

# Ensemble Methods for Boosting Visualization Models

Bruno Baruque<sup>1</sup>, Emilio Corchado<sup>1</sup>, Aitor Mata<sup>2</sup> and Juan M. Corchado<sup>2</sup>

1. Department of Civil Engineering, University of Burgos, Spain.

2. Department of Computer Science and Automatics, University of Salamanca, Spain  
escorchado@ubu.es, bbaruque@ubu.es, aitor@usal.es, corchado@usal.es

**Abstract.** Topology preserving mappings are great tools for data visualization and inspection in large datasets. This research presents a study of the combination of different ensemble training techniques with a novel summarization algorithm for ensembles of topology preserving models. The aim of these techniques is the increase of the truthfulness of the visualization of the dataset obtained by this kind of algorithms and, as an extension, the stability conditions of the former. A study and comparison of the performance of some novel and classical ensemble techniques, using well-known datasets from the UCI repository (Iris and Wine), are presented in this paper to test their suitability, in the fields of data visualization and topology preservation when combined with one of the most widespread of that kind of models such as the Self-Organizing Map.

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

## 1. Introduction

From the range of tools that can be used to treat the high amounts of data that industrial and business operations processes, one of the most useful 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 obtain the results. The present work is centred on one of the 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.

The Self-Organising Map (SOM) [1] is probably the most widely used algorithm making use of this kind of learning. It is based on an adaptive process in which the neurons in a neural network gradually become sensitive to different input categories, or sets of samples in a specific domain of the input space. The SOM was conceived as a visualization tool to enable the representation of high-dimensional datasets on 2-dimensional maps and thereby facilitating data interpretation tasks for human experts.

The principal problem of the models based on competitive learning is, as happens with all ANNs, their instability. This means that even running the same algorithm several times with the same parameters can lead to rather different results. The present research is focused on the comparison and study of some novel and classical ensemble extension versions of the two competitive learning models based on the topology preserving concept. A novel summarization of topology preserving ensembles is presented and included in this comparison. This algorithm aims to obtain a more trustful representation of the datasets by combining the best features of several trained maps. The summarization algorithms are tested for the SOM model in combination with two ensemble techniques such as the Bagging [2] and the AdaBoost [3]. The purpose of this comparison is to verify if the performance of these unsupervised connectionist models can be improved by means of these ensemble meta-algorithms. AdaBoost is applied for the first time in this paper in combination with the WeVoS algorithm.

## 2. Self-Organizing Maps

The Self-Organizing Map (SOM) algorithm [4] 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 [5]. Its aim is to produce a low dimensional representation of the training samples while preserving the topological properties of the input space.

The main feature of the SOM algorithm is that the neighbours on the lattice are also allowed to learn – i.e. to adapt their characteristics to the input - as well as the winning neuron. Thus, the neighbouring neurons gradually come to represent similar inputs, and their representations become ordered on the map lattice.

This updating of neighbourhood neurons in SOM can be expressed as:

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

where,  $w_k$  is the weight vector associated with neuron  $k$ ;  $\alpha(t)$  is the learning rate of the algorithm;  $\eta(v,k,t)$  is the neighbourhood function (usually, the Gaussian function or a difference of Gaussians), in which  $v$  represents the position of the winning neuron in the lattice, or the best matching unit (BMU);  $k$ , the positions of neighbouring neurons and  $x$ , the network input.

## 3. Quality Measures

Several quality measures have been proposed in literature to study the reliability of the results displayed by topology preserving models in representing the dataset that have been trained with [6, 7]. There is not a global and unified one, but rather a set of complementary ones, as each of them asses a specific characteristic of the

performance of the map in different visual representation areas. The three used in this study are briefly presented in the following paragraphs.

**Classification Error** [8]. Topology preserving models can be easily adapted for classification of new samples using a semi-supervised procedure. A high value in the classification accuracy rate implies that the units of the map 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 precisely.

**Topographic Error** [9]. It consists on finding the first two best matching units (BMU) for each entry of the dataset and testing whether the second is in the direct neighbourhood of the first or not.

**Distortion** [10, 11]. When using a constant radius for the neighbourhood function of the learning phase of a SOM; the algorithm optimizes a particular function. This function can be used to quantify in a more trustful way than the previous one, the overall topology preservation of a map by means of a measure, called distortion measure in this work.

**Goodness of Map** [12]. This measure combines two different error measures: the square quantization error and the distortion. It takes account of both the distance between the input and the BMU and the distance between the first BMU and the second BMU in the shortest path between both along the grid map units, calculated solely with units that are direct neighbours in the map.

## 4. Unsupervised Competitive Learning Ensembles

The ultimate goal of constructing an ensemble is to improve the performance obtained by 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 [13]. Many ensemble models and theories have been previously developed and have been applied mainly to models designed specifically for classification, especially 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 [15] field over an unsupervised learning process such as the Competitive Learning.

### 4.1. 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 selected.

Bagging (or bootstrap aggregating) [2] is one of the simplest techniques for ensemble construction. 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 technique provides the ensemble with a balance between variability and similarity.

The idea of AdaBoost [3] is very similar to that of the Bagging. 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 associated 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 belong to two different classes while AdaBoost.M2 is recommended for dataset with more than two different classes.

The Adaboost algorithm requires a measure of accuracy of classification of each of the components of the ensemble. Therefore, a semi-supervised learning technique [8] is applied in this case to enable its use under the frame of topology preserving models.

#### 4.2. Summarizing some Applied Ensembles Techniques

Several algorithms for fusion of maps have been tested and reviewed recently by the authors of this work [16, 17].

In this case, an algorithm devised by the authors of this work is used to generate the final network summarizing the results obtained by the different networks included in the ensemble. It is called Weighted Voting Summarization (WeVoS) [18]. As the SOM is mainly designed as visualization tools, it is desirable that a combination of several of this kind of maps presents a truthful representation of data for visual inspection, based in the parts of the maps that were representing that portion of the data space the most correctly. The WeVoS tries to achieve this by taking into account one of the most important features of these algorithms: topology preservation. To do so, it obtains the final units of the map by a weighted voting among the units in the same position in the different maps, according to a quality measure. This measure can be any of the previously presented or other found in literature, as long as can be calculated in a unit by unit basis.

The voting process used is the one described in Eq. 2:

$$V_{p,m} = \frac{\sum_{m=1}^M b_{p,m} \cdot q_{p,m}}{\sum_{i=1}^M b_{p,i} \sum_{i=1}^M q_{p,i}} \quad (2)$$

where,  $V_{p,m}$  is the weight of the vote for the unit included in map  $m$  of the ensemble, in its in position  $p$ ,  $M$  is the total number of maps in the ensemble,  $b_{p,m}$  is the binary vector used for marking the dataset entries recognized by unit in position  $p$  of map  $m$ , and  $q_{p,m}$  is the value of the desired quality measure for unit in position  $p$  of map  $m$ .

A detailed pseudo-code of the WeVoS algorithm is presented in Table 1.

**Algorithm 1.** Weighted Voting Superposition (WeVoS).

---

```

1: train several networks by using the bagging (re-sampling with replacement) meta-algorithm
2: for each map ( $m$ ) in the ensemble
3:   for each unit position ( $p$ ) of the map
4:     calculate the quality measure/error chosen for the current unit
5:   end
6: end
7: calculate an accumulated quality/error total for each homologous set of units  $Q(p)$  in all maps
8: calculate an accumulated total of the number of data entries recognized by an homologous set of units in all maps  $D(p)$ 
9: for each unit position ( $p$ )
10:  initialize the fused map ( $fus$ ) by calculating the centroid ( $w'$ ) of the units of all maps in that position ( $p$ )
11: end
12: for each map ( $m$ ) in the ensemble
13:   for each unit position ( $p$ ) of the map
14:     calculate the vote weight of the neuron ( $p$ ) in the map ( $m$ ) by using Eq. 2
15:     feed the weight vector of the neuron ( $p$ ), as if it were a network input, into the fused map ( $fus$ ), using the weight of the vote calculated in Eq. 2 as the learning rate and the index of that same neuron ( $p$ ) as the index of the BMU.
        The unit of the composing ensemble ( $w_p$ ) is thereby approximated to the unit of the final map ( $w'$ ) according to its weighting system.
16:   end
17: end

```

---

## 5. 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 are Iris and Wine that were obtained from the UCI repository [19].

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. Then the computation of the combination is performed. Finally, both the ensemble and the combination generated from it are tested employing the test fold.

Visualization results are displayed in Fig.1 while analytical results appear in Fig.2. In Fig. 1 the maps obtained by the different combination of algorithms are displayed.

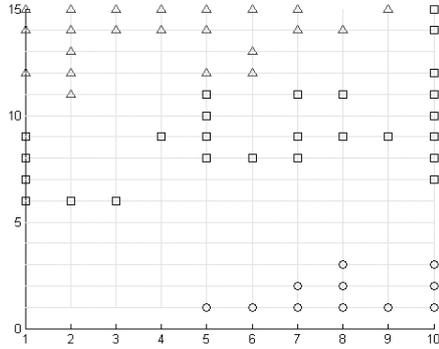


Fig. 1(a). Single SOM

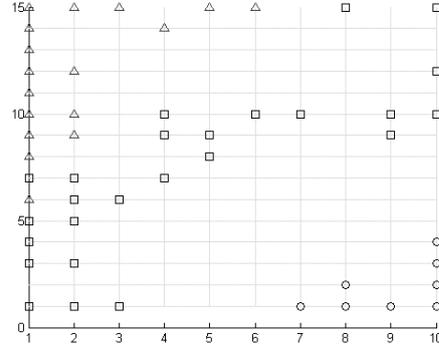


Fig. 1(b). WeVos from an ensemble trained using the Bagging algorithm.

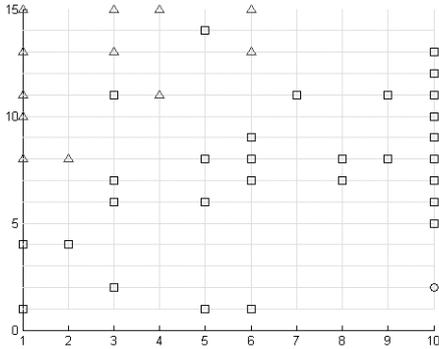


Fig. 1 (c). WeVos from an ensemble trained using the AdaBoost.M1 algorithm.

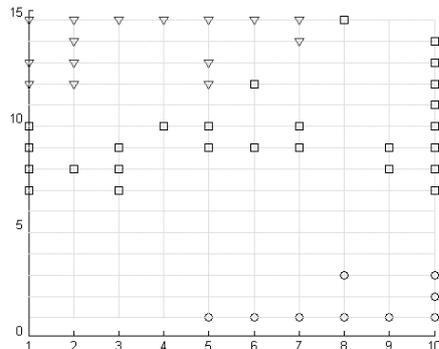


Fig. 1 (d). WeVos from an ensemble trained using the AdaBoost.M2 algorithm.

**Fig 1.** This figure shows the 2D maps representing the Iris dataset. Each figure represents a map obtained by training the ensemble of SOM algorithms using a different meta-algorithm and the applying the WeVoS algorithm to all of them.

Fig. 1 shows the results of applying a different ensemble algorithm to the same dataset with the same topology preserving algorithm. Both Adaboost.M1 and Adaboost.M2 have been tested for the sake of comparison taking into account that first algorithm can be applied also to multi-class datasets, although the second one should be more suitable. All SOMs in all ensembles showed where trained using the same parameters. Fig. 1(a) displays the map obtained by a single SOM. It contains three different classes, one of them (class one, represented by circles) clearly separated from the other two. Fig. 1(b) represents the summary obtained by the WeVoS algorithm over an ensemble trained using the bagging meta-algorithm. In this case, as all dataset entries are considered of the same importance in all iterations a smooth map is obtained. It is worth noting that classes are displayed in a more compact way than in the single SOM. Class 1 appears more distant to class 2 (squares) and class 2 and 3 (triangles) are more horizontally separated in the top of the image, although this separation is not so clear in the middle-left part of the image.

Fig. 1(c) represents the map obtained from an ensemble trained on AdaBoost.M1 algorithm. As this algorithm tries to concentrate in difficult to classify classes, only one neuron is used in the final map to represent class 1, which is obviously the most easy to distinguish from the three of them. This can be considered a desired or not so desired result, regarding to what the final resultant network is going to be used. In the case of intending to use the map for classification purposes, this map can be considered more suitable than the single one. In the case of intending to use the map for visualization purposes, this could be considered quite the contrary. Finally, the result of using the Adaboost.M2 is showed in Fig. 1(d). As this algorithm uses a finer granularity for classification than the previous version, it again represents class 1 in a greater detail than the AdaBoost.M1 (Fig. 1(c)), but showing a bit more compact groups than the single algorithm (Fig. 1(a)) and showing a more clear separation of groups than the Bagging algorithm (Fig. 1(b)) does.

Fig. 2 represents the results of the calculation of the different quality measures previously described over the different algorithms described. All are error measures, so the desired value is always as close to 0 as possible.

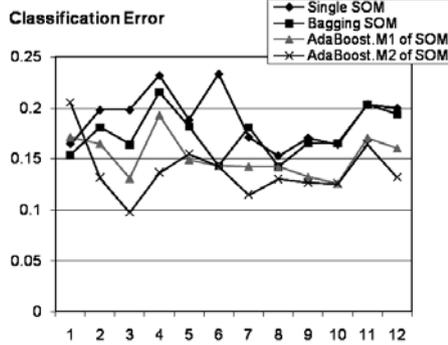


Fig 2(a). Classification Error

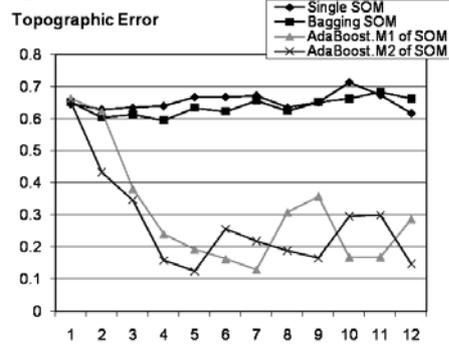


Fig 2(b). Topographic Error

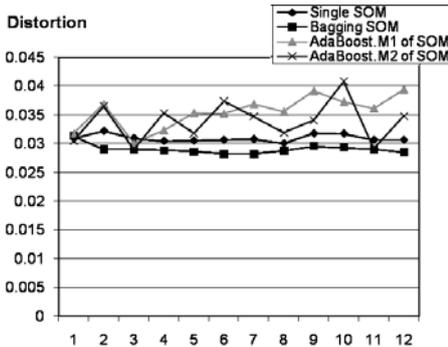


Fig 2(c). Distortion

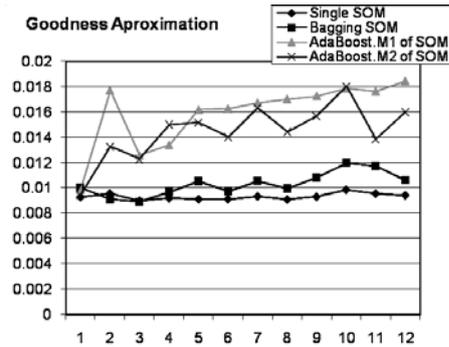


Fig 2(d). Goodness of Approximation

**Fig. 2.** Displays the quality of maps measures for each of the ensemble models presented, along with the corresponding single model. All models were trained and tested using the Wine dataset. The X-axis represents the number of single maps composing each of the ensembles and the Y-axis represents the value of each measure.

In Fig. 2, the quality measure values obtained from the ensemble algorithms are measured from their corresponding WeVoS summarization, while data from single models is obviously obtained directly by that model. The WeVoS algorithm has previously been compared with other summarization methods (Fusion by Euclidean distance [20], Fusion by similarity in Voronoi polygons [21]), showing an improvement in its results [18]. The results of those models are not included in this work for the sake of brevity. Fig. 2(a) represents the classification error of the different variants under study. Obviously, as that is their original purpose, the maps obtained through ensemble algorithms outperform the single model. Fig 2(b) represents the topographic ordering of the final map. Again, ensemble models obtain lower errors than the single one, especially the two variants of AdaBoost. Fig 2(c) measures in a greater detail the topological ordering of the maps and shows a different situation than Fig. 2(b) for the AdaBoost algorithm. This points to an overfitting problem. Finally, Fig. 2(d) represents a measure combining quantization and distortion errors. In this case, the results for the two variants of AdaBoost were expected, as the algorithm tries to concentrate not on the whole dataset, but on the most difficult to classify samples of it, increasing the distance to other samples. The bagging algorithm obtains lower error, although a bit higher than the simple model. This is due to the nature of the WeVoS algorithm, which benefits the topology preservation of the summary over the quantization side of the model, being the first one of the most characteristic features of the family of models under study.

## 6. 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 on the combination of several maps trained over slightly different datasets to form an ensemble of networks with self-organizing capabilities. This idea can be especially useful when a reduced dataset is available. The training of the ensemble of networks has been tested by using the bagging and boosting techniques for their comparison. As an ensemble of maps is impossible to represent, a summarization algorithm is also presented and used.

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. It is reasonable to think that, adapting the AdaBoost algorithm to enhance other capabilities of the SOM, such as topology preservation, these results can be improved.

Also the problem of the overfitting when using the AdaBoost algorithm will be subject of further analysis by authors of this paper.

## Acknowledgments

This research has been partially supported through project BU006A08 by the Junta de Castilla y León.

## References

1. Kohonen, T.: Self-organizing maps. Springer Series in Information Sciences, Vol. 30. Springer, Berlin, Germany (1995)
2. Breiman, L.: Bagging Predictors. *Machine Learning*, **24** (1996) 123-140
3. 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** (1997) 119-139
4. Kohonen, T.: The Self-Organizing Map. *Neurocomputing*, **21** (1998) 1-6
5. Kohonen, T., Lehtio, P., Rovamo, J. et al.: A Principle of Neural Associative Memory. *Neuroscience*, **2** (1977) 1065-1076
6. Pözlbauer, G.: Survey and Comparison of Quality Measures for Self-Organizing Maps. WDA'04, (2004) 67--82
7. Polani, D.: Measures for the organization of self-organizing maps. In: U. Seiffert and L. C. Jain (eds.): *Self-Organizing Neural Networks: Recent Advances and Applications (Studies in Fuzziness and Soft Computing)*, Vol. 16. Physica-Verlag, Heidelberg (2003) 13-44
8. Vesanto, J.: Data Mining Techniques Based on the Self-Organizing Map. (1997) 63
9. Kiviluoto, K.: Topology Preservation in Self-Organizing Maps. (ICNN'96), 1 (1996) 294-299
10. Lampinen, J.: On Clustering Properties of Hierarchical Self-Organizing Maps. *Artificial Neural Networks*, 2, **II** (1992) 1219-1222
11. Vesanto, J., Sulkava, M., Hollmén, J.: On the Decomposition of the Self-Organizing Map Distortion Measure. WSOM'03, (2003) 11-16
12. Kaski, S., & Lagus, K.: Comparing Self-Organizing Maps. ICANN'96, 1112 (1996) 809-814
13. Heskes, T.: Balancing between Bagging and Bumping. (NIPS), 9 (1997) 466-472
14. Schwenk, H., & Bengio, Y.: Boosting Neural Networks. *Neural Computation*, **12** (2000) 1869-1887
15. Freund, Y., & Schapire, R. E.: Experiments with a New Boosting Algorithm. (1996) 148-156
16. Baroque, B., Corchado, E., Yin, H.: Quality of Adaptation of Fusion ViSOM. IDEAL 2007, (2007) 728-738
17. Corchado, E., Baroque, B., Yin, H.: Boosting Unsupervised Competitive Learning Ensembles. (ICANN 2007), 4668 (2007) 339-348
18. Baroque, B., Corchado, E., Rovira, J. et al.: Application of Topology Preserving Ensembles for Sensory Assessment in the Food Industry. (2008) 491-497
19. Asuncion, A., & Newman, D. J.: UCI Machine Learning Repository. 2008 (2007)
20. Georgakakis, A., Li, H., Gordan, M.: An Ensemble of SOM Networks for Document Organization and Retrieval. (AKRR'05), (2005) 6-141
21. Saavedra, C., Salas, R., Moreno, S. et al.: Fusion of Self Organizing Maps. IWANN 2007, 4507 (2007) 227-234