ViSOM Ensembles for Visualization and Classification

Bruno Baruque, Emilio Corchado, and Hujun Yin

Department of Civil Engineering. University of Burgos, Spain
bbaruque@ubu.es, escorchado@ubu.es
School of Electrical and Electronic Engineering. University of Manchester, UK
h.yin@manchester.ac.uk

Abstract. In this paper ensemble techniques have been applied in the frame of topology preserving mappings in two applications: classification and visualization. These techniques are applied for the first time to the ViSOM and their performance is compared with ensemble combination of some other topology preserving mapping such as the SOM or the MLSIM. Several methods to obtain a meaningful combination of the components of an ensemble are presented and tested together with the existing ones in order to identify the best performing method in the applications of these models.

1 Introduction

Topology Preserving Maps [1], were originally created as a visualization tool; enabling the representation of high-dimensional datasets onto two-dimensional maps and facilitating the human expert the interpretation of data. Almost effortlessly this family of algorithms can be modified to serve as data classifiers, exploiting its inner pattern recognition capabilities [2]. This added characteristic can even serve as a certain measure of the stability of the trained network and accuracy of the mapping.

A general way of boosting the classification capabilities of classic classifiers (such as decision trees) is the construction of ensembles of classifiers [3], [4]. Following the idea of a 'committee of experts', the ensemble technique consists of training several identical classifiers on slightly different datasets in order to constitute a 'committee' to classify new instances of data.

This paper presents an application of the ensemble technique on several topology preserving models to improve their classification capabilities as well as their visualization performance.

2 Topology Preserving Mapping

This name comprises a family of techniques with a common target: to produce a low dimensional representation of the training samples while preserving the topological properties of the input space. The best known technique among them is the Self-Organizing Map (SOM) algorithm [5] [6]. It 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 [1].

F. Sandoval et al. (Eds.): IWANN 2007, LNCS 4507, pp. 235–243, 2007.

One interesting extension of this algorithm is the Visualization Induced SOM (ViSOM) [7], [8] 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.

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:

$$w_k(t+1) = w_k(t) + \alpha(t)\eta(v,k,t)\big(x(t) - w_v(t)\big) \tag{1}$$

Update of neighbourhood neurons in ViSOM:

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

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 (MLSIM) [9], [10]. It is similar to the SOM [5] but in this case training is based on the use of a particular Exploratory Projection Pursuit (EPP) model called Maximum Likelihood Hebbian Learning (MLHL) Network [12], [13]. 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,

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

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

These three models can be adapted for classification of new samples using a semi-supervised procedure. Once the network training is completed, the same dataset used in the training stage is presented once again to the network, but this time keeping record of how many times each neuron responds to each of the classes of the dataset. That way, each neuron of the network is labelled with the class it has recognized more consistently. When a new sample is presented to the network, it is classified with the class associated to the neuron that won the competition at that time. 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 [14].

3 Classifiers Combination

The ultimate goal for designing pattern recognition systems is to achieve the best possible classification performance for the task at hand. It has been observed in several studies that although one of the classifiers in an ensemble would yield the best performance, 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 [15]. Competitive learning based networks are inherently instable, due to the nature of the statistical learning algorithms. The effect of this instability may, however, be minimized again by using an ensemble with a voting results scheme [16].

The algorithms to combine classifiers can be divided into two broad classes. The simpler variety of algorithms merely combines, by averaging in some way, the results each of the composing classifiers of the ensemble yields into a final result. More complex types of algorithms try to combine not only the results but the whole set of classifiers in order to construct a single better one that should outperform its individual components. In the case of this paper both of the two approaches have been considered, with comparison purposes.

The second type of classifier combination was originally considered to be applied to classical classification trees. Its main advantage is that it combines the improvement on the classification quality with the simplicity of the handling of only one classifier. In the case of the present work, the emphasis has been equally put in the classification accuracy of the model and the visualization capabilities of it. In this second perspective, the concept of a single "summary" or "synthesis" of the patterns stored within the whole ensemble is essential, to which end different approaches have been applied and evaluated in this work. Our overriding aim is to obtain a unique map that may be seen to represent all of the features of the maps in the ensemble.

3.1 Proposed Combinations

The main thrust of this work is the construction of ensembles of neural topology preserving maps in order to boost their visualization and classification performance. A number of ensemble techniques are applied to the ViSOM and other topological mapping networks such as the SOM [17] or MLSIM [18].

The application of those combination methods in the context of the topology preserving networks classification capability seems a straightforward implementation of the previously described procedures. On the contrary, in the context of the visualization some adaptations are necessary to build a meaningful combination of the maps they represent. The part of the work developed to deal with this issue has two perspectives, which were inspired by SOM bagging [17] in one hand and by SOM fusion [19] on the other.

The procedure is the same for the training of the networks that compose the ensembles. All are trained using typical cross-validation, with the dataset divided into several folders, leaving one of them out to test the classification accuracy. The method obtains *n* subsets of the training dataset through re-sampling with replacement and trains individual classifiers on such re-sampled subsets. This permits to generate *n* trained networks which are used to classify as an ensemble (using bagging) or combined into a final network. In order to obtain the similarity needed to make networks comparable to combine them, the networks of the same ensemble are initialized to the inter-neural weights obtained in the training of the previous network; while is trained on a slightly different portion of the training folds of the dataset to ensure the balance between similarity and diversity needed for an ensemble to perform correctly. The combination of maps is done once all the networks composing the ensemble have finished their training. The different options studied for this are described in the following paragraphs, where the last two new methods have been developed in this work.

Bagging: This combination procedure was originally devised to increment the classification accuracy of tree-based classifiers [20]. This is the reason why it works well for the classification results of the networks, but it has some problems when trying to obtain a visualization "summary" of the ensemble, as it is pointed out in [17]. The aim of this procedure is to reduce the classification error; especially when the number of samples is low in relation with the space of hypothesis. In this paper it is only used when comparing classification accuracy. The whole ensemble is trained in a way that all the resultant maps are comparable, which means that neurons located in a region of a map will be similar to neurons located in the same region of a different map. This is done by initializing the networks in the same way. The option for the voting decision used in this work is the weighted variant. The vote of a network that has consistently recognized a certain class will have more weight than others.

Fusion: This method involves comparing the networks neuron by neuron in the input space. 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. To deal with the high computational complexity of the algorithm it was implemented using dynamic programming. A more detailed description of this procedure can be found in [19]. Here the labelling of the neurons of the fused network, employing again the training dataset, is done in order to obtain 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 is a combination of the two previously described techniques. 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 (in the same way as bagging). 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 is the superposition explained before. The second 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, 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 little number of neurons respond to this re-labelling, giving as a result a more compact map.

4 Experiments and Results

To test these ensemble techniques the well-known iris dataset was used. For comparison purposes, all the ensemble algorithms have been tested employing the SOM and MLSIM models as well as with the ViSOM model for the first time.

The first experiment was performed using an ensemble of SOMs at first with a size of 10x10 and then by increasing the number of neurons to 20x20. The next series of experiments repeated those described above, but this time employing an ensemble of ViSOMs. Initially a size of (20x20) is used, and then it is increased to (30x30); as

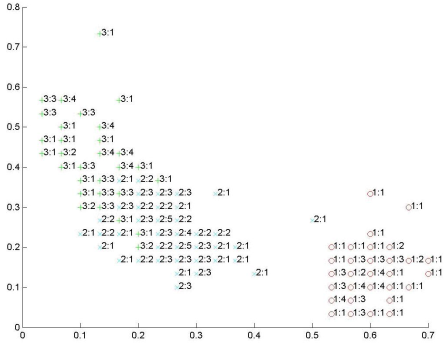

Fig. 1. Visualization of the ViSOM (30x30) ensemble using the ‘**superposition**’ method on the iris dataset

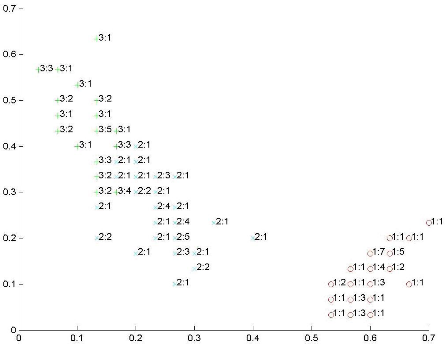

Fig. 2. Visualization of the ViSOM (30x30) ensemble using the ‘**superposition + re-labelling**’ method on the iris dataset

presented in Fig. 1 and Fig. 2. In all figures, in the code number associated to each neuron; the first number identifies the class recognized by it and the second indicates the number of times that neuron has responded to that class.

For comparative purposes the same experiment was performed by employing an ensemble of MLSIMs. As the dataset is not disposed in a radial way, the results are poorer than the first two experiments.

The classifying accuracies of the single best classifying network of the ensemble, the whole ensemble by combination of its outputs (weighted voting), and the three mentioned summaries are showed in Table 1. The experiment was conducted by using a 5-fold cross validation method.

There are some observations worth noting as result of these experiments. The first, the greater the number of neurons on the map, the more defined image of the distribution of data can be obtained, for the SOM, ViSOM and even MLSIM if the neighbouring function is set properly. On the contrary, if attention is paid to the classification accuracy; it can be observed that the greater the number of neurons is used, the more this accuracy descends. This is even more the case in the SOM than in the ViSOM (from a 92% when using only 10x10 neurons to 81% when using 20x20 in the ensemble of SOMs and from a 92% when using 20x20 neurons to 83% when using 30x30 in the ensemble of ViSOM). This may be due to the fact that using a square matrix of 10x10 (100 neurons) is more than enough to classify a dataset of 100 samples.

Table 1. Accuracy in classification of the different models obtained from a SOM, ViSOM and MLSIM ensemble on the iris dataset. The result of the table is the average of the five tests in the cross validation (one with a different fold as testing set).

Type of model	Best Single Netwk.	Ensemble	Superposition	Superp.+ Re-Labelling	Fusion
SOM ensemble (10x10)	78%	94%	92%	73%	75%
SOM ensemble (20x20)	50%	81%	62%	59%	58%
ViSOM ensemble (20x20)	82%	92%	78%	74%	77%
ViSOM ensemble (30x30)	74%	83%	82%	70%	71%
MLSIM ensemble (30x30)	74%	80%	24%	71%	78%

It is interesting to note that the ensemble of ViSOMs seems more stable than the ensembles of SOM; as the latter deteriorates with the increase in the number of neurons form 10x10 to 20x20, while the former does only slightly from 20x20 to 30x30. As might be expected, MLSIM, which responds better to radial-based datasets, obtained the worst results.

The same experiments conducted for the iris dataset were repeated with another well known dataset: the Wisconsin Breast Cancer dataset (UCI Repository) [21]. The results obtained in the visualization part are displayed in Fig. 3 and Fig 4. The classification results are presented in Table 2.

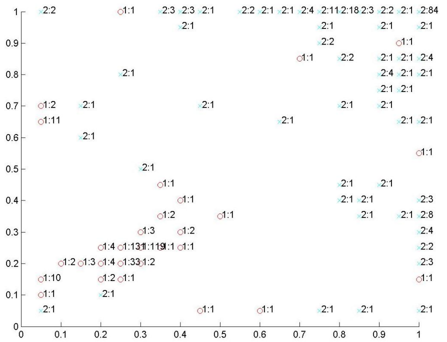

Fig. 3. Visualization of the SOM (20x20) ensemble using the **'superposition+re-labelling'** method on the cancer dataset

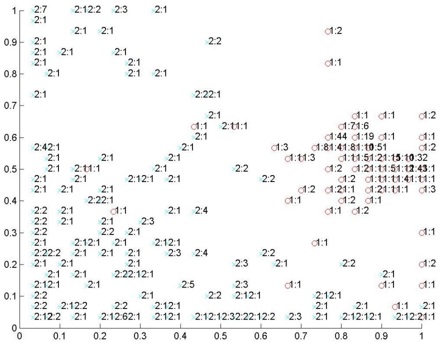

Fig. 4. Visualization of the ViSOM (30x30) ensemble using the '**superposition+re-labelling**' method on the cancer dataset

Table 2. Accuracy in classification of the different models obtained from a SOM, ViSOM and MLSIM ensemble. The results are the average of the five tests in the cross validation (one with a different fold as testing set).

Type of model	Best Single Netwk.	Ensemble	Superposition	Superp. + Re-Labelling	Fusion
SOM ensemble (10x10)	92%	96%	95%	93%	93%
SOM ensemble (20x20)	77%	95%	74%	87%	85%
SOM ensemble (30x30)	69%	92%	70%	82%	76%
ViSOM ensemble (20x20)	94%	96%	96%	94%	94%
ViSOM ensemble (30x30)	94%	97%	94%	95%	95%
ViSOM ensemble (40x40)	91%	96%	94%	92%	92%
MLSIM ensemble (30x30)	79%	94%	56%	78%	84%

Inspecting the results in Table 2, a similar behaviour as in the first series of experiments can be seen, especially for the SOM model: when the number of neuron increases, the representation capacity increases too but the classification accuracy decreases. This time the ViSOM seems to be quite more stable, obtaining results even slightly better when increasing from 20x20 networks to 30x30 networks in the ensemble.

5 Conclusions and Future Work

As it can be seen in the experiments section the ensemble always improves in the classification perspective, as in each test it clearly outperforms the single models or combinations. For 2-D visualization purposes however, the ensemble is not directly displayable. This is why all the different combination procedures were tested. The best performing technique turns out to be the "superposition" which can be considered the most similar to the ensemble philosophy. This is because it combines the neurons of the ensemble into the “superposed” map by a particular weighted voting between the counterpart neurons in all the ensembles. In the re-labelling of the superposed map less neurons respond to the training data, as those neurons that were previously disposed on different networks and were able to win the competition separately (cooperating in the voting process), now compete in the same network. As stated before, when fewer neurons respond to the inputs, a worse visualization is obtained; but as a counterpart better classification accuracy is achieved. The fact that the maps are initialized in a way that same neurons (or very closer ones) in different networks win when presented similar inputs helps this technique (with its two variants) to outperform the “fusion” which fuses neurons regardless their position in the map.

In this work some novel ensemble combination techniques have been presented. The ensembles were applied to the ViSOM and yield good results not only in the visualization of multivariate data, but also in classification. The ensemble methods were also applied to other topology preserving models such as the SOM or MLSIM in order to draw comparisons, proving that these models can benefit these ensemble methods as well.

Future work will focus on application of these techniques to several real datasets to measure their performance in real world problems.

Acknowledgments

This research has been supported by the MCyT project TIN2004-07033 and the project BU008B05 of the JCyL.

References

1. Kohonen, T., Lehtio, P., Rovamo, J., Hyvarinen, J., Bry, K., Vainio, L.: A Principle of Neural Associative Memory. Neuroscience 2, 1065–1076 (1977)
2. Vesanto, J.: Data Mining Techniques Based on the Self-Organizing Map. Engineering Physics and Mathematics. Helsinki University of Technology. Espoo, Finland. (1997)
3. Kuncheva, L. I.: Combining Pattern Classifiers: Methods and Algorithms (2004) ISBN: 0471210781
4. Ron, M., Gunnar, R.: An Introduction to Boosting and Leveraging. Advanced Lectures on Machine Learning: Machine Learning Summer School 2002, Canberra, Australia, February 11-22, 2002. Revised Lectures, pp. 118-183 (2003)
5. Kohonen, T.: The Self-Organizing Map. Neurocomputing 21, 1–6 (1998)
6. Kaski, S.: Data Exploration Using Self-Organizing Maps. Department of Computer Science and Engineering. Helsinki University of Technology. Espoo, Finland (1997)
7. Yin, H.: Data Visualisation and Manifold Mapping Using the Visom. Neural Networks 15, 1005–1016 (2002)
8. Yin, H.: Visom - a Novel Method for Multivariate Data Projection and Structure Visualization. Neural Networks, IEEE Transactions 13, 237–243 (2002)
9. Fyfe, C.: A Scale-Invariant Feature Map. Network: Computation in Neural Systems 7, 269–275 (1996)

10. Corchado, E., Fyfe, C.: The Scale Invariant Map and Maximum Likelihood Hebbian Learning. International Conference on Knowledge-Based and Intelligent Information and Engineering System (2002)
11. Fyfe, C., Corchado, E.: Maximum Likelihood Hebbian Rules. European Symposium on Artificial Neural Networks (ESANN) (2002)
12. 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)
13. Kraaijveld, M.A., Mao, J., Jain, A.K.: A Nonlinear Projection Method Based on Kohonen's Topology Preserving Maps. Neural Networks, IEEE Transactions 6, 548–559 (1995)
14. Heskes, T.: Balancing between Bagging and Bumping. Advances in Neural Information Processing Systems 9. In: Proceedings of the 1996 Conference 9, 466–472 (1997)
15. Ruta, D., Gabrys, B.: A Theoretical Analysis of the Limits of Majority Voting Errors for Multiple Classifier Systems. Pattern Analysis and Applications 5, 333–350 (2002)
16. Petrakieva, L., Fyfe, C.: Bagging and Bumping Self Organising Maps. Computing and Information Systems (2003)
17. Corchado, E., Baruque, B., Gabrys, B.: Maximum Likelihood Topology Preserving Ensembles. In: Proceedings of the Intelligent Data Engineering and Automated Learning - Ideal 2006, Vol. 4224, pp. 1434-1442 (2006)
18. Georgakis, A., Li, H., Gordan, M.: An Ensemble of Som Networks for Document Organization and Retrieval. Int. Conf. on Adaptive Knowledge Representation and Reasoning (AKRR'05) p. 6 (2005)
19. Breiman, L.: Bagging Predictors. Machine Learning 24, 123–140 (1996)
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/~mlearn/MLRepository.html