Clustering, Noise Reduction and Visualization Using Features Extracted from the Self-Organizing Map

Leonardo Enzo Brito da Silva and José Alfredo Ferreira Costa

Universidade Federal do Rio Grande do Norte
Programa de Pós-Graduação em Engenharia Elétrica e de Computação
Natal, RN, Brasil
{leonardoenzob,jafcosta}@gmail.com

Abstract. This paper presents an analysis of a feature space generated by extracting properties related to pattern density and Euclidean distances between neurons from the self-organizing map network. Hence, along with the weight vector, each neuron has a 2-D feature vector associated with it, whose components are extracted from the U-matrix and a hit matrix, where latter is based on hyperspheres centered on each neuron. This collection of feature vectors, that represents the neurons of the network, is partitioned into different groups, and their labels are carried back to the data space as well as the neuron grid, in order to perform the tasks of clustering, noise reduction and visualization. Experiments were carried out using synthetic and real world data sets.

Keywords: self-organizing maps, clustering, noise reduction, visualization, feature extraction.

1 Introduction

The increasingly quantity of data produced in the modern world have maximized the necessity for understanding and exploiting the information existent [1][2], while simultaneously overcoming its quality related problems. The self-organizing maps (SOM) [3][4] are artificial neural networks widely used in the data mining field for partitioning the data into similar groups and as a tool for visualization through low dimensionality projections of multidimensional data, due to the fact that classic clustering algorithms may be applied to the SOM neurons [5], and many visualization techniques associates data characteristics to the topologically ordered neuron grid, so that an insight of the data distribution may be obtained [6].

This paper focuses on performing the clustering task, noise filtering and visualization through partitioning the self-organizing map according to the distribution of feature vectors associated with the neurons, which enclose characteristics related to pattern density and distances between neurons.

The paper is organized as follows. Section 2 provides a brief review of the SOM network and related visualization methods; also the derived feature space is defined. In Section 3, the proposed approach is described. The results of the experiments and discussions are presented in Sections 4 and 5, respectively.

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 242–251, 2013.

2 Self-Organizing Maps and Derived Feature Space

The self-organizing maps are neural networks based on unsupervised learning. They are composed by a lattice of neurons, each one associated with a weight vector $\boldsymbol{w_i}$ in the p-dimensional data space. In this work, the SOM is trained using the *batch mode,* in which the whole data set is fed at once to the network. At each epoch, the BMUs (neurons with the smallest Euclidean distance from their associated weight vectors to the input patterns) for all patterns are determined, so the M neurons of the network can be simultaneously updated according to (1):

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

where t denotes the iteration, $\boldsymbol{w}_j(t)$ is a weight vector associated with the j[th] neuron, $\mathbf{x}_i$ is the i[th] input pattern, N is the total number of patterns, and $h_{j,c}(t)$ is the neighborhood kernel, which is usually a Gaussian function defined by the neighborhood radius σ.

In order to inspect the relative sizes and positions of clusters in a given data set, visualization techniques must be applied to a trained SOM network, which are typically matrix plots of Euclidean distances between neurons or pattern density, such as the popular U-matrix [7] and P-matrix [8], respectively. The U-matrix consists of an image that portrays the Euclidean distances in the data space between the SOM neurons. Regions of small and large distances are often regarded as clusters and their borders, respectively. The P-matrix is generated by counting the number of patterns inside a Pareto hypersphere centered on each neuron.

The proposed feature space consists of the following concept: for each neuron in the grid, besides the weight vector in the data space, there is also associated with it a 2-D feature vector, which carries meaningful information from the data set. The characteristic of the feature vector distribution is analyzed so as to diminish the noise present in the data set, to perform a clustering task or to visualize similar groups of neurons. In this work, the feature vector $\boldsymbol{f_i}$ of a neuron $\boldsymbol{w_i}$ has two components:

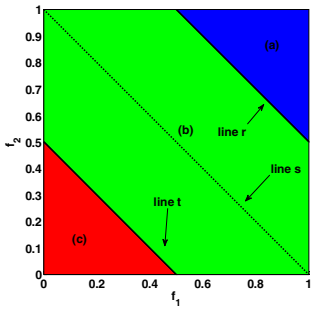
$$\boldsymbol{f_i} = [f_1 \quad f_2] \tag{2}$$

The first component (f_1) is the value extracted from the neuron's corresponding position in U-matrix. However, it is negated with the purpose of becoming a measure of similarity instead of dissimilarity. The second component (f_2) is obtained from the matrix plot in which every position entails the number of patterns inside a hypersphere centered in that neuron, where the radius is determined as the minimum so all neurons have at least one pattern inside its respective hypersphere. This approach is inspired from the P-matrix visualization method. The radius of this hit matrix H is then defined as:

$$r = \max_j \left(\min_i \left\| \mathbf{x}_i - \boldsymbol{w}_j \right\| \right) \tag{3}$$

where $i = (1,2,\ldots,N)$, and $j = (1,2,\ldots,M)$.

It is assumed that, in general, the feature space should comprise three main regions (Fig. 1), which correspond to neurons that are positioned in the core of the clusters (region 1), in their frontier (region 2) or between them (region 3). The values of the components are normalized in the range $[0;1]$.



Components		f_1	
		High	*Low*
f_2	*High*	Cluster's neurons (a)	Boundary neurons (b)
	Low	Boundary neurons (b)	Interpolating neurons (c)

Fig. 1. The three colored regions correspond to regions that enclose interpolating (region 3 - red), boundary (region 2 - green) and cluster's neurons (region 1 - blue). The lines *r*, *s*, and *t* symbolize generic hyperplanes that divide the feature space.

3 Proposed Approach

After training a SOM network, the feature vectors are generated with the values obtained from the U-matrix and the hit matrix H, so an analysis can be carried out to perform noise reduction, clustering or visualization. The main goal is to identify interpolating neurons, i.e. those neurons that do not belong to a cluster and just link close-knit groups of neurons, which represent the core of the clusters.

In order to partition the data of the feature space, the k-means algorithm [9] and a competitive network [10] were used. The assumption described in Section 2 was simplified in the sense that only two regions were considered, that is, the generic line *s* of Fig. 1, so one part of the boundary neurons would be included in the interpolating neurons subset and the other to the clusters' neurons subset. Although not critical to most of the data sets used in the experiments, this simplification of the hypothesis was not functional considering two data sets, and thus it cannot be generally used. If the points in the feature space regarding data sets with noise may be modeled as an exponential function with a reasonable goodness-of-fit statistics of the curve fitting, then the partitioning of the feature vectors into two subsets corresponding to the interpolating neurons and the clusters' neurons may also be accomplished using the common elbow criterion as a threshold. After the segmentation in the feature space, the subset with the largest median of the Euclidean norm is considered as the core of the clusters. For the purpose of reducing the noise, the final step consists of determining the

patterns from which each neuron is related, and then eliminate from the data set the ones that are associated with interpolating neurons. Conversely, when the task being considered is clustering, then the clusters are identified and labeled in the SOM grid using 4-neigborhood connected components labeling (CCL) [11]. Regions that are not connected and are smaller than a percentage of the total number of SOM neurons (α) are disregarded. All the remaining neurons are classified through the k Nearest Neighbor algorithm (kNN) [12], in which the parameter k was set to 1. At last, the patterns are labeled using the partitioned SOM.

In the case that a data set has several close clusters with extremely different densities, the proposed method may not be directly applied as described. Nonetheless, different regions of similar neurons of the SOM may be highlighted by visualizing the partitioning of the feature space into different number of regions.

4 Experiments

The SOM Toolbox [13] was used to implement the SOM networks, which were trained using the batch mode (1000 epochs) and linear initialization. The Table 1 sums up the characteristics of the data sets used in the experiments, which are depicted in Fig. 2. They come from [14-17], and the data set *D2* was artificially generated. Such databases were preprocessed using the linear normalization, i.e. the data sets' attributes were normalized in the $[0;1]^n$ cube.

Table 1. Data sets characteristics

Data set	*Dim.*	*Size*	*Clusters*	*Type*	*Main problem*
DS3	2	10000	9	Synthetic	Clusters with different shapes / noise
Engytime	2	4096	2	Synthetic	Overlapping clusters
Hepta	3	212	7	Synthetic	Clusters with different variances
Tetra	3	400	4	Synthetic	Very close clusters
Twodiamonds	2	800	2	Synthetic	Clusters connected by a bridge
Wingnut	2	1016	2	Synthetic	Clusters with variable densities
D2	4	600	4	Synthetic	Data set with high dimensionality
D3	2	1500	5	Synthetic	Gaussian clusters
Wine	13	178	3	Real World	Data set with high dimensionality

The analysis carried out over the feature space using the proposed approach is presented in detail considering the *DS3* data set through the Figs. 3 to 7. A SOM network of size 100x100 was trained with the parameters previously described and final neighborhood radius equal to zero. The Fig. 3 depicts the U-matrix and P-matrix generated using the SOMVIS Package, as well as the proposed hit matrix H. Considering the Fig. 3 'b' and 'c', both of which contains information of pattern density, the definition of the clusters boundaries are sharper in item 'c', and thus the feature space is generated using the values of the matrixes of the items 'a' and 'c' (Fig. 4a).

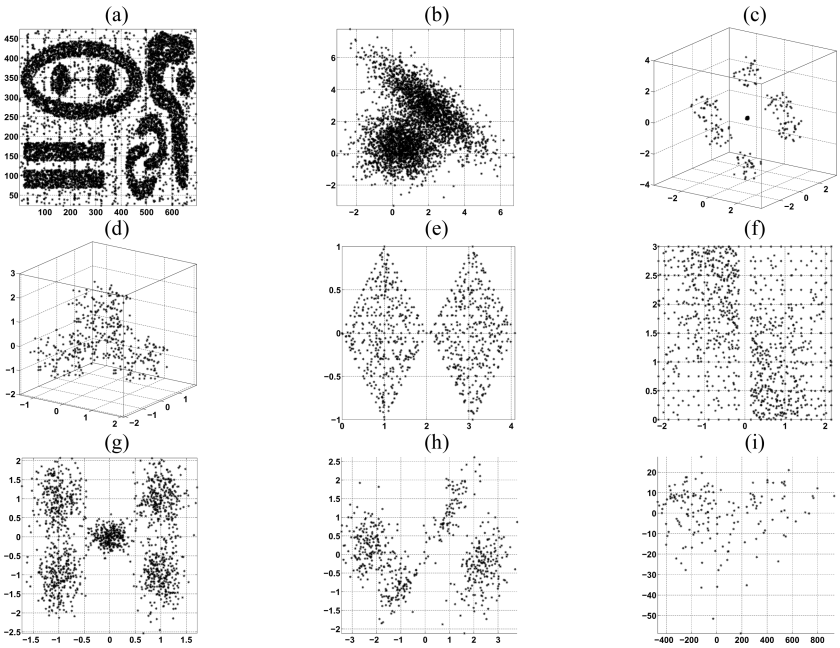


Fig. 2. Illustration of the data sets: (a) *DS3*, (b) *Engytime*, (c) *Hepta*, (d) *Tetra*, (e) *Twodiamonds*, (f) *Wingnut*, (g) *D3*, (h) *D2* and (i) *Wine*, both with a 2-D PCA projection.

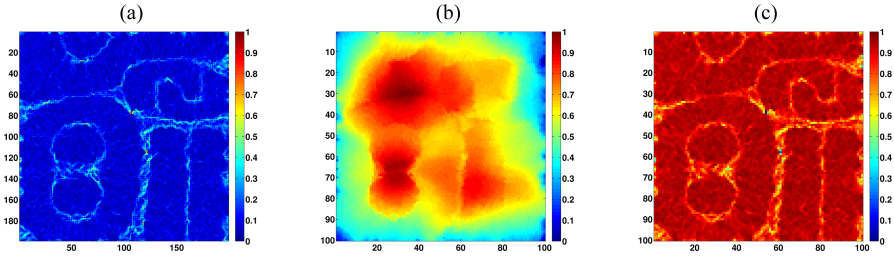


Fig. 3. (a) U-matrix, (b) P-matrix, and (c) H-matrix (with the radius defined by Eq. 3) of the SOM network trained with the *DS3* data set.

The next stage consists of partitioning the feature space into two regions of similar neurons. The region with the largest median of the Euclidean norm is regarded as the one representing the core of the data set's clusters. This may be achieved using any kind of clustering algorithm; in the case of this particular map, the k-means algorithm was applied. If the objective is to perform the noise filtering, then patterns associated with the interpolating neurons (mostly noise) are excluded from the data set (Fig. 4c).

The feature space may reveal different regions of neurons with similar characteristics as a function of the number k of the partitions of this space, and therefore it can be used as a tool for visualization. In Fig. 5, the dependence of the method with the value of the parameter k is depicted. As expected, when partitioning the feature space into 2 regions, the boundary neurons are included in both the interpolating neurons

subset and core of the clusters subset. Considering k equal to 3 leads to the appearance of the boundary neurons as a group of its own. Finally, increasing the value of k turns stricter the neurons' intra-group similarity.

In order to automatically find the clusters of the *DS3* data set, the CCL was applied to the image of Fig. 5a, and the result obtained is depicted in Fig. 6a. After labeling the group of neurons related to the clusters, the interpolating neurons may be labeled by flooding or by the *k*NN algorithm, and in this work the latter is used (Fig. 7a). At last, the segmented SOM is used to label the data set.

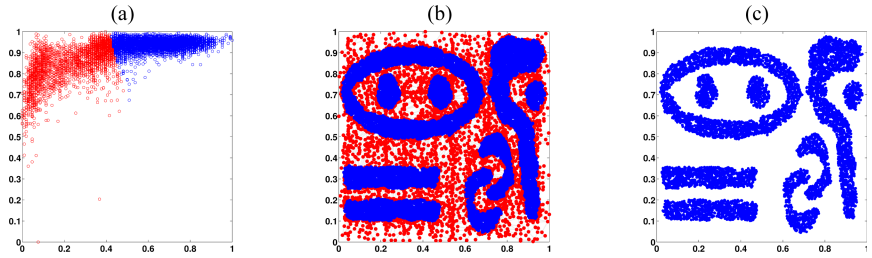


Fig. 4.(a) Feature space data divided into 2 regions using the k-means algorithm. (b) The 2 groups of neurons viewed in the data space. Classified neurons of the SOM network are shown in red (interpolating neurons subset) and blue (cluster's neurons subset). Neurons of the same group are represented with the same colors in items 'a' and 'b'. (c) Filtered data set.

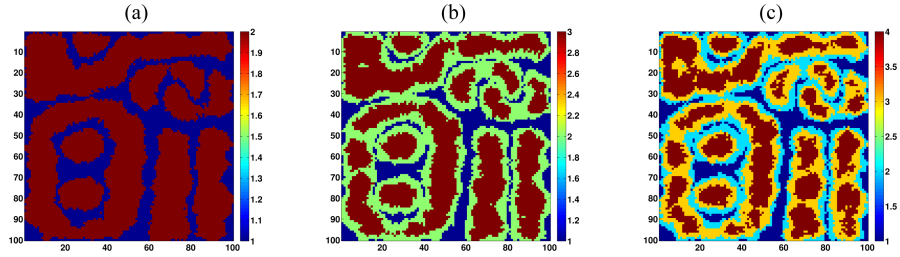


Fig. 5. Matrix plots with the same size as the network lattice. The partitions of the feature space are shown for (a) k = 2, (b) k = 3, and (c) k = 4, in which each color correspond to neurons of the same group.

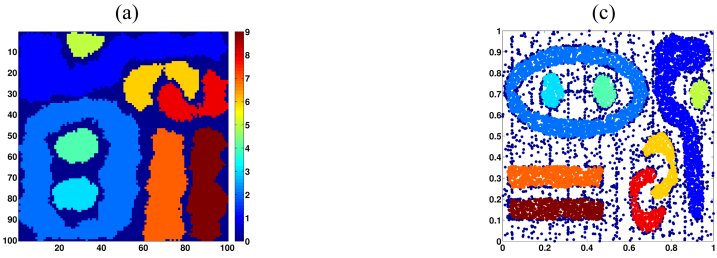


Fig. 6. (a) Labeled neurons of the SOM grid using CCL. (b) Labeled data set.

The Fig. 7 also depicts the results of partitions obtained by the watershed [18] and k-means algorithms, the latter applied over the SOM neurons in the data space, with the parameter k defined by the best value of the Davies-Bouldin index (DBI) [19]. The watershed algorithm was applied to the U-matrix image of Fig. 3a after a morphological image processing (filtering through area open and area close) [20], where the area size was set to half the maximum dimension of the map [17]. The proposed method was able to uncover the 9 clusters of the *DS3* data set, whereas the watershed and k-means algorithms found 8 and 30 clusters, respectively.

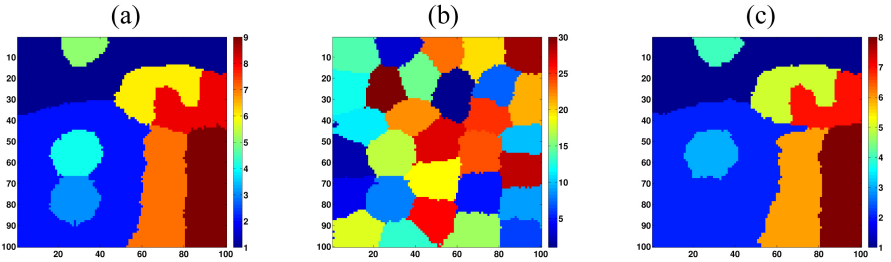


Fig. 7. (a) Labeled SOM neurons trough the (a) k-means in the feature space with posterior CCL, (b) k-means in the data space over the SOM neurons, with the DBI criterion for the choice of the parameter k, (c) watershed algorithm over the U-matrix of Fig. 3a.

In order to perform the clustering task for the remaining data sets, several experiments were carried out using diverse values for the following parameters: map size, grid type, final neighborhood radius (σ_f), the minimum cluster size (α), and number of neuron regions (k). The majority of the map sizes was defined according to [3][5] and ended up to be rectangular, the others were defined as square maps, as they led to better results (Table 2). Different sizes were considered as multiples of the original size. The scale factor S is such that $S = \{0.75,1,1.25,1.5,1.75,2\}$.

Table 2. Summary of the parameters used throughout the experiments

Data set	*Wine*	*Engytime*	*Hepta*	*Tetra*	*Twodiamonds*	*Wingnut*	*D2*	*D3*
Map size (S=1)	8x8	21x15	10x10	10x10	20x7	10x10	18x7	16x12
k	2	2	3	3	2	2	2	2
α (%)	1	1	1	5	1	5	1	1
$*\sigma_f$	1	1	1	1	0	1	1	1

*Final neighborhood radius during the SOM training.

The k-means and competitive network were chosen to perform the partition of the representation of the neurons in the feature space. Due to the random initialization of the k-means algorithm, it was repeated 100 times and the result with the smallest sum of squared errors was selected. The competitive network was trained with 100 epochs. The results obtained were compared to the same methods applied to the SOM neurons in the data space, using the same parameter settings. In this case, the parameter k ranged from 2 to $\sqrt{M}$, where M is the total number of neurons of the network, and again, the one with best DBI value was chosen. The Watershed algorithm applied to

the U-matrix image of each trained SOM. The Fig. 8 depicts the results obtained while varying the mentioned parameters. The classification accuracy (CA) [21] was used for evaluating the results (Fig. 8). The CA consists of the percentage of the properly classified patterns with respect to the complete data set.

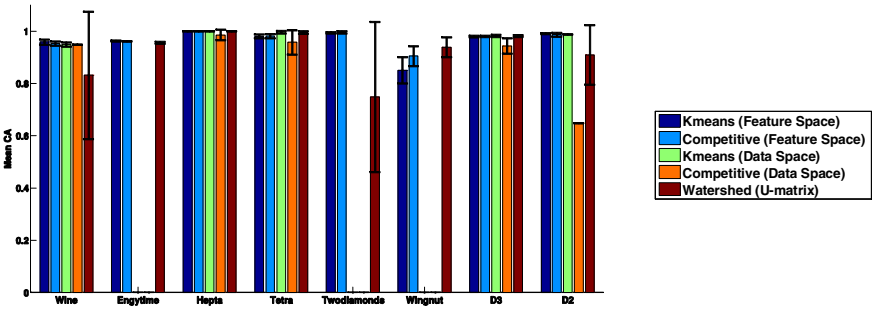


Fig. 8. Mean classification accuracy and standard deviation, for the clustering task performed by the k-means and competitive network both in the data set space and the feature space, as well as the watershed algorithm applied to the U-matrix generated by the trained SOMs.

The k-means and competitive networks applied to the SOM neurons in the data space along with the DBI were not able to find the correct number of clusters in none of the 6 maps (scale factor S) for the data sets *Engytime*, *Twodiamonds* and *Wingnut*. The difficulties encountered by the watershed algorithm for the *Twodiamonds* data set rely in the fact that this data set has connected clusters. Thus, in this case density metrics are more relevant than distance metrics, which is the one displayed in the U-matrix that is used for segmentation. The watershed algorithm was used after the same previously described morphological image processing. It must be noted that improved results using this algorithm may be obtained by thoroughly fine tuning the area size used in the filtering stage.

As several maps were trained, additional measures related to the clustering task were made necessary: the correct clustering frequency (CF) and clustering efficiency (CE). The CF measures the percentage of runs in which the correct number of clusters was found, because the CA is only calculated if and only if the correct number of clusters was found. The CE gives an overall performance measure of the clustering ($CE = Mean\ CA \times CF$), as it takes into account the number of times the right number of cluster was found and how correct was the classification. The CFs obtained are depicted in detail in Table 4 for all the clustering methods used in the experiments.

Although the methods have shown a good overall performance in terms of mean classification accuracy, it must be noted that clustering the feature space and going back to apply the results in the data space demonstrated itself more consistent due to the fact that the correct number of clusters was identified in more simulations than the other methods (see the CE depicted in the radar plot of Fig. 9). For instance, the lowest CFs considering the k-means and the competitive network used for clustering the feature space are 67% and 83% of the runs, respectively, whereas the lowest percentage is 0% for the same algorithms applied to the neurons of the SOM in the data space, or 33% when using watershed. Besides, the mean classification accuracy was above 0.8 for all tested data sets, with comparatively small standard deviations.

Table 3. Classification Frequency

Data set	*Feature Space*		*Data Set Space*		
	k-means	Competitive	k-means	Competitive	Watershed
Wine	1.00	1.00	0.33	0.17	0.67
Engytime	1.00	0.83	0.00	0.00	1.00
Hepta	0.83	0.83	0.67	0.67	1.00
Tetra	1.00	1.00	1.00	0.67	0.83
Twodiamonds	0.67	1.00	0.00	0.00	0.67
Wingnut	1.00	1.00	0.00	0.00	0.83
D3	1.00	1.00	0.83	0.83	1.00
D2	1.00	0.83	0.33	0.17	0.33

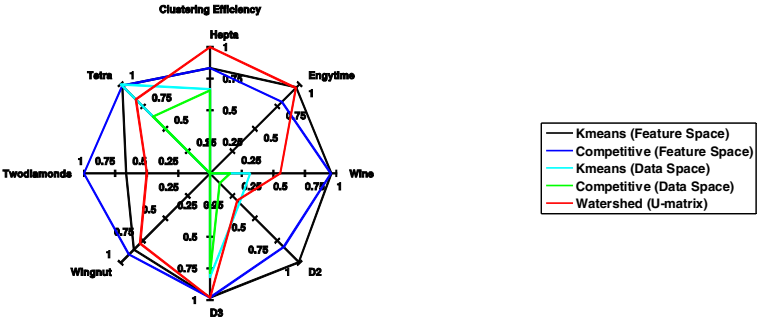


Fig. 9. Classification efficiency for the clustering task performed by the k-means and competitive network both in the data set space and the feature space, as well as the watershed algorithm applied to the U-matrix generated by the trained SOMs.

5 Conclusions

An analysis of the feature space generated by extracting properties related to density and distances between neurons of the SOM network was presented. Applications of the feature space include data filtering so as to reduce noise, as well automatically detecting the number of clusters and used as a tool for visualization.

In order to partition the data set or to reduce the noise, classical clustering algorithms may be used as well as the elbow method if an exponential curve fit is feasible. In this work the k-means algorithm and the competitive network were used with the simplified assumption that in most data sets exist only two types of neurons, however this hypothesis may not be considered universal, as it worked with the great majority, but not all data sets, where the separation in three regions is necessary. The results were compared to the same clustering algorithms applied to the neurons of the SOM network in the data space, as well as with the watershed technique. The proposed method was able to consistently segment the SOM into the correct number of clusters with high classification accuracies. If the clusters present in a given data set have utterly different variances, then the method may not be applied. Nevertheless, neurons with similar characteristics may be depicted as a matrix plot for a given number of partitions of the feature space (k), and therefore provide clues to the data clusters' relative positions and sizes.

References

1. Larose, D.T.: Discovering Knowledge in Data. John Wiley & Sons (2005)
2. Tan, P.-N., Steinbach, M., Kumar, V.: Introduction to Data Mining. Addison Wesley (2006)
3. Kohonen, T.: Self-Organizing Maps, 3rd edn. Springer (2001)
4. Kohonen, T.: Essentials of the self-organizing map. Neural Networks 37, 52–65 (2013)
5. Vesanto, J., Alhoniemi, E.: Clustering of the self-organizing map. IEEE Transactions on Neural Networks 11(3), 586–600 (2000)
6. Vesanto, J.: SOM-based data visualization methods. Intelligent Data Analysis 3(2), 111–126 (1999)
7. Ultsch, A., Siemon, H.P.: Kohonen's self organizing feature maps for exploratory data analysis. In: Proc. of the International Conference on Neural Networks, Paris, pp. 305–308 (1990)
8. Ultsch, A.: Maps for the visualization of high-dimensional data spaces. In: Proc. of the 4th Workshop on Self-Organizing Maps, Kyushu, pp. 225–230 (2003)
9. Jain, A.K.: Data Clustering: 50 Years Beyond K-means. Pattern Recognition Letters 31(8), 651–666 (2010)
10. Haykin, S.: Neural Networks: A Comprehensive Foundation, 2nd edn. Bookman (2001)
11. Haralick, R.M., Shapiro, L.G.: Computer and Robot Vision, vol. 1, pp. 28–48. Addison-Wesley (1992)
12. Duda, R., Hart, P., Stork, D.G.: Pattern classification and scene analysis. John Wiley Professional, Wiley (2000)
13. Vesanto, J., Himberg, J., Alhoniemi, E., Parhankangas, J.: Self-Organizing Map in Matlab: the SOM Toolbox. In: Proc. of the Matlab DSP Conference, Espoo, pp. 35–40 (1999)
14. Ultsch, A.: Clustering with SOM: U*C. In: Proc. of the Workshop on Self-Organizing Maps, Paris, pp. 75–82 (2005)
15. Karypis, G., Han, E.-H., Kumar, V.: Chameleon: A Hierarchical Clustering Algorithm Using Dynamic Modeling. IEEE Computer 32(8), 68–75 (1999)
16. Frank, A., Asuncion, A.: UCI Machine Learning Repository (2010)
17. Costa, J.A.F.: Uma nova abordagem para visualização e detecção de agrupamentos em mapas de Kohonen baseado em gradientes das componentes. Learning and Non Linear Models 9(1), 20–31 (2011)
18. Costa, J.A.F.: Clustering of complex shaped data sets via kohonen maps and mathematical morphology. In: Proc. of the SPIE, Data Mining and Knowledge Discovery, vol. 4384, pp. 16–27 (2001)
19. Davies, D.L., Bouldin, D.W.: A Cluster Separation Measure. IEEE Transactions on Pattern Analysis and Machine Intelligence 1(2), 224–227 (1979)
20. Dougherty, E.R., Lotufo, R.A.: Hands-on Morphological Image Processing. SPIE Publications (2003)
21. Meila, M., Heckerman, D.: An experimental comparison of model based clustering methods. Machine Learning 42(1–2), 9–29 (2001)