Logarithmic Spiral-based Construction of RBF Classifiers

Mohamed Wajih Guerfala
Laboratory of research in Automatic Control, ENIT
University of Tunis El Manar, National Engineering School of Tunis
BP 37, Le Belvédère 1002 Tunis, Tunisia

Amel Sifaoui
Laboratory of research in Automatic Control, ENIT
University of Tunis El Manar, National Engineering School of Tunis
BP 37, Le Belvédère 1002 Tunis, Tunisia

Afef Abdelkrim
Laboratory of research in Automatic Control, ENIT
University of Carthage, National Engineering School of Carthage (ENICarthage), BP 37, Le Belvédère 1002 Tunis, Tunisia

Abstract—Clustering process is defined as grouping similar objects together into homogeneous groups or clusters. Objects that belong to one cluster should be very similar to each other, but objects in different clusters will be dissimilar. It aims to simplify the representation of the initial data. The automatic classification recovers all the methods allowing the automatic construction of such groups. This paper describes the design of a radial basis function (RBF) neural classifiers using a new algorithm for characterizing the hidden layer structure. This algorithm, called k-means Mahalanobis distance, groups the training data class by class in order to calculate the optimal number of clusters of the hidden layer, using two validity indexes. To initialize the initial clusters of k-means algorithm, the method of logarithmic spiral golden angle has been used. Two real data sets (Iris and Wine) are considered to improve the efficiency of the proposed approach and the obtained results are compared with basic literature classifier.

Keywords—Radial Basis Function neural network; classification; k-means; validity index of Davis Bouldin; Mean Squared Error; Mahalanobis distance; Logarithmic spiral; golden angle; golden ratio

I. INTRODUCTION

Clustering is one of the most useful tasks in data mining process for discovering groups and identifying interesting distributions and patterns in the underlying data. The problem with Clustering is about partitioning a given data set into groups (clusters) such that the data points in a cluster are more similar to each other than points in different clusters [1].

In recent years, a number of clustering algorithms has been proposed and is available in literature. The radial basis function (RBF) neural network is one of the most used in data classification.

RBF neural networks consist of three layers: an input layer, a hidden layer and an output layer. The input layer corresponds to the input vector feature space and the output layer corresponds to the pattern classes [2]. So the whole architecture is fixed only by determining the hidden layer and the weights between the middle and the output layers [3].

Its training procedure is usually split into two successive steps: training in the hidden layer followed by training in the output layer [4]. First, the centers of the hidden layer (HL) neurons are selected by clustering algorithms such as k-means [5], [6], support vector machine (SVM) [7] or hierarchical clustering [8]. 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.

One of the used techniques to find the optimal number of this HL is the logarithmic spiral which has seen a significant amount of research on nature-inspired optimization techniques such as neuro-computing in the past 25 years, evolutionary and genetic algorithms, particle swarm optimization. Most recently, a new multipoint meta-heuristics research method has emerged for 2-dimensional continuous optimization problems based on the analogy of spiral phenomena in nature, called 2-dimensional spiral optimization first proposed by Tamura and Yasuda in 2010 [9].

Focused spiral phenomena are approximated to logarithmic spirals, which frequently appear in nature, such as whirling currents, nautilus shells and arms of spiral galaxies. A Two-dimensional spiral optimization uses the feature of Logarithmic Spirals [LS] [9].

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

The aim of this approach consists in transforming the problem of determining the number of hidden layer neurons to a clustering problem. In order to determine the number of clusters in the data of each class, the k-means algorithm is combined with two different validity indexes (the validity index of Davis Bouldin for the first classifier and Mean Square Error for the second classifier).

In k-means algorithm, the used distance corresponds to the Mahalanobis distance. A solution is also given to overcome
the problem of setting the start values for the initial centers needed to start this algorithm using the proposed method "The logarithmic spiral golden angle". Two different real databases are used in order to evaluate the two proposed classifier performances.

Next section presents the problem of the construction of the hidden layer for RBF neural networks. Section 3 describes in detail the logarithmic spiral golden angle. Section 4 is devoted to the elaborated solution to overcome the problem of the k-means algorithm. Section 5 describes the construction of the two proposed RBF classifiers. Experiments and discussions are presented in Section 6, followed by concluding remarks in Section 7.

II. PROBLEM STATEMENT OF USE

The construction of the hidden layer of RBF neural networks is by clustering algorithms such as k-means. K-means clustering algorithm is one of the best-known algorithms used in clustering.

However, it still has some problems one of which is in its initialization step which is generally done randomly by users. Another disadvantage of k-means is that it converges to local optimum, depending on its random initialization.

The k-means algorithm classifies objects to a pre-defined number of clusters, which is given by the user (assume k clusters). The idea is to choose random cluster centers, one for each cluster. These centers are preferred to be as far as possible from each other. The starting points affect the clustering process and results [10]. “Fig. 1”.

Each boot (initialization) is a different solution (local optimum) which can in some cases be far from the optimal solution (global optimum) [11]. A simple solution to this problem is to run the algorithm several times with different initialization and retain the best combination found.

The use of this solution is limited because of its cost and because of the possibility of finding better results in a single execution [12].

However, because of the sensitivity of k-means to its initial points, two solutions have been proposed to this problem. The first one is to initialize the centers of k-means algorithm using the circle method [13]. The obtained results of this algorithm are the initial centers positions $C_i$ represented by the “Fig. 2”.

The second solution is to initialize the centers of k-means algorithm using the logarithmic spiral golden angle in order to improve the clustering performance. In this paper, the second solution will be explained.

![Fig. 2. Tracing of initialization of the $k$ centers maximum on the outline of the circle of radius $r$ and spaced by the angle $\theta$](image)

III. THE LOGARITHMIC SPIRAL GOLDEN ANGLE

The logarithmic spiral golden angle is a specific case of the logarithmic spiral. It represents a plane curve centered in a starting point and parameterized by the radius $r$, the angle $\theta$ and the Golden Ratio $\phi$.

A. The logarithmic spiral

A Logarithmic Spiral is a plane curve for which the angle between the radius vector and the tangent to the curve is a constant [14]. Such spirals can be approximated mathematically defined by the following equation on the 2-dimensional polar coordinate system $(r, \theta)$ as [9]:

$$r = a \ e^{b\theta}$$

(1)

Where $a$ and $b$ are positive real with $a > 0$ and $b \neq 0$.

Equation (1) can be transformed into Cartesian coordinates as follows:

$$\begin{cases} x(\theta) = r(\theta) \ \cos(\theta) = ae^{b\theta} \ \cos(\theta) \\ y(\theta) = r(\theta) \ \sin(\theta) = ae^{b\theta} \ \sin(\theta) \end{cases}$$

(2)

In this work, the factor $b$ of the logarithmic spiral has been set to zero ($b = 0$), it goes back to simplify the polar radius as follows: $r = ae^{b\theta} \Rightarrow r = a$

The following equation of the logarithmic spiral golden angle is obtained:

![Fig. 1. Clustering on a set of 2D points data, 3 clusters](image)
\[
\begin{align*}
    x(\theta) = r \cos(\theta) &= a \cos(\theta) \\
    y(\theta) = r \sin(\theta) &= a \sin(\theta)
\end{align*}
\]  
(3)

B. The Golden ratio

The irrational number, golden ratio is also known as golden section by the ancient Greeks, golden proportion, divine proportion or golden number [15].

The golden ratio \( \phi \), has many properties which people are eager to know. It is a number that is equal to the reciprocal of its own with the addition of 1: \( \phi = \frac{1}{\phi} + 1 \).

Likewise, the ratio of any two consecutive Fibonacci numbers converges to give approximates of 1.618, or its inverse, 0.618. This shows the relationship between Fibonacci numbers and golden ratio [16].

If the possibility of dividing a line in such a way that the ratio of the whole length to the length of the longer segment happens to be equal to the ratio of the length of the longer segment to the length of the shorter segment, then it could be said that the ratio is a golden ratio [15], “Fig. 3”.

![Fig. 3. Dividing of a whole length AC into two segments AB and BC](image)

This gives mean ratio if \( \frac{AB}{BC} = \frac{AC}{AB} \). If the value of AB is set to be \( x \), and use 1 to represent the length of BC, then \( \frac{x}{1} = \frac{1+x}{x} \) is obtained. Then the irrational number is the only positive solution of the equation \( x^2 - x - 1 = 0 \), so \( x = \frac{1+\sqrt{5}}{2} \).

Its value is: \( \phi = \frac{1+\sqrt{5}}{2} \approx 1.6180339887 \) . Where the Greek letter \( \phi \) represents the golden ratio.

C. The Golden angle

In geometry, the golden angle is created by dividing the circumference of a circle \( c \) in two sections, a longer arc of length \( a \) and a smaller arc of length \( b \) such that: \( c = a + b \) and \( \phi = \frac{a+b}{a} = \frac{c}{a} = \frac{a}{b} \), “Fig. 4”.

The angle formed by the arc \( b \) of circle \( c \) is called the golden angle \( \psi \). It derives from the golden ratio \( \phi \).

\[
\psi = 2\pi - \left( \frac{2\pi}{\phi} \right) = 2\pi \left( \frac{\phi - 1}{\phi} \right) \approx 2.391 \text{ rad} \approx 137.5^\circ
\]

IV. PROPOSED INITIALIZATION OF THE K-MEANS ALGORITHM WITH LOGARITHMIC SPIRAL GOLDEN ANGLE

The k-means algorithm aims to minimize the distance between the object and the center of its group. In this section, the k-means algorithm based on the Mahalanobis distance and its proposition for initialization of the centers are presented.

A. The k-means algorithm Mahalanobis distance specifications

There are different types of distances such as: Minkowski distance, the average, the family of metrics, Euclidean Weighted and the Euclidean distance which is the most used, e.g. applied in the RBF Networks. [17]

Moreover, the Mahalanobis distance is a distance measure and its utility is a way to determine the similarity between two multidimensional random variables. It differs from Euclidean distance, because the Mahalanobis distance takes into account the correlation between random variables, [17]. The Mahalanobis distance is defined by:

\[
d(x, y) = \sqrt{(x - y) \times \text{Cov}(X)^{-1} \times (x - y)^t}
\]  
(4)

With \( \text{Cov}(X) \) is the covariance matrix.

If the elements \( x \) and \( y \) are independent, the covariance matrix is the identity and the Mahalanobis distance is equal to the Euclidean distance. The proposed algorithm based on the combination of Mahalanobis distance with k-means is described by the following steps:
Algorithm: Function Kmeans_distance_Mahalanobis (KDM)

Begin
Input: - The database $X = \{x_1, x_2, \ldots, x_n\} \in \mathbb{R}^d$.
- The position of each center $C = \{c_1, c_2, \ldots, c_k\} \in \mathbb{R}^d$.

Output: - The new position of each center $C' = \{c'_1, c'_2, \ldots, c'_k\} \in \mathbb{R}^d$.

Step 1:
- Determine the size $N$ of the data base of $X$.
- Determine the number $k$ of centers to be used in the observation space $C'$.
- Initialize the vector of the new positions of the centers $C'$ to zero.

Step 2: - Determine the covariance matrix $\text{Cov}(X)$ with the following equation:

$$
\text{Cov}(X) = \frac{1}{N-1} \sum_{i=1}^{N} (X_{ij} - \bar{X}_j)(X_{ij} - \bar{X}_j)^T
$$

with $X_{ij} \in X$ ; $i = 1, \ldots, n$ and $j = 1, \ldots, p$.

Where $\bar{X}_j = \frac{1}{N} \sum_{i=1}^{N} X_{ij}$ with $\bar{X}_j$: arithmetic averages.

While: The new centers do not have, a significant displacement Do:

Step 3: - Assign each observation (dot) group nearest center $c_j: x_i \in c_j$ according to the Mahalanobis distance formula:

$$
d(i, j) = \sqrt{(x_i - c_j)^T \text{Cov}(X)^{-1} (x_i - c_j)}
$$

with $i = 1, \ldots, N$ and $j = 1, \ldots, k$.

Step 4: - Recalculate the position of each new center:

$$
c'_j = \frac{1}{N_j} \sum_{x_i \in c_j} x_i
$$

with $N_j$ = the set of points belonging to the center $c_j$ and $j = 1, \ldots, k$.

End While

End

The third is to initialize the values of the logarithmic spiral golden angle on the polar coordinate system $(r, \theta)$: the radius $r = a = a_0 = 0$ and the angle $\theta = \theta_0 = 0$. The angle $\theta$ increases by the factor $d\theta = \psi = 2\pi (\phi - 1)$ and the radius $r$ increases by the factor $da = \frac{R}{k_{\text{max}}} = \frac{Gb}{k_{\text{max}}}$.

Fig. 5. Tracing of the two most distant individuals $(a, b)$ and their medium $G$.

To determine $k_{\text{max}}$, the suggestion of Bezdek was adopted [19] as follows: $k_{\text{max}} = \sqrt{N}$ ($N$ is the size of the database).

The forth step is to determine the positions of the centers of the logarithmic spiral golden angle with center $G$ and radius $r = a$ “Fig. 6”. The calculation of the center position $C_i$ is performed by applying the following formula:

$$
\begin{align*}
C_{ki} &= G_i + a \times \cos(\theta) \\
C_{vi} &= G_i + a \times \sin(\theta)
\end{align*}
$$

With $\theta = \theta + d\theta$; $a = a + da$ and $k = 1, \ldots, k_{\text{max}}$.

Fig. 6. Tracing of the initialization of the $k$ centers maximum on the outline of the logarithmic spiral golden angle.

Thereafter, the positions of these $k_{\text{max}}$ centers in the variable $C_k = \{c_1, \ldots, c_{k_{\text{max}}} \}$ will be saved.
The basic principle of the adopted strategy is summarized in the following algorithm:

**Algorithm: Init_Centres_Kmeans_Logarithmic_Spiral**

**Begin**

**Input:**
- The database $X = \{x_1, x_2, ..., x_N\} \in \mathbb{R}^d$.
- The maximum number of centroids $k_{\text{max}}$.

**Output:**
- The position of each center $C = \{c_1, c_2, ..., c_{k_{\text{max}}}\} \in \mathbb{R}^d$.

**Step 1:** Calculate the maximum distance $D$ between two points belonging to the base $X$.
- Calculate the center $G$ of $D$ and the radius $R = GB$.

**Step 2:** Calculate the golden ratio $\phi$ by applying the following formula: $\phi = \frac{1 + \sqrt{5}}{2}$

**Step 3:** Initialize the values of the logarithmic spiral golden angle on the polar coordinate system $(r, \theta)$: the radius $r = a = a_0 = 0$ and the angle $\theta = \theta_0 = 0$.
- Fix the increment of the angle $\theta$ by the factor $d\theta = \psi = 2\pi \frac{(\phi - 1)}{\phi}$
- Fix the increment of the radius $r$ by the factor $da = \frac{R}{k_{\text{max}}}$

**Step 4:** Determine the positions of the centers belonging to the logarithmic spiral golden angle with center $G$ and radius $r = a$ according to the following formula:
- $C_{x_1} = G_x + a \times \cos(\theta)$
- $C_{y_1} = G_y + a \times \sin(\theta)$

With $\theta = \theta + d\theta$; $a = a + da$ and $k = 1, ..., k_{\text{max}}$.

**Step 5:** Save the positions of the centers found in $C = \{c_1, c_2, ..., c_{k_{\text{max}}}\} \in \mathbb{R}^d$.

**End**

**B. Evaluation Measures**

Using an unsupervised clustering algorithm, such as k-means algorithm, requires the determination of the number $k$ of groups leading to the execution of the algorithm repeatedly for different values of this parameter.

For optimal number of groups, a criterion should be used to evaluate the result of the algorithm. This criterion is known as the validity index [1], [20]-[22], [15, 16, 17, and 18] name based on the notions of compactness and separation.

In literature, there are a lot of validity indexes, most of them are based on the notions of compactness within different groups and the separability between these different groups. In this article, the Davies-Bouldin index and the Mean Squared Error will be used as two validity indexes of neural classifiers.

**C. Davies-Bouldin Index**

This index takes into account both of the compactness and the separability of groups [23]. Its value is much lower than the groups are compact and well separated. It promotes hyperspherical groups and is, therefore, particularly well-suited for a use with the k-means algorithm. The $I_{DB}$ index is defined by the following expression:

$$I_{DB} = \frac{1}{k} \sum_{i=1}^{k} \left[ \frac{d_i(c_i) + d_i(c_j)}{D_{cc}(c_i, c_j)} \right]$$

(6)

Where $d_i(c_i)$ is the average distance between an object and its group $c_i$ following the center and $D_{cc}(c_i, c_j)$ is the distance between the centers of groups $c_i$ and $c_j$ with:

$$d_i(c_i) = \frac{1}{N_i} \sum_{j=1}^{N_i} \|x_i - c_j\|$$

(7)

$$D_{cc}(c_i, c_j) = \|c_i - c_j\|$$

(8)

**D. Mean Squared Error**

The Mean Squared Error is frequently used to assess the risk of an estimator. It is also useful to relay the concepts of bias, precision, and accuracy in statistical estimation. In this work, the MSE was used for groups’ compactness measure for each centroid [24]; it is notably equivalent to the Euclidean function of the k-means algorithm:

$$E = \sum_{i=1}^{N} \sum_{j=1}^{k} \delta_{ij} \|x_i - c_j\|^2$$

(9)

With: $\delta_{ij} \begin{cases} 1 & \text{if } x_i \in c_i \\ 0 & \text{else} \end{cases}$

V. NEW ALGORITHMS FOR THE CONSTRUCTION OF THE HIDDEN LAYER OF THE RBF CLASSIFIER

Two new algorithms were used to characterize the hidden layer classifier i.e. to determine the number of centers of different Gaussian and the value of each center.

In what follows, the principle of the proposed classifiers is presented. It is also explained how the two validity indexes (DB and MSE) are combined with the k-means algorithm Mahalanobis distance to determine automatically the number $k$ of groups.

However, it is necessary to fix a maximum number of centroids $k_{\text{max}}$. The $k_{\text{max}}$ value can be defined by the user if he/she knows the structure of his database. Given that it is not always the case, the Bezdek [19] suggestion is adopted, so the $k_{\text{max}} = \sqrt{N}$ ($N$ is the size of the database) is chosen.

Applying these algorithms to all classes and summing the number of the obtained groups, the number of neurons in the hidden layer is determined. A neuron is then assigned to each group. For this RBF classifier, the database was partitioned $X = \{x_1, x_2, ..., x_N\} \in \mathbb{R}^d$ in individual blocks according to the

www.ijacsa.thesai.org
The database $X_d = \{X_{\Omega_1}, X_{\Omega_2}, \ldots, X_{\Omega_n}\}$ is obtained.

Then, we apply the Principal Component Analysis (PCA) to the data base $X_d$ in order to reduce it to a new basis two-dimension $X'_d = \{x_1, x_2\} \in R^2$.

Principal component analysis (PCA) is a widely used statistical technique for unsupervised dimension reduction. K-means clustering is a commonly used data clustering for performing unsupervised learning tasks [25].

The PCA is based on the calculation of averages, variances and correlation coefficients. The main basis of dimension reduction is that PCA picks up the dimensions with the largest variances.

The PCA allows, in the same time, a reduction of data and an easier interpretation in the treated domain, as the new dimensions are often very significant [26]. In this case, the two largest variances were chosen to represent the new database.

The next step is to determine the number of centers and the center position of each class $C_{\Omega_i} = \{c_i, \ldots, c_k\} \in R^d$ through the classifier based on the k-means algorithm with Mahalanobis distance.

The centers of each class $\Omega_j = 1, \ldots, m$ found in the matrix

$$ C = \begin{pmatrix} C_{\Omega_1} \\ \vdots \\ C_{\Omega_m} \end{pmatrix} $$

are grouped and the k-means algorithm Mahalanobis distance is applied to the new positions of the centers $C' = \{c_1', \ldots, c_k'\} \in R^d$.

To complete the construction of the hidden layer classifier, there is a second parameter to consider in the neurons, which is the width factor $\sigma_j$ for each centroid $c_j$ ($j = 1, \ldots, k$). This factor is calculated using the following formula:

$$ \sigma_j = \frac{1}{N_a} \sum_{i=1}^{N_a} \|x_i - c_j\| $$  (10)

Witch $N_a$ represents the training data

**A. Construction of the RBF classifier KMD-LS-IDB**

The proposed algorithm (KMD-LS-IDB) based on the k-means algorithm with Mahalanobis distance combined with the Davies-Bouldin validity index. This classifier determining the number and the centers values of the hidden layer for each class of the database is described below:

**Begin**

**Input:**
- The block database $X_{\Omega_j} = \{X_{\Omega_1}, \ldots, X_{\Omega_n}\} \in R^d$ of one class of data base $X_d$, taking the case of the block. (The same approach for other classes).

**Output:** - The position of each center $C_{\Omega_i} = \{c_i, \ldots, c_k\} \in R^d$.

**Step 1:** - Determine the size $n$ and the number of characters (the attributes) $p$ of the data base $X_{\Omega_i}$.

**Step 2:** - Initialize the minimum number of centroids $K_{\text{min}} = 2$ and then look for the maximum number of centroids by $K_{\text{max}} = \sqrt{n}$.
- Initialize the variables $a = k = K_{\text{min}}$ and $d = 1$.

**Step 3:** - Apply Algorithm `Init_Centres_Kmeans_Logarithmic_Spiral` which initializes the centers for kmeans algorithm of the data base $X_{\Omega_i}$.

**Step 4:** - Repeat the following steps until $k = K_{\text{max}}$.

**Step 4.1:** - If $k < K_{\text{max}} + 1$ Then:
- Take the following centers positions $C = \{c_1, \ldots, c_a\} \in R^d$.
- Deduce the number of the centers $k$.

**End If**

**Step 4.2:** - Apply k-means algorithm with Mahalanobis distance to determine the new positions of the centers $C' = \{c_1', \ldots, c_k'\} \in R^d$.

**Step 4.3:** - Calculate the compactness and separability of groups with the Davies- Bouldin Index: $I_{\text{DB}}$.

**Step 4.4:** - Save $I_{\text{DB}}$ variable in the table called `Tab_IDB`.

**Step 4.5:** - Increment variables $a = a + 1$, $d = d + 1$.

**Step 5:** - Determine $\beta$: the $I_{\text{DB}}$ lowest index of the table `Tab_IDB`.
- Take the following centers positions $C_{\Omega_i} = \{c_i, \ldots, c_{\beta a}\} \in R^d$ as the optimal number required classifying $\Omega_i$ class centers.

**End**

**B. Construction of RBF classifier KMD-LS-MSE**

The proposed algorithm (KMD-LS-MSE) based on the k-means algorithm with Mahalanobis distance combined with the Mean Squared Error validity index. This classifier determining the number and the centers values of the hidden layer for each class of the database is described below:
Y = H × W \tag{12}

The objective is to determine the matrix W that minimizes an error function, chosen as the square of the sum of classification errors.

The weight of the output layer can be calculated by the following matrix equation:

\[
\begin{bmatrix}
\varphi_{11} & \varphi_{12} & \cdots & \varphi_{1M} \\
\varphi_{21} & \varphi_{22} & \cdots & \varphi_{2M} \\
\vdots & \vdots & \ddots & \vdots \\
\varphi_{N1} & \varphi_{N2} & \cdots & \varphi_{NM}
\end{bmatrix}
\begin{bmatrix}
w_1 \\
w_2 \\
\vdots \\
w_M
\end{bmatrix}
= \begin{bmatrix}
y_1 \\
y_2 \\
\vdots \\
y_N
\end{bmatrix}
\tag{13}
\]

With

\[H : \text{The matrix of Radial Basis Functions}\]
\[Y : \text{The matrix of the output layer}.\]
\[W : \text{The weight matrix of centroids}.\]
\[\varphi_j : \text{The Gaussian widths of the matrix } H.\]

The above equation is giving by:

\[H \times W = Y \Rightarrow W = H^{-1} \times Y\] \tag{14}

Given that the H matrix is rarely square, the pseudo-inverse of the matrix H is applied according to equation (15):

\[W = \left[H^T \times H\right]^{-1} \times H^T \times Y\] \tag{15}

VI. EVALUATION OF RBF CLASSIFIERS

The purpose of this section is to evaluate the performance and efficiency of the proposed RBF neural classifiers. The performance of these RBF neural networks classifiers is tested with two different databases: Iris and Wine among the different data sets available from the machine learning community by the University of California at Irvine (UCI) [27].

The first test is carried out with the Wine database which includes the results of a chemical analysis of types of wine produced in different regions of Italy from different grapes varieties. The concentration of 13 components are indicated for each of the 178 types of wine (patterns) which are analyzed and divided into three classes (59 in Class 1, 71 in Class 2 and 48 in Class 3).

The second test is done with Iris database which is one of the most popular data set 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 Virginica) each having 50 patterns with four features.

To evaluate the proposed classifiers performances, the holdout method is used. It consists on dividing the initial data into two independent sets: one for training and the other for testing the classifier performances.

The results given by the RBF classifier built with this algorithm are compared with those obtained with other neural
classifiers: the Learning Vector Quantization (LVQ) classifier proposed by Kohonen, the RBF neural network classifier for which the hidden layer is obtained using adaptive Pattern Classifier (APCIII) [28], the Multi-Layer Perceptrons classifier (MLP) and with a reference: the K nearest Neighbor (KNN).

The present comparative results of different classifiers over Wine and Iris are illustrated in Table I and Table II.

### Table I. Results of the Recognition Rate over Wine Database

<table>
<thead>
<tr>
<th>Classification algorithms</th>
<th>Database: Wine</th>
</tr>
</thead>
<tbody>
<tr>
<td>KMD-LS-IDB</td>
<td>98.88 %</td>
</tr>
<tr>
<td>KMD-LS MSE</td>
<td>95.55 %</td>
</tr>
<tr>
<td>LVQ</td>
<td>66.14 %</td>
</tr>
<tr>
<td>APCIII</td>
<td>67.04 %</td>
</tr>
<tr>
<td>MLP</td>
<td>73.80 %</td>
</tr>
<tr>
<td>KNN</td>
<td>70.45 %</td>
</tr>
</tbody>
</table>

### Table II. Results of the Recognition Rate over Iris Database

<table>
<thead>
<tr>
<th>Classification algorithms</th>
<th>Database: Iris</th>
</tr>
</thead>
<tbody>
<tr>
<td>KMD-LS-IDB</td>
<td>93.46 %</td>
</tr>
<tr>
<td>KMD-LS MSE</td>
<td>93.46 %</td>
</tr>
<tr>
<td>LVQ</td>
<td>94.00 %</td>
</tr>
<tr>
<td>APCIII</td>
<td>93.33 %</td>
</tr>
<tr>
<td>MLP</td>
<td>96.66 %</td>
</tr>
<tr>
<td>KNN</td>
<td>96.70 %</td>
</tr>
</tbody>
</table>

Considering Wine database, the best recognition rate is obtained by the KMD-LS-IDB proposed classifier and then the proposed classifier KMD-LS-MSE. For Iris database, the best recognition rate is given for the KNN classifier; however, the difference with the two proposed classifiers is not important.

Then, the proposed algorithms give good results in terms of recognition rate but the most powerful of them is the classifier KMD-LS-IDB.

### VII. Conclusion

In this paper, new RBF neural networks classifiers have been designed to classify database. The proposed algorithms aim to deduce the centers of the hidden layer neurons and to calculate the number of the neurons in particular.

The basic idea of this approach is to select the training data from the database class by class and to decide about the optimal number of neurons in each class by using two different validity indexes (the validity index IDB and the MSE). This number is integrated in the k-means algorithm with the Mahalanobis distance.

A solution was also proposed to overcome the problem of initialization of centers necessary for the start of the K-means algorithm using the method of the logarithmic spiral golden angle.

The obtained classifiers results are satisfactory in comparison with other considered classifiers in literature for two real databases (Iris and Wine).

### References


