can greatly improve techniques understanding and usage, and simplify tasks such as data understanding and preparation or results visualization. Clustering techniques are standard tools in data mining [2]. Kohonen neural networks, or Self-Organizing Maps (SOM) [3] are a preferred alternatives to traditional hierarchical clustering methods because of their better performances regarding noise tolerance, speed and robustness [4]. One key idea behind SOM is to transform a n-dimensional input data into a lower dimensional display (typically bi-dimensional) where the clusters are mapped, following the principle that elements that are close in the input space typically belong to neighboring classes. Maps consist of neurons that correspond to reference vectors whose dimensions is the same as the dimension of the input data. However, when dealing with n-dimensional data deciding the boundaries of the clusters first, and then cluster interpretation is often a challenge. Some approaches have been proposed to solve this problem, for example the method of the U-matrix used in conjunction with plane projections [4] to facilitate clusters interpretation; and a growing self-organized map that preserves hierarchical structures in the data is presented [5].

In this paper we use an alternative approach for clustering n-dimensional data based on a competitive, unsupervised learning algorithm enhanced with a tool to change and reshape the visualization space by adding, removing or deleting free neurons in the classification layer of the network. The tool also provides automatic hints for optimal neural space reorganizations. Finally, a visualization facilities help the user to visualize the number of classes, the number of elements in the classes, and the distance of each element of the classes from its center. It also provides explanations as to why certain elements belong to a class, in terms of positive and negative features. In this paper we use this tool to group economic data into classes of similar patterns. The paper is organized as follows. Sect.2 illustrates the proposed approach contrasting it with classical SOM. Sect.3 illustrates the tool's functionalities and visual interface. Sect.4 reviews previous work in economics data mining and summarizes the results from the application of the tool to a previously mined macroeconomics data set. Sect.5 offers some concluding remarks.

II. Self organizing classification

Neural networks are often used to cluster input data. This grouping may be done automatically in an unsupervised way based on data correlations. The network architecture is composed by an input layer with as many nodes as the number of features of the items that must be classified and a competitive layer or output layer with as many nodes as the number of classes or cluster that must be found. Each input node is linked to all output nodes, and weights change during training until a termination criterion is satisfied.

Self-organizing Features Map (SOFM) is a competitive learning algorithm that has been widely used to obtain the mentioned unsupervised classification. At each iteration a winner (one neuron of the output layer) is found. The winner is computed by finding the output neuron with the lowest distance between the input data and the weights. The weights

A. Faro, D. Giordano, and F. Maiorana are with the University of Catania, Department of Computer Engineering and Telecommunications, Via A. Doria, 6, 95127 Catania. Phone: + 39-095-7382372; fax: +39-095-7382397; e-mail: afaro@ diit.unict.it, dgiordan@diit.unict.it, maioranaf@ tiscali.it .

of the winning neuron and the weights of neighboring neuron are updated by the following equation:

$$w(a_i) = w(a_i) + \nu\ \eta\ (m(i) - w(a_i))$$

where:

- **m** is the input matrix having N rows and M columns whose rows represents the input items and whose columns represent their features
- **m(i)** is the i-th row of matrix m representing the i-th input item,
- **w**(a_i) represent the weights linking the features of the input items to the output neurons belonging to a topological area a_i
- η is the learning rate of the network
- ν is a neighborhood function

As an example, assuming that the winning neuron for the current input is the neuron located at place (3,1) of the output bi-dimensional map, fig.1(A) shows the synaptic weights linking the input layer to the winning neuron that will be updated in the learning step. A possible topological area involved in this learning step is also shown. Fig.1 (B) shows that if at the end of the learning phase, the winning neuron for the i-th item is the neuron located at place (1,4) then the item is considered to belong to first of the four classes drawn in the map. The main problem of this algorithm is how to identify the classes, whereas the bi-dimensional map does not allow us to appreciate all the inter-classes relationships

.

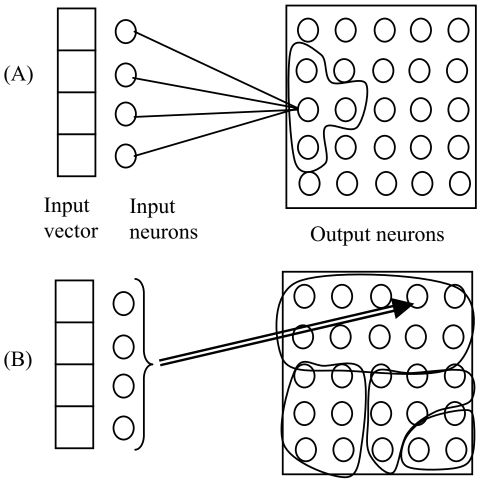


Figure 1 – Self-organizing Classification

To avoid the mentioned limits, we adopt the slightly different neural network outlined in fig. 2 where the neurons involved in a learning step are the winning neurons and other neurons (usually two or three neurons) that are mostly activated by the current item. The classes are given by the output neurons, i.e., if the neuron mostly activated by the i-th item is the j-th neuron, then the item belong to class j. Of course in our approach there is not topological similarity between output neurons since adjacent output neurons do not represent necessarily similar classes.

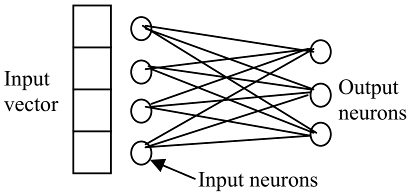


Fig.2 – Unsupervised neural classifier

In particular in our approach we adopt learning formula:

$$w(b_i) = w(b_i) + \nu\ \eta\ (m(i) - w(b_i))$$

Such formula is similar to the SOFM one but with the difference that the output neurons constitute a layer rather than a map, and that the neurons involved in each learning steps do not belong to a topological area but are the set b_i of the most activated neurons.

Since the number of classes is not known a-priori, a suitable strategy has to be adopted in order to find the best class number C. This can be done by resorting to the notion of linking energy per class (L_C) whose formal definition is given below. The maximum number of classes is the one beyond which the linking energy per class E_C does not increase. Since under certain conditions E_C might increase indefinitely thus determining that the final classes consists of only one item, a suitable threshold T could be considered enough to avoid to transform the output layer by adding another output neuron to cluster the original set of items by one more class. However, adding when $E_C < T$ an output neuron whose synaptic weights are randomly initialized, may cause a slow, possibly not effective, convergence towards a new cluster. Moreover, sometimes a better classification could be obtained by simply increasing the steps of the learning phase or by imposing, during the learning process, some small modification of the current synaptic values to avoid the algorithm remains trapped in some relative minimum. Finally a measure to evaluate if two classes are similar or not should be given. To manage automatically all these problems we define the following notion of linking energy:

$$E_c = 1 - L_c$$

$$L_c = \Sigma_j L_j / C \text{ for } j = 1 \text{ to } C$$

where L_j represents the mean distance (comprised between 0 and 1) of the items of the class j from the point that mostly belongs to class j, i.e., the one having for any i ranging over Ij the following weights: $w_{ir} = 0$ (if r # j) and $w_{ir} = 1$ (if r = j). L_j is as follows:

$$L_j = (\Sigma_{i(Ij)}\ (\Sigma_{r,i\#j}\ w_{ir}^{\ 2} + (1-w_{ij})^2\ )\ )^{0.5} / N$$

where i(Ij) is the set of the items belonging to class j and w_{ir} (i.e., the synaptic weight connecting input i and the output neuron r at the end of the learning phase) measures how much the item of the class j belongs also to class r. To find the distances between the classes r and s we calculate the quantities x_{hk} (h = 1 to C, j = k to C) as follows:

$$x_{hk} = \Sigma_{i(Ih)}\ w_{ik} / N_h$$

that represent for each class h the weights of the item P_h that is at the center of the items that belong to that class h. Thus two classes r and s is simply the distance between P_r and P_h as follows:

$$D_{rs} = \Sigma_j (x_{rj} - x_{sj})^{0.5} / C$$

These measures allows us to outline an automated strategy to find the number of classes as follows:

1. if L_c has a high value or some D_{rs} shows a low value then increase the output neurons by one, and put the synaptic weights of this new neuron equal to the values of P_j, where i refers to the class with the highest L_j. If adding a new neuron does not produce an increase of L_c then delete the neuron with the highest L_j in the clustering obtained by C+1 output neurons and follow step 2.
2. if L_c or some D_{rs} have an intermediate value, then try to optimize the clustering first by moving the neuron of the highest L_j between the classes having lowest D_{rs} and then by increasing the number of the learning steps.

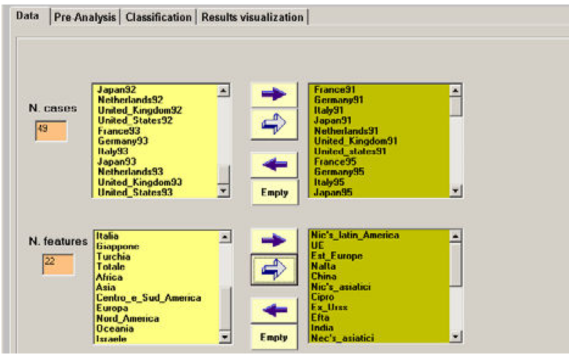
III. TOOL DESCRIPTION

The mining process supported by the tool consists of four phases: in the first phase we load the data from a text file and visualize them in a matrix m. We can choose the items to analyze (the rows of the matrix) and the features of each element (the columns). In the second phase data can be manipulated to improve data quality. In particular the original matrix is transformed into a similarity matrix as follows:

1. For each column j of the matrix m find the maximum Max_j and the minimum value Min_j and then normalize all the value of the column m(i, j) between 0 and 1, i.e., $m_{norm}(i, j) = (m(i, j) - Min_j))/ (Max_j - Min_j)$
2. Compute the similarity matrix s(i,j) whose general element measures the similarity between item I and item J as the cartesian distance between rows i and j
3. find the maximum Smax and the minimum value Smin of the matrix s and then normalize all the value s(i, j) between 0 and 1, i.e., s(i, j) = (s(i, j) - Smin))/ (Smax - Smin)

Of course, if the original matrix represents how much an item is linked to the others, only the mentioned step 3 has to be performed. This shows that the tool is suitable not only to classify items expressed by features/keywords but especially items interrelated by links whose value (usually between 0 and 1) expresses how much an item is influenced or similar to the others. This is common in many domains (e.g., references in literature and scientific production, inspiration in art and design). In the third phase we use the self organizing network presented in the previous section to classify the data.

The tool interface (Fig. 3) allows the user to choose the items and which features to consider. The items are the rows whereas the features are the columns of the matrix m from which the proposed classification method starts. The similarity matrix s is then computed to be passed to the self organizing classification algorithm. The neural network has to be initialized by setting some parameters such as the maximum number of cycles, the number of final classes, the updating neighborhood and the learning rate. The interface displays the current cycles and a progress bar indicates the status of the classification process.



N. cases 49

Japan92
Netherlands92
United_Kingdom92
United_States92
France93
Germany93
Italy93
Japan93
Netherlands93
United_Kingdom93
United_States93

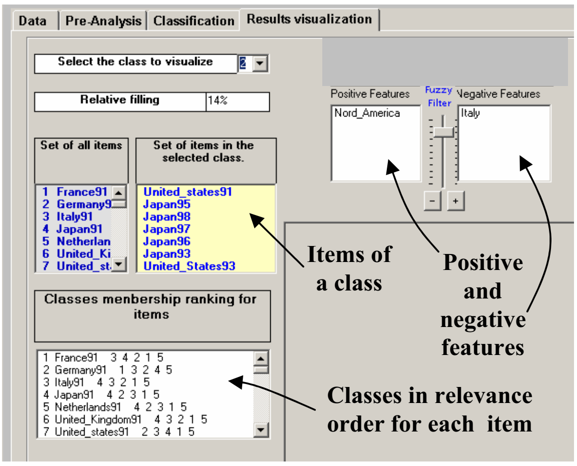
France91
Germany91
Italy91
Japan91
Netherlands91
United_Kingdom91
United_states91
France95
Germany95
Italy95
Japan95

N. features 22

Italia
Giappone
Turchia
Totale
Africa
Asia
Centro_e_Sud_America
Europa
Nord_America
Oceania
Israele

Nic's_latin_America
UE
Est_Europe
Nalta
China
Nic's_asiatici
Cipro
Ex_Urss
Efta
India
Nec's_asiatici

Fig.3 – Items/cases and features available for classification (on the left) and items/features chosen (on the right)



Select the class to visualize 2

Relative filling 14%

Set of all items

1 France91
2 Germany9
3 Italy91
4 Japan91
5 Netherlan
6 United_Ki
7 United_st

Set of items in the selected class.

United_states91
Japan95
Japan98
Japan97
Japan96
Japan93
United_States93

Classes menbership ranking for items

1 France91 3 4 2 1 5
2 Germany91 1 3 2 4 5
3 Italy91 4 3 2 1 5
4 Japan91 4 2 3 1 5
5 Netherlands91 4 2 3 1 5
6 United_Kingdom91 4 3 2 1 5
7 United_states91 2 3 4 1 5

Positive Features: Nord_America

Negative Features: Italy

Items of a class

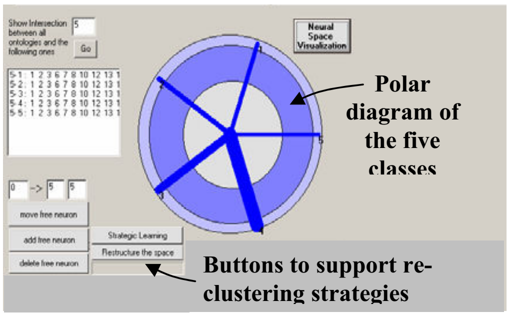
Positive and negative features

Classes in relevance order for each item

Fig.4 - Classification results

The classification results are shown in another window (fig.4) where it is possible to know what items belong to each class and what are the features that have determined the insertion of the item in the class. The relevant features are identified by computing what are for each class the features that have a high or low value (positive and negative features) for all the items of the class. Moreover, for each item the tool shows the classes to which it belongs to in order of relevance. If this item is of some interest this is a simple way to find all the items that may be of relevance too, e.g., the items belonging to the classes that are of first and the second relevance for the item initially retrieved.

The last section of the tool deals with the restructuring strategy (fig.5), i.e., it gives an idea of how many items are belonging to the classes and indicates what is the best action to do for optimizing the clustering. This is obtained by a polar diagram where each line represents a class. The width of these lines represents the number of items. The color black and violet indicates that the class has no item or that it consists of only one item. If the length of the line is within internal circle then the class is dispersed and the insertion of a new neuron is suggested. The restructuring strategy may be implemented by the user following the indications of such diagram or the user may ask the tool to compute automatically and implement the best action to do.



```
5-1: 1 2 3 6 7 8 10 12 13 1
5-2: 1 2 3 6 7 8 10 12 13 1
5-3: 1 2 3 6 7 8 10 12 13 1
5-4: 1 2 3 6 7 8 10 12 13 1
5-5: 1 2 3 6 7 8 10 12 13 1
```

Polar diagram of the five classes

Buttons to support re-clustering strategies

Fig.5 – Polar diagram of the entire classification and strategic learning

IV. Application to Macroeconomics

Data mining usage in economics is a new developing field that uses data mining techniques not only to explore data but also to find a model for the data even if this model is often built without an underlying economic theory. The lack of economic theory in the data mining model doesn't affect the result of the techniques used [6]. Financial information can be divided in economic and business information [5]. Business deals with companies and their profit and expenses (microeconomics), economic data deals with the national level (macroeconomics). A mining analysis of macroeconomics data was performed in Bordoni, Giordano and Spadaro [8] by mining a database with over 400,000 records regarding the import and export of the seven most economically developed countries in the period from 1990 to 1998. The analysis was carried out with IBM Intelligent Miner, by unsupervised neural clustering techniques. As a result, national product specialisation sectors were identified, the evolution of the economic structures of the considered countries was traced and the similarities among the countries were highlighted.

In this paper we take into account the same set of data in order to evaluate the effectiveness of the proposed environment. The result of the classification showed that the best number of classes is five. This solution represents a good compromise with classes containing a significant number of items in respect to the total number of items, and a distance between each element of the class and the class center not too great. The class found contains the following items:

1. class one : Italy in the years 91, 92, 93, 95, 96, 97 and 98; Japan in the years 91 and 92; Netherlands in the years 91, 92, 93, 95, 96, 97 and 98; UK in the years 91, 92 and 93;
2. class two: Germany in the years 91, 92, 95, 96, 97, 98;
3. class three: France in the years 91, 92, 93, 95, 96, 97, 98; UK in the year 95, 96, 97, 98; Germany in 93;
4. class four: United States in the years 92, 95, 96, 97 98;
5. class five: United States in the years 91 and 93; Japan in the years 93, 95, 96, 97, 98.

This classification is in accordance with the one obtained by the Intelligent Miner, but the reasons provided by the tool supporting this clustering are more intelligible; in fact the tool indicates the market areas characterizing each class, allows us to easily select a subclass and go into further sub-clustering in order to better evaluate the competitors; it points out for each class the nearest classes and for each item the two-three class to which it belongs to in order to analyze into details the market sectors of specific interest.

V. Concluding Remarks

In this paper we presented a method that affords the advantage of being simple to use and affords the user interactive control over the classification process, which is important to gain an understanding of the data. Complex regularities may be pointed out by using the tool features thus avoiding the simplification introduced by the bi-dimensional mapping of the SOFM classification or complicated SOFM based approaches that aim at enlarging the dimensions of the clustering space. To complement the method, a powerful visualization tool that allow the user to virtually navigate the N-dimensional classification space generated by the tool has been developed. Usability studies are being carried out to evaluate the advantages of this virtual reality inspired interface for improving the discovery of complex regularities by highly interactive mining sessions.

References

[1] D. Giordano, F. Maiorana. A visual tool for mining macroeconomics data. In A. Zanasi, N.F.F. Ebecken, & C. Brebbia (eds.): *Data mining V*. WIT Press, 2004.

[2] Kohonen, T. *Self-Organizing Maps*. Springer-Verlag, 2001.

[3] Jain, AK, Murty, M.N. Flynn, P.J. Data clustering: a review. *ACM Computing Surveys,* Sept. 1999.

[4] Hautaniemi, S. Yli-HAria, O. Astola, J. et al. (2003). Analysis and visualization of gene expression microarray data in human cancer using self-organizing maps. *Machine Learning* 52, 45-66.

[5] Dittenbach M, Rauber, A. Merkl, D. (2002). Uncovering hierarchical structure in data using the growing hierarchical self-organizing map. *Neurocomputing* 48 (2002) 199-216.

[6] Felders, A. J. Data mining in economic science [Online]. Available at : http://www.cs.uu.nl/people/ad/dmecon.pdf.

[7] Lux, M. "Visualization of financial data" in *Proc. Workshop on New Paradigm in Information Visualization* (1997)

[8] L.Bordoni, D. Giordano, S.Spadaro. Il data mining: un'applicazione agli studi macroeconomici. *Atti del convegno AICA 2002* (Associazione Italiana Calcolo Automatico), pp. 557 – 61, 2002.