Free Projection SOM: A New Method For SOM-Based Cluster Visualization

ABDEL-BADEEH M. SALEM[1], EMAD MONIER[2], KHALED NAGATY[2]
Computer Science Department
Ain Shams University
Faculty of Computer & Information Sciences, Abbassia, Cairo
EGYPT

Abstract: - In this paper an extension to the learning rule of the Self-Organizing Map (SOM) namely the Free Projection SOM (FP-SOM) is presented in order to enhance the SOM projection. The general idea of the FP-SOM is to mirror the movement of weight vectors during the training process allowing their images on the map grid to move more freely between the junctions. The result of the extended training algorithm allows intuitive analysis of the similarities inherent in the input data and most important, intuitive recognition of cluster boundaries. Experiments on artificial and real data sets show the advantages of the proposed extension as a cluster visualization method.

Key-Words: - Data Mining, Cluster Visualization, Self-organizing Maps, High-dimensional Data Projection.

1 Introduction

Data mining is an emerging area of new research efforts, responding to the presence of large databases in com-merce, industry and research. It is also a title for a large number of widely divergent methods ranging from belief networks and relational learning to statistics and neural networks. Data mining is part of a larger framework, Knowledge Discovery in Databases (KDD) [1], whose purpose is to find new knowledge from databases where dimension, complexity or amount of data is prohibitively large for human observation alone. Data mining is an iterative process requiring that the intuition and background knowledge of humans be coupled with the computational efficiency of modern computer technology. For this reason, visualization is a very important part of data mining.

By nature, visualization requires a mapping process from the high dimensional input space to a low dimensional output space. This problem can be attenuated by projection techniques such as the well-known Principal Component Analysis (PCA) [2]. However, PCA is a strictly linear method that is unable to detect nonlinear dependencies between variables. Numerous nonlinear projection methods have been created to address this issue. For example, the nonmetric Multidimensional Scaling (MDS) [3] and Sammon's nonlinear mapping (NLM) [4] are based on the preservation of either pair-wise dissimilarities or Euclidean stances. Neural ver-sions of the NLM, like Curvilinear Component Analysis (CCA) [5], [6] generally show better performance, particularly when they do not use the traditional Euclidean metrics [7], [8]. Finally, nonlinear projection can be achieved by the Self-Organizing Map (SOM) [9]. But the projection implemented by the Self-Organizing Map (SOM) is restricted to the junctions of the map grid, which makes it very crude and raises the necessity to use other computationally expensive projection methods.

The Self-Organizing Map is a neural network algorithm based on unsupervised learning. It has proven to be a valuable tool in data mining and KDD with applications in full-text and financial data analysis. It has also been successfully applied in various engineering applications in pattern recognition, image analysis, process monitoring and fault diagnosis [10], [11]. The use of the SOM in exploratory data analysis is studied in [12], [13], [14], [15].

The SOM has several beneficial features, which makes it a useful method in data mining. It implements an ordered dimensionality-reducing mapping of the training data. The map follows the probability density function of the data and is robust to missing data. It is readily explainable, simple and - most importantly - easy to visualize. Visualization of complex multidimensional data is indeed one of the main application areas of SOM.

In spite of these advantages, the projection implemented by the SOM is restricted to the junctions of the map grid, and therefore it is very crude. To visualize the shape of the SOM in the input space, the prototype vectors of the map are typically projected separately using one of the previously mentioned methods. In this paper an

	Attribute	Dove	Hen	Duck	Goose	Owl	Hawk	Eagle	Fox	Dog	Wolf	Cat	Tiger	Lion	Horse	Zebra	Cow
Is	Small	1	1	1	1	1	1	0	0	0	0	1	0	0	0	0	0
	Medium	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0
	Big	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
Has	2 legs	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
	4 legs	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
	Hair	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
	Hooves	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
	Mane	0	0	0	0	0	0	0	0	0	1	0	0	1	1	1	0
	Feather	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
Likes to	Hunt	0	0	0	0	1	1	1	1	0	1	1	1	1	0	0	0
	Run	0	0	0	0	0	0	0	0	1	1	0	1	1	1	1	0
	Fly	1	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0
	Swim	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0

Fig. 1. The Animal data set.

extension to the learning algorithm of the SOM is presented in order to enhance the SOM projection for the sake of better visualization of clusters in the input data.

We show the applicability and the effect of this extension, referred to as the Free Projection SOM (FP-SOM) approach with a prominent artificial test data set for semantic classification and another real data set. The remainder of the paper is organized as follows. Section 2 gives an overview on the self-organizing map algorithm and the proposed extension of its learning rule. Section 3 presents the implementation and results of the FP-SOM applied on a popular benchmark artificial data set and a real data set. A discussion about the performance and benefits of the FP-SOM is given in section 4. Finally our conclusion is presented in section 5.

2 The Free Projection Self-Organizing Map (FP-SOM)

The SOM consists of neurons located on a regular low-dimensional grid, usually 1- or 2-dimensional. Higher dimensional grids are possible, but they are not generally used since their visualization is problematic. The lattice of the grid can be either hexagonal or rectangular. Each neuron k is represented by an n-dimensional prototype (weight) vector $\boldsymbol{m}_k = [m_{k1}, \ldots, m_{kn}]$, n is the dimension of the input space. On each training step, a data sample $\boldsymbol{x}$ is selected and the nearest unit $\boldsymbol{m}_c$ (the best-matching unit, BMU) is found from the map. The prototype vectors of the BMU and its neighbors on the grid are moved toward the sample vector

$$\boldsymbol{m}_k = \boldsymbol{m}_k + \alpha(t)\, h_{ck}(t)\, (\boldsymbol{x} - \boldsymbol{m}_k) \qquad (1)$$

where $\alpha(t)$ is the learning rate and $h_{ck}(t)$ is a neighborhood kernel centered on the winner unit c. Both learning rate and neighborhood kernel radius decrease monotonically with time. In this paper the Gaussian neighborhood kernel [9] is used. It can be written as:

$$h_{ck}(t) = \exp\left(-\frac{\|r_c - r_k\|^2}{2\sigma^2(t)}\right) \qquad (2)$$

where $r_c \in \Re^2$ and $r_k \in \Re^2$ are the location vectors of neurons c and k respectively, in the grid and $\sigma(t)$ is a monotonically decreasing function of time that defines the width of the kernel.

The basic idea of the FP-SOM is to mirror the movement of prototype vectors during the training process allowing their images on the map grid to move more freely between the junctions in a way that makes the boundaries between related and unrelated input data intuitively recognizable.

Each prototype vector is assigned a position $\boldsymbol{p}$, where $\boldsymbol{p} \in \Re^2$. This position is initialized to random values around 1. At each training cycle the activation of the various prototype vectors is done according to (1). A similar activation is done on the position vectors according to:

$$\boldsymbol{p}_k = \boldsymbol{p}_k + \alpha(t)\, v_{ck}(t)\, (\boldsymbol{r}_c - \boldsymbol{p}_k) \qquad (3)$$

Where $v_{ck}(t)$ is another Gaussian neighborhood kernel defined as:

$$v_{ck}(t) = \exp\left(-\frac{\|x - m_k\|^2}{2\sigma^2(t)}\right) \qquad (4)$$

We note that the position vectors $\boldsymbol{p}_k$ learn from $\boldsymbol{r}_c$ the location of the BMU in the map grid. Also the neighborhood kernel depends on the distance between the sample data vector $\boldsymbol{x}$ and the prototype vector $\boldsymbol{m}_k$. Thus the clustering of units around the winning unit resembles the clustering of the units weight vectors around the presented input signal after the current training cycle.

After convergence of the training process the clusters learned by the self-organizing map can be visualized by using the position vectors of the units for graphical representation.

3 Experimental Results

For the example presented below we used the *Animals* data set (see Fig. 1) as defined in [16] because it is widely used and discussed as a

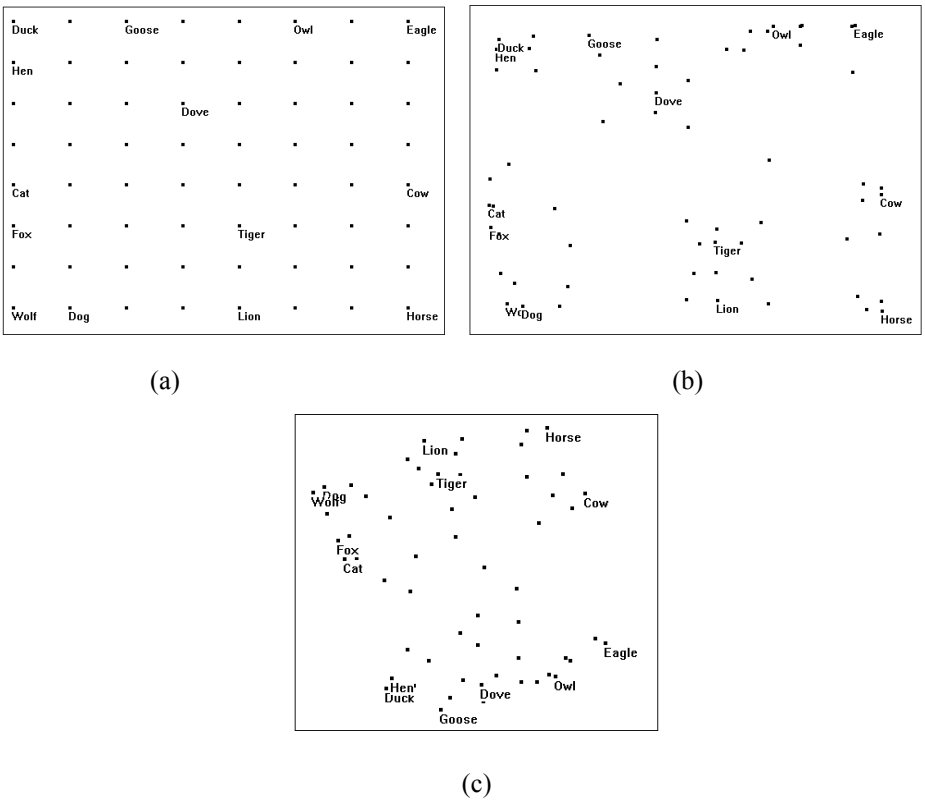

Fig. 2. 8×8 SOM trained on the Animal data set, (a) Standard SOM Representation (b) FP-SOM Representation (c) Sammon nonlinear mapping of the SOM prototype vectors.

reference data set in numerous papers on related topics (e.g. [17]) allowing easy comparison of the results obtained. Furthermore, the data set is simple enough to be presented within the limited space of this paper while still being sufficiently complex to demonstrate the main features of our approach. Please note that we omitted the encoding of the animals' names in the input vector as suggested in [16], hence the animals *Hawk* and *Owl* as well as *Zebra* and *Horse* are mapped onto the same unit respectively since they have equal feature vector representation.

Figure 2(a) depicts a trained 8×8 SOM with the winning units being labeled by the corresponding input signal. In spite of the fact, that the map presents a topographic ordering of the input signals (e.g. all birds are mapped to the upper half of the map) cluster boundaries are not detectable from this standard representation, unless one has sufficient prior knowledge concerning the input data. Fig. 2(c) gives the Sammon's nonlinear mapping of the SOM prototype vectors. It is clear that the cluster boundaries are not recognized easily and also the locations of the clusters are changed which makes it difficult to link with other SOM-based visualizatioins. This is in addition to the long time taken for its computation.

Contrary to that, the representation of FP-SOM as provided in Fig. 2(b) clearly shows the distinct clusters learned by the map, the birds are separated from all other animals in the bottom of the map, the cluster itself being substructured into two groups consisting of the hunting birds *Hawk* and *Eagle* and the non-hunting *Duck, Hen*, *Goose* and *Dove*. *Horse* and *Cow* on the bottom right corner are separated from both the birds and the big hunters *Tiger* and *Lion*. There is another sub-cluster in the upper right corner containing the medium sized carnivore animals *Dog*, *Wolf*, *cat* and *Fox*.

Finally the FP-SOM has been applied on a real data set namely the Iris data set [18]. Iris flower data set contains 3 classes and 150 vectors, 50 in each class, where each class refers to a type of Iris plant, namely, Setosa, Virginica and Versicolor. Each vector has 4 continuos attributes: Septal length, Septal width, Petal length, and Petal width. We refer to the 3 classes as class ***a***, class ***b*** and class ***c*** respectively. Its clear from Fig. 3 (b) that class ***a*** is distant from both class ***b*** and ***c*** and class ***c*** occupies the upper left corner of the map. The Sammon's nonlinear mapping of the map vectors almost give

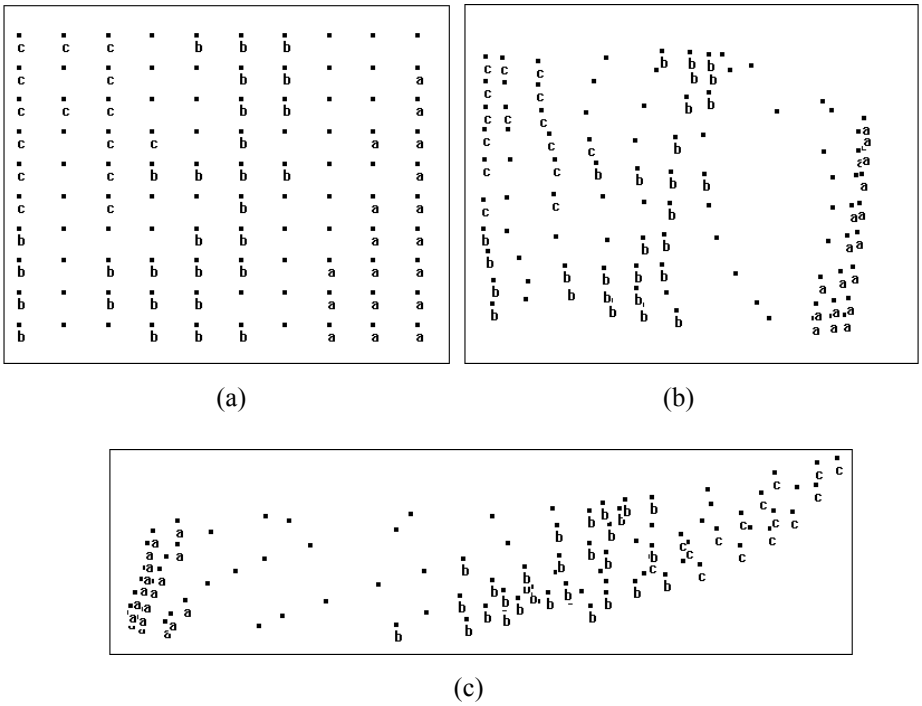

Fig. 3. 10 × 10 SOM trained on the Iris data set, (a) Standard SOM Representation (b) FP-SOM Representation (c) Sammon's nonlinear mapping of the SOM prototype vectors.

the same information but it takes a lot of computational time.

4 Discussion

FP-SOM provides a method to visualize and detect the structures learned by a self-organizing map as an extension to the standard training process. The structure of the input data as analyzed by the resulting mapping is clearly visible as a set of clusters within the map. Each cluster in turn may consist of a set of sub clusters providing a finer granularity for exploratory data analysis. On the other hand the clusters themselves are organized in the map corresponding to their mutual similarity i.e. similar clusters are located more closely to each other than distinct ones. Note that the FP-SOM representation does not reveal new clusters compared to the standard representation of self-organizing maps in the sense that the same mapping is represented. In other words the structure of the FP-SOM presentation is identical to the structure present in the standard SOM as far as the overall topographic ordering is concerned.

The basic differences are that - due to the fixed neighborhood relation within the grid and the fixed distance between units - first, cluster boundaries usually can not be detected satisfactorily and second the degree of similarity is not or only to a limited extent expressed by the basic network topology. To overcome this limitation the extended training algorithm allows units to change their location more freely within the map with respect to their weight vector's movement. Thus the training process leads to varying distances between neighboring units in terms of both network topology and their neighborhood in terms of input space. By being an extension to the basic SOM training procedure FP-SOM provides a supplementary improved visualization method for self-organizing maps instead of replacing the existing architecture. Thus results obtained and experiences gathered so far with the basic SOM learning process are still valid and can further be used. Another benefit of being an extension to the standard fixed grid representation is the ability to link between the resulting visualization and other visualizations -which are based on the fixed grid - by position [13]. Furthermore, the computational cost to obtain the FP-SOM representation is neglectable, consisting in two operations performed on the position vectors: first initialization, which is done once and second an adaptation analogous to the one performed on the map units (see Section 2). With respect to the space complexity, FP-SOM uses only a position vector consists of two attributes (x and y coordinates) for each map unit.

4 Conclusion

FP-SOM has been shown as an extended learning algorithm for self-organizing maps. The development of the extension was motivated by the fact that the projection of the SOM is restricted to the junctions of the map grid, which makes it difficult to detect the cluster boundaries in the standard representation. FP-SOM enables both the visualization of similarity of input data comprising one cluster and the visualization of similarity between different clusters. Furthermore FP-SOM representation does not affect the standard training process or the fixed grid structure of the SOM. The computational cost for obtaining the FP-SOM representation is small compared to other methods that are used separately to project the map units, for example Sammon nonlinear mapping.

References:

[1] U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors. *Advances in Knowledge Discovery and Data Mining*. AAAI Press / The MIT Press, California (1996)

[2] I. T. Jollife. *Principal Component Analysis*. Springer-Verlag, New York (1986)

[3] R. N. Shepard. *The analysis of proximities: Multidimensional Scaling with an unknown distance function*. Psychometrika, 27:125-140, (1962)

[4] W. Sammon, J. A nonlinear mapping algorithm for data structure analysis. *IEEE Transactions on Computers*, CC-18 (5):401-409, (1969)

[5] P. Demartines and J. Herault. Vector Quantization and Projection Neural Network. In A. Prieto, J. Mira, and J. Cabestany, editors, *Lecture Notes in Computer Science*, volume 686, pages 328-333. Springer-Verlag (1993)

[6] P. Demartines and J. Herault. Curvilinear Component Analysis: A self-organizing neural network for nonlinear mapping of data sets. *IEEE Transaction on Neural Networks*, 8(1):148-154, January (1997)

[7] J. A. Lee, A. Lendasse, N. Donckers, and M. Verleysen. A Robust Nonlinear Projection Method. In M. Verleysen, editor, Proceedings of *ESANN'2000, 8th European Symposium on Artificial Neural Networks,* pages 13-20. D-Facto public.,Bruges Belgium, April (2000)

[8] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science,* 290(5500): 2319-2323, (2000)

[9] T. Kohonen. *Self-Organizing Maps.* Springer, Berlin, Heidelberg, (2001)

[10] T. Kohonen, E. Oja, O. Simula, A. Visa, and J. Kangas. Engineering applications of the self-organizing map. *Proceeding of the IEEE,* 84(10):27 pages, October (1996)

[11] O. Simula and J. Kangas. *Neural Networks for Chemical Engineers,* volume 6, of Computer-Aided Chemical Engineering, chapter 14, Process monitoring and visualization using self-organizing maps. Elsevier, Amsterdam, (1995)

[12] G. Deboek and T. Kohonen. V*isual Exploration in Finance using Self-Organizing Maps.* Spriger-Verlage, London, (1998)

[13] S. Kaski. Data Exploration Using Self-Organizing Maps. *PhD Thesis*, Helisinki University of Technology, 1997.

[14] J. Mao and A.K. Jain. Artificial neural networks for feature extraction and multivariate data projection. *IEEE Transaction on Neural Networks,* 6(2):296-317, March (1995)

[15] W. Pedrycz and H.C. Card. Linguistic interpretation of self-organizing maps. *In Proceeding of International Conference on Fuzzy Systems,* (1992)

[16] .H. Ritter. T. Kohonen *Self-organizing semantic maps.* Biological Cybernetics, 54. (1989)

[17] B.Fritzke. Growing Cell Structures: A self-organizing neural network for unsupervised and supervised learning. Neural Networks, 7(9). (1994)

[18] C.L. Blake and C.J. Meretz. *UCI repository of machine learning databases,* (1998)