

Boosting Unsupervised Competitive Learning Ensembles

Emilio Corchado¹, Bruno Baruque¹, and Hujun Yin²

¹ Department of Civil Engineering. University of Burgos, Spain
escorchado@ubu.es, bbaruque@ubu.es

² School of Electrical and Electronic Engineering. University of Manchester, UK
h.yin@manchester.ac.uk

Abstract. Topology preserving mappings are great tools for data visualization and inspection in large datasets. This research presents a combination of several topology preserving mapping models with some basic classifier ensemble and boosting techniques in order to increase the stability conditions and, as an extension, the classification capabilities of the former. A study and comparison of the performance of some novel and classical ensemble techniques are presented in this paper to test their suitability, both in the fields of data visualization and classification when combined with topology preserving models such as the SOM, ViSOM or ML-SIM.

Keywords: topology preserving mappings, boosting, bagging, unsupervised learning.

1 Introduction

Several tools can be used to treat the high amounts of data that industrial and business operations processes. A very useful one is the unsupervised learning, in the field of artificial neural networks (ANNs). For unsupervised learning only the input and the network's internal dynamics are the two elements required. No external mechanism is used to check on the weight setting mechanism.

This paper is based in one of the two major methods of unsupervised learning: competitive learning, where the output neurons of a neural network compete among themselves for being the one to be active. This too mirrors the reality of what happens in the brain in that there are finite resources for learning and so one neuron's gain means another's loss.

The principal problem of the models based on competitive learning is, as happens with all ANNs, their instability. This research is focused on the comparison and study of some novel and classical ensemble extension versions of some competitive learning models based on the topology preserving concept. Therefore the extended models are the Self-organising Map (SOM) [1], the Visualization-induced Self-Organizing Maps (ViSOM) [2] and the Maximum Likelihood Scale Invariant Map (ML-SIM) [3] in combination with some simple ensemble techniques such as the

Bagging [4] or the AdaBoost [5]. The aim is to verify if the performance of these unsupervised connectionist models can be improved by means of these ensemble meta-algorithms.

2 Self-Organizing Maps

The Self-Organizing Map (SOM) algorithm [1] is based on a type of unsupervised learning called competitive learning; an adaptive process in which the neurons in a neural network gradually become sensitive to different input categories, sets of samples in a specific domain of the input space [6]. Its aim is to produce a low dimensional representation of the training samples while preserving the topological properties of the input space.

This study focuses on an interesting extension of this algorithm. It is called Visualization Induced SOM (ViSOM) [2] and it is proposed to directly preserve the local distance information on the map, along with the topology. The ViSOM constrains the lateral contraction forces between neurons and hence regularises the interneuron distances so that distances between neurons in the data space are in proportion to those in the input space [7].

The difference between the SOM and the ViSOM hence lies in the update of the weights of the neighbours of the winner neuron as can be seen from Eqs (1) and (2).

Update of neighbourhood neurons in SOM:

$$w_k(t+1) = w_k(t) + \alpha(t)\eta(v,k,t)(x(t) - w_v(t)) \quad (1)$$

Update of neighbourhood neurons in ViSOM:

$$w_k(t+1) = w_k(t) + \alpha(t)\eta(v,k,t) \left([x(t) - w_v(t)] + [w_v(t) - w_k(t)] \left(\frac{d_{vk}}{\Delta_{vk}\lambda} - 1 \right) \right) \quad (2)$$

where w_v is the winning neuron, α the learning rate of the algorithm, $\eta(v,k,t)$ is the neighbourhood function where v represents the position of the winning neuron in the lattice and k the positions of the neurons in the neighbourhood of this one, x is the input to the network and λ is a “resolution” parameter, d_{vk} and Δ_{vk} are the distances between the neurons in the data space and in the map space respectively.

Another example of a topographic mapping algorithm is the Maximum Likelihood Scale Invariant Map (ML-SIM) [3]. It is similar to the SOM but in this case training is based on the use of a particular Exploratory Projection Pursuit (EPP) model [9] called Maximum Likelihood Hebbian Learning (MLHL) Network [9] [10]. The competitive learning and a neighbourhood function are then used in a similar way as in the SOM. The distinctiveness is that in this case the winner’s activation is then fed back through its weights and this is subtracted from the inputs to calculate the error or residual. Then the MLHL algorithm is used to update the weights of all nodes in the neighbourhood of the winner, which can be expressed as,

$$\mathbf{e}(t) = x(t) - w_v(t) \cdot y_v, (y_v = 1) \quad (3)$$

$$w_k(t+1) = \alpha(t) \cdot \eta(v, k, t) \cdot \text{sign}(\mathbf{e}(t) - w_v(t)) |\mathbf{e}(t) - w_v(t)|^{p-1}, \forall i \in N_c \quad (4)$$

These three models can be adapted for classification of new samples using a semi-supervised procedure [11]. This added feature can also serve as a measure of the stability of the trained network. A high accuracy in the classification rate implies that the neurons of the network are reacting in a more consistent way to the classes of the samples that are presented. As a consequence, the map should represent the data distribution more accurately.

3 Unsupervised Competitive Learning Ensembles

The ultimate goal of constructing an ensemble is to improve the performance obtained of a single working unit. When talking about classification it is generally accepted that the sets of patterns misclassified by the different classifiers would not necessarily overlap. This suggests that different classifier designs potentially offer complementary information about the patterns to be classified and could be harnessed to improve the performance of the selected classifier [12]. Many ensemble models and theories have been developed in the previous years that range from a quite simple technique like Bagging [4] or AdaBoost [5] to the more sophisticated ones such as LPBoost [13] or and many other variants. These techniques have been mainly applied to models designed specifically for classification, specially supervised classifiers [14]. In the present study the central idea is to verify the improvements that an ensemble technique can provide in the multi-dimensional data visualization field over an unsupervised learning process such as the Competitive Learning.

3.1 Bagging and AdaBoosting

Boosting meta-algorithms consists on training a simple classifier in several stages by incrementally adding new capacities to the current learned function. In the case of the present work the decision taken was to begin by implementing simpler boosting algorithm to initially study its effect on some topology preserving algorithms. Bagging and AdaBoost are the two boosting algorithms for ensemble training applied in this work.

Bagging (or bootstrap aggregating) [4] is one of the simplest techniques for ensemble construction. Its aim is to improve classification and regression models in terms of stability and classification accuracy, also reducing variance and helping to avoid overfitting. It consists on training each of the classifiers composing the ensemble separately using a different subset of the main training dataset. This is accomplished by using re-sampling with replacement over the training set. The classification results are obtained by (weighted or majority) voting among its composing classifiers. The technique provides the ensemble with a balance between variability and similarity.

The idea of AdaBoost [5] is to train several different classifiers, one each stage, in slightly different datasets by re-sampling with replacement. The difference is that it is taken into accounts which of the training samples are not correctly classified by the

current classifier. When a sample is not well classified its probability is increased, so there are more chances that it will be presented to the next trained classifier as input. That way, the ensemble concentrates in the samples that are harder to classify, improving its learning capabilities. There have been proposed two slightly different versions of the algorithm [15]. AdaBoost.M1 is recommended for datasets with samples that only belong to two different classes while AdaBoost.M2 is recommended for dataset with more than two different classes. Although this meta-algorithm has been previously used for supervised classification [14], a short number of studies [16], [17], [19] involve unsupervised one such as in the present work.

3.2 Summarizing Some Applied Ensembles Techniques

The models used in this work are mainly designed as visualization tools. Constructing classical ensembles can be considered as a good option when trying to boost their classification capabilities, stabilizing its learning algorithm and avoiding overfitting; but when dealing with its visualization feature an ensemble is not directly displayable [18]. Representing all the networks in a simple image can only be useful when dealing with only 1-dimensional maps [17] but gets messy when visualizing 2-D maps. To overcome this problem some “ensemble combination” algorithms have been devised during this work in order to obtain a unique network that somewhat represents the information contained in the different networks composing the ensemble. Our objective is that this “combination” unites good classification accuracy with truthful representation of data for visual inspection. This part of the work has two approaches, which were inspired by SOM bagging [19] in one hand and by SOM fusion [20] on the other.

For all the tests involving this combination of networks the procedure is the same. A simple n-fold cross-validation is used in order to employ all data available for training and testing the model and having several executions to calculate an average of its performance. In each step of the cross-validation first, an ensemble of networks must be obtained. The way the ensemble is trained does not affect the way the combination is computed. In the case of this study this has been done using the bagging or the adaboost meta-algorithm. Then the computation of the combination is performed. Finally, both the ensemble and the combination generated from it are tested employing the test fold.

The different options studied for combining the network of the ensemble into a single network, summarizing its main characteristics are described in the following paragraphs. The first version is an existing one while the second and third, ‘Superposition’ and ‘Superposition + Re-labelling’, are novel techniques presented for the first time in this study.

Fusion: This method involves comparing the networks neuron by neuron in the input space [20]. This implies that all the networks in the ensemble must have the same size. First, it searches for the neurons that are closer in the input space (selecting only one neuron in each network of the ensemble), then it “fuses” them to obtain the final neuron in the “fused” map. This process is repeated until all the neurons have been fused. A more detailed description of this procedure can be found in [20]. Here the

labelling of the neurons of the fused network, employing again the training dataset, is done in order to get a clear visualization of the map.

Superposition: In order to obtain a visual illustration (in the form of a 2-dimensional map) of the information the whole ensemble contains, this procedure has been designed during the development of this work. It consists of “superposing” the maps formed by the networks composing the ensemble into a final map, on a neuron by neuron comparison (as is done in fusion). Note that the weights of each network in the ensemble are initialized in a way that makes the neurons in the same position of two (or more) networks comparables. A description of the process could be:

1. Selection of the neuron in the same position in all maps (1 neuron for each map).
2. Creation of a neuron in the same position in the final map. Its inter-neuronal weights are the average of the inter-neuronal weights of the neurons selected in 1. Its frequency in recognizing a class is the addition of the frequency of the neurons selected in 1 for each class recognized (This is used in the classification stage).
3. Labelling the neuron in that position according to its most frequently recognized class (This is used in the representation of the map).
4. Repeating 1-3 until all the neurons have been processed.

This way ensures that the resultant “summarized” or “superposed” map represents visually what the majority of the maps composing the ensemble represent in a neuron-by-neuron basis. When using the resultant “superposed” map for classification purposes it returns the class represented by the neuron that is activated when the new sample is presented to the network.

Superposition + Re-labelling: This method has two main phases. The first one is the superposition explained before. The second one consists of testing which class actually recognizes better each neuron after the superposition, instead of relying on the recognition of the neurons in the same position done previously in the individual testing of each of the ensemble networks. So, after the superposition phase, the same dataset used for training is presented to the resultant network of the superposition to check which class is more consistently recognized by each neuron. Usually less number of neurons responds to this re-labelling, giving as a result a more compact map.

4 Experiment Details

Several experiments have been performed to check the suitability of using the previously described boosting and combining techniques under the frame of the mentioned topology preserving models. The datasets selected, the very well known Iris dataset and Wisconsin Breast Cancer dataset, were obtained from the UCI repository [20]. Visualization results are displayed in Fig.1 and Fig.2.

Fig. 1 displays maps representing the iris dataset. The circles represents “iris-setosa”, the squares the “iris-virginica” and the triangles the “iris-versicolor”. As it can be seen, the dataset’s inner structure remains the same in the single map and in the three combinations having a clearly separable group (circles) and two non-linearly separable groups (triangles and squares).

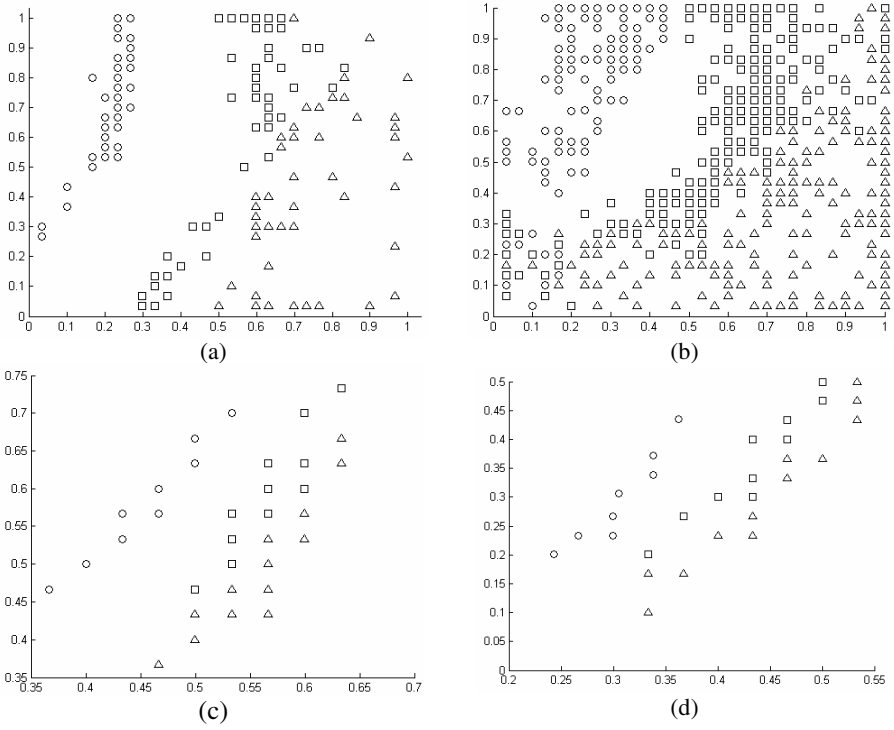


Fig. 1. Iris dataset represented by a ViSOM of 20x20 neurons. Fig 1(a) is obtained from a single ViSOM. Fig 1(b) is obtained from the ‘superposition’ of an ensemble of 10 maps with the same characteristics as Fig 1(a). Fig (c) and (d) are respectively the ‘superposition+re-labelling’ and the ‘fusion’ of the same ensemble.

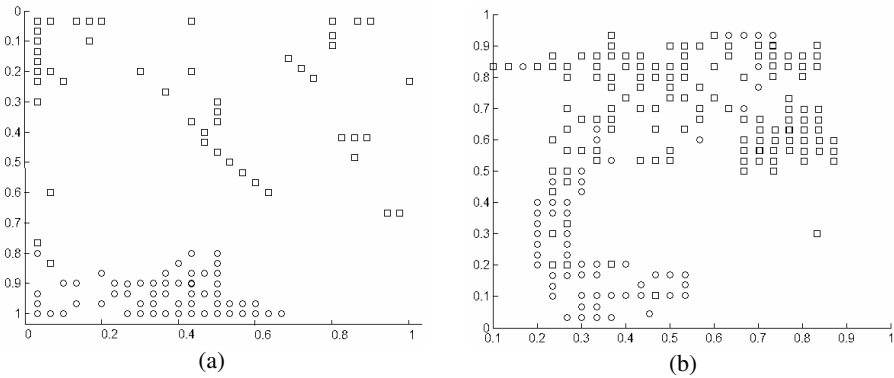


Fig. 2. Wisconsin Breast Cancer dataset represented by a ViSOM of 30x30 neurons. Fig 2(a) is obtained from a single ViSOM. Fig 2(b) is obtained from the ‘superposition’ of an ensemble of 10 maps with the same characteristics as Fig 2(a). Fig 2(c) and Fig 2(d) are respectively the ‘superposition+re-labelling’ and the ‘fusion’ of the same ensemble as Fig 2(b).

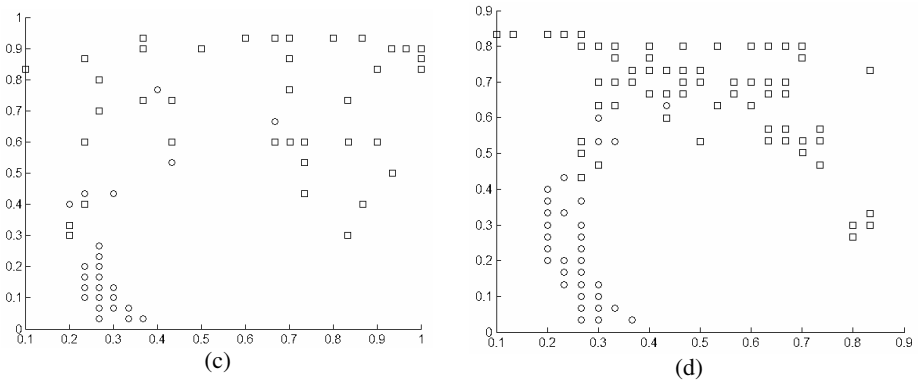


Fig. 2. (continued)

Fig. 2 displays maps representing the Wisconsin breast cancer dataset. The circles represent the “benign” cases, while the squares represent the “malignant” cases. As it can be seen a very condensed cloud of points corresponding to the “benign” cases appears at the bottom of all images, while a much more sparse cloud representing the “malignant” cases occupies the rest of the map.

As it can be seen, the visualization obtained by the combination methods (Fig. 1 b,c,d, Fig.2 b,c,d) keeps the internal structure of the dataset, being able to reproduce that structure with minor variations. The advantage of using an ensemble (by bagging or boosting) is the added stability to the analysis. This means that when dealing with few and scattered data points disposed over the input space the ensemble of maps should be able to reconstruct the structure of the dataset by combining the same “areas” of the different maps that compose the ensemble. If there is an area where no neurons (or very few) were activated in one map, but the neurons in the same position were activated in another map (because those maps were trained in slightly different dataset), there will not be a blank space (or very scattered disposition of neurons) in the resultant combination. This can be easily seen in the “superposition” combination where each neuron represents the data that activated that same neuron in all the maps of the ensemble. The “superposition + re-labelling” makes possible to obtain a map with the same main disposition but much reduced in complexity, as fewer neurons are now represented. This can serve to eliminate some distortions that can appear when are simply superposing all the maps in the ensemble, at cost of lower definition in the representation. This is more obvious in the cancer dataset in Fig 2. The “fusion” combination obtains very similar results to this second technique, as it follows the same combination and then re-labelling idea. The only difference comes by the criteria used to combine the neurons of each map. The classification accuracy obtained in the experiments is shown in Tables 1 – 4.

Table 1. Percentage of correct recognition of samples of the **Iris** dataset using ensembles of 10 networks employing a 10-fold cross validation for the tests. All ensembles were trained using the **bagging** meta-algorithm.

<i>Model</i>	Best Single Netwk.	Ensemble (weighted voting)	Superp.	Superp. + ReLabelling	Fusion
<i>SOM (5x5)</i>	94%	97.33%	84%	94%	94.6%
<i>SOM (10x10)</i>	73.3%	97.3%	80.6%	82%	83.3%
<i>SOM (20x20)</i>	48.6%	90.6%	66.6%	69.3%	58%
<i>ViSOM (10x10)</i>	93.3%	92.6%	61.3%	87.3%	94%
<i>ViSOM (20x20)</i>	87.3%	97.3%	80.6%	89.3%	94%
<i>ViSOM (30x30)</i>	75.3%	94.6%	80.6%	82.6%	85.3%
<i>ML-SIM (20x20)</i>	75.3%	76%	45%	74%	76.6%

Table 2. Percentage of correct recognition of samples of the **Iris** dataset using ensembles of 10 networks employing a 10-fold cross validation for the tests. All ensembles were trained using the **AdaBoost.M2** meta-algorithm.

<i>Model</i>	Best Single Netwk.	Ensemble (weighted voting)	Superp.	Superp. + ReLabelling	Fusion
<i>SOM (5x5)</i>	95.3%	97.3%	77.3%	90.6%	96.0%
<i>SOM (10x10)</i>	75.3%	95.3%	80.6%	83.3%	83.3%
<i>SOM (20x20)</i>	58%	90%	71.3%	71.3%	61.3%
<i>ViSOM (10x10)</i>	92%	94.6%	79.3%	97.3%	90.6
<i>ViSOM (20x20)</i>	86.6%	94.6%	86.6%	88.6%	88.6%
<i>ML-SIM (20x20)</i>	75.3%	77.3%	44%	79.3%	78.6%

Table 3. Percentage of correct recognition of samples of the **Wisconsin Breast Cancer** dataset using ensembles of 10 networks employing a 10-fold cross validation for the tests. All ensembles were trained using the **bagging** meta-algorithm.

<i>Model</i>	Best Single Netwk.	Ensemble (weighted voting)	Superp.	Superp. + ReLabelling	Fusion
<i>SOM (5x5)</i>	97%	97%	89.4%	96.1%	96.2%
<i>SOM (10x10)</i>	93%	96%	93.4%	93.3%	94.7%
<i>SOM (20x20)</i>	77.6%	96%	83.4%	90%	82.4%
<i>ViSOM (10x10)</i>	94.1%	97%	96.1%	93.6%	94.4%
<i>ViSOM (20x20)</i>	80.9%	95.3%	92.3%	83%	84%
<i>ViSOM (30x30)</i>	77.3%	93.3%	91.7%	78.3%	79.6%
<i>ML-SIM (20x20)</i>	73.9%	94.5%	70.1%	85.7%	89.2%

Table 4. Percentage of correct recognition of samples of the **Wisconsin Breast Cancer** dataset using ensembles of 10 networks employing a 10-fold cross validation for the tests. All ensembles were trained using the **AdaBoost.M1** meta-algorithm.

<i>Model</i>	Best Single Netwk.	Ensemble (weighted voting)	Superp.	Superp. + ReLabelling	Fusion
<i>SOM (5x5)</i>	96%	95.8%	93.8%	96%	96.7%
<i>SOM (10x10)</i>	91.6%	96.4%	94.3%	92.8%	94.5%
<i>SOM (20x20)</i>	79.6%	96.6%	94.4%	86.1%	89.3%
<i>ViSOM (10x10)</i>	85.4%	96.9%	93.8%	94%	93.6%
<i>ViSOM (20x20)</i>	84.9%	96.3%	95.1%	86.8%	87.7%
<i>ViSOM (30x30)</i>	77.9%	95.1%	93.3%	83.8%	84.5%
<i>ML-SIM (20x20)</i>	73%	95.4%	66.1%	87.2%	87.5%

There are several aspects worth of noting in relation with these experiments and the results presented from Table 1 to Table 4. The most obvious one is that the best model, both for visualization and for classification is the ViSOM. The second best model is the SOM and the last one the ML-SIM. This was expected, taking into account that the ViSOM was specially developed for data visualization while ML-SIM was specially designed for working with radial-based datasets. The second conclusion is that the amount of neurons composing the map is directly proportional to the clarity and definition of the map, but inversely proportional to the classification accuracy of the network.

Talking about the ensemble algorithms, it can be concluded that the visualization capabilities of the models are slightly enhanced when using the simple superposition technique, as more neurons are represented in only one map, although this can sometimes make the data structures in the map a little less defined. On the contrary, the classification side of the models is really benefited from the ensemble inclusion. Even when the ensemble is constructed using networks of low classification accuracy the ensemble is able to obtain a greater accuracy. This was expected from all the previous work done in ensemble and classification. The combinations of the ensemble networks into a unique one are far more dependant of the accuracy of the networks employed to create them but still they manage to get a classification accuracy that is most of the times equal or superior to the accuracy of the better network of the ensemble, specially the superposition + re-labeling. This can be taken into account when a unique, but more stable network is needed; both for classification and visualization.

5 Conclusions and Future work

This study presents an interesting mixture of techniques for representation of multi-dimensional data in 2-D maps. They are based in the combination of several maps trained over slightly different datasets to form an ensemble of networks with self-organizing capabilities. The training of the ensemble of networks has been tested by using the bagging and boosting techniques for their comparison. The purpose of this combination of maps is to avoid the overfitting and to improve the stability of previously developed models based in unsupervised and competitive learning. They can be especially useful when a reduced dataset is available. As an ensemble of maps is impossible to represent, several different combination algorithms are proposed and tested. They have been applied for the first time in the case of the ViSOM and compared with the SOM and the ML-SIM performance. A novel ensemble combination technique has also been presented, and tested in a very successfully way. As an added effect, the classification capabilities of the models are also increased. These techniques have been tested in two widely known real datasets. Future work will include far exhaustive testing of the presented combination of techniques using several more complex datasets, as well as adapting the present model to other novel boosting meta-algorithms to check if more improvements can be obtained.

Acknowledgments

This research has been supported by the MCyT project TIN2004-07033.

References

1. Kohonen, T.: The Self-Organizing Map. *Neurocomputing* 21, 1–6 (1998)
2. Yin, H.: Data Visualisation and Manifold Mapping Using the Visom. *Neural Networks* 15, 1005–1016 (2002)
3. Corchado, E., Fyfe, C.: The Scale Invariant Map and Maximum Likelihood Hebbian Learning. In: *International Conference on Knowledge-Based & Intelligent Information & Engineering System (KES)* (2002)
4. Breiman, L.: Bagging Predictors. *Machine Learning* 24, 123–140 (1996)
5. Freund, Y., Schapire, R.E.: A Decision-Theoretic Generalization of on-Line Learning and an Application to Boosting. *Journal of Computer and System Sciences* 55, 119–139 (1997)
6. Kohonen, T., Lehtio, P., Rovamo, J., Hyvarinen, J., Bry, K., Vainio, L.: A Principle of Neural Associative Memory. *Neuroscience* 2, 1065–1076 (1977)
7. Yin, H.: Visom - a Novel Method for Multivariate Data Projection and Structure Visualization. *Neural Networks, IEEE Transactions on* 13, 237–243 (2002)
8. Corchado, E., MacDonald, D., Fyfe, C.: Maximum and Minimum Likelihood Hebbian Learning for Exploratory Projection Pursuit. *Data Mining and Knowledge Discovery* 8, 203–225 (2004)
9. Fyfe, C., Corchado, E.: Maximum Likelihood Hebbian Rules. In: *European Symposium on Artificial Neural Networks (ESANN)* (2002)
10. Kraaijeveld, M.A., Mao, J., Jain, A.K.: A Nonlinear Projection Method Based on Kohonen's Topology Preserving Maps. *Neural Networks, IEEE Transactions on* 6, 548–559 (1995)
11. Heskes, T.: Balancing between Bagging and Bumping. In: *Advances in Neural Information Processing Systems 9 - Proceedings of the 1996 Conference* vol. 9, pp. 466–472 (1997)
12. Demiriz, A., Bennett, K.P., Shawe-Taylor, J.: Linear Programming Boosting via Column Generation. *Machine Learning* 46(1-3), 225–254 (2002)
13. Schwenk, H., Bengio, Y.: Boosting Neural Networks. *Neural Computation* 12, 1869–1887 (2000)
14. Freund, Y., Schapire, R.E.: Experiments with a New Boosting Algorithm. In: *International Conference on Machine Learning*, pp.148–156 (1996)
15. Gabrys, B., Baroque, B., Corchado, E.: Outlier Resistant PCA Ensembles. In: Gabrys, B., Howlett, R.J., Jain, L.C. (eds.) *KES 2006. LNCS (LNAI)*, vol. 4253, pp. 432–440. Springer, Heidelberg (2006)
16. Corchado, E., Baroque, B., Gabrys, B.: Maximum Likelihood Topology Preserving Ensembles. In: Corchado, E., Yin, H., Botti, V., Fyfe, C. (eds.) *IDEAL 2006. LNCS*, vol. 4224, pp. 1434–1442. Springer, Heidelberg (2006)
17. Kaski, S.: Data Exploration Using Self-Organizing Maps. Department of Computer Science and Engineering, Helsinki University of Technology. Espoo, Finland (1997)
18. Petrakieva, L., Fyfe, C.: Bagging and Bumping Self Organising Maps. *Computing and Information Systems* (2003)
19. Georgakakis, A., Li, H., Gordan, M.: An Ensemble of SOM Networks for Document Organization and Retrieval. In: *Int. Conf. on Adaptive Knowledge Representation and Reasoning (AKRR'05)*, p. 6 (2005)
20. Newman, D.J., Hettich, S., Blake, C.L., Merz, C.J.: UCI Repository of Machine Learning Databases. University of California, Irvine, Dept. of Information and Computer Sciences (1998), <http://www.ics.uci.edu/~mllearn/MLRepository.html>