Noname manuscript No.
(will be inserted by the editor)

A Weighted Voting Summarization of SOM Ensembles

Bruno Baruque and Emilio Corchado

the date of receipt and acceptance should be inserted later

bbaruque@ubu.es
escorchado@ubu.es

Abstract Weighted Voting Superposition (WeVoS) is a novel summarization algorithm for the results of an ensemble of Self-Organizing Maps. Its principal aim is to achieve the lowest topographic error in the map in order to obtain the best possible visualization of the internal structure of the data sets under study. This is done by means of a weighted voting process between the neurons of the ensemble maps in order to determine the characteristics of the neurons in the resulting map. The algorithm is applied in this case to the most widely known topology preserving mapping architecture: the Self-Organizing Map. A comparison is made between the novel fusion algorithm presented in this work and other previously devised fusion algorithms, along with a new variation of those algorithms, called Ordered Similarity. Although a practical example of the new algorithm was introduced in an earlier work, a rigorous description and analysis is presented here for the first time by comparing the performance of the aforementioned algorithms in relation to three well-known data sets (Iris, Wisconsin Breast Cancer and Wine) obtained from Internet repositories. The results show how this novel fusion algorithm outperforms the other fusion algorithms, yielding better visualization results for ensemble summarization of maps. Data Visualization, Topology Preservation, Ensemble Learning, Self-Organizing Maps

1 Introduction

The extraction of information from the enormous data sets that are generated by modern experimental and observational methods is increasingly necessary in almost all scientific fields [Patra et al., 2006] and business operations processes [Kohonen et al., 1996] nowadays. This "information extraction" is defined as the nontrivial data mining of implicit, previously unknown, and potentially useful information. One of the many techniques used to extract relevant information is data visualization.

Artificial neural networks (ANN) have wide range of applications and are prominent among the great variety of analytical tools for multivariate data analysis and

Dept. Civil Engineering. University of Burgos.

pattern recognition currently under development. In this work, we concentrate on Self-Organizing Maps (SOMs) [Kohonen et al., 1977, Kohonen, 1988, 1995], a network type that is characterized by the use of unsupervised and competitive learning to generate maps that enable visual data inspection.

This family of algorithms was originally created as a visualization tool; enabling high-dimensional data sets to be represented on 2-dimensional maps, which facilitates data interpretation by the human expert [Gianniotis and Tino, 2008]. One of the most well documented problems of all the neural network algorithms is that they are, in general, rather unstable [Heskes, 1997] due to the fact that there are several random processes (weights initialization, random selection of data inputs...) that take place during the training of that kind of models. Taking this into account, running the same algorithm, even when using the same parameters -except the previously mentioned ones, that will be random-, can lead to quite different results, which makes it very difficult to identify any result as objectively better than the others. Another typical problem of these algorithms is overfitting [Ling, 1995, Zhang et al., 1998], when the adaption of the ANN to the data set under study is overly accurate, but it is unable to generalize new data well. Thus, the use of ensembles is rapidly establishing itself as one most widespread techniques for increasing the stability of an analysis model and to avoid overfitting [Johansson et al., 2006, Petrakieva and Fyfe, 2003, Schwenk and Bengio, 2000]. This meta-algorithm consists in training several slightly different models over the same data set and relying on the combination of their results, rather than the results of any single model. It is based on the intuitive idea that a committee of experts, each with their own experience in the same area, working to solve a particular problem would lead to a more reliable solution than having only one single expert dedicated to solving the same problem.

The technique is used in many studies that apply mainly to classification problems. In this work, however, the aim is to obtain the most reliable possible representation of a multidimensional data set over a 2-dimensional map. Classic ensemble summarization techniques are therefore not directly applicable in this case. Some models already exist for the summarization of topographic preserving maps [Georgakis et al., 2005, Petrakieva and Fyfe, 2003, Saavedra et al., 2007], although certain characteristics of the topology preserving models have not been considered. The earlier models calculate the best characteristics vector for each neuron in the fused map from the characteristics vectors of the neurons in the ensemble maps, but they do not take account of the neighbourhood of each neuron. This results in poor topology preservation in the final fused map.

In this study we present both a novel variation of earlier algorithms and a novel method for the fusion of the SOM called WeVoS-SOM that can outperform some of the proposed models in several aspects of visual representation. Although a practical example of the new algorithm was introduced in an earlier work, a rigorous description and analysis is presented here for the first time by comparing the performance of the aforementioned algorithms in relation to three well-known data sets (Iris, Wisconsin Breast Cancer and Wine) taken from the UCI web repository [Asuncion and Newman, 2007].

The rest of this work is organized as follows: Section 2 briefly reviews the widely used Self-Organizing Map algorithm. Section 3 presents certain quality measures proposed in the literature to analyse several aspects and properties of a SOM. Section 4 includes a very brief description of the ensemble meta-algorithm. Section 5 describes the main characteristics of previously proposed algorithms that are used to obtain

ensemble summaries of SOMs. Section 6 describes the novel proposed method for summary calculation in detail: Weighted Voting Superposition (WeVoS). Section 7 sets out and comments on the evaluation of the WeVoS algorithm and the properties of its summaries in comparison with those of the other ensemble summarization methods. Finally, Section 8, summarizes the conclusions and outlines future lines of work. Appendix A also includes a table with full details and figures on the experiments and the results, which are not included in the paper for reasons of brevity. Appendix B includes additional figures obtained in the experimental part of this work.

2 The Self-Organizing Map

Topology preserving mapping 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 is the Self-Organizing Map (SOM) algorithm [Kohonen et al., 1977, Kohonen, 1988, 1995]. 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.

As a result of the learning process, i.e. the presentation of all input vectors and the adaptation of the weight vectors, the SOM generates a mapping from the input space onto the lattice U, in such a way that the topological relationships in the input space are as far as possible preserved in U.

If not only the winning neuron but also its neighbours on the lattice are allowed to learn, that is to adapt its characteristics to those presented as the input; neighbouring neurons gradually specialize to represent similar inputs, and the representations become ordered on the map lattice. This is the main feature of the SOM algorithm. Update of neighbourhood neurons in SOM:

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

where w_k is the winning neuron, also called best matching unit (BMU); α 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 and k the positions of the neurons in the neighbourhood of this one, and x is the network input.

3 Quality Measures

Several quality measures have been proposed in literature [Polani, 2003, Pozlbauer, 2004] to determine the reliability of the results of representing the data set used in the training, which are displayed by Self-Organizing Maps. In the following section, four of these measures are briefly described. Several measures are considered, because each of them aims to measure one specific characteristic of a SOM. Thus, instead of a global, unified measure, there are complementary measures that offer useful information on the performance of the map in different representation areas. These measures have been chosen with the objective of studying and measuring the widest possible range of these characteristics.

As mentioned earlier, this novel model entitled WeVoS aims to obtain a truthful representation of the data set in a map, in order to generate the best results from a visualization point of view. Hence, the most important features to evaluate in this study are the neighbouring relationships of the neurons in the map and the continuity of the map. In order to do so, the main measures to be analysed are topographic error, distortion and to some extent the goodness of the map. The comparison of the models in this research includes the mean square quantization error measure for the sake of completeness.

3.1 Topographic Error

The main characteristic of the SOM is probably topology preservation. To quantify this, a topographic error measure [Kiviluoto, 1996] is used in this work. One of the first and the simplest topology measures, it identifies the first two best matching units for each data set entry and verifies whether or not the second is in the direct neighbourhood of the first. This can be computed as a normalized single value, by calculating the mean error of each neuron, indicating the overall quality of the map; or it can be decomposed to be visualized over each neuron of the map. Its mathematical expression is shown in Eq. 2:

$$E_T = \frac{1}{|D|} \sum_{x_i \in D} u(x_i) \tag{2}$$

where $|D|$ is the number of entries contained in the test data set, D and the value of function u is either 1, if the first and second BMU for input entry x_i are adjacent in the map grid, or 0 otherwise.

3.2 Distortion

As explained in [Lampinen and Oja, 1992], the algorithm optimizes a function known as the distortion measure when using a constant radius for the neighbourhood function of the learning phase of a SOM. This function can be used to measure the overall topology preservation of a map in a more detailed way. It is computed as shown in Eq. 3:

$$E_D = \sum_{x_i \in D} \sum_{w_k \in W} \eta(v_i, k) \left\| x_i - w_k \right\|^2 \tag{3}$$

where x_i represents each entry of the data set;$\eta(v_i, k)$ represents the neighbourhood function between the BMU and every other neuron in the map; (v_i the position of the BMU corresponding to x_i; k the position of each other neuron in the network; and w_k being each of the characteristics vector that compose the network (W). Further discussion can be found in Vesanto et al. [2003]. Special attention is paid to this measure in this research due to its relation with visualization properties.

3.3 Mean Square Quantization Error

The mean square quantization error (MSQE) is a very widely known measure, which can be used in algorithms that perform any kind of vector quantization. In our case, it indicates how well the neurons of the map approximate the data on the data set. Or in other words, how close or how far away the neurons composing the map are from the different data entries they recognize (that is, the extent to which they are considered the best matching unit for that entry), in the input space. It can be easily adapted for use with the SOM, as shown by Eq. 4:

$$E_Q = \frac{1}{|D|} \sum_{x_i \in D} \|x_i - v_i\|^2 \tag{4}$$

where $|D|$ is the number of entries in the data set used D, and w_i the BMU for each presented input x_i of the data set.

3.4 Goodness of a Map

This measure, described in [Kaski and Lagus, 1996] combines two of the previous error measures: the square quantization error and the topographic error. It takes account of the distance between the input and the BMU as well as the distance between the first BMU and the second BMU on the shortest path between both along the grid-map neurons. It is calculated by taking only those neurons into account that are direct neighbours in the map for each input by measuring both the continuity of the mapping from the data set to the map grid, and the accuracy of the map when representing the set. The mathematical expression of this measure for each data entry is shown in Eq. 5:

$$d(x_i) = \|x_i - v_i\| + min \sum_{k=0}^{|K_{v'_i}|-1} \left\| w_{I_{i(k)}} - w_{I_{i(k+1)}} \right\| \tag{5}$$

where, v_i and v'_i represent the weights of the first BMU and the second BMU respectively, corresponding to data entry x_i. $I_{i(k)}$ and $I_{i(k+1)}$ represent indexes of the k^{th} and the k^{th+1} neurons along the minimum path from v_i to v'_i, both neurons being direct neighbours in the map grid. According to that definition $w_{I_i(0)} = v_i$; which is to say, the first neuron in the path is the first BMU for data entry x_i and $w_{I_i(k_{v'_i})} = v'_i$, which is to say, the last neuron in the path that corresponds to the second BMU for data entry x_i. The final goodness of the map is defined as the average of the values obtained by Eq. 5 for all data entries in the test data set.

4 The Ensemble Meta-Algorithm

The use of an ensemble of similarly trained models or algorithms was conceived to improve the performance of classification algorithms [Breiman, 1996]. It has been observed that, although one of the classifiers in an ensemble might yield the best performance, the sets of patterns that are 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. The idea is not to rely on a single decision making scheme, but to use all the designs, or their subsets, for decision making by combining their individual opinions to derive a consensus decision [Ruta and Gabrys, 2005]. The main problem of competitive learning based networks is that they are inherently unstable due to the random nature of their learning algorithm. The main idea of this work is that the effect of this instability may, however, be minimized by the use of ensembles. The learning algorithm of the Self-Organizing Maps specifies that their composing units (or neurons) specialize during the algorithm iterations at recognizing certain types of inputs, which also determines the topology of the map. In a similar way to the classification process, we can infer that the regions of the maps that do not reliably represent the real nature of the data set do not necessarily overlap. Therefore, the visualization of a single map might be improved by adapting each of the composing neurons of a map in the best way possible to the data set under study by using ensemble techniques, as they offer complementary visualizations of the maps [Corchado et al., 2007, Baruque et al., 2007].

The idea of using several maps to improve the results of a single SOM has previously been proposed in different ways. One of the most well-known is the Hierarchical SOM [Blackmore and Miikkulainen, 1993]. This algorithm generates a pyramidal structure of SOMs that creates a hierarchical representation of the data set. The higher level maps are used to discriminate between the lower level maps to decide the one to which the input data will be presented; inducing an initial clustering of data. The lower levels are used to represent the structure of each of the clusters in detail.

In our case, the intention is also to use several maps to improve the results of a single one, but in this case a hierarchical structure will not be observed. Rather than using a division of maps to force a clustering of data, the maps in the ensemble will be summarized into a single one, aiming to preserve the most relevant characteristics of each of its components.

The algorithms that combine classifiers can be divided into two broad classes. The simpler types of algorithms [Breiman, 1996] merely combine, by averaging in some way, the results of each composing classifiers of the ensemble yields into a final result. More complex types of algorithms [Kuncheva, 2004] 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. The second type of classifier combination was initially applied to classic classification trees. Its main advantage is that it combines the improvements to classification quality with the simplicity of having to handle only one classifier [Freund and Schapire, 1996, Kuncheva et al., 2002].

In our case, the conceptual perspective we follow to improve the performance of the model is that of a single "summary" or "synthesis" of the inputs stored within the whole ensemble, that is referred to as 'fusion' throughout this work. The main objective is to obtain a unique map that may be seen to represent as clearly and as reliably as possible the different features contained in the different maps in the ensemble.

5 Self-Organizing Mapping Fusion

The models used in this work are mainly designed as visualization tools. Classical ensemble results combination 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. Representing all the networks in a simple image can only be useful when dealing with only 1-dimensional maps [Petrakieva and Fyfe, 2003] but it is unmanageable when visualizing 2-D maps. To overcome this problem a novel ensemble combination algorithm has been devised during this work in order to obtain a unique network that to some extent represents the information contained in the different networks composing the ensemble. The objective of the combination is to obtain a truthful and stable representation of data for visual inspection. This part of the work has several approaches, two of which were already introduced in previous works on SOM fusion [Georgakis et al., 2005, Saavedra et al., 2007]. These two approaches are described in Section 5.2. A combination of both previous criteria called Ordered Similarity, developed under the present line of work is also presented for the first time. Finally, WeVoS-SOM is described in detail as a novel approach to the fusion of maps and as the main contribution of this study. The description of this algorithm is included in Section 6.

5.1 Common Procedure for Ensemble Training

A network ensemble must firstly be obtained in order to perform any of the map combination algorithms that aim to improve the results of a single map. The way the ensemble is trained does not affect the way the combination is computed. In the case of this study the training of all the map ensembles was done with the bagging [Breiman, 1996] meta-algorithm. The map combination is performed once all of the networks composing the ensemble have finished their training.

Each individual map of an ensemble is trained on one of the re-sampled subsets of the training set (n-1 folds of the whole data set), obtained as required in the bagging algorithm, using exactly the same parameters for training. This generates several slightly different trained networks which are combined into a final network that is expected to outperform all of them individually. In order to force more commonality for all the maps in the same ensemble, firstly an initial map is calculated. If it is considered valid (a quality measure is within a threshold) all other maps in the ensemble are initialized with the weights of that previously trained one, with an order of 10^{-1}. A quality measure is used (the MSQE, in the case of this work) to ensure that the recently trained map yields reasonably similar results to the others, before it is added to the ensemble. The value of that threshold is a parameter that must be tuned according to the data set. In the case of the experiments showed in the study, the value was determined empirically.

The different options that are studied for combining the ensemble network into a single network, which summarizes its main characteristics, are described in the following section (Section 5.2).

5.2 Fusion

In this work the term 'fusion' designates the process that combines several different maps trained as an ensemble (that is, over slightly different portions of the same data set) into a final map. In all cases, the neurons composing the final map are obtained as the fusion of neurons from different maps that are considered "close enough" to be

fused into one, as they seem to represent the same portion of the input data space. This is done by calculating the centroid of the weights of the neurons to fuse (Eq. 6):

$$w_c = \frac{1}{|W_k|} \sum_{w_i \in W_k} w_i \tag{6}$$

where W_k are the weight vectors of the set of neurons to fuse. That process is repeated until all neurons in all of the trained networks are fused into a unique final network. The criterion to determine which neurons are "close enough" to be fused is what differentiates the three algorithms in this comparative study.

Fusion Method 1: Euclidean Distance. This algorithm [Georgakis et al., 2005] first 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. The algorithm may be implemented by using a dynamic program to deal with its high computational complexity, which is done through an initialization of the fused map by calculating the fusion using only two maps of the ensemble. Then, the same calculation is performed again between the resultant fused map and another one of the maps composing the ensemble. This process is repeated until all the maps of the ensemble have been included in the calculation of the fused map.

The main characteristic of this approach is that a pair-wise match of the neurons of each network will always take place, so the final fused network has the same size as the single composing ones. Thus, all maps must have the same size in order to apply this algorithm. It also implies that a certain global neighbouring structure can be kept and reconstructed in the fused network, although it might not be as correct as the neighbouring structure of the single maps, due to the fact that when completing the fusion of two neurons, the neighbouring neurons are not taken into account. As explained before, fusing two neurons will result in a neuron associated with a slightly different characteristic vector. In visual terms, this is the same as "shifting" the position of a neuron in a map. If done without taking account of the neighbouring neurons, two neurons considered neighbours will not necessary be the two closest neurons of the network in the final fused map. The complete algorithm implementing this fusion method is detailed in Algorithm 1.

As the authors state in the original paper [Georgakis et al., 2005], from the pseudo-code it can be inferred that the computational complexity of the algorithm [Brassard and Bratley, 1995] is of $O((M-1) \times N^2)$, in which M is the number of maps in the ensemble, and N the number of neurons in each map.

Fusion Method 2: Voronoi Polygons Similarity [Saavedra et al., 2007]. Each neuron in a Self-Organizing Map can be associated with a portion of the input data space called the Voronoi polygon [Aurenhammer and Klein, 2000]. That portion of the input multi-dimensional space is the portion that contains data for which that precise neuron is the BMU of the whole network. 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. A record may be kept of which data entries activated each neuron as the BMU, to calculate the dissimilarity between the Voronoi polygons of two neurons. This can be easily done by associating a binary vector to the neuron, the length of which is the size of the data set and contains

Algorithm 1 Map Fusion by Euclidean Distance

Input: Set of trained topology-preserving maps: $M_1 ... M_n$
Output: A final fused map: M_{fus}

Select a training set $S = \langle (x_1, y_1) \dots (x_m, y_m) \rangle$
train several networks by using the bagging (re-sampling with replacement) meta-algorithm : M_n
procedure FUSION(M_n)
    initialise M_{fus} with the weight vectors of the first map: $M_{fus} \leftarrow M_1$
    for all $M \in M_n$ **do**
        for all $w'_i \in M_{fus}$ **do**
            calculate Euclidean Distance between w'_i and ALL neurons of map M_i
                ▷ let w^* be the closest neuron in map M_i to the one selected in M_{fus}
            $w^* \leftarrow argmin_i \left(ED(w'_i, w_i)\right)$
            $w_c \leftarrow w'_i + w^*/2$ ▷ applying Eq. 6 to two neurons
            $w'_i \leftarrow w_c$ ▷ replace w_i by the centroid (w_c)
        end for
    end for
end procedure

zeros in the positions where the neuron was the BMU for that sample and zeros in all other positions. The dissimilarity (i.e. the distance) between neurons can therefore be calculated, as shown in Eq. 7:

$$ds(b_r, b_q) = \frac{\sum XOR(b_r, b_q)}{\sum OR(b_r, b_q)} \tag{7}$$

where r and q are the neurons whose dissimilarity will be determined and b_r and b_q the binary vector relating each neuron with the data sample that it recognises.

The main issue with this proximity criterion is that it depends on the recognition of data by the network, rather than on the network 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 set threshold are removed before calculating the similarities between them.

Thus, the neighbouring properties of the whole network are no longer considered. The similarity criteria must be used again to keep a notion of neighbouring between the neurons of the fused network. Neurons with dissimilarity below a certain threshold will be considered as neighbours in the fused network.

This whole procedure implies that the final map will approximate very well to the data set, enhancing the vector quantization feature of the SOM. Its drawbacks are that the number of neurons in the final map is unknown beforehand (and will almost certainly differ from the size of the composing maps) and that the neighbourhood relationships of the composing maps will be ignored in the final one, as the latter will create a new neighbourhood for each neuron based on its dissimilarity with the others. This characteristic means that the algorithm yields very different results in its quality calculations that relate to topology preservation measures. A detailed description is shown in Algorithm 2.

By analysing Algorithm 2, it may be inferred that its computational complexity is also quadratic. In this case, the algorithm performs several stages of calculations that are sequential and for keeping explanations clear are analysed sequentially. First, a record of the data entries recognized by each neuron is constructed: $O(D \times N)$. Than, all neurons in each map are checked according to the usage threshold: $O(M \times N)$. All

Algorithm 2 Map Fusion by Voronoi Polygon Similarity

Input: Set of trained topology-preserving maps: $M_1 ... M_n$,
usage threshold: θ_u , fusion threshold: θ_f, connection threshold: θ_c
Output: A final fused map: M_{fus}

1: Select a training set $S = \langle (x_1, y_1) \dots (x_m, y_m) \rangle$
2: train several networks by using the bagging (re-sampling with replacement) meta-algorithm : M_n
3: let θ_u, θ_f and θ_c be the usage, fusion and connection thresholds respectively
4: **procedure** FUSION($M_1 ... M_n$)
5: **for all** $M_i \in M_n$ **do** ▷ for all maps in the ensemble
6: **for all** $w_j \in W_i$ **do** ▷ for all neurons in each map
▷ accept neurons with a recognition rate higher than a given threshold
7: $W_{fus} \leftarrow w_i \; if \; \sum_i b_r(i) > \theta_u$
8: **end for**
9: **end for**
10: **for all** $w_i \in W_{fus}$ **do**
11: calculate dissimilarity between w_i and ALL neurons in W_{fus} (Eq. 7)
12: $D_i \leftarrow ds(w_i, w_k) \forall w_k \in W_{fus}$
13: **end for**
14: group into different sub-sets (Ws_n) the neurons that satisfy the following conditions:
$\begin{cases} ds(b_r, b_q) < \theta_f & \forall r, q \in Ws_n \\ ds(b_r, b_q) > \theta_f & \forall r, q \notin Ws_n \end{cases}$
15: **for all** Ws_n **do**
16: calculate the centroid (w_c) of the set by using Eq. 6
17: add the centroid to the set of nodes of the final map (W^*_{fus})
18: **end for**
19: **for all** $w_r \in W^*_{fus}$ **do** ▷ for all neurons in the fused map
20: Connect w_r with any other neuron in W^*_{fus} , if they satisfy $\min_{b_r \in Ws_k, b_q \in Ws_l} ds(b_r, b_q) < \theta_c$
21: **end for**
22: **end procedure**

the remaining neurons for each map must be compared to calculate their dissimilarity: $O(N'^2)$, where N' is the total number of neurons of all maps that have recognition rate higher thanθ_u. The same complexity applies to the calculation of which neurons are to be grouped together: $O(N'^2)$. And finally, the same occurs when calculating the final dissimilarity between the fused neurons: $O(S^2)$, where S is number of sets calculated in step 14 of the algorithm, that coincides with the the number of final neurons in the map.

Fusion Method 3: Ordered Similarity. In this study, we present and test a novel criterion based on the cooperative use of the two previously described fusion methods.

The second fusion method (Voronoi Similarity) can obtain a very good adaptation from the neurons of the map to the data set, with a very low quantization error. This is a useful characteristic when the aim of the training is to learn and to represent the topology of a 2-D data set; but it is not a great help when trying to represent the inner structure of a multi-dimensional data set on a 2-D map, as a lot of neighbouring information between neurons is disregarded. Neurons that do not recognize any data entry are left out from the final map, leaving blank spaces in their positions.

This situation may be improved by using either of the two previously discussed criteria for determining the two closest neurons to fuse, depending on the situation. The idea is to calculate the fusion on a neuron-by-neuron basis again, as described in

both previous algorithms. The variation here is that, firstly the most similar neuron in the Voronoi Similarity (Algorithm 2) is identified (see Eq. 7), in order to find a suitable neuron to fuse in a map from the ensemble. If the neuron in the fused map does not recognize any data entry, then the search for the closest neuron in the Euclidean Distance (Algorithm 1) is performed. The fusion of neurons is done in the same way as both other fusion methods.

Algorithm 3 Map Fusion by Ordered Similarity

Input: Set of trained topology-preserving maps: $M_1 ... M_n$
Output: A final fused map: M_{fus}

Select a training set $S = \langle (x_1, y_1) \dots (x_m, y_m) \rangle$
train several networks by using the bagging (re-sampling with replacement) meta-algorithm : M_n
procedure FUSION(M_n)
    initialise M_{fus} with the weight vectors of the first map: $M_{fus} \leftarrow M_1$
    for all $M \in M_n$ **do**
        for all $w'_i \in M_{fus}$ **do**
            calculate Dissimilarity between w'_i and ALL neurons of map M_i (see Eq. 7)
            calculate Euclidean Distance between w'_i and ALL neurons of map M_i
            if $\sum b_i > 0$ **then** ▷ the neuron is recognizing some data
                ▷ let w^* be the neuron in map M_i with the most similar Voronoi polygon to the one selected in M_{fus}
                $w^* \leftarrow argmin_i \left(ds(b_i, b_q)\right)$
            else ▷ let w^* be the closest neuron in map M_i to the one selected in M_{fus}
                $w^* \leftarrow argmin_i \left(ED(w'_i, w_i)\right)$
            end if
            $w_c \leftarrow w'_i + w^*/2$ ▷ applying Eq. 6 to two neurons
            $w'_i \leftarrow w_c$ ▷ replace w_i by the centroid (w_c)
        end for
    end for
end procedure

This way, the blank spaces that the Voronoi similarity fusion method leaves, are filled with neurons that do no recognize any data; but the neighbouring relationship between neurons are maintained and therefore, the topography preservation. This approach was defined and tested for the first time in the present research and the criterion is referred to in this work as "Ordered Similarity".

The computational complexity is quadratic; as it follows the complexity of the main fusion methods on which it is based :the Fusion by Euclidean Distance (see Algorithm 1). Firstly, a calculation of the dissimilarity between all neurons in all maps must be performed: $O(M \times N)$, M being the number of maps in the ensemble and N the total number of neurons in the maps (as it is done in Algorithm 2). Then, the calculation of dissimilarities between all neurons must be determined in which, as in the Fusion by Distance, the complexity is therefore $O((M-1) \times N^2)$.

6 A Novel Fusion Algorithm: Weighted Voting Superposition

The idea behind the novel fusion method presented in this study, WeVoS, is to obtain the final map on a neuron-by-neuron basis. Unlike the three previously explained methods, which establish the best position for a single neuron , it aims to obtain the best

position for a neuron and their neighbours. As a consequence, the final map keeps one of the most important features of this type of algorithms: its topological ordering.

WeVoS is an improved version of an algorithm presented in several previous works on superposition [Baruque et al., 2007, 2008]. In this study, it is applied for the first time to well-know data sets to perform a thorough study and comparison of its capabilities.

The first step in this meta-algorithm is to calculate the "quality" of each of the neurons composing each map, in order to relay some kind of informed decision for the fusion of neurons. This "quality" measure (or rather, error measure) can be any of the many quality of map measures existing in literature regarding the Self-Organizing Maps [Polani, 2003, Pozlbauer, 2004], as long as this measure can be calculated on a neuron-by-neuron basis.

The final map is obtained again on a neuron-by-neuron basis. First, the neurons of the final map are initialized by calculating the centroids of the neurons in the same position of the map grid in each of the trained maps. Then, a recalculation of the final position of that neuron uses the information associated with the neurons in that same position in each map of the ensemble. For each neuron, a sort of voting process is performed, as in Eq. 8:

$$V_{p,m} = \frac{\sum b_{p,m}}{\sum_{i=1}^{M} b_{p,i}} \cdot \frac{\sum q_{p,m}}{\sum_{i=1}^{M} q_{p,i}} \tag{8}$$

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

The weights of the neurons are fed into the final network as with the data inputs during the training phase of a SOM, considering the "homologous" neuron in the final map as the BMU. The weights of the final neuron will be updated towards the weights of the composing neuron. The difference of the updating performed for each "homologous" neuron in the composing maps depends on the quality measure calculated for each neuron. The higher the quality (or the lower the error) of the neuron of the composing map, the stronger the neuron of the fused map updated towards the weights of that neuron. Either a single or a linear combination of several quality measures can be used to determine the quality. The number of data inputs recognized by each neuron is also taken into account in this quantization of the "best suitability" of one neuron or another for the same position in the final map,. In short, the fusion algorithm will consider "more suitable" weights of a composing neuron to be the weights of the neuron in the final map according to both the number of inputs recognized and the adaptation quality of the neuron. The model, called WeVoS is described in detail in the Algorithm 4.

So, in comparison with previously presented methods (Algorithms 1 and 2), when updating the characteristics of a single neuron, this approach takes into account not only the characteristics of that neuron, but also the topographic ordering of its neighbour. It is expected that this new approach will obtain more maps that are more faithful to the inner structure of the data set from a visualization point of view.

In this case, the computational complexity can be considered comparable to the others. As in the Fusion by Similarity there are several sequential processes in the algorithm that are considered separately. Firstly, the quality of each neuron must be calculated, which in the measures presented depends also on the data set used for

Algorithm 4 Weighted Voting Summarization algorithm

Input: Set of trained topology-preserving maps: $M_1 ... M_n$, training data set: S
Output: A final fused map: M_{fus}

1: Select a training set $S = \langle (x_1, y_1) \dots (x_m, y_m) \rangle$
2: train several networks by using the bagging meta-algorithm : $M_1 ... M_n$
3: **procedure** WeVoS($M_1 ... M_n$)
4: **for all** map $M_i \in M_n$ **do**
5: calculate the quality/error measure chosen for ALL neurons in the map
6: **end for**
▷ These two values are used in Eq. 8
7: calculate an accumulated total of the quality/error for each position $Q(p)$
8: calculate recognition rate for each position $B(p)$.
9: **for all** unit position p in M_i **do**
10: initialize the fused map (M_{fus}) by calculating the centroid (w_c) of the neurons of all maps in that position (p) Eq. 6
11: **end for**
12: **for all** map $M_i \in M_n$ **do**
13: **for all** unit position p in M_i **do**
14: calculate the vote weight (V_{p,M_i}) using Eq. 8.
15: feed the weights vector of neuron w_p into the fused map (M_{fus}) as if it was an input to the network.
The weight of the vote (V_{p,M_i}) is used as the learning rate (α).
The position of that neuron (p) is considered as the position of the BMU (v). ▷ This causes the neuron of the fused map (w_p^*) to approximate the neuron of the composing ensemble ($w_{p,m}$) according to the quality of its adaptation.
16: **end for**
17: **end for**
18: **end procedure**

training. So the computational complexity of that first stage is $O(M \times N \times D)$, where M is the number of maps, N the number of neurons on each map and D the size of the data set. The total of the quality of each position for a neuron is also calculated in this tage (as the accumulation of those values). Then, the initialization of the final map is performed as the calculation of the centroids for each neuron's position, which has a complexity of $O(M \times N)$. Finally, a calculation of the new position for the fused neuron is performed by taking account of the weight of the neurons in the other maps, but only those in the same position, which has a complexity of $O(M \times N)$. In this case there is no need to determine the BMU, only the operation of updating the neighbourhood has to be taken into account.

7 Experiments and Results

The results of the experiments performed to test the novel WeVoS algorithm and to compare it with the other existing fusion methods (Fusion by Distance and fusion by Similarity of Voronoi Polygons) and a proposed combination of both (Fusion by Ordered Similarity) are set out below. As explained before, three well-known data sets were used for these tests (Iris, Wine, and Wisconsin Breast Cancer).

7.1 Overview

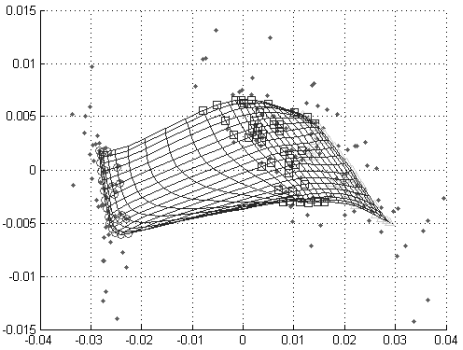
The main objective of this work to obtain a fusion method that preserves the visual representation of a data set. We begin with a visual presentation of the results of the different fusion methods, which will help us to understand the analytical results that are presented later on. The first result presented is the map obtained by a single SOM and the one obtained by each of the other three fusion algorithms. Fig. 1 shows each of the models in a 2D input space. This input space is obtained by calculating the first two Principal Components [Hotelling, 1933] of the Iris data set and projecting all the data over those two axes. The figures depict the grid of the model embedded over the data set, providing an at-a-glance view of the adaptation of the map to the data set. The four fused maps are obtained from the same ensemble of five maps, trained with the same parameters as the single map, which is also presented (see Appendix A), he only difference being the fusion method used to obtain the map.

It can be seen that the single SOM (Fig. 1a) has the regular grid of nodes expected from this algorithm arranged so they can cover as much as the input space as possible. The Fusion by Distance (Fig. 1b) tries to perform the best distribution of the neurons on the input space, without taking their neighbourhood into account. So the result is still a grid, but with an obviously messy layout. Fusion by Similarity (Fig. 1c) therefore yields a structure that is no longer a grid, but rather a set of connected neurons, arranged around the disposition of the data set. Instead of a grid, this fusion method presents a graph as a result. This means that the edges represent a certain similarity to Voronoi polygons, but they are not neighbouring in the 2D output space. As the Fusion by Ordered Similarity (Fig. 1d) follows mainly the Fusion by Distance algorithm, although considering the neurons to be fused in a different way; the results obtained are very similar to those of the Fusion by Distance. A grid structure is present, but the positions of the neurons do not clearly match an ordered grid. Finally, the WeVoS algorithm (Fig. 1e) obtains a clear grid arranged over the data set. It can easily be seen that the grid obtained by WeVoS is spread over the data set more than the single model, thereby obtaining a sparser data display when the data is represented in the output space of the 2D map.

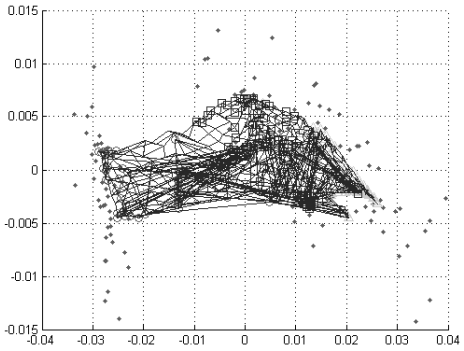
7.2 Stability Test

The experiment consists in training an ensemble of SOMs over a data set several times. Some outliers are also included in the sample in order to include a source of instability in the data set. Each iteration reduces the number of original elements in the data set, while the number of included outliers remains the same. Also, some outliers are included in the sample. The objective is to verify the degree to which the maps obtained by the fusion algorithms are affected by the reduction in the number of samples with which they are trained. For each step in the experiment a 5-fold cross-validation is performed. The data set is a horseshoe shaped 2-D data set, similar to the one used in [Kaski and Lagus, 1996]. It has 570 entries in the first step and 163 in the last one. The Distortion measure obtained in each of the steps for each of the models compared is presented in Fig. 2.

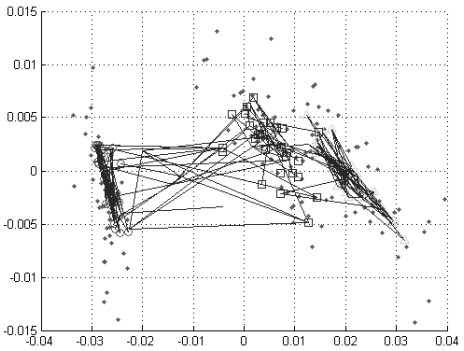
By inspecting Fig. 2, it is obvious that the Distortion of the entire model increases as the number of samples available for training the algorithm decreases. The difference in how this distortion changes with the inclusion of less data in the analysis is a very



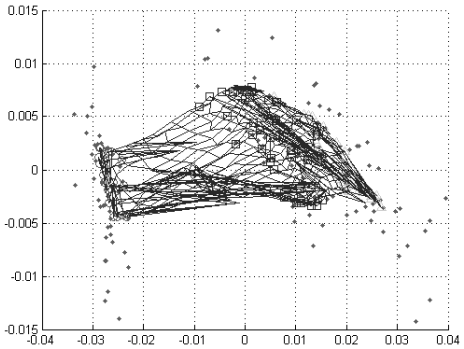
(a) Single SOM



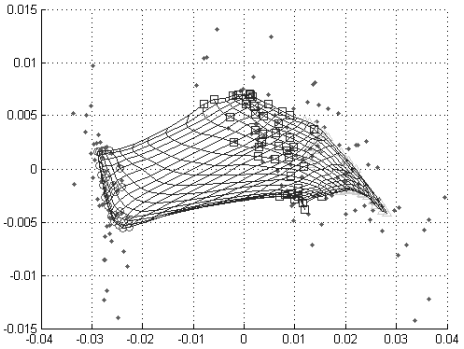
(b) Fusion by Distance



(c) Fusion by Similarity



(d) Fusion by Ordered Similarity



(e) WeVoS

Figure 1: The five models (four ensemble fusion models and the single model) discussed in the paper embedded into a 2D representation of the Iris data set. Both the data set and the grids were projected over the first two Principal Components calculated from the data set. The main point of this visual comparison is to show how the neurons of each map are organized.

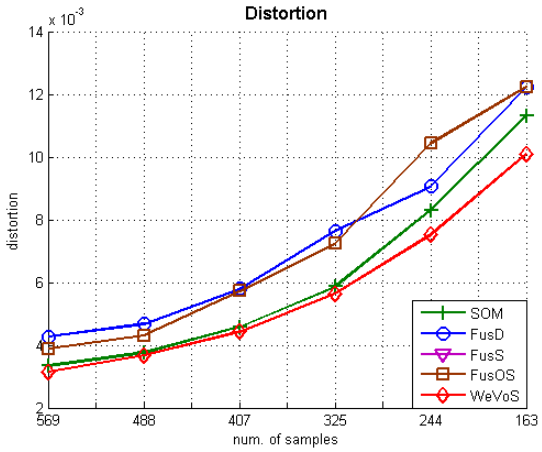


Figure 2: Distortion measured over four different fusion algorithms for a 1-D ensemble of SOMs. Y-axis represents the number of samples taken from the data set where used in the training. The X-axis represents the value of the distortion measure.

interesting feature in the figure. It is easy to see that the WeVoS model has the lowest increase in Distortion. It is followed by the single SOM and by the other two fusion algorithms (Fusion by Distance and Fusion by Ordered Similarity). In Fig. 2 Fusion by Similarity does not appear because its distortion is far higher than 14×10^{-3} and including it into the figure would make difficult to distinguish the difference between the other four fusion methods. This is proof of the additional stability that may be achieved through the use of ensemble meta-algorithms when they are combined with rather unstable models such as the SOM.

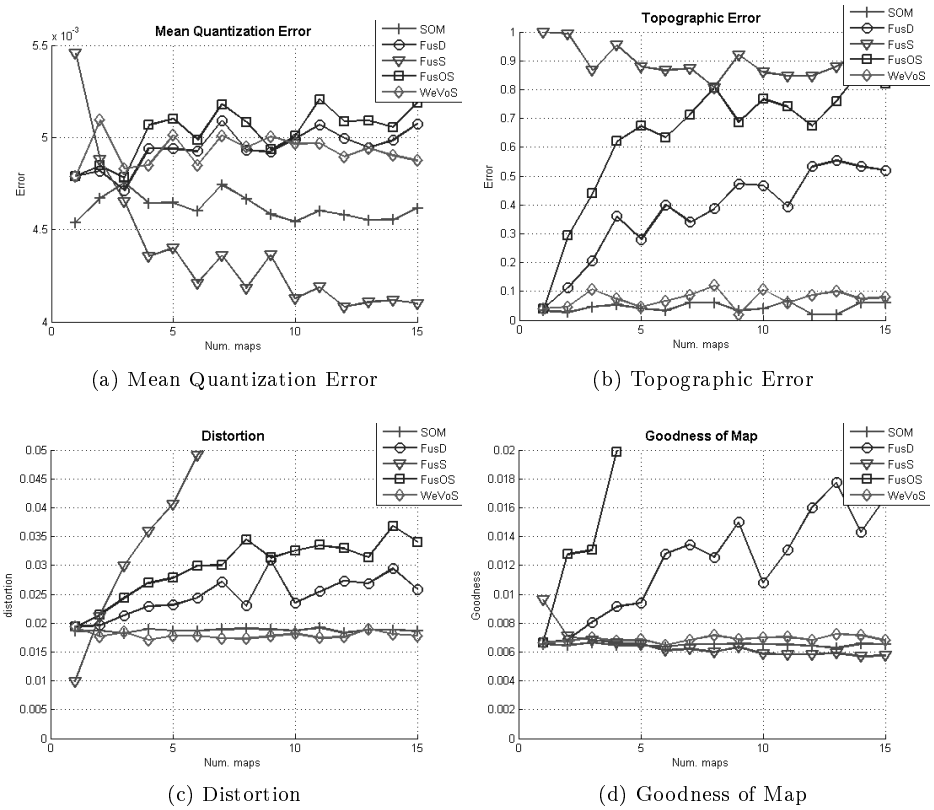
7.3 Experiments Settings

For all the experiments presented in Sections 7.4, 7.6 and 7.5 the settings are the same. A 5-fold cross-validation is used to train and test the maps of the ensembles. The corresponding data set is randomly divided into five different folds of the same size, each of which have the same data distribution for all the classes presented in the ensemble. In each iteration of the cross-validation, four of the five folds are used as the training data set and the remaining fold is used to calculate the measures that are presented. That procedure is repeated five times, each considering a different fold the test fold, so each entry of the data set is used both to train and to test the fusion methods. Finally the mean of the five results (one for each step in cross-validation) obtained for each experiment is calculated to obtain the measure displayed in the figures of Section 7.4, Section 7.6 and Section 7.5. The parameters used in each experiment, both for the training of the SOM and for the fusion of the ensembles, are presented in Table 1 (see Appendix A). All parameters have been chosen empirically.

7.4 Iris Results

Fig. 3 presents the values obtained for each of the four measures described in Section 3 calculated for the four different fusion methods and the single SOM. The charts represent the evolution of the measure of the four measures (Y-axis) with the increasing number of maps used in the ensemble (X-axis). The tests have been performed in this case using the Iris data set. This data set is formed by 150 entries, belonging to three different classes. Each entry is composed of four different dimensions or variables [Asuncion and Newman, 2007] All the measures shown are error measures, so the result is considered to be better the closer it is to 0. Regarding the quantization error (Fig. 3a), it appears that all the ensemble methods obtain higher errors than the single model, except for the Fusion by Similarity. This situation was expected, as none of the fusion algorithms aim to obtain a lower quantization error. On the contrary, one of the ideas behind the algorithms is to reduce the potential overfitting of a single map, whether it is for obtaining better classification accuracy or a better visual representation of the data set. However, a sign of this overfitting would be an overly low quantization error [Ling, 1995]. The case of Fusion by Similarity is different, as the final map will have fewer neurons than the rest of the maps, as neurons with a low activation rate are removed from the final fused map. Therefore, the presence of fewer neurons implies less error, as fewer neurons contribute to the total error obtained by the map. The mean is calculated from the data set input and not from the neuron in the map, so the mean obtained with fewer neurons will be lower than other maps with more neurons.

Concentrating on measures directly related with visualization, it may be observed that, in general, the ensemble fusion methods, with the exception of WeVoS, perform worse than the single map. In Topographic error (Fig. 3b) the three previous fusion algorithms obtain a far higher error than the single SOM. WeVoS obtains a similar error to the SOM, although in most cases it is a bit higher. It is worth noting that this measure is one of the oldest and most simplistic measures for topographic ordering and is not widely regarded as a very trustful one. The results of the next measure -the Distortion (Fig. 3c)- appear to contradict the previous one. In this case we can see that WeVoS consistently presents lower distortion error than the single SOM and the latter a lower error than the other three fusion algorithms. In this case, the advantage of using WeVoS would appear to be worth the added effort of constructing an ensemble. Finally in the case of the Goodness of map (Fig. 3d), which as stated before is a measure that combines quantization and topographic characteristics of the SOM, both WeVoS and the single SOM obtain similar results, although once again it is the single SOM which obtains a slightly lower error. It may be concluded that, although WeVoS can obtain a map with less distortion than the single SOM, the data quantization error means that this measure is higher for WeVoS –which, as may be seen in Fig. 3a, yields a higher quantization error– than for the SOM. This is also the reason why the Fusion by Similarity obtains such a low value for the Goodness of map, even lower than the single SOM. As an overall observation, it is also worth pointing out that WeVoS obtains quite stable results, varying slightly depending on the number of maps; in contrast with the other three fusion algorithms, which yield high variations in their results, and in many cases the error even increases with the number of maps. Obviously the single SOM obtains stable results, as its results are always measured from the training of a single map with the same parameters but with slightly different sets of data inputs.



(a) Mean Quantization Error

(b) Topographic Error

(c) Distortion

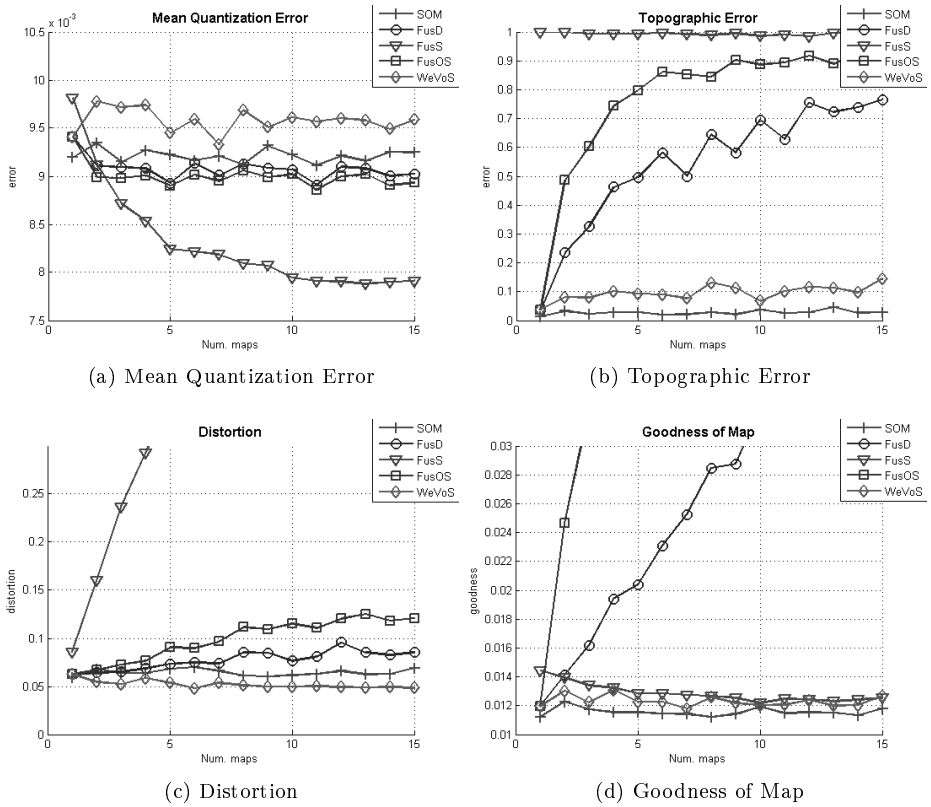
(d) Goodness of Map

Figure 3: Measures for all the four quality measures obtained from the four different fusion algorithms, plus the single SOM. The X-axis represents the number of maps used in the ensemble. The Y-axis represents the value of each measure. The tests were performed using the Iris data set.

7.5 Cancer Results

The following figures (Figs. 5(a)-5(d)) present the values obtained for each of the four error measures described in Section 3 calculated for the four different fusion methods and the single SOM. The charts represent the evolution of the error measures (Y-axis) with the increase of maps used in the ensemble (X-axis). The data set employed for the following tests was the Wisconsin Breast Cancer data set [Asuncion and Newman, 2007]. This data set is formed by 683 entries, belonging to two different classes. Each entry is composed by 9 different dimensions or variables. The results in this case also prove that previously presented results are consistent, as almost the same remarks made for previous experiments are applicable to this one, except for the quantization error.

In this case, the Quantization error (Fig. 4a) results differ greatly from the two previous experiments. The fusion methods, except for WeVoS obtain lower errors than the single SOM. This is probably due to the nature of this particular data set, which is



(a) Mean Quantization Error

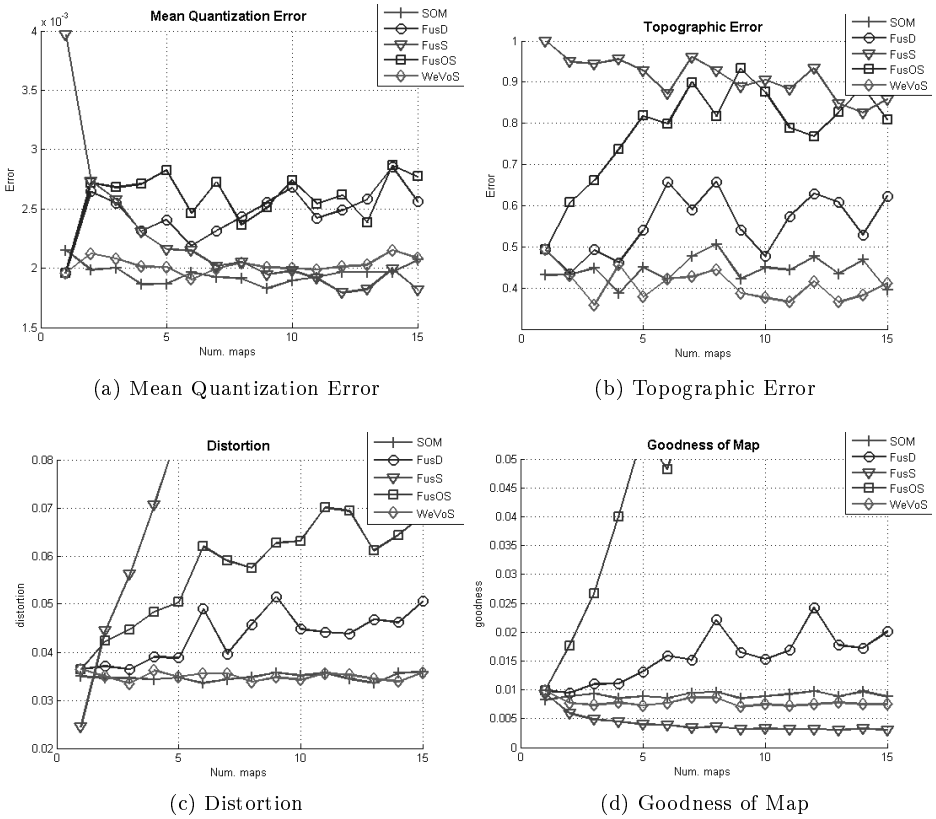
(b) Topographic Error

(c) Distortion

(d) Goodness of Map

Figure 4: Measures for all the four quality measures obtained from the four different fusion methods, plus the single SOM. The X-axis represents the number of maps used in the ensemble. The Y-axis represents the value of each measure. The tests were performed using the Wisconsin Breast Cancer data set.

far less sparse than the previous two; favouring algorithms that disperse their neurons along the input space. The Topographic and Distortion errors (Fig. 4b and Fig. 4c) are similar in this experiment to the first one (using the Iris data set). While the topographic error is lower for the single SOM, WeVoS is the fusion method that obtains the lowest Distortion. The other fusion algorithms obtain higher errors than these two in both measures. Finally, regarding the Goodness of map (Fig. 4d), the performance of the single SOM and WeVoS are again quite similarly, and the single SOM once again obtains a slightly lower error. The difference in this case is that the Fusion by Similarity does not obtain a lower error than the SOM, as in the first case, but a slightly higher error than WeVoS.



(a) Mean Quantization Error

(b) Topographic Error

(c) Distortion

(d) Goodness of Map

Figure 5: Measures for all the four quality measures obtained from the four different fusion algorithms, plus the single SOM. The X-axis represents the number of maps used in the ensemble. The Y-axis represents the value of each measure. The tests were performed using the Wine data set.

7.6 Wine Results

The following figures (Fig. 5) present the values obtained for each of the four measures described in Section 3 calculated for the four different fusion methods and the single SOM. The charts represent the evolution of the error measures (Y-axis) with the increase of the number of maps used in the ensemble (X-axis). The test makes use of the Wine data set [Asuncion and Newman, 2007]. This data set is formed by 178 entries, belonging to three different classes. Each entry is composed by 13 different dimensions or variables.

In this case the Mean Quantization Error (Fig. 5a) yields a slightly different result compared with the results of the Iris or Cancer data set (Fig. 3a). Fusion by Distance and by Ordered Similarity still obtain far higher errors than the rest, but in this case the Fusion by Similarity does not clearly outperform the results of the other fusion methods, obtaining similar results only when the size of the ensemble is of five maps or more (represented in X-axis). In this case too, WeVoS and the single SOM obtain

very similar results, the performance of WeVoS being comparable to the single SOM. With regard to Topographic Error (Fig. 5b), in this case it is the WeVoS algorithm that clearly outperforms the rest, being lower than that of the single SOM and much lower than the rest of the ensemble fusion methods. On the other hand, the WeVoS Distortion measure (Fig. 5c) does not clearly outperform that of the single SOM, although the former still performs much better than the other ensemble fusion methods. Again in this case, contrary to the previous experiments, the Goodness of map (Fig. 5d) is better in the WeVoS algorithm than in the single SOM, and these two results are far better than Fusion by Distance and Fusion by Ordered Similarity. Once again, Fusion by Similarity seems to obtain better results than any other, but as explained earlier, the quantization characteristic of this map will always obtain better results, as it contains less neurons to add to the total error. As the Goodness of maps is a measure involving quantization, Fusion by Similarity obtains very different results compared with the other algorithms.

7.7 Overall Remarks

As seen in the analytical results it can be inferred that the usefulness of the ensemble meta-algorithms and WeVoS fusion in particular depends to a certain degree on the "sparseness" of the data set that is analyzed. By "sparse" data set we mean a data set with a high number of dimensions but a relatively low number of entries. This "sparsity" is intuitively related to the complexity of learning or of extracting correct patterns from that data. Nevertheless, there are several consistent results that are worth noting. The WeVoS-SOM algorithm does not obtain a lower Quantization error than the Single SOM in any experiment, as this model focuses on structure visualization more than on the vector quantization capabilities of the maps they generate. The Topographic error does not give conclusive results, as only in one experiment does it clearly differ from the WeVoS-SOM and the Single SOM. This was to some extent predictable, as this measure is widely considered a very simplistic one.Interestingly, the Distortion measure , which represents the topographical ordering of the maps and is clearly and consistently lower for the WeVoS-SOM than for the single SOM in two out of three experiments. The experiment where this is not true corresponds to the Wine data set (Section 7.5), which is the most complicated data set in the study. This data set includes the highest number of dimensions –13– and relatively few entries –178– to analyse. Compared with the Iris (only 4 dimensions and 150 entries) and the Cancer (9 dimensions, but 683 entries), the nature of the Wine data set makes its analysis and visualization a much more difficult task. The Goodness of Map, a measure combining quantization and topographic preservation measurement, offers rather complementary results to Distortion. For the previously devised ensemble models (Fusion by Distance, Fusion by Similarity) and its combination (Fusion by Ordered Similarity) is generally quite high compared with the WeVoS-SOM and the Single SOM, which have a rather similar low value. In two of the three experiments –Iris and Cancer–, the single SOM obtains better results, due mainly to the quantization capabilities of the WeVoS-SOM, which are not as good. But in the third experiment –Wine– the WeVoS-SOM model obtains consistently better results than the Single SOM, which involves the most complex data set under study. The results in this case are due to the combination of the characteristic low Distortion measure of the WeVoS-SOM and the low Quantization error obtained by WeVoS-SOM in this particular experiment. The described situation suggests that,

as expected, the advantages of using the ensemble meta-algorithm, and particularly the WeVoS-SOM; appear more evident when the data set under study becomes more complex. Ordered Similarity, the other fusion method presented in this study, does not appear to be viable, as in almost every case it obtains higher errors than WeVoS. The main problem with this algorithm seems to be that, by trying to overcome the problems of two previous fusion methods (Fusion by Distance and Fusion by Similarity) it, in turn, ends up being unable to outperform the best aspects of the other methods. So, while it does indeed obtain better results in neighbourhood preservation than Fusion by Similarity (see topographic errors and distortion measures), it never outperforms Fusion by Distance in those measures; and it does not really outperform the best aspect of Fusion by Similarity: the quantization error. As a result, it yields the worst results for the goodness of maps.

8 Conclusions and Future Work

This study presents the capabilities of a novel method for the fusion of an ensemble of Self-Organizing Maps called Weighted Voting superposition (WeVoS), which aims to obtain the most truthful visual representation of a high-dimensional data set in the form of a 2D map. The combination of these algorithms is identified as WeVoS-SOM. Although the method had been devised and was initially presented as a solution to a particular problem, a complete study of its performance using several widely known real data sets has been presented in this study, and analyzed with a variety of analytical quality measures. As may be observed from the results (both the analytical results in Section 7.3 and the visual representations in Figs. 1, and from Fig. 6 to Fig. 10 –in Appendix B–), the main feature of this novel algorithm entitled Weighted Voting Superposition (WeVoS) is a more reliable visual representation of the data set due to enhancement of the topology preservation feature, which is one of the most important for the original model that is to be improved. This characteristic is measured by Distortion, and to an extent by the Goodness of map measures. The WeVoS-SOM model proves its real usefulness when representing a more complex data set rather than a simpler one, as any additional complexity entailed in the calculation of an ensemble of maps, is compensated by improvements in the visualization results. As an added advantage, the algorithm has a lower computational complexity than other previously presented ensemble fusion algorithms. While previous models performed their calculations on the entire extension of the maps that were to be fused, the WeVoS-SOM performs calculations only on "homologous" neurons and their neighbourhoods in the respective maps of the ensemble; resulting in a less extensive algorithm in computational terms. A novel modification of previous algorithms to carry out fusion of Self-Organizing Maps ensembles, called Fusion by Ordered Similarity, has also been presented, analysed and compared with two earlier models on which it is based and WeVoS. Fusion by Ordered Similarity improves the results of both models on which it is based with respect to their respective worst aspects; but fails to outperform either models with respect to their best aspects. Future work will involve upgrading the Weighted Voting Superposition algorithm to improve its results by testing and comparing several different variants. These include the use of boosting meta-algorithms that are adapted to this kind of model or the application of the meta-algorithm presented in this study to other topology preserving algorithms, in order to identify and to test improvements in a wider range of real-world data sets and problems. Also the application of the WeVoS algorithm

to some other specific algorithms of the topology preserving family of models will be explored.

Acknowledgements This research has been partially supported through projects of the Junta of Castilla and León BU006A08; Business intelligent for production within the framework of the technological Institute of Castilla y León and investment and services agency (ADE); project of the Spanish Ministry of Education and Innovation CIT-020000-2008-2. The authors would also like to thank the vehicle interior manufacturer, Grupo Antolin Ingenieria S.A., within the framework of the project MAGNO2008 - 1028.- CENIT Project funded by the Spanish Ministry.

References

A. Asuncion and D. Newman. UCI machine learning repository, 2007. URL `http://www.ics.uci.edu/~mlearn/MLRepository.html`.

F. Aurenhammer and R. Klein. Voronoi diagrams. In *Handbook of Computational Geometry*, number 5, pages 201–290. North-Holland, Amsterdam, Netherlands, 2000.

B. Baruque, E. Corchado, and H. Yin. ViSOM ensembles for visualization and classification. In *9th International Work-Conference on Artificial Neural Networks (IWANN'07)*, volume 4507/2007 of *Lecture Notes in Computer Science*, pages 235–243, San Sebastian, Spain, 2007. Springer Berlin / Heidelberg. doi: 10.1007/978-3-540-73007-1.

B. Baruque, E. Corchado, J. Rovira, and J. Gonzalez. Application of topology preserving ensembles for sensory assessment in the food industry. In *Intelligent Data Engineering and Automated Learning (IDEAL 2008)*, volume 5326/2008 of *Lecture Notes in Computer Science*, pages 491–497. Springer Berlin / Heidelberg, 2008. doi: 10.1007/978-3-540-88906-9.

J. Blackmore and R. Miikkulainen. Incremental grid growing: Encoding high-dimensional structure into a two-dimensional feature map. In *Proceedings of the IEEE International Conference on Neural Networks*, volume 1, pages 450–455, 1993. ISBN 0-7803-0999-5.

G. Brassard and P. Bratley. *Fundamentals of Algorithmics*. Prentice Hall (US ed edition), August 1995. ISBN 0133350681.

L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

E. Corchado, B. Baruque, and H. Yin. Boosting unsupervised competitive learning ensembles. In *Artificial Neural Networks (ICANN 2007)*, volume 4668/2007 of *Lecture Notes in Computer Science*, pages 339–348, Porto, Portugal, 2007. Springer Berlin / Heidelberg. doi: 10.1007/978-3-540-74690-4.

Y. Freund and R. E. Schapire. Experiments with a new boosting algorithm. In *International Conference on Machine Learning*, pages 148–156, 1996.

A. Georgakis, H. Li, and M. Gordan. An ensemble of SOM networks for document organization and retrieval. In *International Conference on Adaptive Knowledge Representation and Reasoning (AKRR'05)*, pages 6–141, Espoo, Finland, 2005.

N. Gianniotis and P. Tino. Visualization of tree-structured data through generative topographic mapping. *IEEE Transactions on Neural Networks*, 19(8):1468–1493, 2008. ISSN 1045-9227.

T. Heskes. Balancing between bagging and bumping. In M. C. Mozer, M. I. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems*, volume 9, pages 466–472, Denver, Colorado, USA, 1997. MIT Press.

H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Education Psychology*, 24:417–444, 1933.

U. Johansson, T. Lofstrom, and L. Niklasson. Obtaining accurate neural network ensembles. In M. Mohammadian, editor, *International Conference on Computational Intelligence for Modelling, Control & Automation Jointly with International Conference on Intelligent Agents, Web Technologies & Internet Commerce, Vol 2, Proceedings*, pages 103–108, 2006.

S. Kaski and K. Lagus. Comparing self-organizing maps. In C. v. d. Malsburg, W. v. Seelen, J. C. Vorbruggen, and B. Sendhoff, editors, *ICANN 96: Proceedings of the 1996 International Conference on Artificial Neural Networks*, volume 1112 of *Lecture Notes In Computer Science*, pages 809–814, London, UK, 1996. Springer-Verlag. ISBN 3-540-61510-5.

K. Kiviluoto. Topology preservation in self-organizing maps. In *IEEE International Conference on Neural Networks (ICNN'96)*, volume 1, pages 294–299, 1996.

T. Kohonen. An introduction to neural computing. *Neural Networks*, 1(1):3–16, 1988. ISSN 0893-6080.

T. Kohonen. *Self-Organizing Maps*. Springer, Berlin, Germany, 1995.

T. Kohonen, P. Lehtio, J. Rovamo, J. Hyvarinen, K. Bry, and L. Vainio. A principle of neural associative memory. *Neuroscience*, 2(6):1065–1076, 1977.

T. Kohonen, E. Oja, O. Simula, A. Visa, and J. Kangas. Engineering applications of the self-organizing map. In *Proceedings of the IEEE*, volume 84, pages 1358–1384, 1996.

L. I. Kuncheva. *Combining Pattern Classifiers: Methods and Algorithms*. Wiley-Interscience, 2004.

L. I. Kuncheva, M. Skurichina, and R. P. W. Duin. An experimental study on diversity for bagging and boosting with linear classifiers. *Information Fusion*, 3(4):245–258, 2002.

J. Lampinen and E. Oja. Clustering properties of hierarchical self-organizing maps. *Journal of Mathematical Imaging and Vision*, 2:261–272, 1992.

C. X. Ling. Overfitting and generalization in learning discrete patterns. *Neurocomputing*, 8(3):341–347, 8 1995.

J. C. Patra, E. L. Ang, P. K. Meher, and Q. Zhen. A new SOM-based visualization technique for dna microarray data. In *International Joint Conference on Neural Networks (IJCNN'06)*, pages 4429–4434, 2006. doi: 10.1109/IJCNN.2006.247044.

L. Petrakieva and C. Fyfe. Bagging and bumping self organising maps. *Computing and Information Systems Journal*, 2003. ISSN 1352-9404.

D. Polani. Measures for the organization of self-organizing maps. In U. Seiffert and L. C. Jain, editors, *Self-organizing Neural Networks: Recent Advances and Applications (Studies in Fuzziness and Soft Computing)*, volume 16, pages 13–44. Physica-Verlag, Heidelberg, 2003.

G. Pozlbauer. Survey and comparison of quality measures for self-organizing maps. In J. P. Rauber, G. Polzlbauer, and Andreas, editors, *5th Workshop on Data Analysis (WDA'04)*, pages 67—82. Elfa Academic Press, 2004.

D. Ruta and B. Gabrys. Classifier selection for majority voting. *Information Fusion*, 6(1):63–81, 2005.

C. Saavedra, R. Salas, S. Moreno, and H. Allende. Fusion of self organizing maps. In *9th International Work-Conference on Artificial Neural Networks (IWANN 2007)*, Lecture Notes in Computer Science, pages 227–234. Springer Berlin / Heidelberg, 2007. doi: 10.1007/978-3-540-73007-1.

H. Schwenk and Y. Bengio. Boosting neural networks. *Neural Computation*, 12(8): 1869–1887, 2000.

J. Vesanto, M. Sulkava, and J. Hollmen. On the decomposition of the self-organizing map distortion measure. In *Proceedings of the Workshop on Self-Organizing Maps (WSOM'03)*, pages 11–16, 2003.

G. Zhang, B. E. Patuwo, and M. Y. Hu. Forecasting with artificial neural networks:: The state of the art. *International Journal of Forecasting*, 14(1):35–62, 3/1 1998.

A Experiments Parameters

Experiments performed, with their corresponding data sets, algorithms and training parameters and the figures displaying the results for each experiment.

Data Set	SOM Parameters		Algorithm	Fusion Parameters		Figures
Iris	size	15x10	SOM	num. inputs	105	1a, 3, 8a
	iterations	1500	Fusion Distance	num. inputs	105	1b, 3, 8b
	learning rate	0.1	Fusion Similarity	num. inputs	105	1c, 3
	neighb. func.	c:1		σ_u	0.02	
		σ:10		σ_f	0.7	
				σ_c	0.8	
			Fusion Ordered Similarity	num. inputs	105	1d, 3, 8c
			WeVoS	num. inputs	105	1e, 3, 8d
				quality measure	goodness of map	

Table 2: Parameters for the experiments with the Iris data set

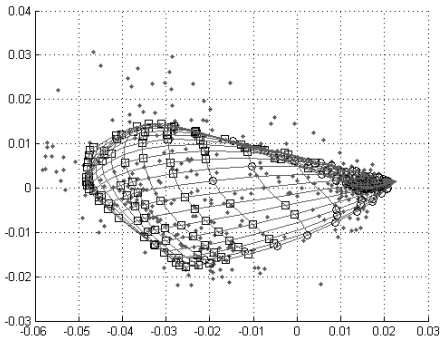
Data Set	SOM Parameters		Algorithm	Fusion Parameters		Figures
Cancer	size	25x20	SOM	num. inputs	525	4, 6a, 9a
	iterations	5000	Fusion Distance	num. inputs	525	4, 6b, 9b
	learning rate	0.1	Fusion Similarity	num. inputs	525	4, 6c
	neighb. func.	c_1:2	Fusion Ordered Similarity	num. inputs	525	4, 6d, 9c
		σ_1:18		σ_u	0.02	
		c_2:1		σ_f	0.7	
		σ_2:20		σ_c	0.8	
			WeVoS	num. inputs	525	4, 6e, 9d
				quality measure	goodness of map	

Table 4: Parameters for the experiments with the Cancer data set

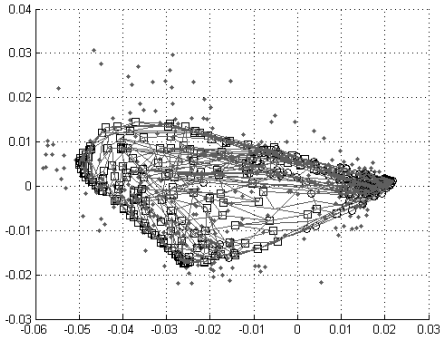
Data Set	SOM Parameters		Algorithm	Fusion Parameters		Figures
Wine	size	25x20	SOM	num. inputs	125	5, 7a, 10a
	iterations	2000	Fusion Distance	num. inputs	125	5, 7b, 10b
	learning rate	0.1	Fusion Similarity	num. inputs	125	5, 7c
	neighb. func.	c:1		σ_u	0.015	
		σ:5		σ_f	0.7	
				σ_c	0.8	
			Fusion Ordered Similarity	num. inputs	125	5, 7d, 10c
			WeVoS	num. inputs	125	5, 7e, 10d
				quality measure	goodness of map	

Table 6: Parameters for the experiments with the Wine data set

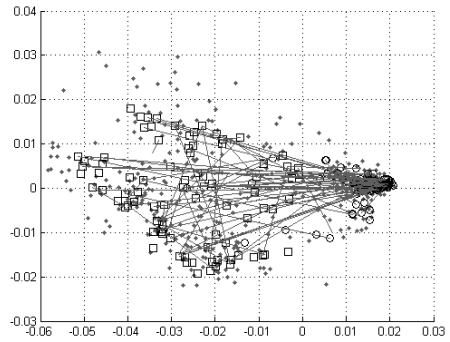
B Additional Figures



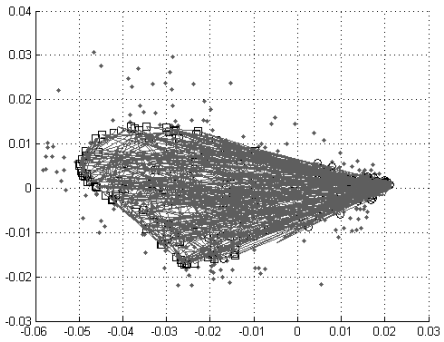
(a) Single SOM



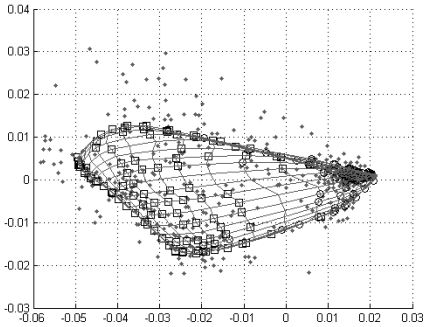
(b) Fusion by Distance



(c) Fusion by Similarity

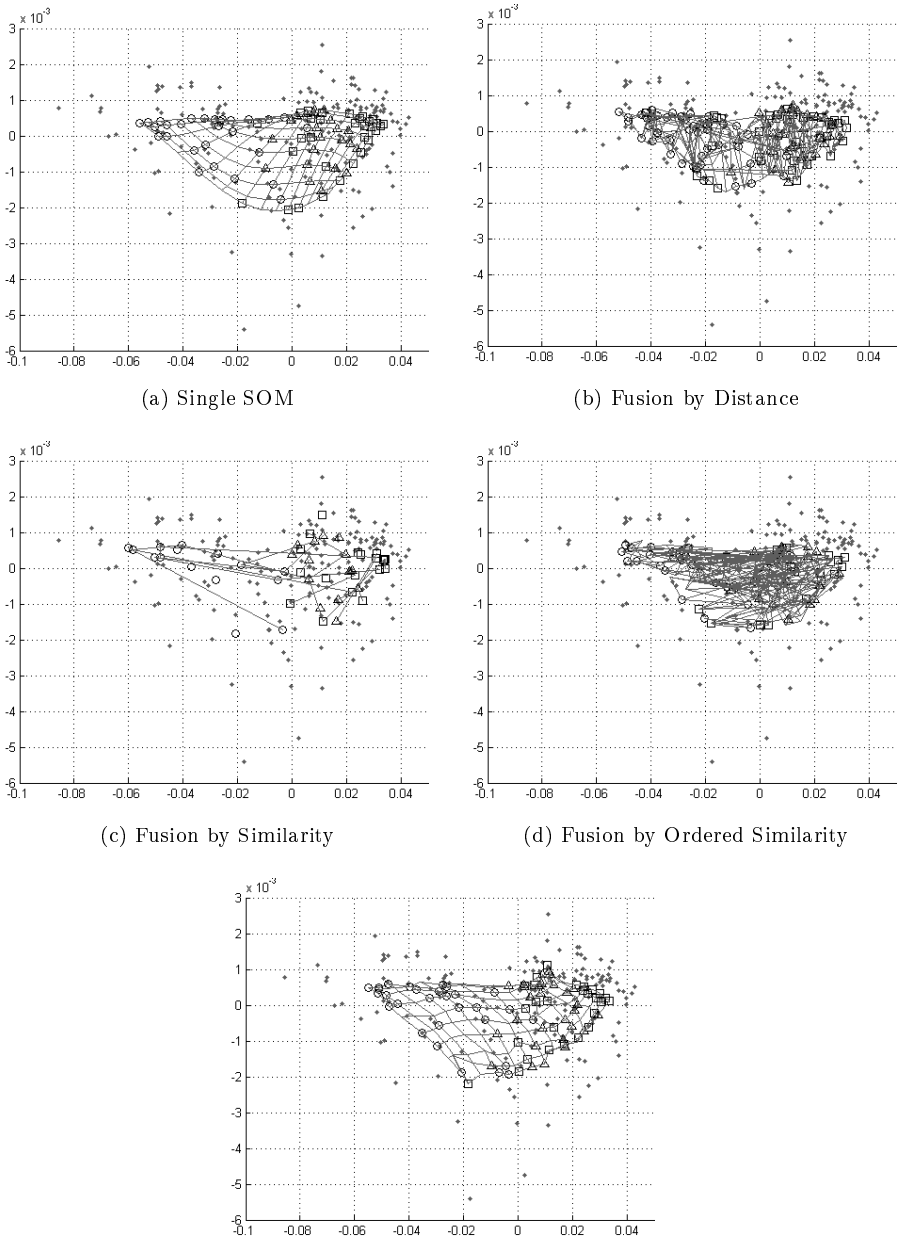


(d) Fusion by Ordered Similarity



(e) WeVoS

Figure 6: The five models (four ensemble fusion models and the single model) discussed in the paper embedded into a 2D representation of the Wisconsin Breast Cancer data set. Both the data set and the grids were projected over the first two Principal Components calculated from the data set.



(a) Single SOM

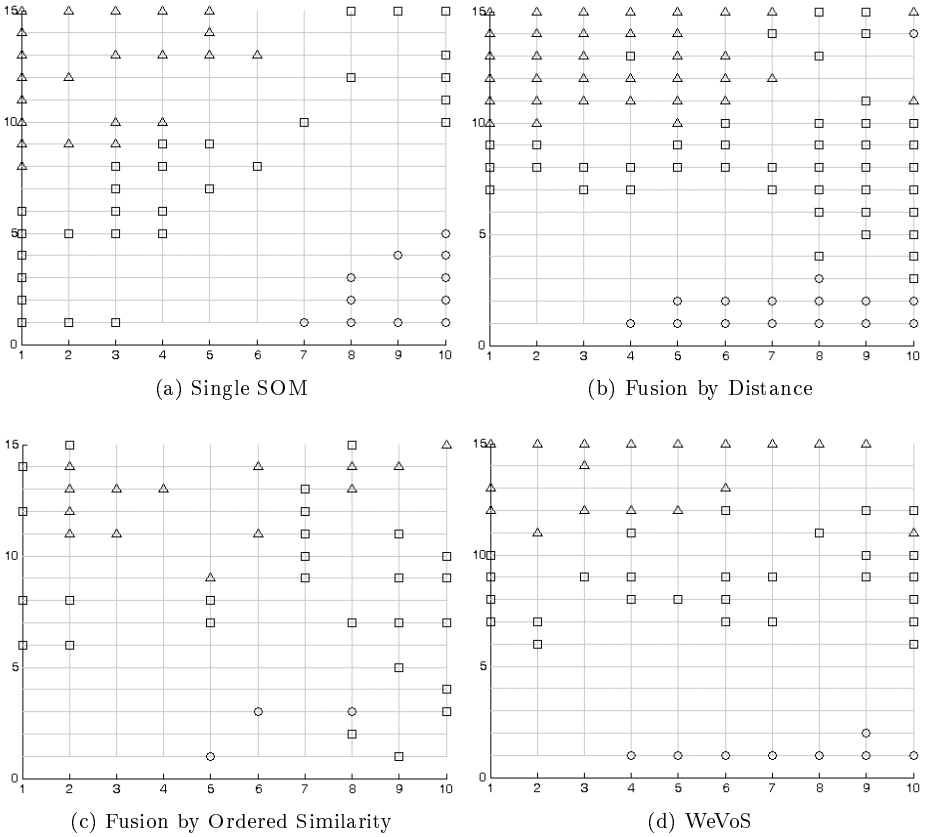
(b) Fusion by Distance

(c) Fusion by Similarity

(d) Fusion by Ordered Similarity

(e) WeVoS

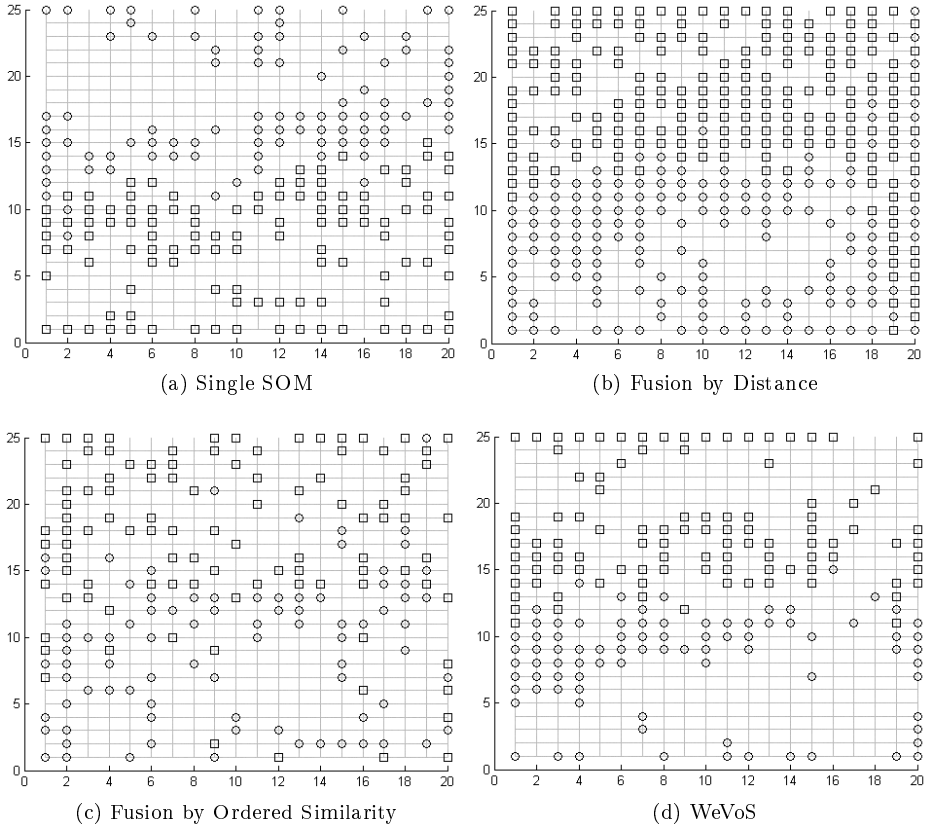
Figure 7: The five models (four ensemble fusion models and the single model) discussed in the paper embedded into a 2D representation of the Wine data set. Both the data set and the grids were projected over the first two Principal Components calculated from the data set.



(a) Single SOM (b) Fusion by Distance

(c) Fusion by Ordered Similarity (d) WeVoS

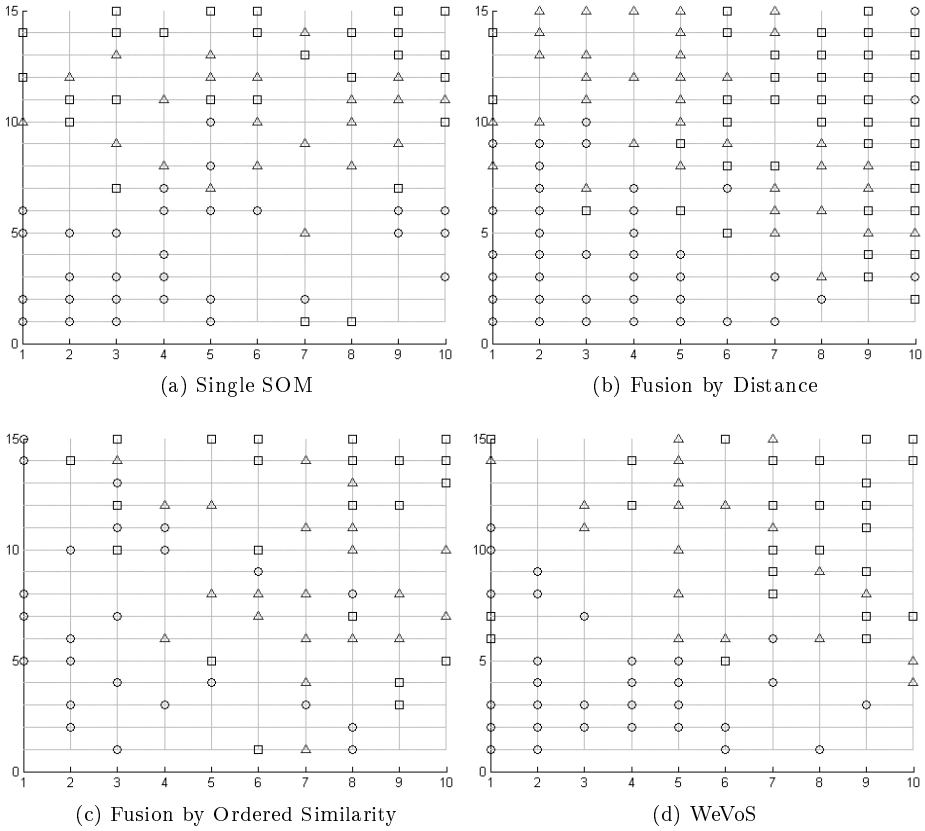
Figure 8: Iris data set represented over the 2D map obtained by the single SOM and the three different fusion algorithms under study.



(a) Single SOM (b) Fusion by Distance

(c) Fusion by Ordered Similarity (d) WeVoS

Figure 9: Cancer data set represented over the 2D map obtained by the single SOM and the three different fusion algorithms under study.



(a) Single SOM (b) Fusion by Distance

(c) Fusion by Ordered Similarity (d) WeVoS

Figure 10: Wine data set represented over the 2D map obtained by the single SOM and the three different fusion algorithms under study.