Information Sciences 280 (2014) 307–321

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

WeVoS scale invariant map

Bruno Baruque [a,*], Emilio Corchado [b]

[a] Department of Civil Engineering, University of Burgos, Spain
[b] Departamento de Informática y Automática, University of Salamanca, Spain

ARTICLE INFO

Article history:
Received 9 July 2011
Received in revised form 25 April 2014
Accepted 3 May 2014
Available online 15 May 2014

Keywords:
Artificial neural network
Self-organization
Ensemble training
Information fusion
Radial dataset

ABSTRACT

A novel method for improving the training of some topology preserving algorithms characterized by its scale invariant mapping is presented and analyzed in this study. It is called Weighted Voting Superposition (WeVoS), and in this research is applied to the Scale Invariant Feature Map (SIM) and the Maximum Likelihood Hebbian Learning Scale Invariant Map (Max-SIM) providing two new versions, the WeVoS–SIM and the WeVoS–Max-SIM. The method is based on the training of an ensemble of networks and the combination of them to obtain a single one, including the best features of each one of the networks in the ensemble. To accomplish this combination, a weighted voting process takes place between the units of the maps in the ensemble in order to determine the characteristics of the units of the resulting map. To provide a complete comparative study of these new models, they are compared with their original models, the SIM and Max-SIM and also to probably the best known topology preserving model: the Self-Organizing Map. The models are tested under the frame of two ad hoc artificial data sets and a real-world one, characterized for having an internal radial distribution. Four different quality measures have been applied for each model in order to present a complete study of their capabilities. The results obtained confirm that the novel models presented in this study based on the application of WeVoS can outperform the classic models in terms of organization of the presented information.

1. Introduction

Among the great variety of tools for multi-dimensional data visualization, several of the most widely used are those belonging to the family of the topology preserving maps [19]. Probably the best known among those algorithms is the Self-Organizing Map (SOM) [20]. Two interesting models that will be mainly discussed along the present study are the Scale Invariant Map (SIM) [12] and the Maximum Likelihood Scale Invariant Map (Max-SIM) [7]. Both are designed to perform their best with radial datasets; that is, which have their samples distributed in a radial way, by being situated at a similar multi-dimensional distance from a given central position, such as a circumference or sphere. The main difference between this mapping and the first mentioned model is that this mapping is scale invariant. When topology preserving maps are trained, they approximate a Voronoi tessellation of the input space [21]. The scale invariant map, however, creates a mapping where each neuron captures a “pie slice” of the data according to the angular distribution of the input data. The present research is manly focused on the enhancement of some of the characteristics of these two models.

[*] Corresponding author. Tel.: +34 947 25 9513; fax: +34 947 25 8910.
E-mail addresses: bbaruque@ubu.es (B. Baruque), escorchado@usal.es (E. Corchado).

http://dx.doi.org/10.1016/j.ins.2014.05.005

The analysis of the scale-invariant features of a dataset are a much more useful and common task than could be expected at first thought. One of the areas where these features are more decisive is the image analysis. In order to perform a reliable matching between different views of an object or scene, it is very important that the image can be analyzed without having to take into account the distance from which the image was acquired, but rather it inherent characteristics and organization Bronstein and Kokkinos [6]. For example, if a system is designed to recognize the fingers of a human hand that a given person is showing to a camera (i.e. for tracking gestures in novel human–machine interfaces); a desired requirement could easily be that this system should work regardless the exact distance from where the user is standing. This is also true when designing navigation algorithms for mobile robots: it is important to distinguish the surroundings of the robot independently from the distance from where the robot is located at a given time, so it greatly simplifies these algorithms Zhang [33]. A neural network algorithm capable of analyzing these kind of data features has lots of applications in these fields.

A general problem that affects the majority of the neural network algorithms is that they are rather unstable, potentially yielding quite different results in spite being trained with the same data set and parameters [16]. This effect comes from the fact that, in practical applications, this kind of algorithms must be initialized in a random way and data must be presented also in random order; to not induce a biased training and benefit the same data samples in different runs of the algorithm. So, although these algorithms can be stable in a probabilistic sense, they may have quite different results when repeating the training/test process. This specific behavior can be observed when the dataset used to train is composed by a low number of data samples or has some regions where data samples are sparse. In those cases, the effect of removing or adding a sample or presenting it in a different moment of an algorithm's training procedure is magnified by the fact that there are few samples in the vicinity of that one to compare to. This is especially true in algorithms that use the Voronoi regions for their analysis. This well-known problem is the one address in the present work.

The use of ensembles is one of the most spread techniques for increasing the stability and performance of an analysis model [4,17]. This meta-algorithm consists on training several slightly different models over the same data set and relying on their combined results, rather than in the results of a single model. A lot of previous studies and algorithms have been proposed for this task, ranging from the simple ones – Bagging [4], Random Forest [5] – to more complex ones – Adaboost [11] Stacking Generalization [31]. All these techniques have the objective of increasing the diversity of the general classifier, that is the ensemble, in order to improve the generalization capabilities of the simple models and therefore, being able to better classify new samples [26].

In order to easily work with the results of an ensemble, one of the most spread techniques is the calculation of a final summary of all the components of the ensemble. This summary – or fusion – usually has the same structure as that of the single models composing the ensemble, but it is expected to outperform the models that are trained individually. The main idea developed in this study is to profit from several different trainings of the same algorithm by fusing them to calculate a final model that includes the best performing aspects of each of them.

There are many combination algorithms in classification ensembles literature; but few of them, up to the knowledge of the authors, are directly applicable to topology preserving algorithms. Several algorithms for topographic maps summarization have been previously proposed [24,13,28], although there are some characteristics of the topology preserving models that have not been taken into account. This study presents a novel algorithm to overcome the main pitfalls of two of them. The first one, called fusion based on Voronoi Polygons Similarity [28] is characterized by determining the units of different maps that are suitable to be fused by comparing the input space covered by each unit. That is, comparing what are called the Voronoi polygons of each unit [30]. This summary performs very well when recognizing and adapting its structure in the input space of the data set, but it is not really able to represent that same data set in a 2-D map; thus being of no use for dimensional reduction and visualization tasks. The second one, called fusion based on Euclidean Distance [13] uses the classic Euclidean Distance between units to determine their suitability to be fused, instead. In this case the model is able to represent the data set as a 2-D map, but the way it computes the neurons to fuse is an approximate one, so is prone to errors in the topology preservation of the map.

In this research we present and analyze a new fusion algorithm called Weighted Voting Superposition (WeVoS) [1] applied for the first time to the SIM and Max-SIM. The study reports the application of these algorithms under two artificial data sets, created accordingly to the main characteristic of the models under study.

The rest of the study is organized as follows. The topology preserving models used in the study are introduced in Section 2. Section 3 describes the measurements used to analyze the results of the tests performed. The novel fusion algorithm applied to the scale invariant map is described in detail in Section 4. Section 5 presents the experiments performed to test the model's qualities. Finally, Section 6 contains some conclusions and lines of future work.

2. Topology preserving algorithms

The main target of the family of topology preserving maps [22] is to produce low dimensional representations of high dimensional data sets maintaining the topological features of the input space.

2.1. Self-Organizing Maps

The best known technique among them is the Self-Organizing Map (SOM) algorithm [20,9]. The basic SOM consists of m units located on a regular low-dimensional grid, U, usually 1- or 2-dimensional. The learning process used by this kind of network 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 or sets of samples in a specific domain of the input space [21].

The update of the neuron weights depending on their neighborhood in SOM is represented by:

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

where x denotes the network input, w_k the characteristics vector of each neuron; α, is the learning rate of the algorithm; and $\eta(v,k,t)$ is the neighborhood function, in which v represents the position of the winning neuron (BMU) in the lattice, and k the positions of the neurons in its neighborhood. The most common function is a Gaussian function centered on the position of the BMU; although other functions, such as the difference of Gaussians, are also widely used.

2.2. Scale-invariant maps

Another example of a topographic mapping algorithm is the Scale Invariant Map (SIM) [12]. The SIM uses a simple network which uses negative feedback activation and simple Hebbian learning to self-organize. By adding neighborhood relations to its learning rule, it creates a feature map which has the property of retaining the angular properties of the input data, i.e. vectors of similar directions are classified similarly regardless of their magnitude. A SIM is also a regular array of nodes arranged on a lattice. Competitive learning and a neighborhood function are used in a similar way as with the SOM. The input data (x) is fed forward to the outputs y_i in the usual way. After selection of a winner, the winner, c, is deemed to be firing ($y_c = 1$) and all other outputs are suppressed ($y_i = 0$). The winner's activation is then fed back through its weights and this is subtracted from the inputs to calculate the error or residual e as shown in Eq. (2):

$$e = x - W_c y_c, \ (y_c = 1) \quad (2)$$

The Maximum Likelihood Scale Invariant Map (Max-SIM) [7] is an extension of the SIM based on the application of the Maximum Likelihood Hebbian Learning (MLHL) [8]. The main difference with the SIM is that the MLHL is used to update the weights of all nodes in the neighborhood of the winner, once this has been updated as in Eq. (2). This can be expressed as in Eq. (3):

$$\triangle W_i = h_{ci}\eta(x - W_c)^{p-1}, \ \forall i \in N_c \quad (3)$$

By giving different values to p, the learning rule is optimal for different probability density functions of the residuals. h_{ci} is the neighborhood function as in the case of the SOM and N_c is the number of output neurons. Finally, η represents the learning rate of the learning rule. During the training of the SIM or the Max-SIM, the weights of the winning node are fed back as inhibition at the inputs, and then in the case of the Max-SIM, MLHL learning is used to update the weights of all nodes in the neighborhood of the winner as explained above. The Max-SIM implements therefore, a general family of rules in which the original SIM is included. It can be considered that the updating of neurons used in a SIM is the same used for the Max-SIM, but with the p parameter having the invariant value of 2.

3. Quality measures for topology preserving models

Several quality measures have been proposed in literature to study the reliability of the results displayed by topology preserving models in representing the data set that have been trained with [3,25,27]. 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 four measures used in this study are briefly presented in the following paragraphs.

Classification error. Using its inherent pattern matching characteristics, the topology preserving maps in general terms can be used for classification tasks. Intuitively, the samples activating the same neuron of the network are very likely to belong to the same class. When a new sample is presented to the network, the sample can be classified in the same class which the majority of samples activating the same neuron belong to. A consistent behavior when classifying samples points to a correctly trained map. So, although this is not the main function of this kind of networks, the measure of how many samples are wrongly classified has been used, to an extent, to asses the quality of the final map in numerous previous studies.

Quantization error. This error is related to all forms of vector quantization and clustering algorithms. Thus, this measure completely disregards map topology and alignment. It is computed by determining the average distance of the test data set entries to the cluster centroids by which they are represented. In case of the SOM, the cluster centroids are the map units.

Distortion [23,29]. When using a constant radius for the neighborhood 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 research.

Goodness of Map [18]. 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 Best Matching Unit (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 neighbors in the map. In other words, it is a mixture of the classic Mean Quantization Error measure and a topographic ordering measure.

4. Weighted Voting Superposition (WeVoS)

The WeVoS [1] is an algorithm that tries to overcome the problems outlined in Section 1 for the previously presented fusion algorithms for topology-preserving models [13,28] and it is applied for the first time to two scale-invariant maps to analyze its response when dealing with radial data sets [12]. The principal idea is to obtain 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 measure 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. (4):

$$V_{p,m} = \frac{\sum b_{p,m}}{\sum_{i=1}^{M} b_{p,i}} \cdot \frac{q_{p,m}}{\sum_{i=1}^{M} q_{p,i}} \tag{4}$$

where $V_{p,m}$ is the weight of the vote for the unit included in map m of the ensemble, in its 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 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.

b is a binary vector of the same length as data samples are in the dataset; that is used to store the samples recognized by a single unit. So, the first term of the equation accounts for the recognition rate of a unit in a map in relation with the total of recognition of the units in the same position on all the maps of the ensemble. Following the same reasoning, the second term accounts for the quality of a unit in a map in relation with the overall quality obtained by the units in the same position in all maps.

The main objective of the WeVoS algorithm is to generate a final map processed unit by unit. Instead of trying to obtain the best position for the units of a single map trained on a single data set, it aims to generate several maps over different parts of the data set. It then obtains a final summarized map, calculating by consensus which is the best set of characteristics vector for each unit position in the map. In order to perform this calculation, the meta-algorithm must first obtain the *quality* of every unit that composes each map [27], so that it can relay in some kind of informed resolution for the fusion of neurons.

A schematic diagram of this situation is depicted in Fig. 1, and a detailed description of the algorithm can be found in Algorithm 1.

Algorithm 1. Weighted Voting Summarization algorithm

```
Input: Set of trained topology-preserving maps: $M_1 \ldots M_n$, training data set: $S$
Output: A final fused map: $M_{fus}$
Select a training set $S = \langle (x_1, y_1) \ldots (x_m, y_m) \rangle$
train several networks by using the bagging meta-algorithm: $M_1 \ldots M_n$
procedure WeVoS $M_1 \ldots M_n$
  for all map $M_i \in M_n$ do
    calculate quality/error measure for ALL neurons in the map
  end for
  calculate an accumulated total of the quality/error for each position $Q(p)$
  calculate recognition rate for each position $B(p)$.
              ▷ These two values are used in Eq. (4).
   for all unit position $p$ in $M_i$ do
      initialize the fused map ($M_{fus}$): calculate the centroid ($w_c$) of the neurons of all maps in position ($p$)
   end for
   for all map $M_i \in M_n$ do
      for all unit position $p$ in $M_i$ do
         calculate the vote weight ($V_{p,M_i}$) using Eq. (4).
         feed the weights vector of neuron $w_p$ into the fused map ($M_{fus}$)
      end for
   end for
         ▷ The weight of the vote ($V_{p,M_i}$) is used as the learning rate ($\alpha$).
       ▷ The position of the neuron ($p$) is considered as the position of the BMU ($v$).
       ▷ The neuron of the fused map $\left(w_p^*\right)$ approximates the neuron of the composing ensemble ($w_{p,m}$) according to the
    quality of its adaptation.
end procedure
```

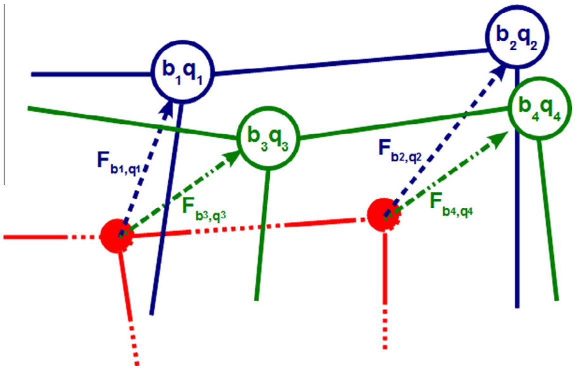


Fig. 1. Schematic diagram of the weighted voting in WeVoS in a 2-D map.

Briefly, the WeVoS algorithm (see Algorithm 1) functions in the following way: first of all an ensemble of maps is trained. Then, the chosen quality/error measure is calculated for each of the neurons in all the ensemble maps. The fused map is initialized by calculating the centroids of the neurons in the same position of all the maps, that is, by calculating the superposition of the ensemble. For each of the neurons in the fused map, the average neuron quality as well as the number of total samples recognized in that position for the ensemble maps are calculated. The weight of the vote for each neuron can be calculated with this information by using Eq. (4). To modify the position of the neuron in the fused map, the weights of each of the neurons in that position are fed to the final map. The learning rate in each case will be the weight of the vote for that neuron.

The intuitive idea behind this process is to minimize the effect of sparse datasets or outlier data samples. Let us say there is an outlier sample on the analyzed dataset. If a single map is trained over the dataset, it will probably be affected by this situation and will include a "defect" on its lattice: a badly situated unit. This situation would be magnified if the dataset is quite sparse, because there are few "normal" samples on the vicinity of the outlier to compare. If several maps are calculated on the dataset using a bagging algorithm – which involves sub-sampling the original dataset – it is quite probable that the outlier sample will not be included in a majority of the data sub-sets; so the majority of the maps will not include the defect on their units. Finally, by applying the WeVoS algorithm, the set of maps is merged into a single one that adjusts the position of each of its units by measuring the level of adaptation to the data of the same corresponding unit in all maps. In this case is possible to detect a defect on the lattice by comparing the adaptation of the same unit along the set of maps. When this is detected, the influence of the "defective" unit on the final map will be minimized. That way, the effect of outlier or sparse samples can be detected and minimized, as they only affect units in a minority of the maps.

The effect of this whole process is a sort of "smoother" lattice expanding over the dataset. This can be clearly seen on the results shown in Fig. 6. The smoother lattice means a better overall adaptation over the dataset while avoiding the overfitting, thus capturing better its main features and leaving out small defects.

5. Experiments and results

In order to test the novel two models presented in this research, two artificial data sets with a radial layout have been created [12]. The first one consists in a simple 2-dimension circular set, while the second one is a 3-dimension spherical data set. In addition, the application of the model to a real-world data set related to image segmentation – obtained from the UCI repository [10] – is included.

Tests were run using a classic fivefold cross-validation in order to use the complete data set for training and testing. The ensembles were trained using one of the simplest meta-algorithms for ensemble training: the bagging meta-algorithm Breiman [4].

What is studied in these series of experiments is the effect of reducing the amount of data available – number of samples – for the training in each of the models, in each of the four quality measures presented in Section 3.

In each of the tests the data set has been reduced in a 1/5 of its original size, and a single model and an ensemble of 5 maps is calculated for each one. Comparing the performance of the models over datasets with the same inner structure, but different levels of sparsity on each; the stabilizing effect of the ensemble models as opposed to single algorithms is better studied. This series of experiments checks if ensemble models are able to keep adapting correctly to a less clearly defined structure. Also, as the three data sets contain a few outliers, the effect of decreasing the number of samples implies an increase on the relative amount of noisy samples in the data set and therefore, an increase of difficulty of the adaptation of the map to the data set increases.

Table 1
Parameters use for all networks trained over the 2D data set.

	SOM	SIM	Max-SIM
Size	25	25	25
Epochs	1000	1000	1000
Learning rate	0.01	0.02	0.03
Neighborhood	$\sigma_1 : 10$	$\sigma_1 : 10$	$\sigma_1 : 10,\ p : 1.9$

5.1. A 2-D artificial data set

This artificial 2-dimensional data set was created for testing and comparing the different algorithms described in this study. It was generated by using classic Gaussian distributions. The Gaussian distributions were centered along six different points of the 2-D data space. The centers were placed forming a “circular shape”. So although the clusters of data contain the same amount of samples, the data set is not uniformly distributed. Also some outlier samples (3% of total of samples) are included. For the measurement of classification error, the data samples were grouped into 6 classes one for each of the data centers, with the same number of samples, corresponding with each one of the centers previously mentioned.

Table 1 shows the parameters used for the base learners throughout all the experiment, both for the single and ensemble models. Parameters have been chosen in an experimental process of trial and error. Being parameter selection a tasks that is very dependent on the dataset to use, several preliminary experiments were conducted with a range of combinations of those parameters. The best performing set of parameters was selected to later conduct all the experiments detailed in the comparison.

Regarding the Classification error (Fig. 2), it can be concluded that for such a simple data set the use of ensembles does not improve the results of simple models (SOM, SIM or Max-SIM). Quite on the contrary, the effect of using ensembles to try to improve good classifiers has the opposite effect, obtaining worse performing networks. This effect had already been observed in [2,4]. The exception of this situation can be seen on Fig. 2 where, when the number of samples is very low (120) the Max-SIM and the WeVoS–Max-SIM outperform their SOM counterparts. When the data set is “harder” to analyze, the use of more complex models becomes advantageous.

Also, as the data classification is not one of the primary objectives of the topology preserving models; their results are very unstable in general for all models, varying a lot depending on the size of the data set.

In Fig. 3 it can be seen how in this case the ensemble models improves their respective single model. Especially, the WeVoS algorithm is useful when used with the SOM, as the single model is not the best one to deal with the data set used, and therefore the use of WeVoS can improve a not optimally performing algorithm. In the case of the Max-SIM, the use of the WeVoS algorithm improves the performance of the single Max-SIM, although in this case only slightly.

In such a simple data set, the use of ensembles is not giving any improvement respect the single model in the Distortion error (Fig. 4). This measure gives an idea of the topographic ordering of a map. In the case of a 1-dimensional map, where units are considered on a single line, it is very unlikely that an ensemble algorithm (WeVoS or other) improves the single models, as the tests prove.

As happened with the quantization error, all models behave in a very stable way, regardless of the size of the data set.

The Goodness of Map (Fig. 5) is a measure combining both quantization and topology preservation. In this case the WeVoS–Max-SIM proves its usefulness, as is the best performing one of all models. Also the WeVoS-SOM performs better than the single SOM. This is probably due to the best quantization characteristics obtained by the WeVoS in these tests, rather than to the topology preservation features. As was observed in other measures, the quantization measure favors the WeVoS model (Fig. 3), while the Distortion is worse for this presented model than for the single ones (Fig. 4).

In this case, this error measure decreases for all models with the decrease of the size of the data set, because as the data set becomes more sparse, the samples that serve as reference are lower in number. Still all models maintain their relative position.

5.2. A 3-D artificial data set

In these experiments, the data set used consists on an uniform distribution in the shape of a sphere. Samples are distributed uniformly forming an sphere, but the samples in the inside of it have been removed (those that where within a radius of 2/3 of the total radius) resulting in a hollow sphere. Each sample was assigned to one of eight different classes, being each class composed of the samples within each one of the eight uniform parts of the sphere. These parts were obtained by dividing the sphere twice using two perpendicular planes. The maps in this case are 2-dimensional.

Table 2 shows the parameters used for the base learners throughout all the experiment, both for the single and ensemble models. Parameters have been chosen in an experimental process of trial and error. The best performing set of parameters was selected to later conduct all the experiments detailed in the comparison.

For these experiments, the final adaptation of the network grid to the data set is showed in Fig. 6. The figure presents several interesting situations. The most straightforward observation is how the SIM and Max-SIM based models adapt better

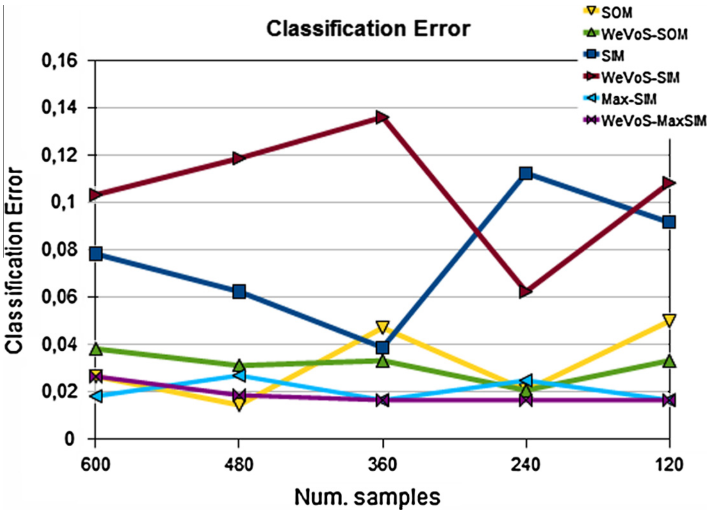


Fig. 2. Classification error for SOM, SIM and Max-SIM and their respective WeVoS ensemble over the circular data set.

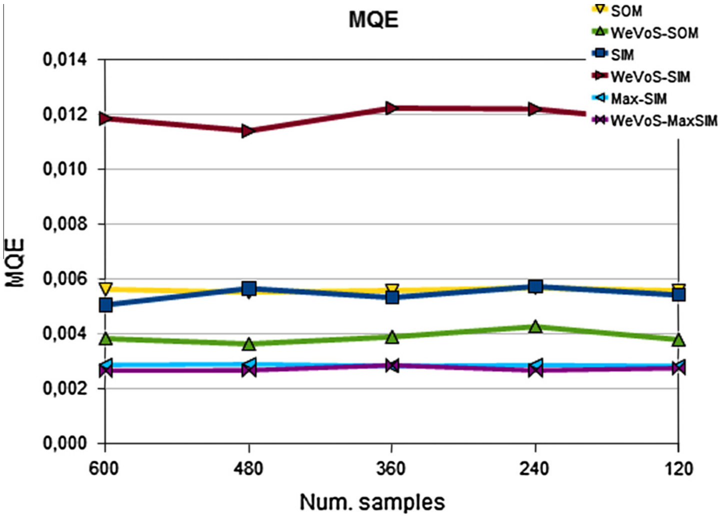


Fig. 3. Mean Quantization Error for SOM, SIM and Max-SIM and their respective WeVoS ensemble over the circular data set.

their structure to the spherical data set than the SOM based ones as expected [12]; as the first mentioned models adopt a more spherical shape and the latter mentioned adopts a more twisted shape. The shape adopted by the SIM and Max-SIM-based models, is considered a better adaptation of the data set, as the data sphere is hollow. This means that in the case of the SOM-based models, all units that are in the center of the map are either "dead" units or they are recognizing arbitrary samples from the outside of the sphere.

The second interesting observation is that in the SIM-based models, the use of the WeVoS algorithms helps to obtain a much clearer grid than the single models thanks to the way the algorithms determines the final position of each unit, taking into account the position of the neighboring ones. Finally, it is also interesting to point out that the SIM and WeVoS–SIM yield as a result a grid that wraps the data set in a spherical shape, therefore being in more close contact with the more exterior data samples. The Max-SIM and WeVoS–Max-SIM yield a similar spherical shape, but thanks to its more fine detail in the tuning of its parameters, they are able to adapt theirselves more closely to the data set, obtaining a sphere that is embedded into the data sphere, rather than wrapping it; being in contact not only with the most external samples, but the internal ones as well.

The effects of this situation into the analytical results are discussed in the remaining of this section.

In Fig. 7 it can be seen that the three ensemble models improve the performance of all the three single models. It can be seen then that, in a more complex data set, the use of the WeVoS algorithm becomes more interesting. Also in this case the Max-SIM and WeVoS–Max-SIM perform better than the respective SIM and SOM algorithms.

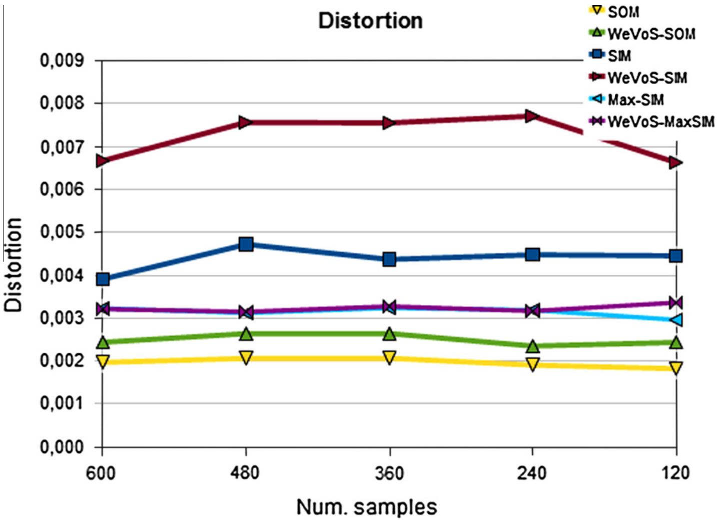


Fig. 4. Distortion for SOM, SIM and Max-SIM and their respective WeVoS ensemble over the circular data set.

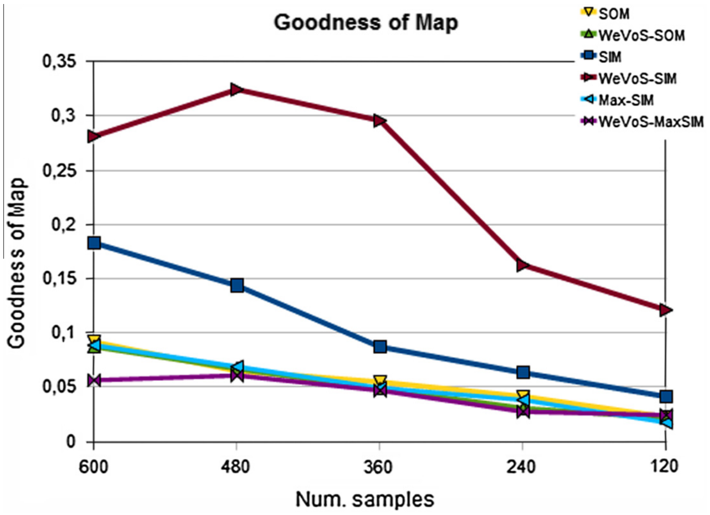


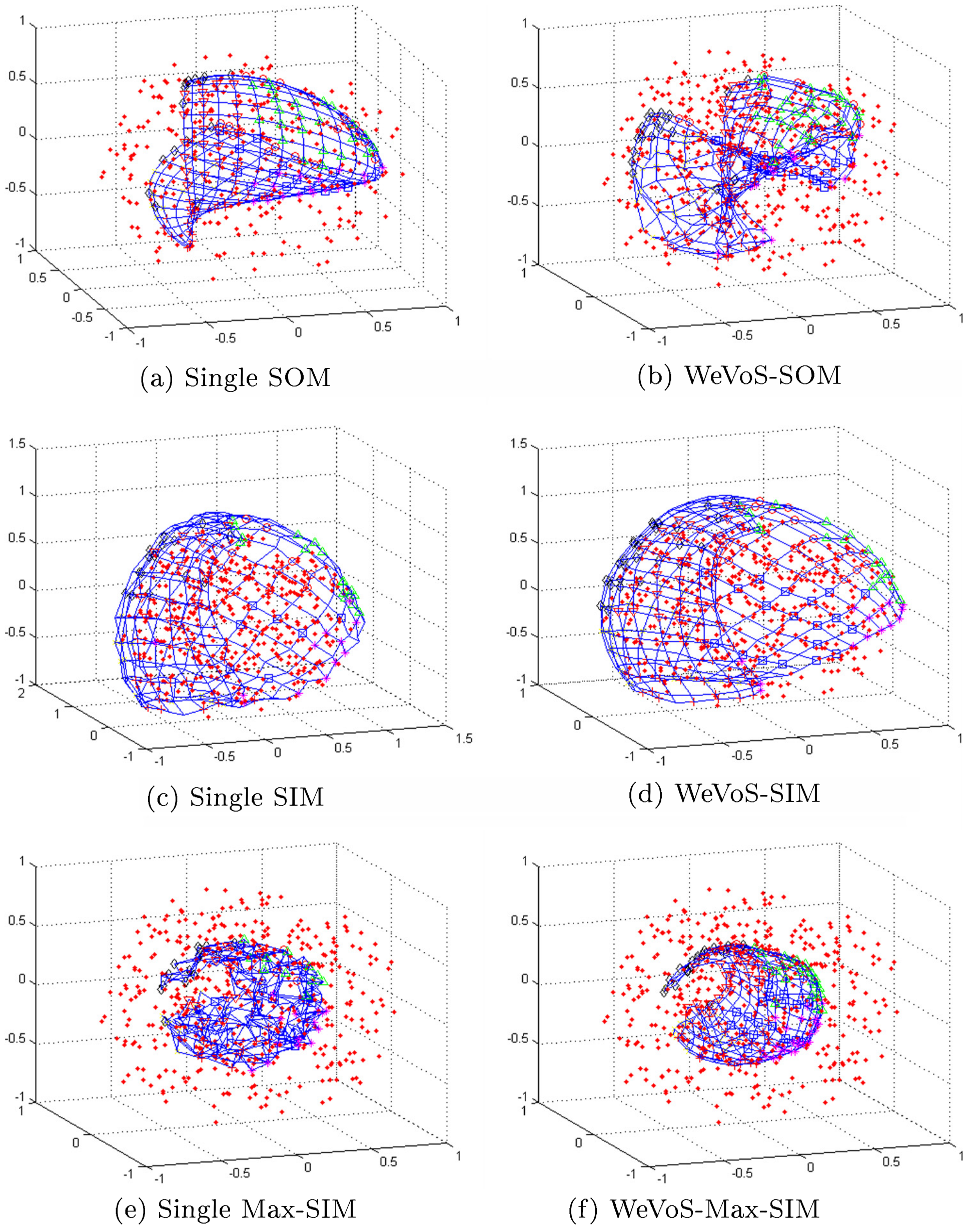
Fig. 5. Goodness of Map for SOM, SIM and Max-SIM and their respective WeVoS ensemble over the circular data set.

Table 2
Parameters use for all networks trained over the 3D data set.

	SOM	SIM	Max-SIM
Size	15 × 20	15 × 20	15 × 20
Epochs	4000	4000	4000
Learning rate	0.01	0.02	0.06
Neighborhood	$\sigma_1 : 8,\ \sigma_2 : 10$	$\sigma_1 : 10$	$\sigma_1 : 10, p : 1.7$

On the contrary of what happened in the 2-dimensional experiments (Fig. 3), in the 3-dimensional case (Fig. 8) the WeVoS–SIM improves the quality of this measure, compared with the single SIM; whereas the WeVoS–Max-SIM obtains worse results when compared with its single counterpart. Still, the SOM and WeVoS-SOM are outperforming the rest of the models.

Regarding the topological order of the networks, represented by the Distortion measure (Fig. 9), the WeVoS–SIM and WeVoS–Max-SIM outperform their respective single models, on the contrary of the 2D case (Fig. 4). The data set is now more complex and being all 2D networks, the chance to commit errors in the topology representation of the data set are higher, so



(a) Single SOM

(b) WeVoS-SOM

(c) Single SIM

(d) WeVoS-SIM

(e) Single Max-SIM

(f) WeVoS-Max-SIM

Fig. 6. The network grid of each of the models discussed represented over the 3-dimensional spherical data set.

the use of ensemble has more room to improve the results of single maps. In this case the best result is obtained by the WeVoS–Max-SIM as expected.

As a kind of summary of the previous observations, as it involves quantization and topology errors, the Goodness of Map is presented in Fig. 10. When comparing the use of the WeVoS meta-algorithm, it is worth noting that while the SOM obtains better results than the SIM; – the WeVoS–SIM improves the performance of the single SIM but the WeVoS-SOM obtains worse results than the single one. This may be due to the fact that, being the adaptation of the SIM less stiff, although adapting a spherical shape, obtains a more "disordered" grid than the SOM. This is added to the fact that in its more simple form, the training of the SIM is more difficult to adapt to a homogeneous data set, obtaining also worse quantization results.

On the contrary, the Max-SIM obtains much better results and even the WeVoS–Max-SIM is the best performing model of the six compared (see Fig. 10). In this case, obtaining a correct spherical shape and a lower Distortion error, the Max-SIM is outperforming the rest of the models.

The previously described behavior is consistent with the expected behavior of the WeVoS algorithm. It is important to emphasize that the main objective of the algorithm is to improve topology preservation by increasing the stability of which

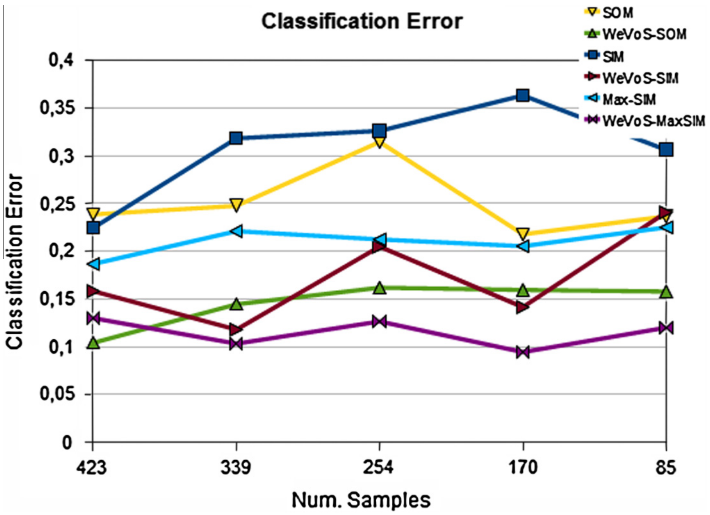


Fig. 7. Classification error for SOM, SIM and Max-SIM and their respective WeVoS ensemble over the spherical data set.

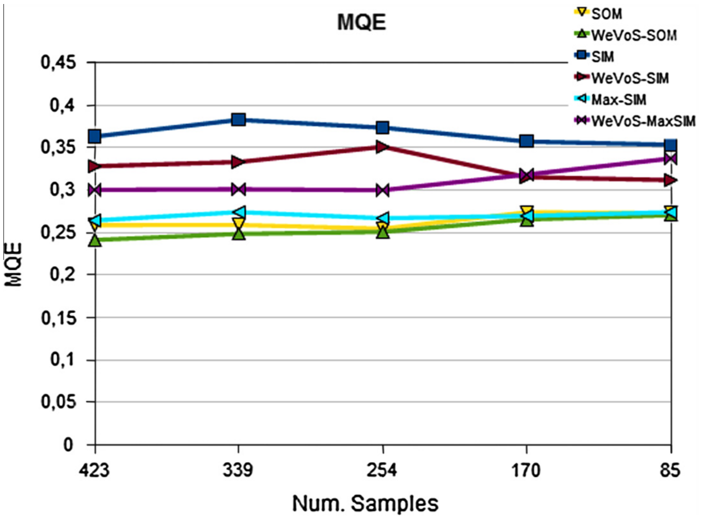


Fig. 8. Mean Quantization Error for SOM, SIM and Max-SIM and their respective WeVoS ensemble over the spherical data set.

the single models lacks. This is done by prioritizing the organization and ordering of the composing units of a map over the quantization error.

5.3. The Image Segmentation data set

In the third set of experiments a real data set is used. It has been obtained form the UCI Internet Repository [10] and consists in an Image Segmentation problem. The application of Topology-preserving models to these kind of tasks has been previously proved interesting [15,14].

The data set consists in 208 samples representing pixels extracted randomly form 7 different outdoor photographs. Each of the samples is composed of 19 dimensions. This is an interesting application for models such as the SIM or Max-SIM, as Image Segmentation is an scale-invariant problem. Although the set includes 2300 samples for testing, for the purpose of this study, only 208 samples have being randomly selected from the dataset; and 10-fold cross-validation was used in all experiments. This is a much more complex task for all the neural models to complete, so the experiments are performed maximizing the size of the training data partitions, as opposed to the simpler previous experiments.

Table 3 shows the parameters used for the base learners throughout all the experiments, both for the single and ensemble models. Parameters have been chosen in an experimental process of trial and error. The best performing set of parameters was selected to later conduct all the experiments detailed in the comparison.

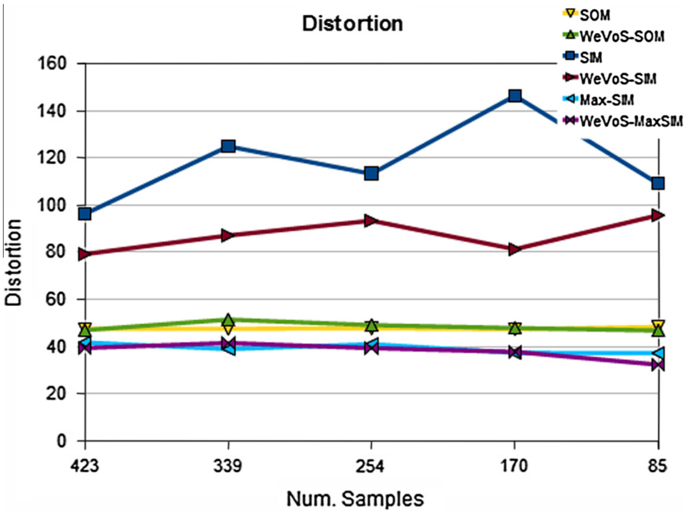


Fig. 9. Distortion for SOM, SIM and Max-SIM and their respective WeVoS ensemble over the spherical data set.

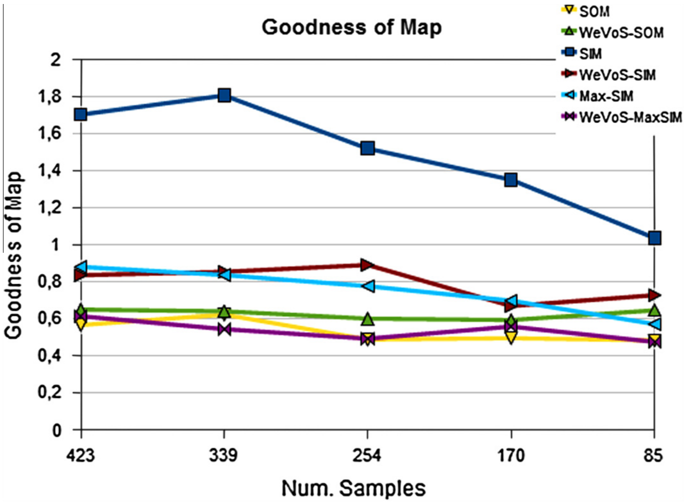


Fig. 10. Goodness of Map for SOM, SIM and Max-SIM and their respective WeVoS ensemble over the spherical data set.

Table 3
Parameters use for all networks trained over the Image Segmentation data set.

	SOM	SIM	Max-SIM
Size	15×10	15×10	15×10
Epochs	2000	3000	3000
Learning rate	0.01	0.005	0.01
Neighborhood	$\sigma_1 : 5,\ \sigma_2 : 8$	$\sigma_1 : 5,\ \sigma_2 : 8$	$\sigma_1 : 8,\ \sigma_2 : 10, p : 2$

Fig. 11 shows the results for the classification error obtained for the Image Segmentation data set. Again, the classification errors are very unstable, varying in a significant way depending on the instability of the data set. The WeVoS algorithm does not clearly improve the results obtained by the single algorithms.

Fig. 12 shows the results for the comparison of the quantization error between the different models. As with the artificial data sets, the model that clearly lowers its quantization error by the use of the WeVoS meta-algorithm is the Max-SIM. Again, the SIM based models seem to fit better the structure of the data set.

Fig. 13 shows results obtained by the models for the Distortion measure. On the contrary to previous results, this measure clearly differentiates the compared models. The SIM-based models obtain a clear lower error than the SOM-based ones.

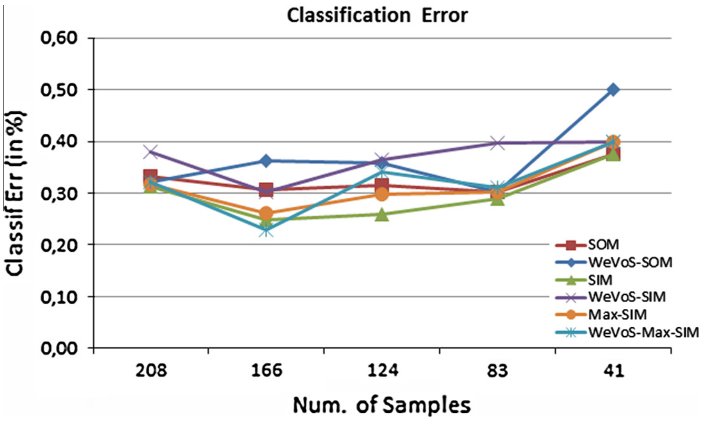


Fig. 11. Classification Error for SOM, SIM and Max-SIM and their respective WeVoS ensemble over the Image Segmentation data set.

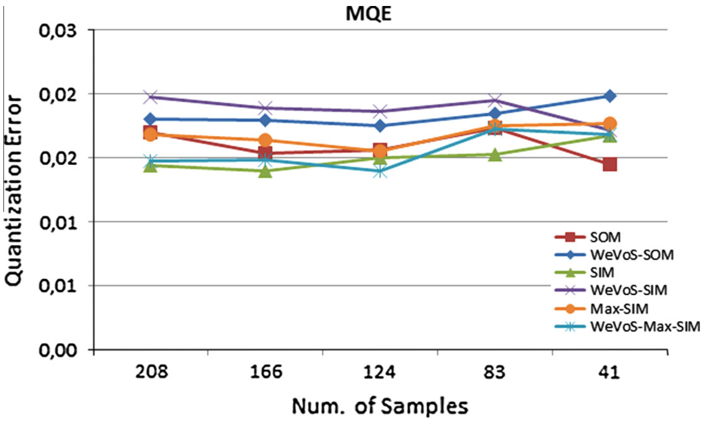


Fig. 12. Mean Quantization Error for SOM, SIM and Max-SIM and their respective WeVoS ensemble over the Image Segmentation data set.

Fig. 14 shows the differences of the compared models in relation with the Goodness of Map measure. Again in this case a clear difference between models can be observed. Also, the single SIM-based models perform quite well, compared with the single SOM. Similarly to other cases, the Max-SIM performance is improved with the use of the WeVoS algorithm as opposed to the other two models.

5.4. Summary

Some aspects that could be interesting to remark about the presented quality measures include the following:

The classification error is not the main objective of the Self-organizing models. As such, the results are quite unstable and cannot really used to compare performances. The WeVoS meta-algorithm presented is aimed to boost one of the most characteristic features of this kind of maps: to obtain a more truthful representation of the overall structure of the data set. In this context, a too close representation of the data samples could even be considered as an overfitting situation. The fact that not a clear improvement is shown in this area, is therefore an expected result.

More or less the same reasoning can be applied to the Mean Quantization Error. A too low quantization error can even point to an overfitting problem. The MQE can serve, to an extent, to asses the quality of a map but it does not really give a clear idea about the structure, or the twists in the map lattice. This is the reason why other, more complex, measures were proposed. Still, in the case of the Image Segmentation data set, a clear improvement can be seen when comparing this measure for the Max-SIM and the WeVoS–Max-SIM.

The Distortion measure is more interesting for the comparison of the models, as it was designed to measure the ordering of the map lattice. A progressive improvement of this measure by the use of WeVoS can be seen as the complexity of the data set increases. With the 2D data set, the best performing algorithm is the SOM; while in the case of the 3D data set, the Max-SIM and WeVoS–Max-SIM outperform the SOM, although remain very close to each other.

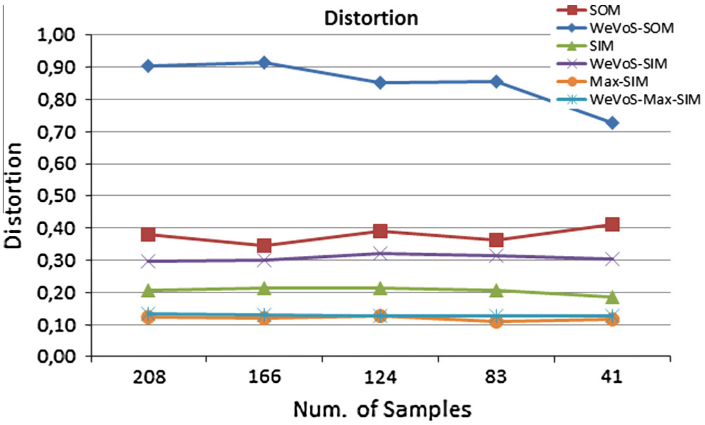


Fig. 13. Distortion for SOM, SIM and Max-SIM and their respective WeVoS ensemble over the Image Segmentation data set.

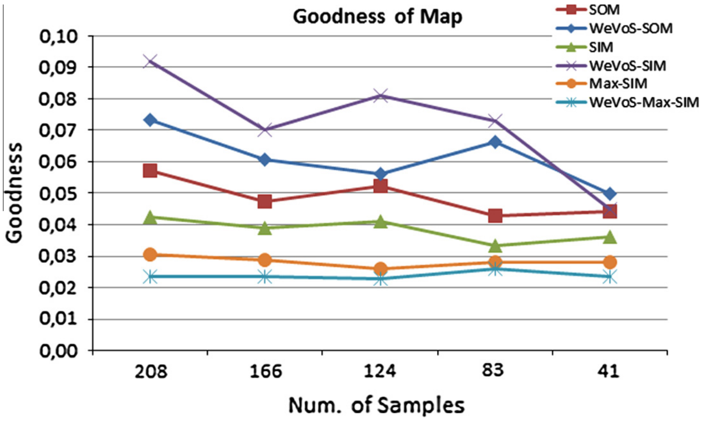


Fig. 14. Goodness of for SOM, SIM and Max-SIM and their respective WeVoS ensemble over the Image Segmentation data set.

The Goodness of Map measure is the more complete one in the sense that tries to account for quantization, but more importantly, also for the organization between the map units. In the case of the Goodness of Map measure the mentioned progression is clearer; being the case of the Image Segmentation the one in which the WeVoS–Max-SIM more clearly outperform the rest of models.

As overall results from the experiments performed, regarding the model comparison; three final tables including the ranking of each of the models, from the lowest error (1) to the highest error (6) have been included in this section.

A common feature of all tests is that the worst performing algorithms tend to be the SOM based, while the best tend to be the Max-SIM based. This was expected, as the latter is especially designed to perform better in the kind of datasets used in the experiments.

Another issue detected in the experiments is that the SIM algorithm is much more unstable than the Max-SIM. The latter has an extra parameter – parameter p – that enables to control the opening of the map over the data set. The WeVoS meta-algorithm, as many of ensemble fusion algorithms, relays in a relative balance between variance and similarity of the fused models. Very different models will fuse more roughly than similar ones. That idea is reinforced in this study by the fact that the WeVoS–Max-SIM manages to improve the results of the single Max-SIM, but that the WeVoS–SIM improves the single SIM in fewer occasions.

The task represented by the first experiment is a very simple one. As such, the difference between using single models or WeVoS is very modest. Even in some cases the WeVoS-based models obtain worst results than single ones. This is an expected effect of the ensemble architectures: when single models are performing in an optimal way, the use of ensembles adds nothing to their capacities. As a common result, the Max-SIM and WeVoS–Max-SIM models are the best performing ones. Its corresponding summary table can be seen at Table 4.

The the second experiment presents a much more difficult task, as the maps calculated are 2-dimensional, so the ordering of the map's lattice becomes substantially more difficult to achieve. In this case the use of the WeVoS algorithm is able to improve the results of the SOM and the Max-SIM in most of the measures and conditions tested. As explained before, when

Table 4
Ranking of the models (from lower to higher error) on the Artificial 2D dataset.

	Single			WeVoS		
	SOM	SIM	Max-SIM	SOM	SIM	Max-SIM
Classif	4	5	2	3	6	1
MQE	5	4	2	3	6	1
Distort.	1	5	4	2	6	3
Goodness	4	5	3	2	6	1

Table 5
Ranking of the models (from lower to higher error) on the Artificial 3D dataset.

	Single			WeVoS		
	SOM	SIM	Max-SIM	SOM	SIM	Max-SIM
Classif	5	6	4	2	3	1
MQE	2	6	3	1	5	4
Distort.	4	6	2	3	5	1
Goodness	2	6	4	3	5	1

Table 6
Ranking of the models (from lower to higher error) on the Image Segmentation dataset.

	Single			WeVoS		
	SOM	SIM	Max-SIM	SOM	SIM	Max-SIM
Classif	4	1	3	5	6	2
MQE	3	1	4	5	6	2
Distort.	5	3	1	6	4	2
Goodness	4	3	2	5	6	1

the task is more complex, there is more room to improve with the WeVoS meta-algorithm. The best performing algorithms is the WeVoS–Max-SIM, as happened in the first experiment.

Although it is a not so well performing algorithm, the improvements obtained by the WeVoS meta-algorithm is even more evident in the case of the SIM model, where the difference between single and WeVoS model can be clearly seen in all quality measures. The fact that the SIM is performing worse than the SOM points again to a high instability inherent to this algorithm. Still, the WeVoS algorithm manages to improve its results. The comparison of the models can be found on Table 5.

Finally, the last experiment represents a task on a real-life dataset. This is the most difficult of the three tasks, and as can be seen, offers some mixed results. When the base algorithm is not well suited for the dataset as happens with the SOM, the use of the WeVoS ensemble does not help to improve its results, but rather it worsens them. This is because, for the WeVoS algorithm to perform correctly, the maps to fuse must be rather similar between them. That is, neurons in the same position of the map must be comparable. Otherwise, the average calculated from them is very deviated from the optimal one and the final fused map does not represent the underlaying data as it should. This is the case in this experiment.

On the contrary, when the base map algorithm is performing correctly, the maps obtained in the ensemble have consistent features, and therefore, they are similar enough to be fused with good results. This is the case of the Max-SIM algorithm: as can be seen (see Table 6), the WeVoS meta-algorithm is able to improve the results obtained by the single map, becoming again the best performing model.

6. Conclusions

A novel algorithm to summarize an ensemble of topology preserving maps has been presented in this study for the SIM and Max-SIM. This algorithm aims to obtain the best topology preserving summary as possible, in order to be used as a reliable tool in data visualization. Due to the inherent capabilities of the SIM and Max-SIM, their combination with the WeVoS meta-algorithm especially helps to improve adaptation and visualization of datasets with a radial structure, as has been showed in the tests. The main improvement of the algorithm is the added stability compared to the single model used.

As can be seen in the results showed, the effective of this meta-algorithm depends on the complexity of the data set. With very simple analysis the added improvement of the ensemble is almost insignificant, or in some cases can even deteriorate the performance. On the contrary, its usefulness has been proved in the case of more complex data sets, where the extra complexity of the calculation of the ensemble is compensated by the considerable increase of performance.

Future work includes the application of this algorithm to other topology preserving models, as the Visualization Induced SOM (ViSOM) [32] and its combination with the use of other ensemble generation algorithms to boost its performance. Also real life data sets will be used in order to assess its usefulness for real problems.

Acknowledgments

This research has been partially supported through project of the Spanish Ministry of Science and Innovation TIN2010-21272-C02-01 (funded by the European Regional Development Fund). The authors would also like to thank the vehicle interior manufacturer, Grupo Antolin Ingenieria S.A., within the framework of the MAGNO2008-1028.-CENIT Project also funded by the MICINN.

References

[1] B. Baruque, E. Corchado, A weighted voting summarization of SOM ensembles, Data Min. Knowl. Disc. 21 (3) (2010) 398–426, http://dx.doi.org/10.1007/s10618-009-0160-3.
[2] C. Bishop, Neural Networks for Pattern Recognition, Oxford, 1995.
[3] E. de Bodt, M. Cottrell, M. Verleysen, Statistical tools to assess the reliability of self-organizing maps, Neural Netw. 15 (8–9) (2002) 967–978.
[4] L. Breiman, Bagging predictors, Mach. Learn. 24 (1996) 123–140.
[5] L. Breiman, Random forests, Mach. Learn. 45 (2001) 5–32, http://dx.doi.org/10.1023/A:1010933404324. <http://portal.acm.org/citation.cfm?id=570181.570182>.
[6] M. Bronstein, I. Kokkinos, Scale-invariant heat kernel signatures for non-rigid shape recognition, in: 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010, pp. 1704–1711. http://dx.doi.org/10.1109/CVPR.2010.5539838.
[7] E. Corchado, C. Fyfe, The scale invariant map and maximum likelihood Hebbian learning, in: International Conference on Knowledge-Based & Intelligent Information & Engineering System, 2002, pp. 245–249.
[8] E. Corchado, D. MacDonald, C. Fyfe, Maximum and minimum likelihood Hebbian learning for exploratory projection pursuit, Data Min. Knowl. Disc. 8 (2004) 203–225.
[9] M. Cottrell, J.C. Fort, G. Pages, Theoretical aspects of the SOM algorithm, Neurocomputing 21 (1–3) (1998) 119–138.
[10] A. Frank, A. Asuncion, UCI machine learning repository, 2010. <http://archive.ics.uci.edu/ml>.
[11] Y. Freund, R.E. Schapire, Experiments with a new boosting algorithm, in: International Conference on Machine Learning, 1996, pp. 148–156.
[12] C. Fyfe, A scale-invariant feature map, Network: Comput. Neural Syst. 7 (1996) 269–275.
[13] A. Georgakis, H. Li, M. Gordan, An ensemble of SOM networks for document organization and retrieval, in: International Conference on Adaptive Knowledge Representation and Reasoning (AKRR'05), 2005, pp. 6–141.
[14] A. Gonzalez, M. Graña, M. Cottrell, Basic competitive neural networks as adaptive mechanisms for non-stationary color quantization, Neural Comput. Appl. 8 (1999) 347–367.
[15] A.I. Gonzalez, M. Graña, A. D'Anjou, F. Albizuri, M. Cottrell, A sensitivity analysis of the self organizing map as an adaptive one-pass nonstationary clustering algorithm: the case of color quantization of image sequences, Neural Process. Lett. 6 (1997) 77–89.
[16] T. Heskes, Balancing between bagging and bumping, in: M.C. Mozer, M.I. Jordan, T. Petsche (Eds.), Advances in Neural Information Processing Systems, Denver, Colorado, USA, 1997, pp. 466–472.
[17] U. Johansson, T. Lofstrom, L. Niklasson, Obtaining accurate neural network ensembles, in: M. Mohammadian (Ed.), Proceedings, International Conference on Computational Intelligence for Modelling, Control & Automation Jointly with International Conference on Intelligent Agents, Web Technologies & Internet Commerce, vol. 2, 2006, pp. 103–108.
[18] S. Kaski, K. Lagus, Comparing self-organizing maps, in: C.v.d. Malsburg, W.v. Seelen, J.C. Vorbruggen, B. Sendhoff (Eds.), Lecture Notes in Computer Science, Springer, Berlin, 1996, pp. 809–814.
[19] T. Kohonen, An introduction to neural computing, Neural Netw. 1 (1988) 3–16.
[20] T. Kohonen, Self-Organizing Maps, vol. 30, Springer, Berlin, Germany, 1995.
[21] T. Kohonen, P. Lehtio, J. Rovamo, J. Hyvarinen, K. Bry, L. Vainio, A principle of neural associative memory, Neuroscience 2 (1977) 1065–1076.
[22] T. Kohonen, E. Oja, O. Simula, A. Visa, J. Kangas, Engineering applications of the self-organizing map, in: Proceedings of the IEEE, 1996, pp. 1358–1384.
[23] J. Lampinen, E. Oja, Clustering properties of hierarchical self-organizing maps, J. Math. Imag. Vis. 2 (1992) 261–272.
[24] L. Petrakieva, C. Fyfe, Bagging and bumping self organising maps, Comput. Inform. Syst. J. (2003).
[25] D. Polani, Measures for the organization of self-organizing maps, in: U. Seiffert, L.C. Jain (Eds.), Self-Organizing Neural Networks: Recent Advances and Applications (Studies in Fuzziness and Soft Computing), vol. 16, Physica-Verlag, Heidelberg, 2003, pp. 13–44.
[26] R. Polikar, Ensemble based systems in decision making, IEEE Circ. Syst. Mag. 6 (2006) 21–45.
[27] G. Pozlbauer, Survey and comparison of quality measures for self-organizing maps, in: J. Paralic, G. Polzlbauer, R. Andreas (Eds.), Fifth Workshop on Data Analysis (WDA'04), Elfa Academic Press, 2004, pp. 67–82.
[28] C. Saavedra, R. Salas, S. Moreno, H. Allende, Fusion of self organizing maps, in: 9th International Work-Conference on Artificial Neural Networks (IWANN 2007), 2007, pp. 227–234.
[29] J. Vesanto, M. Sulkava, J. Hollmen, On the decomposition of the self-organizing map distortion measure, in: Proceedings of the Workshop on Self-Organizing Maps (WSOM'03), 2003, pp. 11–16.
[30] G. Voronoi, Nouvelles applications des paramétres continus á la théorie des formes quadratiques, in: J. Math (Ed.), Reine Angew, vol. 133, 1907, pp. 97–178.
[31] D.H. Wolpert, Stacked generalization, Neural Netw. 5 (1992) 241–259.
[32] H. Yin, ViSOM – a novel method for multivariate data projection and structure visualization, IEEE Trans. Neural Netw. 13 (2002) 237–243.
[33] H. Zhang, Borf: loop-closure detection with scale invariant visual features, in: 2011 IEEE International Conference on Robotics and Automation (ICRA), 2011, pp. 3125–3130. <http://dx.doi.org/10.1109/ICRA.2011.5980273>.