2014 International Joint Conference on Neural Networks (IJCNN)
July 6-11, 2014, Beijing, China

Clustering of the Self-Organizing Map using Particle Swarm Optimization and Validity Indices

Leonardo Enzo Brito da Silva and José Alfredo Ferreira Costa
Department of Electrical Engineering
Federal University of Rio Grande do Norte
Natal-RN, Brazil
Email:{leonardoenzob, jafcosta}@gmail.com

***Abstract*—In this paper, an automatic clustering algorithm applied to self-organizing map (SOM) neurons is presented. The connections of the SOM grid are pruned according to a weighted sum of a set of measures of connection strength between adjacent neurons. The coefficients of the weighted sum are obtained through particle swarm optimization (PSO) search in the multidimensional problem space, where the fitness function is the composed density between and within clusters (CDbw) validity index of strongly connected groups of neurons, while scanning through different values of the minimum cluster size so as to find stable regions with a reasonable trade-off between their length and their mean CDbw value. Simulation results are further presented to show the performance of the proposed method applied to synthetic and real world datasets.**

I. Introduction

Nowadays, technological advances have enabled the acquisition and storage of a plethora of data from sources such as industrial processes, medical data, costumer behaviour, financial data, market segmentation, social networking profiles, among others [1][2]. This increasing amount of data produced in the modern world maximizes, for both scientific and commercial applications, the need to understand the existing information and extract valuable knowledge, as well as simultaneously overcome the problems related to outliers, scalability, dimensionality and heterogeneity.

In this context, clustering in the computational intelligence field can be broadly defined as the search for natural groups existent in datasets. Traditionally, a clustering method is used to partition the data so that the intra-cluster and inter-cluster similarities are maximized and minimized, respectively [3]. A wide variety of methods to accomplish this task have been proposed in the literature [4], and the self-organizing map (SOM)[5] is one of the most used artificial neural network for this purpose. Among the reasons, it may be cited the mapping from an input space of higher dimension (data space) to an output space of lower dimension (grid of neurons), while preserving the topology information and compressing data [6]. In this sense, the SOM network can be seen as a non-linear generalization of principal component analysis [7]. Along with those characteristics, there is also the reduced computational cost when dealing with SOM neurons instead of the entire dataset (vector quantization); the readily usage of classic clustering algorithms applied directly to it [8], as well as its associated visualization techniques, what transforms the self-organizing maps into a powerful tool for exploratory data analysis. Since it has been proposed, the SOM has been used in a wide range of applications [9], such as pattern recognition, image processing (remote sensing, image compression), control and monitoring systems processes (including detection and fault tolerance).

Recently, a swarm-intelligence-based clustering algorithm has been proposed to perform the clustering task [10], by using validity indices as fitness functions in a combination of particle swarm optimization (PSO) [11][12] and differential evolution (DE) [13][14] to avoid local minima. The proposed approach consists of a series of steps that intercalates PSO and DE sequentially, where each particle carries the position of cluster centroids and its activation threshold as a part of the solution. The use of several validity indices in order to evaluate a clustering solution, given their bias towards specific structures, is advised.

This paper focuses on partitioning the SOM through the application of an algorithm that prunes the SOM grid based on coefficients obtained by the PSO to weight different measures of pairwise connection strength between neurons. The paper is organized as follows. The section II provides a brief review of the SOM network while section III discusses the basics of particle swarm optimization. In section IV, a few clustering validity indices are concisely described. The proposed method is defined in section V. In the section VI, the experiments carried out are described. The discussions and conclusions are presented in sections VII and VIII, respectively.

II. Self-Organizing Maps

Self-organizing maps consist of a set of topologically ordered neurons arranged in a grid. The grid is usually referred to as the output space, whereas the data space is known as the input space ($\mathbb{R}^n$ space of the patterns $\mathbf{x}$). Each neuron has an associated weight vector ($\mathbf{w}$) in the input space, so that a mapping from a continuous higher dimensional space to a lower dimensional discrete space (the grid) is performed. This grid of neurons can have rectangular or hexagonal topology, differing in the number of immediate neighbours. In principle, there is no advantage nor drawback in using the rectangular (4 neighbours) over the hexagonal (6 neighbours) topology [15]. Despite the fact that networks of large dimensionality are possible, due to visualization issues, typically the ones with one-dimensional or two-dimensional output grids are used.

The learning in the SOM network is unsupervised. During the training process, each input pattern is assigned a winner neuron, which is the one with the smallest Euclidean distance

978-1-4799-1484-5/14/$31.00 ©2014 IEEE

between its weight vector and that input pattern. The winner is denoted as the best matching unit (BMU), being designated with the index c as follows:

$$||\mathbf{x}_i - \mathbf{w}_c|| = \min_l ||\mathbf{x}_i - \mathbf{w}_l|| \ , \ l = (1, 2, ..., m) \quad (1)$$

where $||\cdot||$ is the Euclidean distance, $\mathbf{x}_i$ is a pattern from the data set, $\mathbf{w}$ is a weight vector and m is the total number of neurons.

The SOM networks include three principles [16]: competition, cooperation and adaptation. The winner neurons move toward patterns carrying with them their neighbouring neurons. By training a neural network adjustments are made to the neurons' weights. For each pattern presented to the network, the neurons compete with each other so that the winner is the closest according to a given similarity metric, in this case, the Euclidean distance. Regarding the characteristics of learning in SOM networks, assume the i^{th} input vector

$$\mathbf{x}_i = [x_1 \quad x_2 \quad \cdots \quad x_n]^T \quad (2)$$

and the weight vector of neuron j

$$\mathbf{w}_j = [w_1 \quad w_2 \quad \cdots \quad w_n]^T . \quad (3)$$

In the batch training algorithm, which is the one used in this work, the whole data set is presented at once. At each epoch, the BMUs for all patterns are calculated, and the weight vectors are concurrently updated according to:

$$\mathbf{w}_j(t+1) = \frac{\sum_{i=1}^{N} h_{j,c}(t)\mathbf{x}_i}{\sum_{i=1}^{N} h_{j,c}(t)} \quad (4)$$

where N is the dataset cardinality and $h_{j,c}(t)$ is the value of the j^{th} neurons neighborhood kernel at the location of the BMU of the pattern $\mathbf{x}_i$. The latter monotonically decreases during the training process, and is a function of the distance between neurons j and c in the SOM neuron grid, usually being defined as a Gaussian function:

$$h\left(||\mathbf{r}_c - \mathbf{r}_j||, t\right) = e^{\left(-\frac{||\mathbf{r}_c - \mathbf{r}_j||^2}{2\sigma^2(t)}\right)} \quad (5)$$

where $\mathbf{r}_j$ and $\mathbf{r}_c$ are the positions of neuron j and the winner neuron c in the grid, and σ is the neighborhood radius that monotonically decreases as the learning progresses.

During the training period, the SOM behaves like an elastic network which conforms to the intrinsic shape of the data. Due to the magnification factor, the arrangement of neurons in the input space reflects the density distribution of the data set, so that there is a larger population of neurons in areas where there is a greater amount of patterns, and vice-versa for low density regions.

In general, the SOM size is chosen so that its dimensions a and b (number of rows and columns of the SOM grid) are proportional to the two largest eigenvalues of the data covariance matrix (λ_1 and λ_2) [17]:

$$\frac{a}{b} \approx \sqrt{\frac{\lambda_1}{\lambda_2}} \quad (6)$$

Additionally, the total number of neurons in a SOM network is typically set to $m \approx 5\sqrt{N}$ [8]. The initialization of the neurons' positions in the data space is usually linear, that is, the neurons are allocated across the hyperplane spanned by the eigenvectors related to λ_1 and λ_2.

The characteristics of the data combined with the topologically ordered grid of neurons from the self-organizing maps originate visualization techniques that provide an initial idea of the data distribution [18], which is a key resource for understanding the structure of the data used in knowledge discovery and data mining. Generally, they consist of matrix plots of pairwise similarity measures between neurons and pattern density, both calculated in the input space and transferred to the ordered grid of the output space.

The U-matrix [19] is one of the most well-known visualization techniques. It depicts the Euclidean distances between neurons that are adjacent in the SOM grid. One of its drawbacks concerns its resolution when applied to decreasing map sizes, i.e., on small maps the visualization is generally compromised, while on large maps the definition of clusters becomes increasingly clear in data sets where distance metrics are relevant figures of merit.

Another property commonly depicted is the pattern density, which is taken into account by the P-matrix [20] visualization technique. The latter aims to estimate the data probability density by estimating the Pareto density. The value at each position of the P-matrix represents the number of patterns inside a hypersphere centered in the neuron that occupies that respective position. The radius is a constant given a particular dataset and SOM set-up, being regarded as the Pareto radius.

The pattern to neuron ratio (PNR), which can be defined as

$$PNR = \frac{N}{m} \quad (7)$$

and therefore the size of the map, influence directly the quality of some visualization techniques and clustering algorithms. The PNR is closely related to the figure of merit neurons' utilization N_u [21]:

$$N_u = \frac{1}{m}\sum_{k=1}^{m} u_k \quad (8)$$

where u_k is equal to one if neuron k has an associated pattern, and zero otherwise.

III. Particle Swarm Optimization

Particle swarm optimization (PSO) is a swarm intelligence algorithm that aims to search in the multidimensional problem space so as to maximize or minimize a given multiple input single output (MISO) cost function. Each particle i updates its own position $\mathbf{z}_i$ according to its own best position $\mathbf{p}_i$ (cognitive component), the whole swarm best position $\mathbf{p}_g$ (social component) and a momentum component:

$$\mathbf{z}_i = [z_1 \quad z_2 \quad \cdots \quad z_c]^T \quad (9)$$

$$\mathbf{v}_i(t+1) = momentum(t) + cognitive(t) + social(t) \quad (10)$$

$$momentum(t) = W(t) \times \mathbf{v}_i(t) \quad (11)$$

$$cognitive(t) = c_1 \times \phi_1 \times (\mathbf{p}_i - \mathbf{z}_i(t)) \quad (12)$$

$$social(t) = c_2 \times \phi_2 \times (\mathbf{p}_g - \mathbf{z}_i(t)) \quad (13)$$

$$\mathbf{z}_i(t+1) = \mathbf{z}_i(t) + \mathbf{v}_i(t+1) \quad (14)$$

where t denotes the iteration, $W(t)$ is the inertia function, $\mathbf{v}_i(t)$ is the velocity of particle i at time t. The parameters c_1 and c_2 are the acceleration constants, ϕ_1 and ϕ_2 are values drawn from an uniform distribution in the range $[0; 1]$. Typically, the inertia function is set as a constant or as a function monotonically decreasing with time.

The PSO algorithm takes into account local and global information in the evolution process due to the cognitive and social components, which represent the movement towards $\mathbf{p}_i$ and $\mathbf{p}_g$ respectively, what also prevents loss of information (memory containing good solutions found during the search).

IV. Validity Indices

There is no clustering algorithm suitable for all possible datasets' structures [22]. In this context, validity indices are concerned with the quantitative evaluation of the results obtained by clustering algorithms: geometry, size, density, and number of clusters. Typically, different clustering algorithms favour different types of data structures. Similarly, the validity indices themselves also tend to favour specific types of data distribution, while considering factors such as clusters' compactness, homogeneity, separation, and so on. In this work, the Composed Density between and within clusters (*CDbw*) relative validity index [23][24] is used as the fitness function for the PSO algorithm, and the external validity indices Rand and Adjusted Rand Indices [25], which use the groundtruth labels (classes) of the data, in order to evaluate the final partitions of the data obtained with the proposed clustering algorithm.

A. CDbw Index

Given a dataset partition resulting from the application of a clustering algorithm, the *CDbw* index defines, for each one of the N_{cl} clusters found, subsets containing r representative patterns $\mathcal{V}_i = \{\mathbf{v}_{i1}, \mathbf{v}_{i2}, \cdots, \mathbf{v}_{ir}\}$, where $(1 \leq i \leq N_{cl})$. The clustering validity is assessed by evaluating the intra-cluster density ($Intra$) and clusters' separation (Sep):

$$CDbw = Intra \times Sep \tag{15}$$

The intra-cluster density is defined as

$$Intra = \frac{1}{N_{cl}} \sum_{i=1}^{N_{cl}} \frac{1}{r} \sum_{j=1}^{r} \frac{density_1(\mathbf{v}'_{ij})}{\bar{\sigma}} \tag{16}$$

where $\bar{\sigma}$ is the mean of the standard deviation of the clusters, $\mathbf{v}'_{ij}$ is the j^{th} representative pattern of cluster i shifted towards $\mathbf{v}_{ij}$ by a user defined parameter, and

$$density_1(\mathbf{v}'_{ij}) = \sum_{l=1}^{n_i} f_1(\mathbf{x}_l, \mathbf{v}'_{ij}) \tag{17}$$

where n_i is the number of patterns of the i^{th} cluster, to which $\mathbf{x}_l$ belongs. The function $f_1(\cdot)$ is given by

$$f_1(\mathbf{x}, \mathbf{v}'_{ij}) = \begin{cases} 1, & \text{if } \left\|\mathbf{x} - \mathbf{v}'_{ij}\right\| \leq \bar{\sigma} \\ 0, & \text{otherwise} \end{cases} \tag{18}$$

Finally, the inter-cluster separation defined as:

$$Sep = \frac{\sum_{i=1}^{N_{cl}} \sum_{\substack{j=1 \\ j \neq i}}^{N_{cl}} \|rep_i - rep_j\|}{1 + Inter} \tag{19}$$

where the inter-cluster density is given by

$$Inter = \sum_{i=1}^{N_{cl}} \sum_{\substack{j=1 \\ j \neq i}}^{N_{cl}} \frac{\|rep_i - rep_j\|}{\sigma(i) + \sigma(j)} \times density_2(\mathbf{u}_{ij}) \tag{20}$$

where the parameters rep_i and rep_j represent the closest pair of representative patterns of clusters i and j, $\mathbf{u}_{ij}$ is the mean point between these patterns. The parameters $\sigma(i)$ and $\sigma(j)$ are the standard deviation of clusters i and j respectively, and

$$density_2(\mathbf{u}_{ij}) = \sum_{k=1}^{n_i+n_j} \frac{f_2(\mathbf{x}_k, \mathbf{u}_{ij})}{n_i + n_j} \tag{21}$$

where n_i and n_j are the number of patterns of the clusters i and j respectively, $\mathbf{x}_k$ is a pattern that belongs to cluster i or j, and the function $f_2(\cdot)$ is defined as

$$f_2(\mathbf{x}, \mathbf{u}_{ij}) = \begin{cases} 1, & \text{if } \|\mathbf{x} - \mathbf{u}_{ij}\| \leq \frac{\sigma(i)+\sigma(j)}{2} \\ 0, & \text{otherwise} \end{cases} \tag{22}$$

The *CDbw* validity index is defined for $1 < N_{cl} < N$.

B. Rand and Adjusted Rand Indices

Considering the partition returned by a clustering algorithm and a given groundtruth partition of the data, the Rand Index (R) and Adjusted Rand Index (AR) are given by:

$$R = \frac{tp + tn}{tp + fp + fn + tn} \tag{23}$$

$$AR = \frac{\binom{N}{2}(tp+tn) - [(tp+fp)(tp+fn)+(fn+tn)(fp+tn)]}{\binom{N}{2}^2 - [(tp+fp)(tp+fn)+(fn+tn)(fp+tn)]} \tag{24}$$

where N is the dataset cardinality, tp, tn, fn and tn stand for true positive, true negative, false negative and true negative, respectively, based on whether a pair of data objects are within the same partition or not considering both the output of the clustering algorithm and the groundtruth.

V. Method Description

Before introducing the proposed method, it is important to first define the heuristics used to measure how strong is a given connection between two adjacent neurons i and j of the SOM. Specifically, regarding the evaluation of the connection strengths between the neuron units of the SOM in the matrix output space, the dissimilarities may be based on distances between neurons, and conversely, the similarities may be based on the cardinality of the data subsets they share (pattern density), such as the CONNvis [26] [27] [28], which is based on receptive fields (RF) and is closely related to the pattern to neuron ratio (PNR).

In the context regarding the number of neurons, there are strands of thought that promote the use of small SOM network

sizes, whereas others promote the use large networks, such as *ViSOM* (Visualization-Induced Self-Organizing Maps) [29] and *ESOM* (Emergent SOM) [30], stating that the true properties of the data emerge when a large number of neurons is used, and small network sizes approximate the behaviour of classical k-means [31].

When the neuron utilization starts to decrease giving increasing sizes of the SOM, i.e., when the pattern to neuron ratio starts to decrease significantly (hits dissolution phenomenon), the approach considering Voronoi regions to form subsets of data starts to generate increasingly sparse connection matrices, which translates into the existence of many small connected regions within each data cluster, as there is not enough patterns available for every neuron of the (competitive before cooperative) SOM network [32].

Thus, in this work we consider the hypershere approach instead of Voronoi regions, in order to create the subsets associated with each neuron, as this approach is reasonable in both sides of the spectrum, viz. small and large maps: for small maps with large PNR, the difference of the two approaches is not relevant (Number of non-null connections). However, when increasing the size of the map, the number of connections with the hypersphere approach is perceptibly greater than the one with Voronoi regions.

A. Connection strength between neurons

Let $\mathcal{H}_i$ and $\mathcal{H}_j$ be the subsets of patterns from the data set $\mathcal{U}$ associated with the neurons i and j. The hypersphere approach [33] is inspired by the P-matrix generation. However, the radius r is defined so that every neuron have at least one pattern inside the hypersphere centered at that neuron ($\mathcal{H}_i \subset \mathcal{U} | \mathcal{H}_i \neq \emptyset, \forall i$):

$$\mathcal{H}_i = \{\mathbf{x}_l \in \mathcal{U} \mid \|\mathbf{x}_l - \mathbf{w}_i\| \leq r\} \tag{25}$$

and

$$r = \max_j \left[\min_l \left(\|\mathbf{x}_l - \mathbf{w}_j\| \right) \right] \tag{26}$$

where $l = (1, 2, ..., N)$, $j = (1, 2, ..., m)$, N is the total number of patterns of the dataset and m is the total number of neurons. Therefore, the intersection (shared patterns set), exclusive disjunction (complement set) and union sets $\mathcal{I}_{i,j}$, $\mathcal{M}_{i,j}$ and $\mathcal{H}'_{i,j}$ are defined as:

$$\mathcal{I}_{i,j} = \mathcal{H}_i \cap \mathcal{H}_j \tag{27}$$

$$\mathcal{M}_{i,j} = \mathcal{H}_i \oplus \mathcal{H}_j \tag{28}$$

$$\mathcal{H}'_{i,j} = \mathcal{H}_i \cup \mathcal{H}_j \tag{29}$$

The subsets $\mathcal{I}_i$, $\mathcal{I}_j$, $\mathcal{M}_i$, $\mathcal{M}_j$, $\mathcal{H}'_i$ and $\mathcal{H}'_j$ are obtained by computing whether neuron i or j is the BMU of the patterns belonging to the sets $\mathcal{I}_{i,j}$, $\mathcal{M}_{i,j}$ and $\mathcal{H}'_{i,j}$, respectively.

Thus, the connection strength (or weakness) $s(i, j)$ between two neighbouring neurons i and j may be measured as:

1) The Euclidean distance between neurons i and j:

$$s_1(i, j) = ||\mathbf{w}_i - \mathbf{w}_j|| \tag{30}$$

2) The Euclidean distance between the centroids c of the subsets $\mathcal{I}_i$ and $\mathcal{I}_j$:

$$s_2(i, j) = ||c_{\mathcal{I}_i} - c_{\mathcal{I}_j}|| \tag{31}$$

3) The Euclidean distance between the centroids c of the subsets $\mathcal{H}'_i$ e $\mathcal{H}'_j$:

$$s_3(i, j) = ||c_{\mathcal{H}'_i} - c_{\mathcal{H}'_j}|| \tag{32}$$

4) The Euclidean distance between the centroids c of the subsets $\mathcal{H}_i$ and $\mathcal{H}_j$:

$$s_4(i, j) = ||c_{\mathcal{H}_i} - c_{\mathcal{H}_j}|| \tag{33}$$

5) The Jaccard coefficient [34]:

$$s_5(i, j) = \frac{|\mathcal{I}_{i,j}|}{|\mathcal{H}_i \cup \mathcal{H}_j|} \tag{34}$$

6) The cardinality of the subset $\mathcal{I}_{i,j}$:

$$s_6(i, j) = |\mathcal{I}_{i,j}| \tag{35}$$

7) The Euclidean distance between the centroids c of the subsets $\mathcal{M}_i$ and $\mathcal{M}_j$:

$$s_7(i, j) = ||c_{\mathcal{M}_i} - c_{\mathcal{M}_j}|| \tag{36}$$

Most of the measures described here were used in a committee machine and discounted-reward with minimum path search algorithm, both in order to partition the self-organizing map [35] [32].

B. Proposed Approach

The proposed method is an automatic clustering algorithm that aims to partition the self-organizing map, and subsequently the dataset by relating the patterns to the BMUs in each SOM partition. In order to achieve this objective, the PSO is used to search the coefficients a_r that reflect the importance of each type of connection strength s_r described in the previous subsection, where $(r = 1, 2, ..., 7)$. The PSO fitness function used consists of the validity index *CDbw*. Therefore, the PSO algorithm attempts to search in the multidimensional problem space, the values of a_r that generates strong connected neuron regions that remains in the undirected SOM graph, while weak connections are pruned.

In the first phase of the proposed approach, a SOM network is trained using its standard batch algorithm. In a second phase, for each pair of neurons i and j that share patterns in a 8-neighborhood, their connection strength is measured according to (30)-(36) so as to generate a new dataset S:

$$S = \begin{bmatrix} — & \left(\mathbf{s}^{(1)}\right)^T & — \\ — & \left(\mathbf{s}^{(2)}\right)^T & — \\ & \vdots & \\ — & \left(\mathbf{s}^{(q)}\right)^T & — \end{bmatrix} \tag{37}$$

where

$$\mathbf{s}^{(k)} = \begin{bmatrix} s_1(i, j) \\ s_2(i, j) \\ s_3(i, j) \\ s_4(i, j) \\ s_5(i, j) \\ s_6(i, j) \\ s_7(i, j) \end{bmatrix}, \; i \neq j \; \forall k \tag{38}$$

The constraint of the second phase is used to enforce that all $\mathbf{s}^{(k)}$ has its components $s_r^{(k)}(i,j) \neq 0, \forall r$. The data set S is normalized using z-score so as each feature s_r has mean zero and variance equal to one. The third phase consists of the search performed by the PSO that maximizes the *CDbw* index. Each particle $\mathbf{z}_i$ of the PSO algorithm is a solution to the logistic regression:

$$\upsilon^{(k)} = \sum_{i=0}^{7} a_i \times s_i^{(k)} \tag{39}$$

$$y^{(k)} = \phi(\upsilon^{(k)}) \tag{40}$$

$$\phi(\upsilon) = \frac{2}{1+e^{-2\upsilon}} - 1 \tag{41}$$

and

$$\mathbf{z}_i = [a_0 \quad a_1 \quad a_2 \quad a_3 \quad a_4 \quad a_5 \quad a_6 \quad a_7]^T \tag{42}$$

where $\phi(\cdot)$ is the tan-sigmoid transfer function $\in [-1;1]$, $s_0 = 1$ and a_0 are the bias term and bias coefficient, respectively. The experiments were carried out with and without the bias term and coefficient.

For each PSO particle i, the feedforward propagation is applied by calculating $Y = \phi(S \times z_i^T)$, where the entry $y^{(k)}$ of the vector $Y_{q\times 1}$ is the value of $\phi(\cdot)$ for the sum of the k^{th} row of $S_{q\times 7}$ weighted by that PSO solution. Next, it is assumed the hypothesis that all connections for which $y^{(k)} > 0$ should remain in the SOM graph, otherwise pruned. Next, Considering the remaining connections, connected groups of neurons in the graph are extracted [36]. Finally, a filtering phase that removes groups of neurons smaller than a user defined parameter α are discarded (used to remove interpolating neurons). The same filter is applied to groups of neurons that represents a fraction smaller than α of the data set (minimum cluster size). The following phase consists of calculating the *CDbw* index of the strongly connected groups of neurons that remained. This process is repeated by each particle at each iteration of the PSO algorithm.

As the *CDbw* tends to return higher scores for a great quantity of small connected regions, the process described is very dependant of the parameter α. That is why, the value of α is scanned from 5% to 50%, where it is reasonable to admit a minimum size of cluster equal to 5% of the data. The minimum number of clusters equal to 2 is obtained by setting the maximum value of a α equal to 50%. Therefore, the existence of at least two clusters is enforced. This is in fact another stopping criteria: if the algorithm finds only one big cluster left, it stops, even if α has not yet reached 50%. Overall this is an hierarchical approach to clustering via the α parameter.

In order to determine the final partition, the function of the number of clusters versus the value of α is analysed, so as to search for regions of stability, which are the ones that remain with the same number of clusters while varying α. An approach regarding the inspection of constant intervals of the number of clusters found taken as a function of a clustering algorithm parameter can be found in [37]. As mentioned previously, the *CDbw* value of a large number of small clusters is, in general, higher than the one with few large clusters. Thus, in order to obtain a trade-off between the value of the *CDbw* and the length of the stable region, where the latter tends to be larger for large clusters, the following score function was analysed for each stability interval:

$$F_j = \mu_j L_j^{\beta} \tag{43}$$

where μ_j is the mean value of the *CDbw* within a given stability region j and L_j is its length. Only regions that are at least half the length of the largest region are considered. The values of μ and L are normalized in the range $[1;2]$ and $[0;1]$, respectively. The parameter β implies the importance given to the *CDbw* index or to the length of the stability region.

The interval j with the largest score F_j is selected as the appropriate region, and the output of the PSO for the largest α within that regions is selected, because it implies the same number of clusters with more connected neurons, so that there is lesser neurons to be assigned to the clusters found, process which is susceptible to the assigning algorithm implicit assumptions and metrics. In this work, the remaining neurons were assigned to the clusters by flooding, using Single and Ward linkages [38] [39] constrained to the SOM neighbourhood.

VI. Experiments

In this work, the following toolboxes were used in the experiments:

- SOM Toolbox [40];
- Particle Swarm Optimization Toolbox (PSOt) [41];
- Cluster Validity Analysis Platform (CVAP) [42].

A. Datasets

The proposed method was applied to synthetic and real world data sets from the Fundamental Clustering Problem Suite (FCPS) [30], the UCI Machine Learning Repository [43], as well as artificially generated and inspired from [44]. The Fig. 1 depicts the data sets used in the experiments. As a preprocessing step, linear normalization was applied to the datasets in order to normalize their attributes in the range $[0;1]$ and to prevent problems related to different scales.

B. Parameters Setting

In this work, the experiments were carried out using the following parameters: the map sizes were defined according to (6), linear initialization of the SOM was made in the subspace spanned by the eigenvectors corresponding to the two largest eigenvalues of the covariance matrix of the data (λ_1 and λ_2 of (6)). The maps were trained using the batch mode, as this setting requires the adjustment of less parameters and leads to a faster convergence [9]. The Tables I and II sum up the characteristics of the data sets used in the experiments and depict the summary of the parameters held in common throughout the training of all SOM networks.

The number of particles in the swarm depends on whether the logistic regression includes the bias term or not. If the bias term is included, there are 16 particles in the swarm, or 14 otherwise. The initial positions of particles are set inspired by the superposition theorem: if a component of particle r is

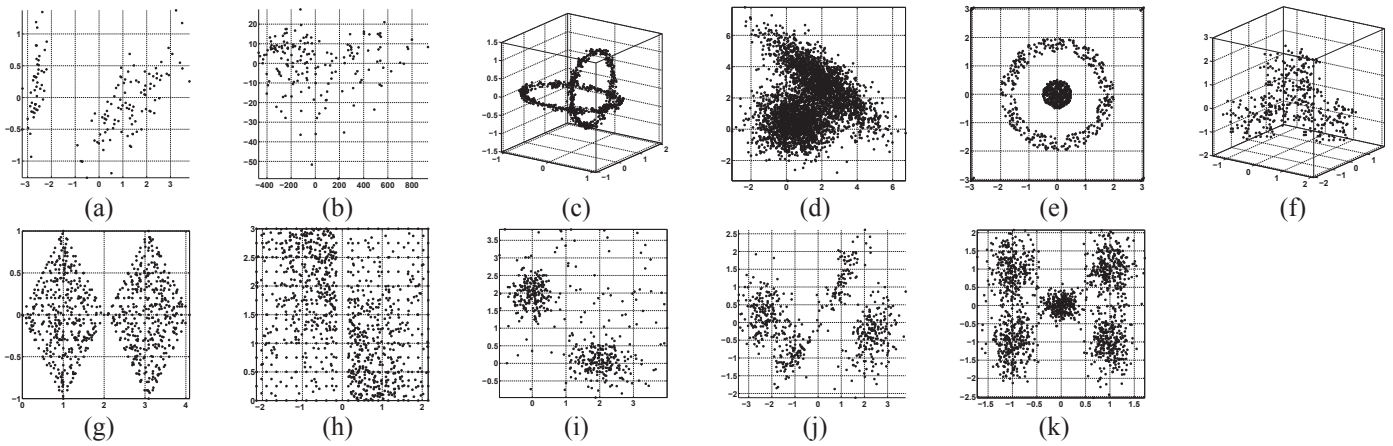


Fig. 1. Illustration of datasets used in the experiments: (a) *Iris*, (b) *Wine*, (c) *Chainlink*, (d) *Engytime*, (e) *Target*, (f) *Tetra*, (g) *Twodiamonds*, (h) *Wingnut*, (i) *D1*, (j) *D2*, and (k) *D3*. The datasets *Iris*, *Wine* and *D2* are depicted using principal component analysis (PCA) projection.

TABLE I. DATASETS' CHARACTERISTICS AND SOM SIZES.

Dataset	Dim.	N	# clusters	Type	Map Size
Iris[1]	4	150	3	Real World	16×4
Wine[1]	13	178	3	Real World	8×8
Chainlink[2]	3	1000	2	Synthetic	18×9
Engytime[2]	2	4096	2	Synthetic	21×15
Target[2]	2	770	2	Synthetic	13×11
Tetra[2]	3	400	4	Synthetic	11×9
Twodiamonds[2]	2	800	2	Synthetic	20×7
Wingnut[2]	2	1016	2	Synthetic	16×10
D1[3]	2	500	2	Synthetic	16×7
D2[3]	4	600	4	Synthetic	18×7
D3[3]	2	1500	5	Synthetic	16×12

[1]UCI
[2]FCPS
[3]Artificially generated

TABLE II. SOM PARAMETER SUMMARY

Parameter	Description
Initialization of neurons	Linear
Training mode	Batch
Neighborhood function	Gaussian
Number of epochs	10^3
Final Radius (σ_f)	1

TABLE III. PSO PARAMETER SUMMARY

Parameter	Description
Maximum Particle Velocity	4
Range of each input variable	$[-100; 100]$
Acceleration Constant 1 (c_1)	2
Acceleration Constant 2 (c_2)	2
Minimum Error	10^{-3}
Number of epochs	500
Inertia Function	Linearly Decreasing from 0.9 to 0.4

set to 100 or -100 for a coefficient a_i, it simultaneously set to zero all the other coefficients a_j $(j \neq i)$, so as only the output of measure s_i is observed, and then the swarm moves toward the most influential measure s_i according to the *CDbw* index. The inertia function decreases linearly from 0.9 to 0.4. until iteration 400. As a stopping criteria, if the error does not change over 50 epochs, the search ends.

C. Results

The results obtained with the proposed method while using the parameters illustrated on Tables I to III are presented in detail for the *Tetra* dataset through the Figs. 2 and 3. For the remaining datasets only the performance is depicted in Tables IV and V. After the strongly connected groups (Fig. 3a) are found within the graph for the largest α of the stable region (Fig. 2), the remaining neurons are assigned to the clusters found by flooding with Single (CDbw S) and Ward linkages (CDbw W) in Figs. 3b and 3c, respectively. Next, their labels are carried back to data set.

The results obtained with the proposed system were compared to the ones obtained by the watershed [45] algorithm. The watershed algorithm was applied to the U-matrices generated by each trained SOM after an image processing (morphological opening and closing) [46], in which the area size was set to half the maximum dimension of the map [47].

The Tables IV and V present the simulation results. The number of clusters found (N_{cl}) using the proposed method is depicted in Table IV, as well as the targeted external validity indices Rand Index (R) and Adjusted Rand (AR): given the vector quantization provided by the SOM, patterns are associated with a given class according to the majority

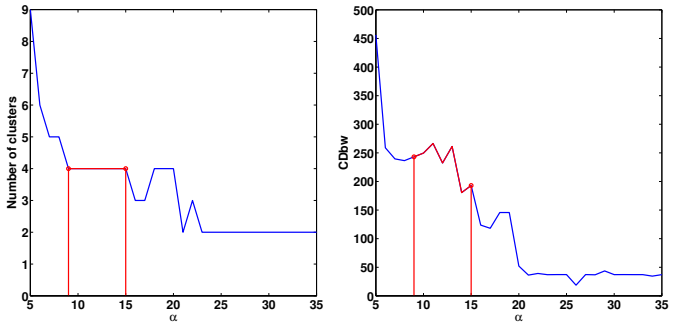
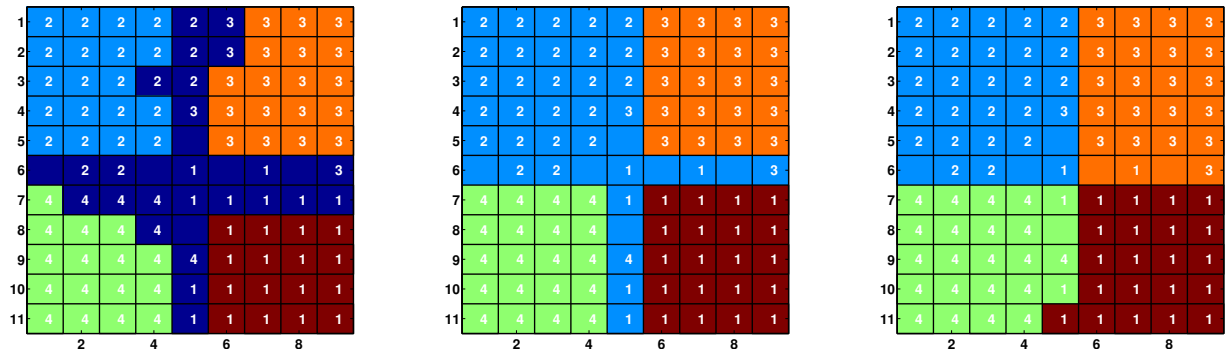


Fig. 2. (a) Number of clusters versus α and (b) *CDbw* versus α. The stable region is depicted in red for the number of clusters and also in red for the associated values of the *CDbw* index.



(a) (b) (c)

Fig. 3. (a) Output of the proposed algorithm for the largest α of the selected stable region (*Tetra* dataset), representing the strongly connected neurons of each cluster. Output of the flooding algorithm using (b) Single Link and (c) Ward Link for the same dataset. The label obtained by simple voting is also depicted for all three matrix plots.

TABLE IV. RESULTS

Dataset	Target Solution		CDbw						Watershed
			With Bias			Without Bias			
	R	AR	β	P	N_{cl}	β	P	N_{cl}	N_{cl}
Iris	0.9656	0.9222	[0.2; 1.0]	0.7031	2	[0.3; 1.0]	0.7813	3	2
Wine	0.9691	0.9310	[0.1; 1.0]	0.9219	3	[0.1; 1.0]	0.8594	3	3
Chainlink	1.0000	1.0000	[0.3; 1.0]	0.9938	2	—	—	—	2
Engytime	0.9358	0.8716	[0.3; 1.0]	0.7111	2	[0.4; 1.0]	0.6540	2	2
Target	1.0000	1.0000	[0.1; 1.0]	0.9790	2	[0.2; 1.0]	0.5385	2	2
Tetra	0.9950	0.9867	[0.1; 0.4]	0.6970	4	[0.1; 1.0]	0.6566	4	4
Twodiamonds	1.0000	1.0000	[0.2; 1.0]	1.0000	2	[0.1; 1.0]	0.8286	2	2
Wingnut	0.9961	0.9921	[0.1; 1.0]	0.6750	2	[0.1; 1.0]	0.4375	2	4
D1	1.0000	1.0000	[0.1; 1.0]	0.9196	2	[0.1; 1.0]	0.7589	2	2
D2	0.9961	0.9904	[0.1; 0.2]	0.7778	4	[0.1; 1.0]	0.8571	4	4
D3	0.9900	0.9686	[0.1; 0.9]	0.7917	5	[0.1; 1.0]	0.7917	5	5

TABLE V. RESULTS

Dataset	With Bias				Without Bias					
	CDbw S		CDbw W		CDbw S		CDbw W		Watershed	
	R	AR	R	AR	R	AR	R	AR	R	AR
Iris	0.7763	0.5681	0.7763	0.5681	**0.9267**	**0.8340**	0.9124	0.8017	0.7763	0.5681
Wine	**0.9349**	**0.8538**	0.9349	0.8537	0.9220	0.8248	0.9349	0.8537	0.9220	0.8249
Chainlink	**1.0000**	**1.0000**	**1.0000**	**1.0000**	0.7175	0.4349	0.6756	0.3510	**1.0000**	**1.0000**
Engytime	0.8182	0.6365	0.9120	0.8239	0.8572	0.7143	0.9053	0.8107	**0.9173**	**0.8346**
Target	**0.9851**	**0.9702**	**0.9851**	**0.9702**	0.6292	0.2643	0.5005	0.0011	**0.9851**	**0.9702**
Tetra	0.9603	0.8938	**0.9803**	**0.9473**	0.9603	0.8938	**0.9803**	**0.9473**	0.9776	0.9401
Twodiamonds	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	0.6076	0.2155
Wingnut	**0.9882**	**0.9765**	0.8006	0.6011	**0.9882**	**0.9765**	0.7481	0.4961	0.7271	0.4539
D1	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
D2	**0.9928**	**0.9820**	0.9895	0.9738	**0.9928**	**0.9820**	0.9895	0.9738	0.8544	0.6652
D3	0.9757	0.9240	0.9879	0.9879	0.9757	0.9240	0.9879	0.9620	**0.9889**	**0.9654**

of the classes to which its BMU is associated. By voting, all patterns associated with a neuron are therefore taken as the same class. The parameter β was varied between 0.1 and 1, and the values to which the correct number of clusters was found are also depicted. The parameter P is the ratio of neurons belonging to the clusters found to the total number of neurons m, conversely, $(1 - P)$ consists of the proportion of neurons not assigned to any cluster. The R and AR were used in order to evaluate the results obtained in the experiments, which are summed up in Table V.

The performance achieved by the proposed method and the watershed algorithm were comparable considering the *Chainlink*, *Target* and *D1* datasets. Considering the dataset *Chainlink*, the parameter β was set to 0.3 for the simulation without bias. For the datasets *Wine*, *Tetra* and *D2*, the proposed method performed slightly better. Moreover, for the *Iris*, *Twodiamonds* and *Wingnut* the results obtained with the proposed method

were significantly better. The Watershed algorithm achieved slightly better results for the *Engytime* and *D3* datasets. The poor performance considering the *Twodiamonds* and *Wingnut* datasets may be due to the fact that solely the Euclidean distances between neurons may be not enough to accurately partition the data with bridges and varying density within the clusters' separation frontier. It must be noted that enhanced results using the watershed algorithm may be achieved by carefully fine tuning the area size used in the morphological image processing.

Although the compared methods have shown a good overall performance in terms of R and AR, the proposed method demonstrated itself more consistent due to the fact that the correct number of clusters was identified in more simulations than the other method, and considering that the best results (with and without bias and both types of flooding) were above 0.92 for the Rand Index and 0.83 for the Adjusted Rand Index.

VII. Analysis and Discussion

As expected, the logistic regression that includes the bias term has a majority of negative coefficients for the measures based on distances (a_1, a_2, a_3 and a_7) and a majority of positive coefficients for the measures based on density (a_5 and a_6). This conforms with the fact that clusters have high density and small distances between its patterns (Fig 4). Surprisingly the majority of the coefficients a_4 are positive. On the other hand, for the logistic regression that does not include the bias term, the previous statement hold true only for the coefficients a_2, a_4, a_5 and a_6, which also conforms with the fact that the coefficients have to find an adjustment to compensate the absence of the bias term in the best possible way while being subjected to the constraint of having to pass through the origin (Fig. 5).

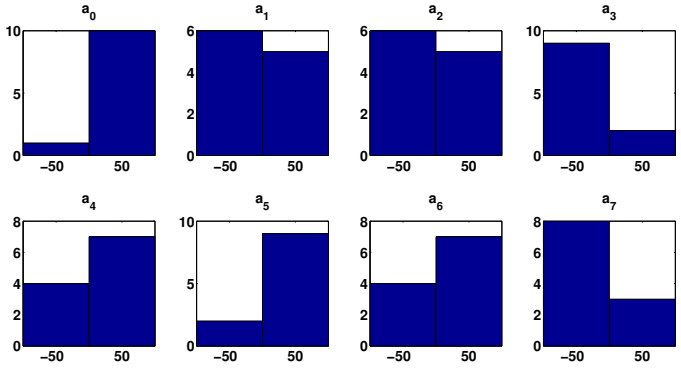


Fig. 4. Histogram of the values of a_i divided between positive and negative for the range $[-100; 100]$ when the bias term was used.

Although no generalization can be made, values of β between $[0.2; 0.3]$ were suitable to the majority of the datasets. It is noticeable that the flooding algorithm, that is, the algorithm that assigns the non-labelled neurons to the clusters found, plays an important role in the performance of the proposed method, mainly when the parameter P is low: even if the strongly connected groups of neurons are correctly found as the core of the clusters (with high R and AR), the process of assigning the remaining neurons, when they are a non negligible part of the SOM, leads to very different

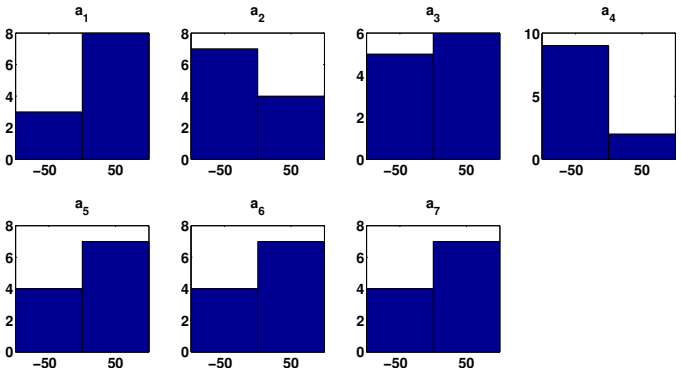


Fig. 5. Histogram of the values of a_i divided between positive and negative for the range $[-100; 100]$ when the bias term was removed.

performances, as can be observed for the datasets *Engytime* and *Wingnut* in Table IV, that although have found the two clusters cores reasonably accurately, flooding with Single link and Ward has led to different performances, as each of which has approximately 30% of the map as non assigned neurons.

VIII. Conclusions and Future Work

A clustering of the self-organizing map using particle swarm optimization with fitness function set as the *CDbw* validity index was presented. The particles of the PSO algorithm contain the coefficients to which each type of measure s_r is multiplied in a linear combination of all the seven measures defined with the subsets of patterns inside hyperspheres centered in each neuron. The system aims to find a stable partition of the map by analysing the trade-off between the length of regions of stability defined by varying the minimum size of the clusters (parameter α) and the mean value of the *CDbw* in that region, which is done by tuning the parameter β. The final result of the proposed method is dependent on flooding algorithm used to assign unlabeled neurons to the clusters found when the parameter P is low.

Future works will focus on examining other validity indices as the fitness function. As the method aims to find global coefficients a_r that multiplies all connections s_r, current focus consists of examining local coefficients, so that each pair of neurons has its own set of coefficients. The influence of a feedforward neural network with hidden layers so as to form more complex functions is also being considered, as well as the analysis of the influence of the map size in the performance of the method.

References

[1] D. T. Larose, *Discovering knowledge in data : an introduction to data mining*. John Wiley & Sons, 2005.

[2] P.-N. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining*. Addison Wesley, 2006.

[3] W. E. Wright, "Gravitational clustering," *Pattern Recognition*, vol. 9, no. 3, pp. 151–166, 1977.

[4] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," *ACM Computing Surveys*, vol. 31, no. 3, pp. 264–323, Sep. 1999.

[5] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biological Cybernetics*, vol. 43, no. 1, pp. 59–69, 1982.

[6] J. A. F. Costa and H. Yin, "Gradient-based SOM clustering and visualisation methods," in *Neural Networks (IJCNN), The 2010 International Joint Conference on*, July 2010, pp. 1–8.

[7] H. Ritter, "The handbook of brain theory and neural networks," M. A. Arbib, Ed. MIT Press, 1995, ch. Self-organizing feature maps: Kohonen maps, pp. 846–851.

[8] J. Vesanto and E. Alhoniemi, "Clustering of the self-organizing map," *Neural Networks, IEEE Transactions on*, vol. 11, no. 3, pp. 586–600, May 2000.

[9] T. Kohonen, "Essentials of the self-organizing map," *Neural Networks*, vol. 37, no. 0, pp. 52 – 65, 2013.

[10] R. Xu, J. Xu, and D. C. Wunsch II, "A Comparison Study of Validity Indices on Swarm-Intelligence-Based Clustering," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 42, no. 4, pp. 1243–1256, Aug 2012.

[11] R. C. Eberhart and Y. Shi, "Particle swarm optimization: developments, applications and resources," in *Evolutionary Computation, 2001. Proceedings of the 2001 Congress on*, vol. 1, 2001, pp. 81–86 vol. 1.

[12] J. Kennedy, R. C. Eberhart, and Y. Shi, *Swarm Intelligence*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001.

[13] R. Storn and K. Price, "Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[14] K. V. Price, R. M. Storn, and J. A. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization*, ser. Natural Computing Series, G. Rozenberg, T. Bäck, J. N. Kok, H. P. Spaink, and A. E. Eiben, Eds. Germany: Springer-Verlag Berlin Heidelberg, 2005.

[15] A. Ultsch and L. Herrmann, "The Architecture of Emergent Self-Organizing Maps to Reduce Projection Errors," in *Proceedings of the European Symposium on Artificial Neural Networks (ESANN'2005)*, 2005, pp. 1–6.

[16] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd ed. Prentice Hall, 1999.

[17] T. Kohonen, *Self-Organizing Maps*, 3rd ed., ser. Springer Series in Information Sciences, T. S. Huang, T. Kohonen, and M. R. Schroeder, Eds. Springer-Verlag Berlin Heidelberg New York, 2001, vol. 30.

[18] J. Vesanto, "SOM-based data visualization methods," *Intelligent Data Analysis*, vol. 3, no. 2, pp. 111 – 126, 1999.

[19] A. Ultsch and H. P. Siemon, "Kohonen's self organizing feature maps for exploratory data analysis," in *Proceedings of International Neural Networks Conference (INNC)*. Kluwer Academic Press, 1990, pp. 305–308.

[20] A. Ultsch, "Maps for the visualization of high-dimensional data spaces," in *Proceedings of the Workshop on Self-Organizing Maps (WSOM'03)*, 2003, pp. 225–230.

[21] Y. ming Cheung and L. Law, "Rival-Model Penalized Self-Organizing Map," *Neural Networks, IEEE Transactions on*, vol. 18, no. 1, pp. 289–295, Jan 2007.

[22] J. Kleinberg, "An impossibility theorem for clustering," in *Advances in Neural Information Processing Systems*, S. Becker, S. Thrun, and K. Obermayer, Eds. MIT Press, 2002, pp. 446–453.

[23] M. Halkidi and M. Vazirgiannis, "Clustering Validity Assessment Using Multi Representatives," in *Proc. SETN Conf*, 2002, pp. 237–249.

[24] ——, "A density-based cluster validity approach using multi-representatives," *Pattern Recognition Letters*, vol. 29, no. 6, pp. 773 – 786, 2008.

[25] L. Hubert and P. Arabie, "Comparing partitions," *Journal of Classification*, vol. 2, no. 1, pp. 193–218, 1985.

[26] K. Tasdemir and E. Merenyi, "Exploiting data topology in visualization and clustering of self-organizing maps," *Neural Networks, IEEE Transactions on*, vol. 20, no. 4, pp. 549–562, April 2009.

[27] K. Tasdemir, "Graph based representations of density distribution and distances for self-organizing maps," *Neural Networks, IEEE Transactions on*, vol. 21, no. 3, pp. 520–526, March 2010.

[28] K. Tasdemir, P. Milenov, and B. Tapsall, "Topology-based hierarchical clustering of self-organizing maps," *Neural Networks, IEEE Transactions on*, vol. 22, no. 3, pp. 474–485, March 2011.

[29] H. Yin, "ViSOM - a novel method for multivariate data projection and structure visualization," *Neural Networks, IEEE Transactions on*, vol. 13, no. 1, pp. 237–243, Jan 2002.

[30] A. Ultsch and F. Mörchen, "ESOM-Maps : tools for clustering , visualization , and classification with Emergent SOM," University of Marburg, Germany, Tech. Rep. 46, 2005.

[31] A. K. Jain, "Data clustering: 50 years beyond k-means," *Pattern Recognition Letters*, vol. 31, no. 8, pp. 651 – 666, 2010.

[32] L. E. B. da Silva and J. A. F. Costa, "Clustering the self-organizing map based on the neurons associated pattern sets," in *Proceedings of 1st BRICS Countries Congress and 11th Brazilian Congress on Computational Intelligence BRICS CCI & CBIC 2013*, 2013.

[33] L. E. B. Silva and J. A. F. Costa, "Clustering, noise reduction and visualization using features extracted from the self-organizing map," in *Intelligent Data Engineering and Automated Learning IDEAL 2013*, ser. Lecture Notes in Computer Science, H. Yin, K. Tang, Y. Gao, F. Klawonn, M. Lee, T. Weise, B. Li, and X. Yao, Eds. Springer Berlin Heidelberg, 2013, vol. 8206, pp. 242–251.

[34] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*. Prentice Hall, 1988.

[35] L. E. Brito da Silva and J. A. Ferreira Costa, "Clustering the self-organizing map through the identification of core neuron regions," in *Neural Networks (IJCNN), The 2013 International Joint Conference on*, Aug 2013, pp. 1–8.

[36] R. E. Tarjan, "Depth-first search and linear graph algorithms," *SIAM Journal on Computing*, vol. 1, no. 2, pp. 146–160, 1972.

[37] M. Goncalves, M. De Andrade Netto, J. Ferreira Costa, and J. Zullo, "Data clustering using self-organizing maps segmented by mathematic morphology and simplified cluster validity indexes: an application in remotely sensed images," in *Neural Networks, 2006. IJCNN '06. International Joint Conference on*, 2006, pp. 4421–4428.

[38] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. John Wiley & Sons, 2000.

[39] J. H. Ward, "Hierarchical grouping to optimize an objective function," *Journal of the American Statistical Association*, vol. 58, no. 301, pp. 236–244, 1963.

[40] J. Vesanto, J. Himberg, E. Alhoniemi, and J. Parhankangas, "Self-Organizing Map in Matlab: the SOM Toolbox," in *Proceedings of the Matlab DSP Conference*, 1999, pp. 35–40.

[41] B. Birge, "PSOt - a particle swarm optimization toolbox for use with Matlab," in *Swarm Intelligence Symposium, 2003. SIS '03. Proceedings of the 2003 IEEE*, April 2003, pp. 182–186.

[42] K. Wang, B. Wang, and L. Peng, "CVAP: Validation for Cluster Analyses," *Data Science Journal*, vol. 8, pp. 88–93, 2009.

[43] K. Bache and M. Lichman, "UCI machine learning repository," 2013. [Online]. Available: http://archive.ics.uci.edu/ml

[44] D. Hamad, C. Firmin, and J.-G. Postaire, "Unsupervised pattern classification by neural networks," *Mathematics and Computers in Simulation*, vol. 41, no. 12, pp. 109 – 116, 1996.

[45] J. A. F. Costa, "Clustering of complex shaped data sets via kohonen maps and mathematical morphology," in *Proceedings of the SPIE, Data Mining and Knowledge Discovery*, B. Dasarathy, Ed., vol. 4384, 2001, pp. 16–27.

[46] E. R. Dougherty and R. A. Lotufo, *Hands-on Morphological Image Processing*, ser. Tutorial Texts in Optical Engineering. SPIE Publications, Jul. 2003, vol. TT59.

[47] J. A. F. Costa, "Uma nova abordagem para visualização e detecção de agrupamentos em mapas de kohonen baseado em gradientes das componentes," *Learning and NonLinear Models*, vol. 9, pp. 20–31, 2011.