On New RBF Neural Network Construction Algorithm for Classification

Amel Sifaoui[1,2]

Amel.Sifaoui@enit.rnu.tn

Afef Abdelkrim[1,3]

Afef.abdelkrim@esti.rnu.tn

Sonia Alouane[2]

alouane_sonia@yahoo.fr

Mohamed Benrejeb[1,2]

mohamed.benrejeb@enit.rnu.tn

[1]U.R.Automatique, ENIT

[2]École Nationale d'Ingénieurs de Tunis (ENIT)

BP 37, Le Belvédère 1002 Tunis, Tunisie

[3]Ecole Supérieure de Technologie et d'Informatique (ESTI)

Abstract: The proposed method to construct a Radial Basis Function (RBF) neural network classifier is based on the use of a new algorithm for characterizing the hidden layer structure. This algorithm, called HNEM-k-means, groups the training data class by class in order to calculate the optimal number of clusters in each class, using new global and local evaluations of the partitions, obtained by the k-means algorithm. Two examples of data sets are considered to show the efficiency of the proposed approach and the obtained results are compared with previous existing classifiers.

Keywords: Radial Basis Function neural network, classification, k-means, validity indexes.

Amel Sifaoui has obtained the Master of Science in Computer Science in 1998 at Faculty of Sciences of Tunis (FST) and the PhD in Electrical Engineering at the National School of Engineers of Tunis (ENIT) in 2008; she is now a Laboratory Assistant at the Faculty of Sciences of Gafsa. Her research interests are in the artificial neuronal network domain.

Afef Abdelkrim has obtained a Diploma of Engineer in 2000 and the PhD in Electrical Engineering at the National School of Engineers of Tunis (ENIT) in 2005; she is now an Assistant Professor at the High School of Technology and Computer Science (ESTI). Her research interests are in handwriting process modelling and synthesis and in the artificial neuronal network domain.

Sonia Alouane has obtained a Graduate Diploma in Computer Science applied to management in 1994 at Institut Supérieur de Gestion Tunis and a post Graduate Diploma a specified for teaching Computer Science and technologie in National School of Computer Science (ENSI) called "Technologue" in 2000. Presently, she is teaching as "Assistante technologue" at the National School of Engineers of Tunis (ENIT), her research interests are in the artificial neuronal network domain.

Mohamed Benrejeb has obtained the Diploma of "Ingénieur IDN" (French "Grande Ecole") in 1973, the Master degree of Automatic Control in 1974, the PhD in Automatic Control of the University (USTL) of Lille in 1976 and the DSc of the same University in 1980. He is currently a full Professor at the Ecole Nationale d'Ingénieurs de Tunis (Tunisia) and an Invited Professor at the Ecole Centrale de Lille (France). His research interests are in the area of analysis and synthesis of complex systems based on classical and non conventional approaches and recently in discrete event system domain.

1. Introduction

Introduced into the neural network literature by Broomhead and Lowe [1], the radial basis function neural networks have been widely used for function approximation, pattern classification and recognition due to their structural simplicity and faster learning abilities [2, 3]. However, their design still remains a difficult task due to the absence of systematic method giving an optimal architecture. A too small network cannot well learn the problem, but a too large network will lead to over-fitting and poor generalization performance [4].

A very important step for the RBF network training is to decide a proper number of hidden neurons (number of basis functions), because it controls the complexity and the generalization ability of RBF networks, so many works have proposed RBF classifier conception algorithms [5-11].

The radial basis function network is three layers feedback network, typically used for supervised classification. Its training procedure is usually split into two successive steps. First, the centers of the hidden layer neurons are selected by clustering algorithms such as k-means [11, 12], vector quantization [13], decision trees [14], and then the widths are calculated [15]. Second, the weights connecting the hidden layer with the output layer are determined by Singular Value Decomposition (SVD) or by Least Mean Squared (LMS) algorithms [16].

In this paper, a new learning algorithm is proposed for construction of the radial basis function networks solving classification problems. It determines the proper number of hidden neurons and calculates the centers values of the radial basis functions. After the selection of the centers of the hidden neurons, the widths of the nodes are determined by the P-nearest neighbour heuristic, and the weights between the hidden layer and the output layer calculated by the pseudo-inverse matrix.

The proposed approach consists of combining new evaluation measurement with the k-means algorithm, which have led to a new algorithm, called HNEM-k-means, allowing the automatic determination of the clusters number in the data of each class. Two different real databases are used in order to evaluate the classifier performances.

2. Construction of RBF Neural Network Classifier – Problem Statement

Classification systems are either supervised or unsupervised, depending on whether they assign new inputs to one of a finite number of discrete supervised classes or unsupervised categories, respectively [12].

In supervised classification, classes exist a priori. The mapping from a set of input data vectors, to a finite set of discrete class labels ($y \in 1,\dots,m$) is modeled in terms of some mathematical function $y = y(x, \mathrm{T})$, where T is an adjustable parameters vector [12], x_l ($l = 1,\dots,N,$ N size of data set) an input data, and m the total number of class types.

In unsupervised classification, also called clustering, data set is partionned into groups (clusters) such that the data points in a cluster are more similar to each other than points in different clusters [17].

The RBF neural network architecture [11] consists of one input layer, one hidden layer and an output layer. Each input neuron corresponds to an element of an input vector which is fully connected to the hidden layer neurons. Again, each of the hidden layer neurons is fully connected to the output neurons.

The output of a hidden layer neuron i is usually generated by a gaussian function φ_i specified by its center c_i and its width σ_i :

$$\varphi_i(x_l) = \exp\left\{-\frac{\|x_l - c_i\|^2}{2\sigma_i^2}\right\},\ i = 1,\dots,n \qquad (1)$$

where $\|.\|$ is the Euclidean distance between the input vector x_l and the center vector c_i.

The number of neurons in the output layer is equal to the possible classes of the given problem. Each output layer neuron s_j computes a linear weighted sum of the outputs of the hidden layer neurons as follows:

$$s_j = f\left(\sum_{i=1}^{n} w_{ij}\varphi_i(x_l)\right) \qquad (2)$$

where the w_{ij} parameters are the synaptic weights connecting the hidden nodes to those of the output layer.

In this paper, the problem of finding a proper number of hidden neurons is transformed into a clustering problem. The patterns of each class are partitioned into clusters using the k-means algorithm and then the obtained partition is evaluated using the proposed evaluation measurements to reach the optimal solution.

The k-means algorithm [12] aims to optimize an objective function given by:

$$J = \frac{1}{2}\sum_{i=1}^{k}\sum_{l=1}^{N}\delta_{il}\left\|x_l - c_i\right\| \qquad (3)$$

with

$$\delta_{il}\begin{cases}1 \text{ if } x_l \in c_i\\ 0 \text{ else}\end{cases}$$

where J represents the sum of square-error for all objects in the data set, x_l a point in space representing a given object, and c_i the center (both x_l and c_i are multidimensional).

Initially, k-means algorithm randomly selects k objects, which initially represent a cluster centers'. The remaining objects are assigned to clusters to which they are most similar, based on the distance between the object and the cluster mean. Then the new mean for each cluster is calculated. This process iterates until the criterion function converges.

This algorithm consists of the following steps.

Step 1: Initialize a k-partition randomly: $c_1, c_2, \ldots, c_k$

Step 2: Assign each object in the data set to the nearest cluster c_j

$\left\|x_l - c_j\right\| \prec \left\|x_l - c_i\right\|$ for $l = 1, \ldots, N,\ j \neq i$ and $j = 1, \ldots, k$

Step 3: Compute the new cluster centers:

$$c_j^* = \frac{1}{N_j}\sum_{x_l \in c_j} x_l$$

where N_j is the number of samples in c_j

Repeat steps 2 and 3 until there is no change for each cluster

The k-means algorithm is very simple and can be easily implemented in solving many practical problems. It can work very well for compact and hyperspherical clusters and can be used to cluster large data sets. One of the drawbacks of k-means is that there is no efficient and universal method for identifying the initial partitions and the number of clusters.

The centers' convergence varies with different initial points. A general strategy for the problem is to run the algorithm many times with random initial partitions [12]. In order to solve this problem, the k-means algorithm is executed many times and then the centers giving the minimal value of objective function are used as initial centers for the algorithm's starting.

3. Proposed Evaluation Measurement

The use of the clustering algorithm such as k-means, requires to specify k, the number of clusters, in advance. However, it is difficult to pre-assign k. So the problem in clustering is to decide of the optimal number of clusters that fits a data set. In order to obtain this number, the partitioning result must be evaluated. Usually, this clustering result is measured by a criterion known as the cluster validity index. The two measurement criteria for evaluating and selecting an optimal clustering scheme are: compactness and separation. Several validity indexes have been proposed in the literature [17-20].

In this paper, new validity indexes permitting to evaluate, globally and locally, the obtained partition are proposed.

3.1 Proposed global evaluation

The global evaluation gives information about the quality of the whole generated partition obtained by the k-means. In this part, new formulations are proposed to define the global compactness and separation in a partition.

Definition 1. Global compactness: Given the groups scheme $K = \left\{K_1, K_2, \ldots, K_k\right\}$ for a set of patterns $X = \left\{x_1, x_2, \ldots x_N\right\}$ and $K' = \left\{K_* / K \text{ and } K_* \text{ not a singleton}\right\}$, $j = 1, 2, \ldots, M$, $M = card\left(K'\right)$, k the number of groups and M the number of non singleton groups.

The global compactness, CP is given by:

$$CP = \frac{M}{\sum_{j=1}^{M} Var_j} \qquad (4)$$

whith Var_j the variance of the group.

The variance of each cluster is obtained by summing over only the members of each cluster, its expression is:

$$Var_j = \frac{\sum_{x_l \in C_{pj}} \left\| x_i - c_j \right\|^2}{\left(Card\left(K_*\right)\right)^2} \qquad (5)$$

where c_j is the center of the group K_*.

Definition 2. Global separation: the global separation Sep of a groups scheme $K = \{K_1, K_2, \ldots, K_k\}$ for a set of patterns $X = \{x_1, x_2, \ldots x_N\}$ is defined by:

$$SP = \left(\frac{\sum_{n=1}^{k} \left(\left\| c_n - c_l \right\| \right)}{k} \right)^2 \qquad (6)$$

Definition 3. Global separation-compactness: Given the groups scheme $K = \{K_1, K_2, \ldots, K_k\}$ for a set of patterns $X = \{x_1, x_2, \ldots x_N\}$ and $K' = \{K_* / K \text{ and } K_* \text{ not a singleton}\}$, $j = 1, 2, \ldots, M$, $M = card\left(K'\right)$, k the number of groups and M the number of non singleton groups.

The global separation-compactness, SC, of the groups scheme K, is formulated as:

$$SC = \frac{M}{k} \times SP \times CP \qquad (7)$$

Consequently the best data partition is obtained by maximizing the measure SC.

3.2 Proposed local evaluation

The local evaluation is used to evaluate individual groups. It identifies the worst group to be merged with others; each pattern of this group is assigned to the nearest group. Then the groups' centers are recalculated. To identify this worst group, separation and compactness measurements for local evaluation of groups are employed.

Definition 4. Local compactness: Given the groups scheme $K = \{K_1, K_2, \ldots, K_k\}$ for a set of patterns $X = \{x_1, x_2, \ldots x_N\}$ if each $K_l \in K$ is not a singleton, the local compactness of the group K_l, cp_l, is given by the following expression:

$$cp_l = \frac{\left(card\left(K_l\right)\right)^2}{\sum_{x_i \in K_l} \left\| x_i - c_l \right\|^2} \qquad (8)$$

Definition 5. Local separation: Given the groups scheme $K = \{K_1, K_2, \ldots, K_k\}$ for a set of patterns $X = \{x_1, x_2, \ldots x_N\}$ if each $K_l \in K$ is not a singleton, local separation of the group K_l, sp_l , is given by the following expression:

$$sp_l = \min_{\substack{1 \le j \le K \\ j \ne l}} \left\| c_j - c_l \right\|^2 \qquad (9)$$

Definition 6. Local separation-compactness: Given the groups scheme $K = \{K_1, K_2, \ldots, K_k\}$ for a set of patterns $X = \{x_1, x_2, \ldots x_N\}$ if each $K_l \in K$ is not a singleton, local separation of the group K_l, sp_l , is given by the following expression:

$$sc_l = sp_l \times cp_l \qquad (10)$$

The smallest value of sc_l indicates the worst group to be merged with others.

4. New Algorithm Constructing the Hidden Layer of FRB: HNEM-k-means

In this part, a new algorithm determining the characteristics of the hidden layer of RBF neural network is proposed. In fact, it calculates the number of hidden neurons and the centers' values of the gaussian functions.

The proposed validity indexes measures are integrated into the k-means algorithm which leads to a new algorithm, called HNEM-k-means algorithm (Hidden Neuron Enhanced Merging k-means) that automatically determines the number of groups in the data set of each class. In this algorithm, there is no need to specify the number of groups a priori.

However, it's needed to specify an upper limit for this number:

$K_{\max}^{\Omega_j}, \left(j = 1, 2, \ldots, m \right)$.

Hidden layer characterization

In order to specify the hidden layer of an RBF neural network, it is necessary to decide the number of hidden neurons and their activation functions. A simple method consists in creating a neuron for each training pattern. However, this method is not practical since in most applications there is a large number of training patterns and the dimension of the input space is very large. Therefore, it is more practical to classify the training patterns to a reasonable number of groups by using a clustering algorithm such as k-means and then to assign a neuron to each cluster.

Such an algorithm is applied when the class of each training pattern is unknown. However, the RBF neural network is used for a supervised classification. So, it takes advantage of the information about the class membership in order to obtain a better clustering result. Hence, it is proposed to cluster the patterns class by class instead of clustering the entire patterns at the same time. The number of groups can be automatically obtained by combining the proposed validity indexes with the k-means algorithm.

4.1 Proposed HNEM-k-means algorithm

The proposed HNEM-k-means algorithm is based on pruning strategy, i.e., it starts with a maximum number of groups $K_{\max}^{\Omega_j}$. This number is decreased during the iterations until obtaining the optimal value of the number of groups $K_{opt}^{\Omega_j}$ for each class $\Omega_j, \left(j = 1, 2, \ldots, m \right)$.

The number K_{Tot} of the hidden layer's neurons is the sum of the number of groups $K_{opt}^{\Omega_j}$ for each class. A neuron is then related to each group. The $K_{\max}^{\Omega_j}$ value can be defined a priori, if the data base structure is known, but this is not always possible. That's why, the Bezdek suggestion is adopted [21]: $K_{\max}^{\Omega_j} = \sqrt{N}$ (*N* dataset size of class Ω_j).

In this algorithm, every iteration aims to maximize the global separation-compactness index SC, obtained for the different calculated number of groups. So, in every iteration the worst group is identified and merged with the other groups. The optimal number of groups correspond to the highest SC, calculated during the different iterations of the algorithm.

The proposed HNEM-k-means algorithm, described in the following, determines the centers values and the $K_{opt}^{\Omega_j}$ number of groups.

```
Input:    Set of data
          $X = \left\{ x_1, x_2, \ldots, x_N \right\} \in R^d$
Output:   Optimal cluster number $K_{opt}^{\Omega_j}$ and
          the centers values
Step 1:   Initialize the parameters related to
          the k-means, $K_{\max}^{\Omega_j} = K^{\Omega_j} = \sqrt{N}$ , $K_{\min}^{\Omega_j} = 2$
Step 2:   Apply the k-means algorithm
Step 3:   Compute and store the global
          separation-compactness $SC$ for
          the crisp partition obtained in step
          2
Step 4:   Repeat
              4.1 Apply the merging procedure
              4.2 Decrease the number of clusters
                  $K^{\Omega_j} \leftarrow K^{\Omega_j} - 1$
              4.3 Compute the global separation-
                  compactness for new clusters and denote
                  it $SC'$
              4.4 Compare the current $SC'$ with
                  the previous $SC$ and keep the maximum
                  one
          Until $K^{\Omega_j} \le K_{\min}^{\Omega_j}$
Step 5:   Output optimal number of clusters
          $K_{opt}^{\Omega_j}$ and their cluster center
          $c = \left\{ c_1, c_2, \ldots, c_{K_{opt}^{\Omega_j}} \right\}$
```

The merging procedure described in the following, is used to identify the worst group which has the smallest sc_l value.

```
Input:  Set of clusters
$K_*^{\Omega_j} = \left\{K_1^{\Omega_j}, K_2^{\Omega_j}, \ldots, K_{K+1}^{\Omega_j}\right\}$
Output:    New set of clusters
$K^{\Omega_j} = \left\{K_1^{\Omega_j}, K_2^{\Omega_j}, \ldots, K_K^{\Omega_j}\right\}$
Step 1: Calculate the local separation-
        compactness $sc_l$ for each cluster
        of $K_*^{\Omega_j}$ using expression (10)
Step 2: Delete the worst cluster which have
the minimal value of $sc_l$
Step 3: Assign each object of this worst
cluster to other clusters
Step 4:    Calculate the new centers of
different clusters using the following formula
        $c_{K_i^{\Omega_j}} = \frac{1}{N_i} \sum_{x \in K_i^{\Omega_j}} x$
Step 5:    Apply the k-means algorithm to
find the optimal cluster centers
```

Applying the HNEM-k-means algorithm to all classes $\Omega_j, \left(j = 1, 2, \ldots, m\right)$ and summing the obtained numbers of clusters $K_{opt}^{\Omega_j}$ gives the hidden neurons number K_{Tot}. A neuron is attributed to each cluster and the centers of clusters become the centers of the radial basis function hidden neurons.

To accomplish the hidden layer construction, the width $\sigma_i \left(i = 1, 2, \ldots, K_{Tot}\right)$ of each gaussian function is calculated using the P-Nearest-Neighbor heuristic (PNN) [15]:

$$\sigma_i = \frac{1}{P}\left(\sum_{r=1}^{P} \left\| c_i - c_r \right\|\right)^{\frac{1}{2}} \qquad (11)$$

where the c_r are the P-Nearest Neighbors of center c_i.

After determining the hidden nodes positions and widths, the training process is completed by calculating the synaptic weights w_{ij} connecting the hidden nodes to those of the output layer using the pseudo-inverse matrix.

The created RBF classifier has m outputs, one for each class Ω_j, the pattern x_l belongs to the class Ω_t, if $t = \max_j \left\{s_j\left(x_l\right)\right\}$. The desired output of a point which belongs to Ω_j is set to 1 while the others are set to 0.

5. Evaluation – Case study

The aim of this section is to evaluate the performances of the elaborated RBF classifier. The performances of the RBF neural networks classifier are tested with two different databases namely, iris and diabetes [22].

The first test is carried out with the iris database which is one of the most popular data sets to examine the performance of novel methods in pattern recognition and machine learning. It is composed of three classes (i.e., iris setosa, iris versicolor and iris virginical) each having 50 patterns with four features. Iris setosa can be linearly separated from iris versicolor and iris virginical, while iris versicolor and iris virginical are not linearly separable.

The second test is done with the diabetes database which contains two classes with 768 patterns representing results of a diabetes test given to Pima Indians. The class 1 has 500 patterns and the class 2 has 268. Each class has 8 features. This database comprises partially superimposed classes.

The results given by the RBF classifier built with our algorithm are compared with those obtained with three other neural classifiers: the Learning Vector Quantization (LVQ) classifier proposed by Kohonen, the RBF neural net work classifier for which the hidden layer is obtained using the Adaptive Pattern Classifier (APCIII) [23], the Multi-Layer Perceptrons classifier (MLP) and with a reference one, the K Nearest Neighbor (KNN).

The results obtained with the holdout method which divides the initial data into two independent sets: one for training and the other for testing the classifier performances. The same training sets and tests are used for all classifiers. The global measurement used also to evaluate the classifier performance is the recognition rate.

The present comparative results of different classifiers over iris and diabetes databases are illustrated in table1 and table 2.

Considering iris database, the best recognition rate is obtained by RBF-HNEM-k-means proposed classifier. For diabetes database, the best recognition rate is given for the KNN classifier; however the difference with the proposed classifier is not important. Then, the proposed algorithm gives good results in term of recognition rate.

Table 1: Results of the recognition rate over iris database

Classification algorithms	Database : iris
HNEM-k-means	97,33%
LVQ	94,00%
APCIII	93,33%
MLP	96,66%
KNN	96,70%

Table 2: Results of the recognition rate over diabetes database

Classification algorithms	Database : diabetes
HNEM-k-means	76,56%
LVQ	74,00%
APCIII	71,00%
MLP	76,00%
KNN	77,00%

6. Conclusion

In this paper, a new algorithm, to design RBF neural networks classifiers and to select the centers of the hidden layer neurons in particularly, is proposed.

Based on pruning technique, it aims to construct the hidden layer of an RBF neural network and starts with the maximum number of groups, which is decreased during the different iterations of the algorithm.

The basic idea of this approach is to gather the training data class by class and to decide of the optimal number of groups in each class by using the proposed validity indexes measure which are integrated in the k-means algorithm.

The obtained classifier results are satisfactory in comparison with other considered classifiers in the literature for two real databases.

REFERENCES

1. BROOMHEAD, D. S., LOWE D., **Multivariable functional interpolation and adaptive networks**, Complex Syst., vol. 2, 1988, pp. 321–355.
2. ZHANG, G.P., **Neural Netwotks for Classification: A survey**, IEEE Transaction on Systems, Man, and Cybernetics, part C: Application and reviews, vol. 30, no. 4, 2000.
3. VERLEYSEN, M., HLAVACKAVA K., **An optimized RBF network for approximation of functions**, European Symposium on Artificial Neural Networks, Brussels, 1994, pp. 175-180.
4. KWOK, T.Y., YEUNG D.Y., **Constructive algorithms for structure learning in feed forward neural networks for regression problems**, IEEE Trans. on Neural Networks, vol. 8, no. 3, 1997, pp. 630-645.
5. SING J.K., BASU D.K., NASIPURI M., KUNDU M., **Improved K- means Algorithm in the Design of RBF Neural Networks**, IEEE Conf. on Convergent Technologies for the Asia-Pacific, vol. 2, Tencon 2003, Bangalore, pp. 841-845.
6. KARAYIANNIS N.B., RANDOLPH-GIPS, M.M., **On the Construction and Training of Reformulated Radial Basis Function Neural Networks**, IEEE Trans. on Neural Networks, vol. 14, no. 4, 2003, pp. 835-845.
7. MAO, K.Z., HUANG G.B., **Neuron selection for RBF neural network classifier based on data structure preserving criterion**, IEEE Trans. on Neural Networks, vol. 16, Issue: 6, 2005, pp. 1531-1540.
8. YU, B., HE X., **Training Radial Basis Function Networks with Differential Evolution**, Proceedings of World Academy of Science Engineering and Technology, vol. 11, 2006, Helsinki, Finland, pp. 157-160.
9. BELLOIR, F., FACHE A., BILLAT A., **A General Approach to Construct RBF**

Net-Based Classifier, European Symposium on Artificial Neural Networks, Bruges, 1999, pp. 399-404.

10. SARATCHANDRAN, P. SUNDARARAJAN N., **A generalized growing and pruning RBF (GGAP-RBF) neural network for function approximation**, IEEE Trans. on Neural Networks, vol. 16, Issue 1, 2005, pp. 57-67.

11. MOODY, J. DARKEN C.J., **Fast learning in networks of locally-tuned processing units**, Neural Computation, 1989, pp. 281-294.

12. XU, R., WUNSCH D., **Survey of Clustering Algorithms**, IEEE Trans. on Neural Networks, vol. 16, no. 3, 2005, pp. 645-678.

13. VOGT, M., **Combination of Radial Basis Function Neural Networks with Optimized Learning Vector Quantization**, IEEE Int. Conf. on Neural Networks, San Francisco, 1993 pp. 1841-1846.

14. KUBAT, M, **Decision Trees Can Initialize Radial-Basis Function Networks**, IEEE Trans. on Neural Networks, vol. 9, Issue 5, 1998, pp. 13-821.

15. BENOUDJIT, N., ARCHAMBEAU C., LENDASSE A., LEE J., VERLEYSEN M., **Width optimization of the Gaussian kernels in Radial Basis Function Networks**, European Symposium on Artificial Neural Networks, Bruges, 2002, pp. 425-432.

16. BORNE, P., BENREJEB M., HAGGEGE J., **Les réseaux de neurones: Présentation et application**, Ed. Technip, Paris, 2007.

17. SHENA, M.J., CHANGA S.I., LEEA E.S., DENGB Y., BROWNB S.J., **Determination of cluster number in clustering microarray data**, Applied Mathematics and Computation, vol. 169, Issue 2, 2005, pp. 1172-1185.

18. HALKIDI, M., BATISTAKIS Y. , VAZIRGIANNIS M., **Clustering Algorithms and Validity**, 13th Int. Conf. on **Scientific and Statistical Database Management**, 2001, Virginia, pp. 3-22.

19. HALKIDI, M., BATISTAKIS Y., VAZIRGIANNIS M., **On Clustering Validation Techniques**, J. of Intelligent Info. Syst., vol. 17, 2001, pp. 107-145.

20. SASSI, M., GRISSA-TOUZI A., OUNELLI H., **Using Gaussians Functions to Determine Representative Clustering Prototypes**, 17th IEEE Int. Conf. on Database and Expert Systems Applications, 2006, pp. 435-439.

21. BEZDEK, J.C., Convergence Theory for Fuzzy C-Means: Counter Examples and Repairs, IEEE Trans. on Systems, 1987, pp. 873-877.

22. UCI Machine Learning Repository, http://www.ics.uci.edu/learn/MLRepository.html.

23. HWANG, Y.S., BANG S.Y., **An Efficient Method to Construct a Radial Basis Function Network Classifier,** Neural Networks, vol. 10, no. 8, 1997, pp. 1495-1503.