Aggregating Self-Organizing Maps with Topology Preservation

Jérôme Mariette and Nathalie Villa-Vialaneix

INRA, UR 0875 MIA-T, BP 52627, 31326 Castanet Tolosan cedex, France
`jerome.mariette@toulouse.inra.fr`, `nathalie.villa@toulouse.inra.fr`

Abstract. In the online version of Self-Organizing Maps, the results obtained from different instances of the algorithm can be rather different. In this paper, we explore a novel approach which aggregates several results of the SOM algorithm to increase their quality and reduce the variability of the results. This approach uses the variability of the algorithm that is due to different initialization states. We use simulations to show that our result is efficient to improve the performance of a single SOM algorithm and to decrease the variability of the final solution. Comparison with existing methods for bagging SOMs also show competitive results.

Keywords: Self-Organizing Maps, aggregation, topology preservation

1 Introduction

Self-Organizing Maps (SOM), [1] have been shown to be powerful methods for analyzing high dimensional and complex data (see, for instance, [2] for applications of the method to many different areas). However, the method suffers from its lack of good convergence properties. In its original version, the theoretical convergence of the algorithm has only be proved in very limited cases [3] and even in the modified version in which the training of the SOM is expressed as an energy minimization problem [4], different runs of the algorithm give different results, that can be very dependent on the initialization. This problem is even more critical when the data set to be analyzed is complex or high dimensional.

This paper addresses the issue of aggregating several results of the SOM algorithm, all obtained on the same data set. Several attempts to combine SOMs while preserving their topological properties have been proposed in the literature [5–9]. In this paper, we present a novel method to combine several SOMs while preserving their topology. The proposed method combines several ideas taken from the different methods and allows to explore initialization states. It is both simple and efficient. We present a full comparison of the different options to aggregate the results of different SOMs and discuss the most relevant choices. Finally, we show that our approach is a competitive alternative to the existing methods on real data applications.

The remainder of the paper is organized as follows: in Section 2, an overview of aggregation methods for SOMs is presented. In Section 3, the proposed method is described. Finally, Section 4 presents experimental results and comparisons.

2 An overview of aggregation methods for SOMs

Suppose that B results of the SOM algorithm are given for the items $(x_i)_{i=1,\ldots,n}$, $(\mathcal{M}^b)_{b=1,\ldots,B}$. Each of these results, $\mathcal{M}^b$ is well defined by its set of prototypes $(p_u^b)_{u=1,\ldots,U}$ and comes with an associated clustering function $\phi^b : x \in \mathbb{R}^d \to \arg\min_{u=1,\ldots,U} \|x - p_u^b\|^2$. For the b-th SOM, the clusters will be denoted by $(\mathcal{C}_u^b)_{u=1,\ldots,U}$, where $\mathcal{C}_u^b = \{x_i : \phi^b(x_i) = u\}$. The purpose is to build a *fused* or a *merged* map, $\mathcal{M}^*$, with prototypes $(p_u^*)_{u=1,\ldots,U}$ and a clustering function ϕ^* which improves and summarizes the B maps into a unique consensual map. Note that all SOMs have been trained from the same data $(x_i)_{i=1,\ldots,n}$ or from a subset (*e.g.*, a bootstrap sample) of this data set. They can also have been trained from different descriptors of the observations (*e.g.*, from different sets of variables observed on the same items): in this case, the fused map thus corresponds to a map integrating the different descriptors. However, for the sake of simplicity, we will restrict our description and simulation to the first case (same observations, or eventually, bootstrap samples from the same observations and same descriptors).

As already explained in [5] in the context of a one-dimensional grid, there is no ground truth for cluster labelling in the unsupervised framework. A first strategy to overcome this issue is to perform a re-labelling of the clusters based on the clustering only: [6] merge together the clusters of different maps with a majority vote scheme. A "fused" prototype is defined as the centroid of the grouped cluster prototypes over $b = 1, \ldots, B$ and a topology is deduced posterior to the definition of the clusters. Another approach that uses the different maps in an indirect way is described in [10]: in this paper, we proposed to use a subset of $(x_i)_i$, using the most representative observations of the set of B maps, to train a final SOM from a simpler and more robust data set. This method is well suited to handle very large data sets. However, both approaches do not necessary produce a map with a topology similar to the B merged SOMs and make use of only a small part of the information provided by the B learned SOMs.

Several attempts to explicitly take advantage of the prior (common) structure of the maps have been proposed in the literature. A first method consists in constraining the B SOMs to be as similar as possible by a common initialization. This initialization can be derived, for instance, from a PCA of $(x_i)_i$. Then, the different maps are fused by averaging the prototypes of the clusters situated at the same position the B SOMs [7] or by using a majority vote scheme to classify the observations [5]. Alternatively, [5, 8] also propose to make the B SOMs similar by initializing the b-th SOM with the final prototypes of the previous one. [8] improves this approach by weighting the averaging of the prototypes by a cluster quality index. Similarly, [11] uses a similar strategy to handle streaming or large data sets, splitting the data into several patches that are sequentially processed by a different SOM algorithm initialized with the result of the previous one. However, these methods do not allow to explore the possibilities of different initializations, which can be an issue in SOM. Moreover, a sequential initialization of the B SOMs prevents from training them in parallel, which can be an important issue if B is large: using a large B is advised for stabilizing the result of the algorithm.

Another approach to preserve the topology property of the map is to align the different maps on one of them, which serves as a reference for the topology: in [12], the map is chosen arbitrarily, and the other maps are fused sequentially to this first one, averaging the prototypes $(p_u^b)_u$ of the current map to the closest prototypes of the current fused map $(p_u^*)_u$. To leverage the problem of the choice of the map that is used to align the other maps, [9] proposes to choose a reference map that is the best one according to a given clustering quality criterion. However, this method makes the result strongly dependent on the choice of the first map because only its topology is used, whereas the topologies of the next maps are not utilized as such.

3   Description of the optimal transformation method

It is well known that the quality of the SOM strongly depends on its initialization. Given different maps obtained from different (random) initializations, we propose to find the "best" transformation that can be used to obtain two comparable results between two distinct maps. The optimal one-to-one transformation between prototypes in general might be difficult to define so we restrict ourselves to transformations that strictly preserve the topology of the map, *i.e.* the set of linear isometric transformations (rotation and/or symmetry). To do so, only square maps with m rows and columns are considered (*i.e.*, using the notations introduced in the previous section, $U = m^2$): in these maps, the clusters are supposed to be positioned on a 2D grid at coordinates $\{(k_1, k_2)\}_{k_1, k_2 = 1, \ldots, m}$.

Then, $\mathcal{T}$ denotes the set of all transformations, $T : \mathbb{R}^2 \to \mathbb{R}^2$, that let the map globally invariant: more precisely, $\mathcal{T}$ is composed of the set of rotations $\{r_\theta\}_{\theta \in \{0, \pi/2, \pi, 3\pi/2\}}$ and of the transformations $\{r_\theta \circ s\}_\theta$, with s the symmetry with respect to the axis passing by the points $\left(\frac{m+1}{2}, 0\right)$ and $\left(\frac{m+1}{2}, m\right)$. For a given map $\mathcal{M}$ with prototypes $(p_u)_u$ and a given $T \in \mathcal{T}$, the transformed map $T(\mathcal{M})$ is the map in which the unit u, with coordinates (k_1^u, k_2^u) in $\mathbb{N}^2$, has a prototype denoted by p_u^T which is the prototype $p_{u'}$ of the original map, u' being the unit located at $T^{-1}(k_1^u, k_2^u)$.

When comparing two maps, the mean of the squared distances (in $\mathbb{R}^d$) between the prototypes of the two maps that are located at the same position is calculated. For two maps $\mathcal{M}$ and $\mathcal{M}'$, with respective prototypes $(p_u)_u$ and $(p'_u)_u$, we define a distance between two maps as the distance between their respective prototypes positionned at the same coordinates:

$$D\left(\mathcal{M}, \mathcal{M}'\right) = \frac{1}{m^2} \sum_{u=1}^{m^2} \|p_u - p'_u\|^2. \tag{1}$$

The best transformation between the current fused map and the next map to be fused is chosen according to this distance. The two maps are then fused using the optimal transformation before they are merged, as described in Algorithm 1. The optimal transformation is found by computing the distance between the maps to be fused, $T_b^*(\mathcal{M}^b)$, and a reference map, which can be the first of the

---

Algorithm 1 Optimal transformation

Initialization $\mathcal{M}^{*,1} \leftarrow \mathcal{M}^1$
for $b : 2 \rightarrow B$ **do**
    Optimal transformation

$$T_b^* := \arg\min_{T \in \mathcal{T}} D\left(\mathcal{M}^{*,b-1}, T(\mathcal{M}^b)\right)$$

    Fusion between $\mathcal{M}^{*,b-1}$ and $T_b^*(\mathcal{M}^b)$. **Provides**: $\mathcal{M}^{*,b}$
end for
Return $\mathcal{M}^* := \mathcal{M}^{*,B}$

---

list, $\mathcal{M}^1$, for instance[1]. The fusion between the map is performed as suggested in [7] by averaging the prototypes located at the same position:

$$\forall\, u = 1, \ldots, m^2, \qquad p_u^* := \frac{1}{B} \sum_{b=1}^{B} p_u^{b,T}. \tag{2}$$

In the method described in the previous section, all maps are fused in an arbitrary order. However, as pointed out in [9], the maps may have very different qualities and may also be very different: merging a very peculiar map with a poor quality might lead to deterioration of the the results instead of improving them. In this section, two strategies are presented to leverage this problem.

The first one uses a measure of quality of the maps and first rank the maps from the one with the best quality to the one with the worse quality: $\mathcal{M}^{(1)}$, ..., $\mathcal{M}^{(B)}$. Standard quality measures for SOM can be used to perform this ranking [13]: i) the *quantization error* (QE), $\sum_{u=1}^{m^2} \sum_{i:\, x_i \in \mathcal{C}_u^*} \|x_i - p_u^*\|^2$, which is a clustering quality measure, disregarding the map topology; ii) the *topographic error* (TE) which is the simplest of the topographic preservation measure: it counts the ratio of second best matching units that are in the direct neighborhood on the map of the best matching units for every $(x_i)_i$. However, for small maps and relatively simple problems, this measure has a small variability and can lead to many equally ranked maps.

Therefore, another approach is introduced to make a trade-off, while ranking the maps, between clustering and topographic qualities: the average rank of the maps is computed as:

$$r^b = \frac{r_{\text{quanti}}^b + r_{\text{topo}}^b}{2} \tag{3}$$

where r_{quanti}^b is the rank of the map $\mathcal{M}^b$ according to its quantization error (the best map is ranked first) and similarly for r_{topo}^b with the topographic error and the maps were finally ranked by increasing order of $(r^b)_b$.

---

[1] The current fused map, $\mathcal{M}^{*,b-1}$ has also been used as a reference map, with no difference in the final result. Using $\mathcal{M}^1$ is thus a better strategy, because optimal transformation can be computed in parallel.

Taking advantage of this ordering of the maps, the previous method can be modified using two different strategies:

1. the *similarity strategy*: following an idea similar to [9], the maps are merged by similarity: the merging process is initialized with the best map: $\mathcal{M}^{*,1} \leftarrow \mathcal{M}^{(1)}$. Then, this map is merged only with the maps that resemble this reference map. To do so, a simple ascending hierarchical clustering is performed between the maps $(T_b^*(\mathcal{M}^b))_{b=1,\ldots,B}$, with $(T_b^*)_b$ obtained by comparison with the reference map $\mathcal{M}^1$. This clustering is based on the distance introduced in (1) and the hierarchical tree is cut using the method described in [14]. Finally, the maps in the same cluster as $\mathcal{M}^{(1)}$ are fused to $\mathcal{M}^{*,1}$;

2. the *ordering strategy*: an alternative approach is performed sequentially by merging the maps by increasing rank $\mathcal{M}^{(1)}, \mathcal{M}^{(2)}, \ldots$ The merging process is stopped at $\mathcal{M}^{(B')}$ with $B' \leq B$ (and usually $B' < B$) when the quality of the fused map $\mathcal{M}^{*,B'}$ would not increase anymore by merging it with $\mathcal{M}^{(B'+1)}$ (actually, two strategies are investigated: stopping when the quality measure is not increasing or stopping when the quality measure has not increased for the last $5\%B$ fused maps).

4 Simulations

Methodology. In all the simulations, $B = 100$ maps are generated using the standard SOM. The optimal B has not been investigated in this paper and the number of fused maps was simply taken large enough so that the fusion makes sense. All maps were built with approximately $m = \sqrt{\frac{n}{10}}$ units and $5 \times n$ iterations of the stochastic algorithm and equipped with a Gaussian neighborhood controlled with the Euclidean distance between units on the grid. The size of the neighborhood was progressively decreased during the training. All simulations have been performed using the R package **SOMbrero**[2]. The 100 maps are then fused using one of the strategies described below and the performance of the different methods are finally assessed using various quality criteria for the resulting maps $\mathcal{M}^*$: i) two criteria already mentioned in Section 3 that are standard to measure the quality of the SOM: i) QE and TE; ii) a criterion which uses the ground truth, when available (*i.e.*, an *a priori* group for the observations), the normalized mutual information (NMI) [15] between the unit of the map and the *a priori* group. This criterion quantifies the resemblance between the *a priori* group and the clustering provided by the SOM (it is comprised between 0 and 1, a value of 1 indicating a perfect matching between the two classifications). Note that this criterion must be interpreted with care because if the *a priori* groups are split between several units of the map, each of these units being composed of one group only (which is expected for SOM results), the criterion can be lower than when the groups are split between less units which are all composed of several groups (which would be a less expected result). Thus, this criterion has to be interpreted only together with the QE and the TE values.

---

[2] http://cran.r-project.org/web/packages/sombrero, version 1.0.

The performance of the method is also assessed in term of *stability*. It is expected that several runs of one aggregating method give similar (thus stable) results. This stability is estimated in terms of: i) the distance between two final maps obtained from two different runs of the same method. If $\mathcal{M}^*$ and $\widetilde{\mathcal{M}}^*$ are two maps, the quantity $D(\mathcal{M}^*, T^*(\widetilde{\mathcal{M}}^*))$, where D is defined as in (1) and $T^* := \arg\min_{T \in \mathcal{T}} D(\mathcal{M}^*, T(\widetilde{\mathcal{M}}^*))$, is computed. This gives an estimation of the dissemblance between two maps from the prototype (hence the topological) perspective. If calculated over 250 different final maps, this quantity helps to quantify the stability of the final prototypes provided by a given aggregation method; ii) the NMI between the final classes of two final maps obtained from two different runs of the same method. This gives an estimation of the dissemblance from the clustering perspective for a given aggregation method.

250 fusions for each method are performed using the methodology described above. This permits to compute average quality as stability criteria as well as to have an overview of the distribution of these criteria when the method is repeated.

Compared methods. The comparisons performed in this section aim at comparing our approach to existing ones (which are described in Section 2) as well as to investigate several options of the method (as discussed in Section 3).

First, our method, which merges several maps obtained from several initialization states, is compared to the standard bagging approach, in which several maps are trained from bootstrap samples from the similar initialization states. More precisely, bootstrap strategies are:

- the method denoted by **B-Rand**, which uses a common random initialization to learn $B = 100$ maps from 100 bootstrap samples coming from the original data set. Then, the prototypes that are positioned at the same coordinates, are averaged to obtain the final map $\mathcal{M}^*$ (as suggested in [7]);
- the method denoted by **B-PCA**, which uses a common PCA initialization to learn $B = 100$ maps from 100 bootstrap samples coming from the original data set (as suggested by [5]). The PCA initialization consists of initializing the prototypes by regularly positioning them along the coordinates of the projection of the data set on the first two axis of the PCA. Then, the prototypes that are positioned at the same coordinates, are averaged to obtain the final map $\mathcal{M}^*$;
- the method denoted by **B-Seq**, which uses a sequential initialization of the $B = 100$ maps: the first map is initialized randomly and trained with a bootstrap sample and the b-*th* map is initialized with the final prototypes of the $(b-1)$-*th* map and trained with another bootstrap sample. Finally, the final map $\mathcal{M}^*$, is obtained by averaging the prototypes of the $B = 100$ maps, that are positioned at the same coordinates, as suggested in [8].

These strategies are compared with our method and its bootstrap version, respectively denoted by **RoSyF** (for "Rotation and Symmetry Fusion") and **B-RoSyF**. **RoSyF** learns $B = 100$ maps, each from a different random initial state and using the whole data set $(x_i)_{i=1,\ldots,n}$ and **B-RoSyF** learns $B = 100$ maps from 100 bootstrap samples coming from the original data set.

Table 1. Method performance comparison (mean and standard deviation of different quality criteria; QE has been multiplied by 100)

	B-Rand	B-PCA	B-Seq	B-RoSyF	RoSyF	Best-R	Best-QE	Best-TE
"Glass"								
mean QE	855.10	855.93	854.97	609.84	597.81	595.09	**560.69**	593.68
sd QE	10.30	9.43	9.24	23.10	9.82	15.52	**5.45**	13.96
mean TE	11.95%	12.42%	11.77%	0.01%	0.01%	0.10%	0.04%	**0.00**%
sd TE	6.09%	6.53%	6.45%	0.04%	0.07%	0.24%	0.17%	**0.00**%
mean NMI	15.80%	15.77%	16.00%	**18.92**%	17.86%	15.64%	16.37%	15.87%
sd NMI	3.38%	3.15%	3.30%	2.09%	**1.38**%	2.20%	2.03%	2.21%
"Vowel"								
mean QE	847.57	847.73	847.91	550.78	545.88	547.44	**531.30**	548.23
sd QE	11.82	10.88	11.63	5.18	**1.01**	7.10	2.39	6.72
mean TE	5.89%	6.06%	5.80%	0.07%	0.07%	0.19%	0.20%	**0.00**%
sd TE	3.62%	3.46%	3.37%	0.10%	0.08%	0.14%	0.14%	**0.00**%
mean NMI	7.11%	6.76%	7.03%	9.47%	9.57%	**9.64**%	9.53%	9.53%
sd NMI	1.44%	1.37%	1.49%	0.12%	**0.11**%	0.66%	0.54%	0.72%

Finally, we also compare **RoSyF** with the approach consisting in selecting only one map from the B maps, the map supposed to be the best for instance. More precisely, using the $B = 100$ maps generated during the training of the **RoSyF** method, we selected one of the $B = 100$ maps i) randomly (this method is denoted by **Best-R**), ii) with the smallest QE (this method is denoted by **Best-QE** or iii) with the smallest TE (this method is denoted by **Best-TE**).

Datasets and results. This section compares the results obtained on two datasets coming from the UCI Machine Learning Repository[3] as available in the R package **mlbench**[4]. More precisely, the data "Glass" ($n = 214$, $d = 10$ and 7 *a priori* groups) [16] and the data "Vowel" ($n = 990$, $d = 10$ and 11 *a priori* groups) [17] are used. The SOM parameters are set to $m = 5$ and 1 000 iterations for "Glass" and $m = 10$ with 5 000 iterations for "Vowel". The different strategies, and especially the relevance of using different initial states instead of different bootstrap samples with the same initialization, is evaluated. The results are provided in Table 1.

First, note that for almost all quality criteria and datasets, **RoSyF** obtain better results than the methods based on different bootstrap samples (all differences are significant according to Wilcoxon test, risk 5%). **B-RoSyF** slightly deteriorates **RoSyF** performances. [18, 19] reported that the SOM algorithm is highly insensitive to initialization if run on the same data set as compared to

---

[3] http://archive.ics.uci.edu/ml
[4] http://cran.r-project.org/web/packages/mlbench

Table 2. Method stability comparison (mean and standard deviation of different stability criteria; D has been multiplied by 10 000)

	B-Rand	B-PCA	B-Seq	B-RoSyF	RoSyF	Best-R	Best-QE	Best-TE
				"Glass"				
mean D	70.85	67.22	**67.06**	149.65	67.07	2047.14	1302.27	1581.49
sd D	38.62	32.32	**31.24**	335.14	310.74	1557.08	1170.39	1186.28
mean NMI	64.77%	65.60%	65.88%	83.54%	**87.47%**	49.15%	54.41%	49.86%
sd NMI	6.37%	6.32%	6.23%	5.83%	**5.11%**	10.81%	9.57%	10.26%
				"Vowel"				
mean D	59.89	61.33	59.21	15.30	**11.07**	681.87	535.32	716.81
sd D	31.19	33.33	31.42	5.77	**3.87**	275.23	185.06	343.41
mean NMI	57.32%	56.83%	57.70%	90.83%	**92.39%**	72.53%	74.94%	72.11%
sd NMI	5.32%	5.21%	5.20%	1.59%	**1.33%**	3.29%	2.66%	3.37%

what is obtained if bootstrap samples are used. However, it seems that the quality of the aggregated map is much better when different initial states are used on the same data set rather than different bootstrap samples with a common initial state, whatever this initial state is. Second, the TE obtained by **RoSyF** is always the lowest, just after the one obtained by **Best-TE** (which always selects the map with the lowest TE) but with a better QE and a better NMI. Again, all these differences are significant according to Wilcoxon tests (risk: 5%). On a clustering quality point of view, **RoSyF** is the method that obtains the second lowest quantization error, just after **Best-QE** which is designed to select the map with the lowest QE. Also, from a classification point of view, its performance is also very good: in average, **RoSyF** ranks first for the NMI criterion. Also note that all quality criteria have a low variability: the standard deviations is almost always the lowest: **RoSyF** is the method which has the best coefficient of variation (mean divided by the standard deviation) for all quality criteria.

Table 2 (and Figure 1 for the dataset "Vowel") provides a comparison of the stability criteria. For this data set, **RoSyF** has the best stability, either in term of prototype stability (even though **B-PCA** and **B-Seq** also have a good prototype stability) and even more in term of class stability. These differences are significant according to Wilcoxon tests (risk: 5%). The results indicate that the method is indeed appropriate to improve the quality of the final map but also that it is very stable and gives very similar results if used several times, with different initializations of the prototypes and different training of the merged maps.

The relevance of stopping the merging process before all the maps have been fused has also been evaluated[5]. This comparison shows that there is only a small benefit in stopping the merging process before all maps have been used: most

---

[5] For the sake of paper length, detailed results are not reported but only described.
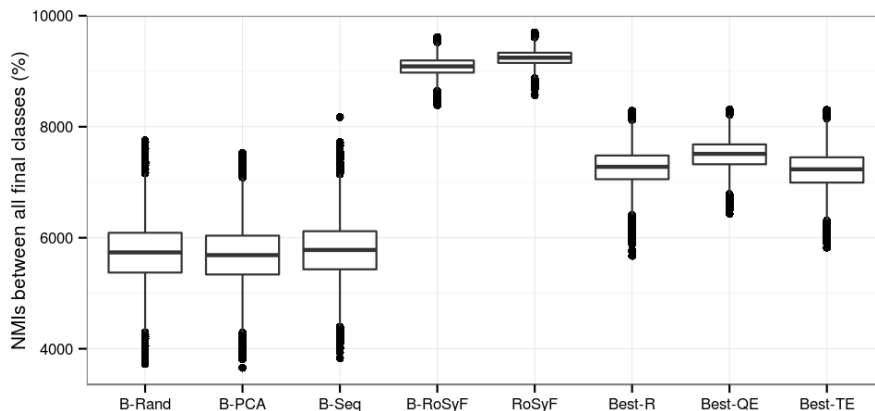
Fig. 1. Normalized mutual information (NMI) between pairs of clusterings obtained from the 250 final maps generated by the different approaches.

strategies lead to an highly deteriorated TE. Only stopping the training process when TE increases (**TE-Inc**) or based on the similarity strategy described in 3 are valid approaches in terms of quality criteria. However, a stability analysis shows that all these strategies strongly deteriorate the stability of the final map: merging all maps is the approach that provides the best stability, either in term of prototype comparison than in term of class comparison, except for **TE-Inc** which provides a slightly more stable clustering but very different prototypes. All these strategies use only few maps (less than 10 maps in average), except again **TE-Inc** which uses 89.4 maps in average for the "Glass" dataset and is thus very close to the maximum number of available maps (100). Actually, additional simulations (not shown for the sake of paper length) merging more than 100 maps proved that the stability increases with the number of fused maps (up to a certain number which was for our dataset between 500 to 1000 maps). A trade-off has thus to be found between computational time required to generate a large number of maps and stability of the results. This question is still under study.

Conclusion

Although most work on SOM ensembles are based on bootstrapping techniques, this paper presents an approach allowing to explore different initial states for the map. The method improves the stability of the fused map, both in term of prototypes and in terms of clustering. We are currently investigating how to choose an optimal number B of maps to fuse as well as weighting schemes based on various quality criteria: this approach is already promising to improve the results, especially the stability of the final map.

References

1. Kohonen, T.: Self-Organizing Maps, 3rd Edition. Volume 30. Springer, Berlin, Heidelberg, New York (2001)
2. Kohonen, T.: MATLAB Implementations and Applications of the Self-Organizing Map. Unigrafia Oy, Helsinki, Finland (2014)
3. Cottrell, M., Fort, J., Pagès, G.: Theoretical aspects of the SOM algorithm. Neurocomputing **21** (1998) 119–138
4. Heskes, T.: Energy functions for self-organizing maps. In Oja, E., Kaski, S., eds.: Kohonen Maps. Elsevier, Amsterdam (1999) 303–315
5. Petrakieva, L., Fyfe, C.: Bagging and bumping self organising maps. Computing and Information Systems Journal **9** (2003) 69–77
6. Saavedra, C., Salas, R., Moreno, S., Allende, H.: Fusion of self organizing maps. In: Proceedings of the 9th International Work-Conference on Artificial Neural Networks (IWANN 2007). (2007)
7. Vrusias, B., Vomvoridis, L., Gillam, L.: Distributing SOM ensemble training using grid middleware. In: Proceedings of IEEE International Joint Conference on Neural Networks (IJCNN 2007). (2007) 2712–2717
8. Baruque, B., Corchado, E.: Fusion methods for unsupervised learning ensembles. Volume 322 of Studies in Computational Intelligence. Springer (2011)
9. Pasa, L., Costa, J., Guerra de Medeiros, M.: Fusion of Kohonen maps ranked by cluster validity indexes. In Polycarpou, M., de Carvalho, A., Pan, J., Woźniak, M., Quintian, H., Corchado, E., eds.: Proceedings of the 9th International Conference on Hybrid Artificial Intelligence Systems (HAIS 2014). Volume 8480., Salamanca, Spain, Springer International Publishing Switzerland (2014) 654–665
10. Mariette, J., Olteanu, M., Boelaert, J., Villa-Vialaneix, N.: Bagged kernel som. In: Proceedings of WSOM, Mittweida, Germany (2014) Forthcoming.
11. Hammer, B., Hasenfuss, A.: Topographic mapping of large dissimilarity data sets. Neural Computation **22**(9) (September 2010) 2229–2284
12. Georgakis, A., Li, H., Gordan, M.: An ensemble of som networks for document organization and retrieval akrr (2005). In: Proceedings of International Conference on Adaptive Knowledge Representation and Reasoning (AKRR 2005). (2005)
13. Polzlbauer, G.: Survey and comparison of quality measures for self-organizing maps. In Paralic, J., Polzlbauer, G., Rauber, A., eds.: Proceedings of the Fifth Workshop on Data Analysis (WDA'04), Sliezsky dom, Vysoke Tatry, Slovakia, Elfa Academic Press (2004) 67–82
14. Langfelder, P., Zhang, B., Horvath, S.: Defining clusters from a hierarchical cluster tree: the dynamic tree cut package for R. Bioinformatics **24**(5) (2008) 719–720
15. Danon, L., Diaz-Guilera, A., Duch, J., Arenas, A.: Comparing community structure identification. Journal of Statistical Mechanics (2005) P09008
16. Towell, G., Shavlik, J.: Interpretation of artificial neural networks: mapping knowledge-based neural networks into rules. Proceedings of Advances in Neural Information Processing Systems **4** (1992)
17. Niranjan, M., Fallside, F.: Neural networks and radial basis functions in classifying static speech patterns. Computer Speech & Language **4**(3) (1990) 275–289
18. Cottrell, M., de Bodt, E., Verleisen, M.: A statistical tool to assess the reliability of self-organizing maps. In Allinson, N., Yin, H., Allinson, J., Slack, J., eds.: Advances in Self-Organizing Maps (Proceedings of WSOM 2001), Lincoln, UK, Springer Verlag (2001) 7–14
19. de Bodt, E., Cottrell, M., Verleisen, M.: Statistical tools to assess the reliability of self-organizing maps. Neural Networks **15**(8-9) (2002) 967–978