Quality of Adaptation of Fusion ViSOM

Bruno Baruque[1], Emilio Corchado[1], Hujun Yin[2]

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

Abstract. This work presents a research on the performance capabilities of an extension of the ViSOM (Visualization Induced SOM) algorithm by the use of the ensemble meta-algorithm and a later fusion process. This main fusion process has two different variants, considering two different criteria for the similarity of nodes. These criteria are Euclidean distance and similarity on Voronoi polygons. The capabilities, strengths and weakness of the different variants of the model are discussed and compared more deeply in the present work. The details of several experiments performed over different datasets applying the variants of the fusion to the ViSOM algorithm along with same variants of fusion with the SOM are included for this purpose.

1. Introduction

A general way of boosting the stability and classification capabilities of classic classifiers (such as decision trees) is the construction of ensembles of classifiers [4], [5]. 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.

Topology Preserving Maps [1], which include the Self-Organizing Maps (SOM) [2] and the Visualization Induced SOM (ViSOM) [3], 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. The main problem of these unsupervised techniques is their inherent instability. Even running the same algorithm using the same parameters and dataset can yield quite dissimilar results.

The ensemble meta-algorithm approach can be used on several topology preserving models to improve their stability and visualization performance. This is done by the training of several complementary networks and computing a fusion of them that outperforms each of its composing individually. This work main objective of this research is to present a study of the characteristics and performance of two different variants of the fusion process.

The rest of the paper is organized as follows: In section 2 the basics of the Self-Organizing Maps and its extension the Visualization Induced SOM, along with some quality measures for this kind of maps are presented. Section 3 is dedicated to the explanation of the ensemble training and the latter fusion process (which includes two variants). Section 4 includes the details of several experiments performed using several real datasets over the two different variants, with comparison of the strengths and weakness of each one. Finally, section 5 contains the conclusions and directions of future work extracted form the present work.

2. Quality Measures for Topology Preserving Models

2.1 The ViSOM learning algorithm

In this study, two different models will be applied: the Self-Organizing Map (SOM) and its variant the Visualization Induced SOM (ViSOM). Both the SOM and the ViSOM models belong to 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 is the Self-Organizing Map algorithm [2]. 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].

An interesting extension of this algorithm is the Visualization Induced SOM [3], [6] 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 Eq. (1) and Eq. (2).

Update of neighbourhood neurons in SOM:

$$w_k(t+1) = w_k(t) + \alpha(t)\eta(v,k,t)\left(x(t) - w_v(t)\right) \quad \mathbf{(1)}$$

Update of neighbourhood neurons in ViSOM:

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

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.

2.2. Quality Measures

To evaluate the quality of the adaptation of the map to the dataset represented several different measures have been devised [7]. A very widely known measure, used to indicate how good the units (neurons) of the map approximate the data on the dataset, is the mean square quantization error (MSQE). It is widely used as a measure of the quality of vector quantization algorithms, but it can be easily adopted for SOM and ViSOM algorithms as represented in Eq. 3:

$$MSQE = \frac{1}{|D|}\sum_{x_i \in D}\left\| x_i - m_{b(x_i)} \right\|^2 \qquad \mathbf{(3)}$$

where $|D|$ is the number of data in the dataset D, and $m_{b(x_i)}$ is the best matching unit of the map to the data sample x_i of the dataset.

The other main characteristic of the Self-Organizing Maps is their topology preservation. As explained in [8], when using a constant radius for the neighbourhood function of the learning phase of a SOM; there exists a function that the algorithm optimizes. This function, called distortion measure in this work, can be used to measure the overall topology preservation of a map. It is computed as shown in Eq. 4,

$$E_d = \sum_{i=1}^{n}\sum_{j=1}^{m} h_{b_i j}\left\| x_i - m_j \right\|^2 \qquad \mathbf{(4)}$$

where $h_{b_i j}$ represents the neighbourhood function between the best matching unit and every other unit in the map.

Also the topology error measure will be mentioned in this work. It is one of the first and simplest topology measures. It consists on finding the first two best matching units for each entry of the dataset and testing whether the second is in the neighbourhood of the first or not. This can be computed as a normalized single value, indicating the overall quality of the map or decomposed to be visualized over each neuron of the map [7].

3. Topology Preserving Mapping Fusion

3.1 Use of the Ensemble Meta-Algorithm

This technique was in its conception aimed at improving the performance of classification algorithms. 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 [4]. The main problem of competitive learning based networks is that are inherently unstable due to the nature of their learning algorithm. The leading idea of this work is that the effect of this instability may, however, be minimized by the use of ensembles [9]. The learning algorithm of the topology preserving maps family specifies that their composing units (or neurons) specialize during the algorithm iterations in recognizing a certain type of patterns, which determines also the topology of the map. Similarly to classification process, we can infer that the regions of the maps that do not represent trustfully the real nature of the dataset do not necessarily overlap. Therefore, the visualization of a single map might be improved by adapting each of the composing units of a map in the best way possible to the dataset under study by using ensemble techniques, as they offer complementary visualizations of the maps.

Among the ensemble algorithms, the most complex types try to combine not only the results but the whole set of classifiers in order to construct a single better one that can outperform its individual components. In the case of this paper this perspective, the concept of a single "summary" or "synthesis" of the patterns stored within the whole ensemble, is the one followed to improve the model performance. The main objective is to obtain a unique map that may be seen to represent in the clearest and most reliable way as possible the different features contained in the different maps in the ensemble.

3.2 Fusion variants under study

Several of ensemble techniques have been applied to the SOM [10], ViSOM [11] and other topological mapping networks, mainly for classification purposes.

Under the context of the visualization however, some adaptations are necessary to build a meaningful combination of the maps they represent. In this work a main algorithm for mapping fusion with two different variants is used for the first time in combination with the ViSOM. The objective is the comparison of the two in order to obtain conclusions that can be used in further studies to generate a more accurate model.

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. To train the network ensemble the meta-algorithms called bagging [12] is used. It consists on obtaining *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* different trained networks which are combined into a final network that is expected to outperform each of them individually. The combination of maps is done once all the networks composing the ensemble have finished their training.

This combination is done in a neuron by neuron basis. That is, neurons that are considered 'near enough' one to the other are fused to obtain a neuron in the final fused network. This is done by calculating the centroid of the weights of the neurons to fuse:

$$w_{neuAvg} = 1/n \cdot \sum_{i=1}^{n} w(i) \tag{5}$$

That process is repeated until all neurons in all trained networks are fused into a unique final one. The criteria to determine which neurons are 'near enough' to be fused is what determines the two variants of the main algorithm.

Criterion 1: *Voronoi Polygons***.** Each neuron in a Self-Organizing Map can be associated with a portion of the input data space called the Voronoi polygon. That portion of the input multi-dimensional space is the portion that contains data for which that precise neuron is the Best Matching Unit (BMU) of the whole network [1]. It is therefore a logical conclusion to consider that neurons that are related to similar Voronoi polygons can be considered similar between them, as they should be situated relatively close in the input data space.

To calculate the dissimilarity between two neurons, a record of which data entries activated each neuron as the BMU can be kept. This can be done easily associating a binary vector to the neuron which length is the size of the dataset and contains ones in the positions where the neuron was the BMU for that sample and zeros in the rest of positions. The dissimilarity (i.e. the distance) between neurons can therefore be calculated as:

$$d(v_r, v_q) = \frac{\sum_{l=1}^{n} XOR(v_r(l), v_q(l))}{\sum_{j=1}^{n} OR(v_r(j), v_q(j))} \tag{6}$$

being r and q the neurons to determine their dissimilarity and v_r and v_q the binary vectors relating each of the neurons with the samples recognized by it. A much more detailed explanation can be found in [13].

The main problem with this proximity criterion is that it depends on the recognition of data by the network, rather than on the network definition itself. This means that a neuron that does not react as the BMU for any data could be considered similar to another neuron with the same characteristic, although they can be relatively far from each other in the input data space. To avoid this, all neurons with a reacting rate lower than a threshold are removed before calculating the similarities between them. This implies that the neighbouring properties of the whole network are no longer considered. To keep a notion of neighbouring between the neurons of the fused network the similarity criteria must be used again. Neurons with dissimilarity less than a threshold will be considered as neighbours in the fused network.

Algorithm 1. *Fusion based on Similarity of the Voronoi Polygons*

```
1:  Being nNet the number of networks to be fused
     fus the resultant fused network
     $\theta$u, $\theta_f$ and $\theta_c$ the usage, fusion and connection thresholds respectively
2: for i=1:nNet
3:   remove from the network the neurons that have a recognition rate lower than a usage
     threshold ( $\Sigma_i v_r(i) < \theta_u$ )
4:   add all the rest of the neurons of network(i) to the set of all nodes of ensemble
5: end
6:   calculate the dissimilarity between all neurons contained in the set obtained in 3-6 using
     Eq. 5
7:   group in different sub-sets the nodes that satisfy that the dissimilarity between all of them
     is lower than the dissimilarity threshold and the distance between each of them and the
     rest of nodes in other sub-set is higher than that threshold.
     $\begin{cases} ds(v(n_r), v(n_q)) < \theta_f & \text{for all } n_r, n_q \in S_k \\ ds(v(n_r), v(n_q)) \geq \theta_f & \text{for all } n_r \in S_k,\ n_q \in S_l,\ k \neq 1 \end{cases}$
     The result will be a set of sub-sets (S).
8:   'fuse' all the nodes in each sub-set to form a node of the final fused network by
     calculating the centroid of the nodes in each sub-set (see Eq. 4). The fused network will
     have as many nodes as sub-sets are contained in S.
9:   create the final network (fus) including in it all the fused nodes
10:  create connections between fused nodes in the fused network (fus) to represent neuron
     neighbourhood. Connections will be established if the distance between fused nodes is
     lower than the connection threshold, considering this distance as:
     $\min_{n_r \in N_r, n_q \in N_r} ds(v(n_r), v(n_q)) < \theta_c$
```

Criterion 2: *Euclidean Distance.* 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 can be implemented using dynamic programming. A more detailed description of this procedure can be found in [14].

The difference with the previous criteria is that, in this case, a pair wise match of the neurons of each network is always possible, so the final fused network has the same size as the single composing ones. This implies that a certain global neighbouring structure can be kept and reconstructed in the fused network.

Algorithm 2. *Fusion based on Euclidean Distance*

```
Train several networks by using the bagging (re-sampling with replacement) meta-algorithm
Being nNet the number of networks to be fused
   nNeur the number of neurons composing each network
   fus the resultant fused network
Initialize fus with the neuron weights of the first network
for i=2:nNet
  for j=1:nNeur
    neuFus : neuron (j) of the fus network
    calculate Euclidean Distance (ED) between neuFus and ALL neurons of network(i)
    neuNet: neuron with the minimum ED
    calculate neuAvg: neuron whose weights are the average of the weights of neuFus and
    neuNet i.e. the centroid of both neurons' weights (see Eq. 4).
    remove neuNet from the set of neurons of the network
    replace neuFus by neuAvg in the fus network (in position j of the network)
  end for
end for
```

4. Performance Experiments

To test the characteristics and capabilities of the fusion of ViSOM and compare both of its variants several real datasets have been employed. Data is extracted form the UCI repository [15] and include the Iris, the Wisconsin Breast Cancer and the Wine datasets.

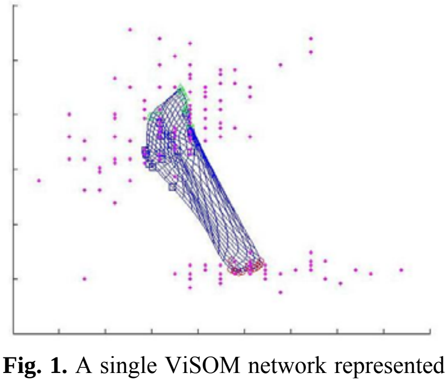


Fig. 1. A single ViSOM network represented over the iris dataset.

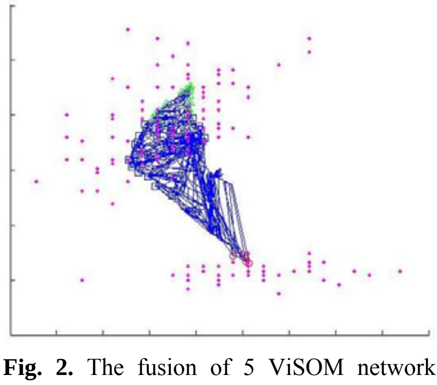


Fig. 2. The fusion of 5 ViSOM networks using the Euclidean Distance criterion

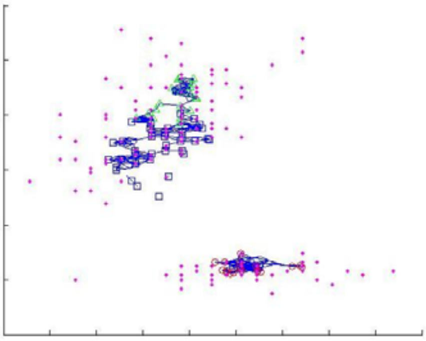


Fig. 3. The fusion of 5 ViSOM networks using the Voronoi polygon similarity criterion

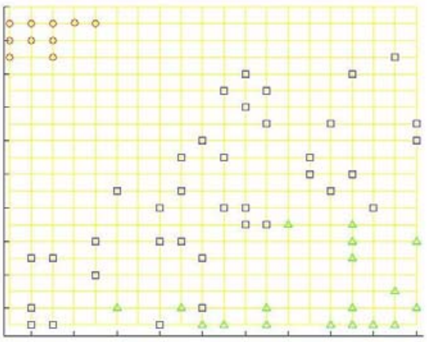
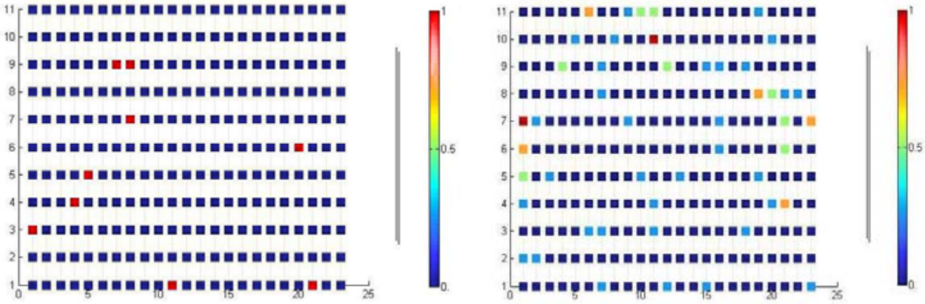


Fig. 4. The 2D map data representation of the fused network appearing on Fig. 2

Figures 1 to 3 depict a single ViSOM and the two variants of fusion represented over the iris dataset. Figure 4 represents the dataset plotted in 2 dimensions over the map obtained of 'unfolding' the network in Figure 2.

It is important to note a structural difference of the two variants of fusion. As explained before, the Euclidean distance variant enables the pair wise fusion of nodes of networks, so topology preservation is still valid in the fused network. This allows obtaining 2D maps such as the one showed in Fig. 4, which can be also easily obtained from single maps. This is impossible to do with the Voronoi similarity, as some neurons are not related to others and a position in the map in relation with the rest can not be determined.



(a) (b)

Fig. 5. Topographic error calculated over the single ViSOM of Fig 1 is shown in Fig 5(a) and over the Fusion by Euclidean distance of Fig 2 is shown in Fig 5 (b).

As reflected visually in Figures 1 and 2 the main problem of the Fusion by Euclidean distance is that it introduces distortions in the neighbourhood of the map. As can be seen in Figure 5a the single ViSOM includes few neurons with high topographic error that are located in a specific region of the map (coinciding with the gap between the linearly separable group and the other two), while in Fig 5b the number of neurons with a medium and high topographic are more numerous and more scattered along the whole map. This is a characteristic that should be corrected, maybe by a re-calculation of the neighbouring after the fusion.

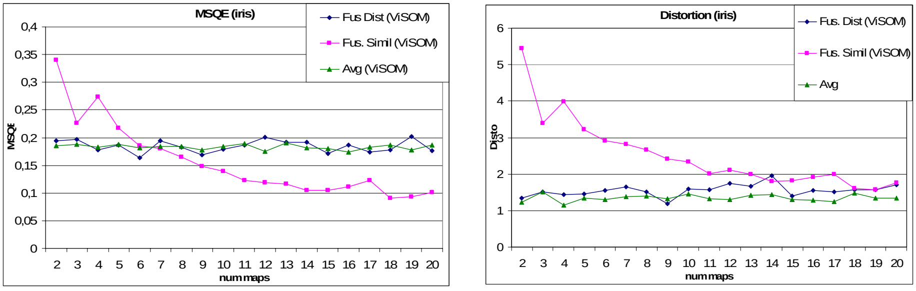
Table 1. Comparison of the two topology preserving models using an ensemble of 10 maps to calculate the **MSQE** for: the average of all 10 maps, the fusion of the 10 maps using the distance criterion and the fusion of the maps using the Voronoi similarity criterion.

	SOM			***ViSOM***		
	Avg	*Fus. Dist*	*Fus. Simil*	*Avg*	*Fus. Dist*	*Fus. Simil*
Iris	0,196	0,200	0,142	0,183	0,179	0,139
Cancer	1.959	1.931	1.161	1,746	1,544	1,231
Wine	9,912	10,406	4,420	9,401	9,138	4,067

Table 2. Comparison of the two topology preserving models using an ensemble of 10 maps to calculate the **Distortion** for: the average of all 10 maps, the fusion of the 10 maps using the distance criterion and the fusion of the maps using the Voronoi similarity criterion.

	SOM			***ViSOM***		
	Avg	*Fus. Dist*	*Fus. Simil*	*Avg*	*Fus. Dist*	*Fus. Simil*
Iris	1,354	1,500	2,127	1,451	1,593	2,336
Cancer	19,03	25,12	43,52	15,46	19,23	41,98
Wine	69,12	71,50	60,93	65,82	55,81	62,81

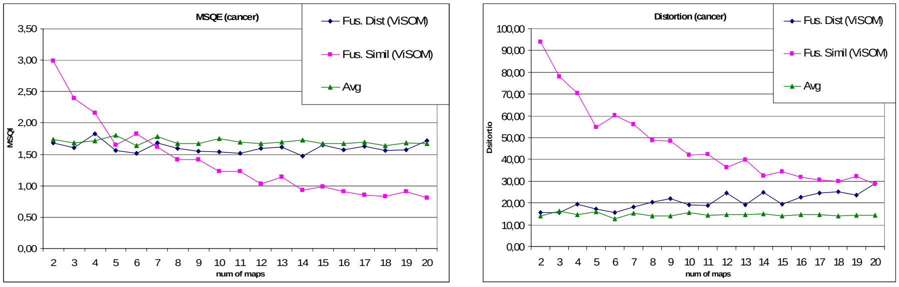
When comparing the two variants of the Topology preserving algorithms used, it can be inferred that the ViSOM obtains better results than the SOM, both for MSQE (Table 1) and Distortion (Table 2) measures, by a small margin, in the three datasets used. This is due to its updating of inter-neuron weights procedure, as it forces the map to adapt its inter-neuron distances to the inter-data distances of the input space; improving its adaptation to the dataset.



(a) (b)

Fig. 6. Results for the Iris Dataset using ViSOM maps. Fig 6(a) shows the MSQE of the different models (Fusion by Distance, Fusion by Voronoi Similarity and Average of Simple Model) according to the number of maps of the ensemble. Fig 6(b) shows the Distortion results for the same model. Both are calculated using the Iris dataset.

The comparative results obtained by the two fusion variants according to the number of networks trained in the ensemble, along with the average of the corresponding measures considering each network of the ensemble individually, are shown in Figures 6 and 7. In Fig 6 the results refer to the iris dataset, while Fig 7 represents the cancer dataset results. Regarding the MSQE measure (that is, how well the map units approximate the data entries in the dataset) it can be seen that the results for the fusion by distance are very similar to those of the average of the ensemble networks (even, in the cancer case, the error is consistently a bit lower). On the other hand, the fusion by similarity in Voronoi polygons obtains quite better results when surpassing the number of 7 or 8 maps in the ensemble (in both iris and cancer datasets). This is due to the nature of the algorithm that enables the fused map to adapt better to the dataset by ignoring the neighbouring of the neurons that are not within the range of a certain threshold. So, while the final map does only keep the neighbouring between certain regions of the map (as it can be seen in Fig 3), this separated regions of the map can approximate better to data in the input space.



(a) (b)

Fig. 7. Results for the Cancer Dataset using ViSOM maps. Fig 7(a) shows the MSQE of the different models (Fusion by Distance, Fusion by Voronoi Similarity and Average of Simple Model) according to the number of maps of the ensemble. Fig 7(b) shows the Distortion results for the same model. Both are calculated using the Cancer dataset.

Regarding the distortion measure (that is taking into account the quality both of quantization and neighbourhood); the fusion by distance of neurons is quite similar to the average of the networks (although it is consistently a bit higher in both cases). Both the average and the fusion by distance have lower errors than the fusion by Voronoi similarity, due again to the way it preserves the neighbourhoods of units in the final map. A neuron is considered to be neighbour of all other neurons that are within a certain distance. So while in a classic SOM with a square-shaped neighbouring a neuron can have between 4 and 2 neighbours (depending of its position in the map); in a fused SOM using the Voronoi polygons similarity a neuron does not have restrictions of the number of neighbours; altering the distortion of the final map. Despite this condition, when the number of neurons approximates to 20 the distortion becomes closer to that of the other two models. This could be due to the obvious increase in the number of neurons with the increase in the number of maps. Being more neurons to connect is more difficult to have isolated neurons or groups of neurons. In opinion of the authors of the paper, this phenomenon needs further study.

5. Conclusions and future work

The current work presents a study of a technique developed to obtain more reliable 2-dimensional maps representing multi-dimensional datasets. This is achieved by the use of the ensemble theory and a fusion process. Two different variants of the algorithm are considered and studied, by means of several widely used measures for quantifying the quality of this type of maps. It seems that the first presented option, the Euclidean distance criterion for fusion is the model that best preserves the main characteristics of the Self-Organizing Maps: both data quantization and topology preservation. Despite on this, the variant of the Voronoi polygon similarity has some interesting advantages that should be studied with the aim of being adopted to improve the final fused model.

Future work should include possible further study of the capabilities of each one of the variants, with the final objective of the proposition of a fusion algorithm developed to bring together the best characteristics of both variants. Also application and comparison of this new fusion algorithm with some other topology preserving ensemble models could be performed.

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

References

1. Kohonen, T., Lehtio, P., Rovamo, J., Hyvarinen, J., Bry, K. and Vainio, L.: A Principle of Neural Associative Memory. Neuroscience, Vol. 2 (1977) p. 1065-1076.
2. Kohonen, T.: The Self-Organizing Map. Neurocomputing, Vol. 21 (1998) p. 1-6.
3. Yin, H.: Visom - a Novel Method for Multivariate Data Projection and Structure Visualization. Neural Networks, IEEE Transactions on, Vol. 13 (2002) p. 237-243.
4. Kuncheva, L. I.: Combining Pattern Classifiers: Methods and Algorithms. (2004) pages. 0471210781: 0471210781
5. Ron, M. and 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, Vol. (2003) 118-183 pages.
6. Yin, H.: Data Visualisation and Manifold Mapping Using the Visom. Neural Networks, Vol. 15 (2002) p. 1005-1016.
7. Pölzlbauer, G.: Survey and Comparison of Quality Measures for Self-Organizing Maps. Fifth Workshop on Data Analysis (WDA'04). Elfa Academic Press. Vol (2004) p. 67--82
8. Lampinen, J. and Oja, E.: Clustering Properties of Hierarchical Self-Organizing Maps. Artificial Neural Networks, 2, Vol. II (1992) p. 1219--1222.
9. Ruta, D. and Gabrys, B.: A Theoretical Analysis of the Limits of Majority Voting Errors for Multiple Classifier Systems. Pattern Analysis and Applications, Vol. 5 (2002) p. 333-350.
10. Petrakieva, L. and Fyfe, C.: Bagging and Bumping Self Organising Maps. Computing and Information Systems, Vol. (2003)
11. Baruque, B., Corchado, E. and Yin, H.: Visom Ensembles for Visualization and Classification. International Work Conference on Artificial Neural Networks (IWANN'07). Springer-Verlag, Berlin. Vol San Sebastián, Spain. (2007)
12. Breiman, L.: Bagging Predictors. Machine Learning, Vol. 24 (1996) p. 123-140.
13. Saavedra, C., Salas, R., Moreno, S. and Allende, H.: Fusion of Self Organizing Maps. 9th International Work-Conference on Artificial Neural Networks, IWANN 2007. Springer. Vol 4507. San Sebastián, Spain. (2007) p. 227--234
14. Georgakis, A., Li, H. and Gordan, M.: An Ensemble of Som Networks for Document Organization and Retrieval. Int. Conf. on Adaptive Knowledge Representation and Reasoning (AKRR'05), Vol. (2005) p. 6.
15. Newman, D. J., Hettich, S., Blake, C. L. and 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