Evolutionary Generalized Radial Basis Function neural networks for improving prediction accuracy in gene classification using feature selection

Francisco Fernández-Navarro, César Hervás-Martínez, Roberto Ruiz, Jose C. Riquelme

Keywords:
Generalized Radial Basis Function
Generalized Gaussian Distribution
Evolutionary algorithm
Gene classification
Feature selection

A B S T R A C T

Radial Basis Function Neural Networks (RBFNNs) have been successfully employed in several function approximation and pattern recognition problems. The use of different RBFs in RBFNN has been reported in the literature and here the study centres on the use of the Generalized Radial Basis Function Neural Networks (GRBFNNs). An interesting property of the GRBF is that it can continuously and smoothly repro-duce different RBFs by changing a real parameter τ. In addition, the mixed use of different RBF shapes in only one RBFNN is allowed. Generalized Radial Basis Function (GRBF) is based on Generalized Gaussian Distribution (GGD), which adds a shape parameter, τ, to standard Gaussian Distribution. Moreover, this paper describes a hybrid approach, Hybrid Algorithm (HA), which combines evolutionary and gradient-based learning methods to estimate the architecture, weights and node topology of GRBFNN classifiers. The feasibility and benefits of the approach are demonstrated by means of six gene microarray classi-fication problems taken from bioinformatic and biomedical domains. Three filters were applied: Fast Correlation-Based Filter (FCBF), Best Incremental Ranked Subset (BIRS), and Best Agglomerative Ranked Subset (BARS); this was done in order to identify salient expression genes from among the thousands of genes in microarray data that can directly contribute to determining the class membership of each pattern. After different gene subsets were obtained, the proposed methodology was performed using the selected gene subsets as new input variables. The results confirm that the GRBFNN classifier leads to a promising improvement in accuracy.

1. Introduction

In traditional RBFNN, the Gaussian function is selected as the network activation function [1,2], although other functional forms have been used for the RBFNNs including some types of thin-plate spline functions, multi-quadratic functions and sigmoidal functions [3].

Nevertheless, there are still some problems in standard Gaussian RBFNN. First, if, the underlying curve representing training pat-terns is nearly constant in a specific interval, it is difficult to utilize a Gaussian function to approximate this constant valued function unless its width tends to infinity. In this case, a RBFNN would be an inefficient model to approximate constant valued functions.

Second, in high-dimensional space, all pairwise distance between patterns seem to be very similar, i.e., the distances to nearest and furthest neighbours look nearly identical. Therefore, the widely used Gaussian kernel and Euclidean distance are not necessarily appropriate functions to quantify similarity in high dimensional spaces because distances in this kind of problem are concentrated and the Gaussian kernel loses its interpretation in terms of locality around its centre [4,5]. Despite this, the use of Euclidean distance in high-dimensional space is not questioned in the machine learning community since it corresponds to distance as we define it in our three-dimensional world.

Third, several papers have included this as future experimentation to be performed with more general RBF models that achieve a good compromise between low training effort and flexible modelling capabilities [6–9].

In order to take care of these problems, a new activation function is presented in this paper. The proposed RBF is based on Generalized Gaussian Distribution (GGD) [10]. GGD adds a shape parameter τ to the normal distribution. In the same way that GGD adds a τ shape parameter to Gaussian Distribution, the novel RBF proposed, called Generalized Radial Basis Function (GRBF), also adds a τ shape parameter to standard Gaussian RBF (SRBF). The GRBF allows better matching between the shape of the kernel and the distribution of the distances, since the τ parameter provokes concavity or convexity around the point where the distance is the radii of the kernel, r.

On the other hand, although a great number of algorithms have been developed to estimate the parameters of a RBFNN for a fixed topology, most of them are hill-climbing procedures, which usually fall in a local optimum. Evolutionary algorithms (EAs) have proved to be very effective and robust search methods for locating zones in the search space where good solutions can be found, even if this space is large and contains multiple local optimums. The application of the EA to optimize RBFNN is justified by a smaller overall computational cost, compared to the methods of trial and error, and their robustness as opposed to constructive/pruning methods.

In general EAs are less efficient than local search techniques in finding the local optimum, so it is convenient to allow the EA to select initial solutions in good areas of the search space, and to locate local optimum in these areas afterwards. For that reason, a Hybrid Algorithm (HA) was employed to estimate the parameters of the model proposed. The HA combines both a global search procedure, EA, and a local improvement procedure based on gradient descent, the *iRprop+* algorithm. Thus, the EA optimizes RBFNN parameters so that RBFNN parameters are located in an area of global optimum and, the *iRprop+* algorithm refines its parameters to improve the results reported by the EA.

The performance of the proposed methodology was evaluated in six well-known deoxyribonucleic acid (DNA) microarray classification problems. DNA microarray allows relative levels of ribonucleic acid (RNA) or messenger RNA (mRNA) abundance to be determined in a set of tissues or cell populations for thousands of genes simultaneously.

The importance of the use of Artificial Neural Networks (ANNs) in the classification of microarray gene expression [11] as an alternative to other techniques was stated in several research works [12,13] due to their flexibility and the high degree of accuracy in their fit to experimental data. These datasets were selected to justify the use of this kernel model in the classification of high dimensionality problems. Furthermore, other soft computing techniques have been implemented to address this problem [14–16].

The motivation for applying feature selection (FS) techniques has shifted from being optional to becoming a real prerequisite for model building. A typical microarray dataset may contain thousands of genes but only a small number of samples (often less than two hundred). Theoretically, having more genes should give us more discriminating power. However, this can cause several problems: increased computational complexity and cost; too many redundant or irrelevant genes; and estimation degradation in the classification error.

There are two ways to group feature selection algorithms, depending on the evaluation measure chosen: one, according to the model used (filter or wrapper) and two, according to the way in which the features are evaluated (individually or by subsets). The filter model evaluates features according to heuristics based on overall data characteristics, notwithstanding the classification method applied, whereas the wrapper uses the behaviour of the target classification algorithm as the feature evaluation criterion.

Based on the generation procedure, FS can be divided into individual feature ranking (FR) and feature subset selection (FSS) [17,18]. FR measures feature-class relevance, then ranks features by their scores and selects the top-ranked ones. In contrast, FSS attempts to find a set of features that performs well. It integrates the metrics for measuring feature-class relevance and feature–feature interactions.

A hybrid model was proposed to handle large datasets to take advantage of the above two approaches (FR, FSS). These methods decouple relevance analysis and redundancy analysis, and have proven to be more effective than ranking methods and more efficient than subset evaluation methods in many traditional high-dimensional datasets. In this framework, FCBF (Fast Correlation-Based Filter) [19], BIRS (Best Incremental Ranked Subset) [20] and BARS (Best Agglomerative Ranked Subset) [21] are the methods proposed to obtain relevant features and to remove redundancy. These features are considered input variables in the network models (GRBFNN) that we propose in this paper

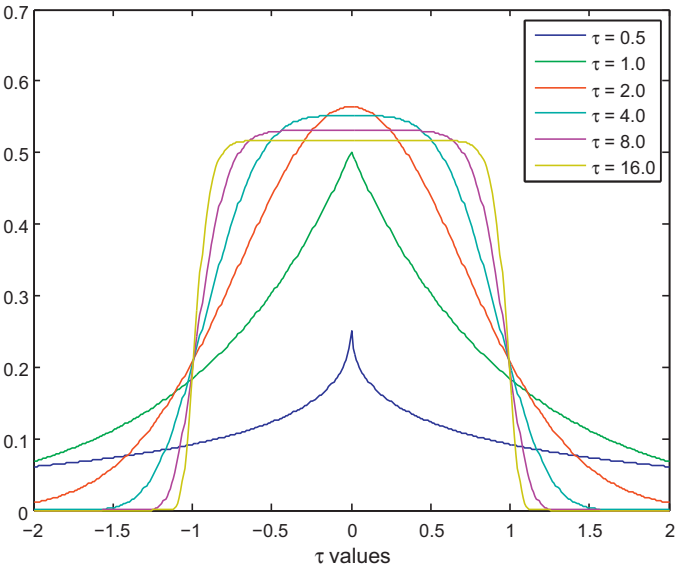

Fig. 1. Probability density function of the Generalized Gaussian Distribution (GGD) with different values of τ, $c=0$ and $r=1$.

One of the major advantages of the proposed method is the reduced number of features and GRBFs included in the final expression, since the HA reduces its complexity by pruning connections and hidden nodes. This can result in a better interpretability of the model, which is especially important when dealing with real problems. Therefore, using the proposed approach, the feature selection is performed in two stages: firstly, in preprocessing by means of the feature selector and secondly, in the HA by pruning connections.

This paper is organized as follows: Section 2 formally presents the GRBF model considered in this work and the main characteristics of the algorithm used for training the model. Section 3 introduces the feature selection algorithms used in this paper. Section 4 describes the experiments carried out and discusses the results obtained. Finally, Section 5 completes the paper with the main conclusions and future directions suggested by this study.

2. Classification method

2.1. Generalized Gaussian Distribution

Although Gaussian Distribution has a principal role in statistical applications, the analysis of real data often leads to rejecting the hypothesis that data have been generated by normal distribution. In these circumstances the adoption of more flexible models allowing the representation of data generated by distributions in an area near the Gaussian one may be appropriate. In particular, models which embed Gaussian distribution as a special case are of great interest, since their use permits deviations to be dealt with from normality, while preserving the possibility to test the adequacy of Gaussian distribution to the data. Generalized Gaussian Distribution (GGD) adds only one additional parameter to Gaussian distribution, the shape parameter τ. The GGD can approximate a large class of statistical distributions by modifying this parameter, for instance: the Gaussian distribution is obtained for $\tau=2$, the Laplacian distribution for $\tau=1$, and by making $\tau\to 0$ we can obtain a distribution close to uniform distribution (Fig. 1). The analytical equation for the

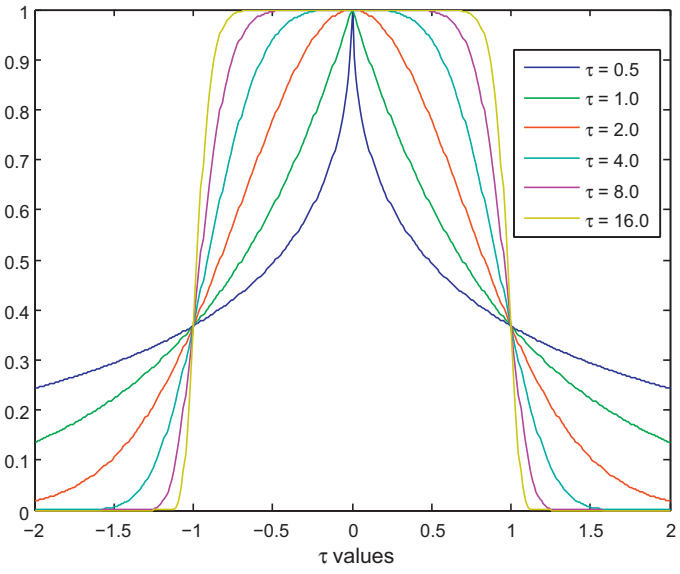

Fig. 2. Radial unit activation in one-dimensional space with $c=0$ and $r=1$ for the Generalized RBF (GRBF) with different values of τ.

probability density function of GGD, for a unidimensional input variable is given by

$$p_j(x;\tau_i,c_j,r_j)=\frac{\tau_j}{2r_j\Gamma(1/\tau_j)}\exp\left(-\frac{\|x-c_j\|^{\tau_j}}{r_j^{\tau_j}}\right),\quad j=1,\ldots,J \qquad (1)$$

where J is the number of classes in the problem, c_j, r_j and $\tau_j>0$ are the mean, the scale or width and a shape parameters, respectively, and $\tau_j>0$ of the ith class-conditional distribution, respectively. $\Gamma(z)=\int_0^\infty \tau^{z-1}e^z dz$, for $z>0$. The scale parameter r_j that expresses the width of the distribution is related to the normal standard deviation by the equation:

$$r_j=\sigma_j\sqrt{\frac{\Gamma(1/\tau_j)}{\Gamma(3/\tau_j)}} \qquad (2)$$

Based on the GGD probability distribution, we define a novel RBF, by removing the constraints of a probability function, that can generalize to the Standard Gaussian RBF (SRBF) by adding a new parameter τ which can relax or contract the basis functions. In this way, the Generalized Radial Basis Function (GRBF) is defined using the following expression, for a K-dimensional input space:

$$\phi(\mathbf{x})=\exp\left(-\frac{\|\mathbf{x}-\mathbf{c}_j\|^{\tau_j}}{r_j^{\tau_j}}\right) \qquad (3)$$

where K is the number of inputs, $\mathbf{x}_i=(x_{1i},\ldots,x_{Ki})$ is the vector of measurements, r_j the width of the GRBF, $\mathbf{c}_j=(c_{j1},\ldots,c_{jK})$ the centre and τ_j the exponent of the jth GRBF. Fig. 2 presents the radial unit activation for the GRBF for different values of τ.

From Fig. 2, another observation can be made: the shape of is approximately rectangular for high values of τ. This implies that the GRBF kernel should be a good candidate for approximating constant functions at specific intervals. Moreover, as shown in Fig. 2, GRBF has a unique maximum at radial symmetry, and a local support property which complies with the fundamental properties of RBF used in Artificial Neural Networks (ANNs). Finally, due to the τ parameter of the GRBF kernel, concavity or convexity is provoked around the point where the distance is the radii of the kernel, so the GRBF demonstrates appropriate kernel functions to quantify similarity in high dimensional spaces. When a normal distribution is

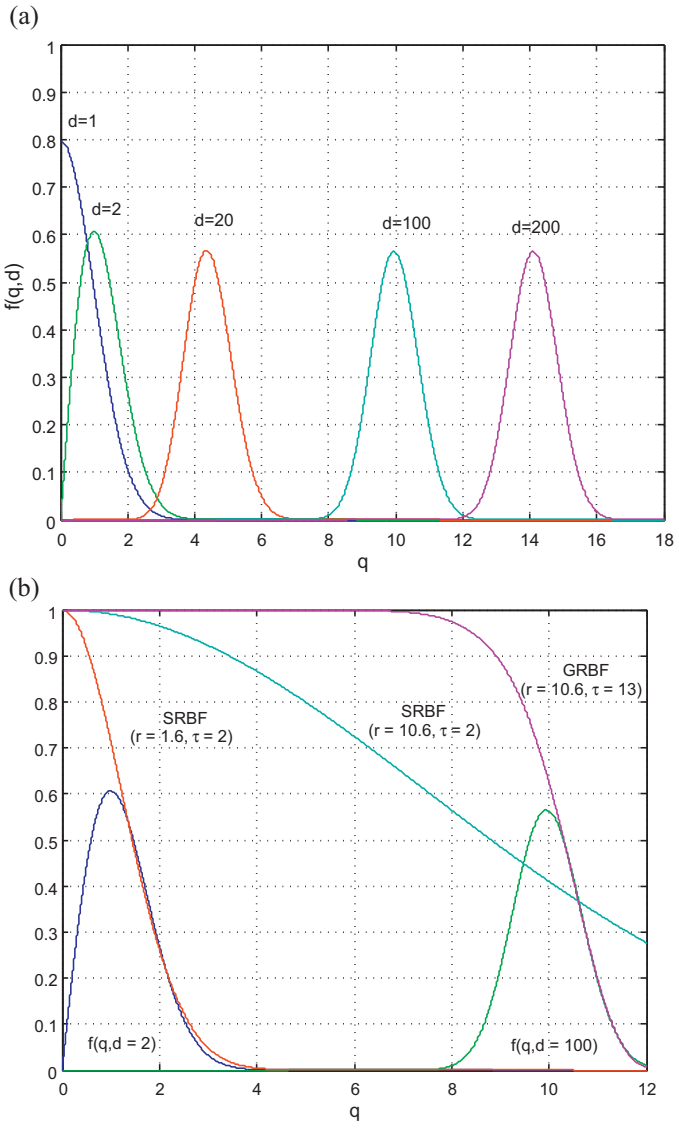

Fig. 3. Probability density of a point from a normal distribution to be at distance q.

assumed, the probability density function to find a point at distance q from the mean of the distribution is given by:

$$f(q,d)=\frac{q^{d-1}}{2^{(d/2)-1}}\times\frac{e^{-q^2/2}}{\Gamma(d/2)}(\sigma=1), \qquad (4)$$

where d is the input data dimension. For one dimension ($d=1$), $f(q, d)$ is maximum at the mean, but when the dimension grows, $f(q, d)$ moves away from the mean (see Fig. 3a). Therefore, the probability of finding patterns near the mean, when the dimension is high, is almost zero. In conclusion, SRBF are no longer local in higher dimensions, and those models that have been seen as sums of local kernels do not behave as such in high dimensions.

Fig. 3b shows that the GRBF provides better matching to the patterns in high dimensional spaces, because the τ parameter allows modification of the shape of the GRBF curvature. As can be observed in Fig. 3b when $d=100$, the SRBF needs a high value of r to model these patterns. When the SRBF has a high value of r, it has a slightly pronounced curvature which causes the SRBF to assign very similar grades of membership to the patterns located far from the cluster centre.

This effect can be observed in Fig. 3b: when $d=2$, the SRBF shows its ability to fit distance distribution, assigning membership values

1: **Hybrid Algorithm**:
2: Generate a random population of size N
3: **repeat**
4: Calculate the fitness of every individual in the population
5: Rank the individuals with respect to their fitness
6: The best individual is copied into the new population
7: The best 10% of population individuals are replicated and they substitute the worst 10% of individuals
8: Apply parametric mutation to the best $(p_m)\%$ of individuals
9: Apply structural mutation to the remaining $(100 - p_m)\%$ of individuals
10: **until** the stopping criterion is fulfilled
11: Apply *iRprop+* to the best solution obtained by the EA in the last generation.

Fig. 4. Hybrid Algorithm (HA) framework.

in the interval [0, 1]; however when $d = 100$, the SRBF assigns membership values in the interval [0.27–0.57], while the GRBF assigns membership values in the interval [0–1]. In our opinion and based also on the experimental results (Section 4.4), this justifies our considering GRBF to be an appropriate kernel to quantify similarity in high dimensional spaces.

2.2. Base classifier: probabilistic Generalized Radial Basis Functions

In a classification problem, measurements x_i, $i = 1, 2, \ldots, K$, of a single individual (or object) are taken, and the individuals are to be classified into one of the J classes based on these measurements. A training sample $D = \{(\mathbf{x}_n, \mathbf{y}_n); n = 1, 2, \ldots, N\}$ is available, where $\mathbf{x}_n = (x_{1n}, \ldots, x_{kn})$ is the random vector of measurements taking values in $\Omega \subset \mathbb{R}^K$, and $\mathbf{y}_n$ is the class level of the nth individual, where the common technique of representing class levels using a "1-of-J" encoding vector is adopted, $\mathbf{y} = (y^{(1)}, y^{(1)}, \ldots, y^{(J)})$, and the Correctly Classified Rate or accuracy of the classifier is defined by $C = (1/N)\sum_{n=1}^{N} I(C(\mathbf{x}_n) = \mathbf{y}_n)$, where $I(.)$ is the zero-one lost function.

In order to tackle this classification problem, the outputs of the GRBF model have been interpreted from the point of view of probability through the use of the softmax activation function, which is given by:

$$g_l(\mathbf{x}, \theta_l) = \frac{\exp f_l(\mathbf{x}, \theta_l)}{\sum_{j=1}^{J} \exp f_j(\mathbf{x}, \theta_j)}, \quad l = 1, 2, \ldots, J \tag{5}$$

where J is the number of classes in the problem, $f_j(\mathbf{x}, \boldsymbol{\theta}_l)$ is the output of the j output neuron for pattern $\mathbf{x}$ and $g_l(\mathbf{x}, \boldsymbol{\theta}_l)$ is the probability a pattern $\mathbf{x}$ has of belonging to class j. The model to estimate the function $f_l(\mathbf{x}, \boldsymbol{\theta}_l)$ is defined by the following equation:

$$f_l(\mathbf{x}, \theta_l) = \beta_0^l + \sum_{j=1}^{M} \beta_j^l \exp\left(-\frac{\|\mathbf{x} - \mathbf{c}_j\|^{\tau_j}}{r_j^{\tau_j}}\right), \quad l = 1, 2, \ldots, J \tag{6}$$

where M is the number of GRBFs or number of nodes in the hidden layer.

Using the softmax activation function presented in Eq. (5), the class predicted by the NN corresponds to the node in the output layer with the greatest output value.

The function used to evaluate a GRBF Neural Network (GRBFNN) is the function of cross-entropy error and it is given by the following expression:

$$l(\theta) = \sum_{n=1}^{N}[-\sum_{l=1}^{J} y_n^{(l)} f_l(\mathbf{x}_n, \theta_l) + log \sum_{l=1}^{J} \exp f_l(\mathbf{x}_n, \theta_l)] \tag{7}$$

where $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_J)$.

The error surface associated with the model is very convoluted with numerous local optima and the Hessian matrix of the error function $l(\boldsymbol{\theta})$ is, in general, indefinite. Moreover, the optimal number of basis functions in the model (i.e. the number of hidden nodes in the neural network) is unknown. Thus, we estimate the parameters $\boldsymbol{\theta}$ by means of a Hybrid Algorithm (HA).

2.3. Hybrid Algorithm

The proposed Hybrid Algorithm (HA) is composed by two stages. In the first stage, an Evolutionary Algorithm is used as a global stochastic search algorithm which generates candidate RBFNNs. In the second stage, the *iRprop+* algorithm performs a local optimization procedure to the best RBFNN individual of the last generation. Fig. 4 describes the procedure to estimate the parameters of GRBFNN.

The basic framework of the EA is the following: the search begins with an initial population of RBFNNs and, in each iteration, the population is updated using a population-update algorithm which evolves both its structure and weights. The population is subject to operations of replication and mutation. The main characteristics of the algorithm are the following:

1. *Representation of the individuals*. The algorithm evolves architectures and connection weights simultaneously, each individual being a fully specified RBFNN. The neural networks are represented using an object-oriented approach and the algorithm deals directly with the RBFNN phenotype. Each connection is specified by a binary value indicating if the connection exists, and the real value representing its weights.
2. *Error and fitness functions*. We consider $l(\boldsymbol{\theta})$ (Eq. 7) as the error function of an individual g in the population. The fitness measure needed for evaluating the individuals is a strictly decreasing transformation of the error function $l(\boldsymbol{\theta})$ given by $A(\boldsymbol{\theta}) = (1/1 + l(\boldsymbol{\theta}))$, where $0 < A(\boldsymbol{\theta}) \leq 1$.
3. *Initialization of the population*. The initial population is generated trying to obtain RBFNNs with the maximum possible fitness.

First, 5000 random RBFNNs are generated. The centres of the radial units are firstly defined by the k-means algorithm for different values of k, where $k \in [M_{min}, M_{max}]$, M_{min} and M_{max} being the minimum and maximum numbers of hidden nodes allowed for any RBFNN model in the HA. The widths of the RBFNNs are initialized to the geometric mean of the distance to the two nearest neighbourhood and the τ parameter to 2, since when $\tau = 2$ the GRBF reduces to the standard Gaussian RBF (SRBF). A random value in the $[-I, I]$ interval is assigned for the weights between the hidden layer and the output layer. The individuals obtained are evaluated using the fitness function and the initial population is finally obtained by selecting the best 500 RBFNNs.

4. *Parametric and structural mutations*. Parametric mutation consists of a simulated annealing algorithm [22]. Structural mutation implies a modification in the structure of the RBFNNs and allows the exploration of different regions in the search space, helping to keep the diversity of the population. There are four different structural mutations: hidden node addition, hidden node deletion, connection addition and connection deletion. These four mutations are applied sequentially to each network. More information about the genetic operators proposed can be seen in [23]. It is important to note the structural and parametric mutations of τ:
   - Structural mutation: If the structural mutator adds a new node in the RBFNN, the τ parameter is assigned to 2, since when $\tau = 2$ the GRBF reproduces to the Gaussian RBF.
   - Parametric mutation: The τ parameter is updated by adding a ε value, where $\varepsilon \in [-0.25, 0.25]$, since the modification of the GRBF is very sensitive to the τ variation (Fig. 2).
5. *iRprop+Local Optimizer*. The local optimization algorithm used in our paper is the *iRprop+* [24] optimization method. The *iRprop+* is believed to be a fast and robust learning algorithm. This algorithm applies a backtracking strategy (i.e. it decides whether to take a step back along a weight direction or not by means of a heuristic). In the methodology proposed, we run the EA and then apply the local optimization algorithm to the best solution obtained by the EA in the last generation. Further details about the adaptation of the *iRprop+* local improvement procedure to the softmax activation function can be seen in Appendix A of the paper.

Since the accuracy of the GRBFNN model was evaluated with bioinformatic datasets and a typical microarray dataset may contain thousands of genes, applying the FS techniques are a prerequisite for building the GRBFNN model in this context. The methodology proposed is shown in detail in Fig. 5.

3. Hybrid-generation feature selection

3.1. Introduction

The limitations of both the approaches introduced in Section 1 (FR and FSS) in high-dimensional spaces, clearly suggest the need for a hybrid model. The three methods used in this work can be labelled as this kind of framework, Hybrid-Generation Feature Selection.

In feature subset selection, it is a fact that two types of features are generally perceived as being unnecessary: features that are irrelevant to the target concept, and features that are redundant due to other features.

The purpose of a feature subset algorithm is to identify relevant features according to a definition of relevance. However, the notion of relevance in machine learning has not yet been rigorously defined by common agreement [25].

On the other hand, notions of feature redundancy are normally in terms of feature correlation. It is widely accepted that two features are redundant to each other if their values are completely correlated. There are two widely used types of measures for the correlation between two variables: linear and non-linear. In the linear, the Pearson correlation coefficient is used, and in the case of non-linear, many measures are based on the concept of entropy, or the measurement of the uncertainty of a random variable. Symmetrical uncertainty (SU) [26] is frequently used, defined as

$$SU(\mathbf{x}, \mathbf{y}) = 2\left[\frac{IG(\mathbf{x}|\mathbf{y})}{H(\mathbf{x}) + H(\mathbf{y})}\right]$$

where $H(\mathbf{x}) = -\sum_i^K P(x_i)\log_2(P(x_i))$ is the entropy of a variable $\mathbf{x}$ and $IG(\mathbf{x}|\mathbf{y}) = H(\mathbf{x}) - H(\mathbf{x}|\mathbf{y})$ is the information gain from $\mathbf{x}$ provided by $\mathbf{y}$. Both of them are between pairs of variables. However, it may not be as straightforward in determining feature redundancy when one is correlated with a set of features.

CFS [27] is one of well-known techniques to rank the relevance of features by measuring correlation between features and classes and between features and other features. The heart of the CFS (Correlation-based Feature Selection) algorithm contains a heuristic for evaluating the worth or merit of a subset of features. This heuristic takes into account the usefulness of individual features for predicting the class label, along with the level of intercorrelation among them. The hypothesis on which the heuristic is based is: *Good feature subsets contain features that are highly correlated with the class, yet uncorrelated with one another*.

$$Merit_S = \frac{k \times \overline{r_{cf}}}{\sqrt{k + k \times (k-1) \times \overline{r_{ff}}}}$$

where $Merit_S$ is the heuristic of a feature subset S containing k features, $\overline{r_{cf}}$ the average feature-class correlation, and $\overline{r_{ff}}$ the average feature–feature intercorrelation. For discrete class problems, CFS first discretizes numeric features and then uses symmetrical uncertainty to estimate the degree of association between the discretized features.

Due to the high computational cost in a high dimensional domain, we discard the wrapper approach, and the three methods use correlation concepts as relevance and redundancy criteria.

3.2. Fast Correlation-Based Filter (FCBF)

Aiming to achieve high efficiency, FCBF calculates SU–correlation between any F_i feature and class C generating a list in descending order, and heuristically decides a F_i feature to be relevant if it is highly correlated with class C, i.e., if $SU_{i,c} > \delta$, where δ is a relevance threshold which can be determined by users. The relevant features selected are then subject to redundancy analysis. Similarly, FCBF evaluates the SU-correlation between individual features for redundancy analysis based on an approximate Markov blanket concept. For two relevant F_i and F_j $(i \neq j)$ features, F_j can be eliminated if $SU_{i,c} \geq SU_{j,c}$ and $SU_{i,j} \geq SU_{j,c}$. The iteration starts from the first element in the ranking and continues as follows. For all the remaining features, if F_i happens to form an approximate Markov blanket for F_j, F_j will be removed from the list. After one round of filtering features based on F_i, the algorithm will take the remaining feature right next to F_i as the new reference to repeat the filtering process. The algorithm stops when no more features can be eliminated. Fig. 6 describes the FCBF method.

3.3. Best Incremental Ranked Subset (BIRS)

BIRS considers that relevance and redundancy concepts are included in the following "incremental usefulness" definition: given a sample of data, an evaluation measure L, a feature space

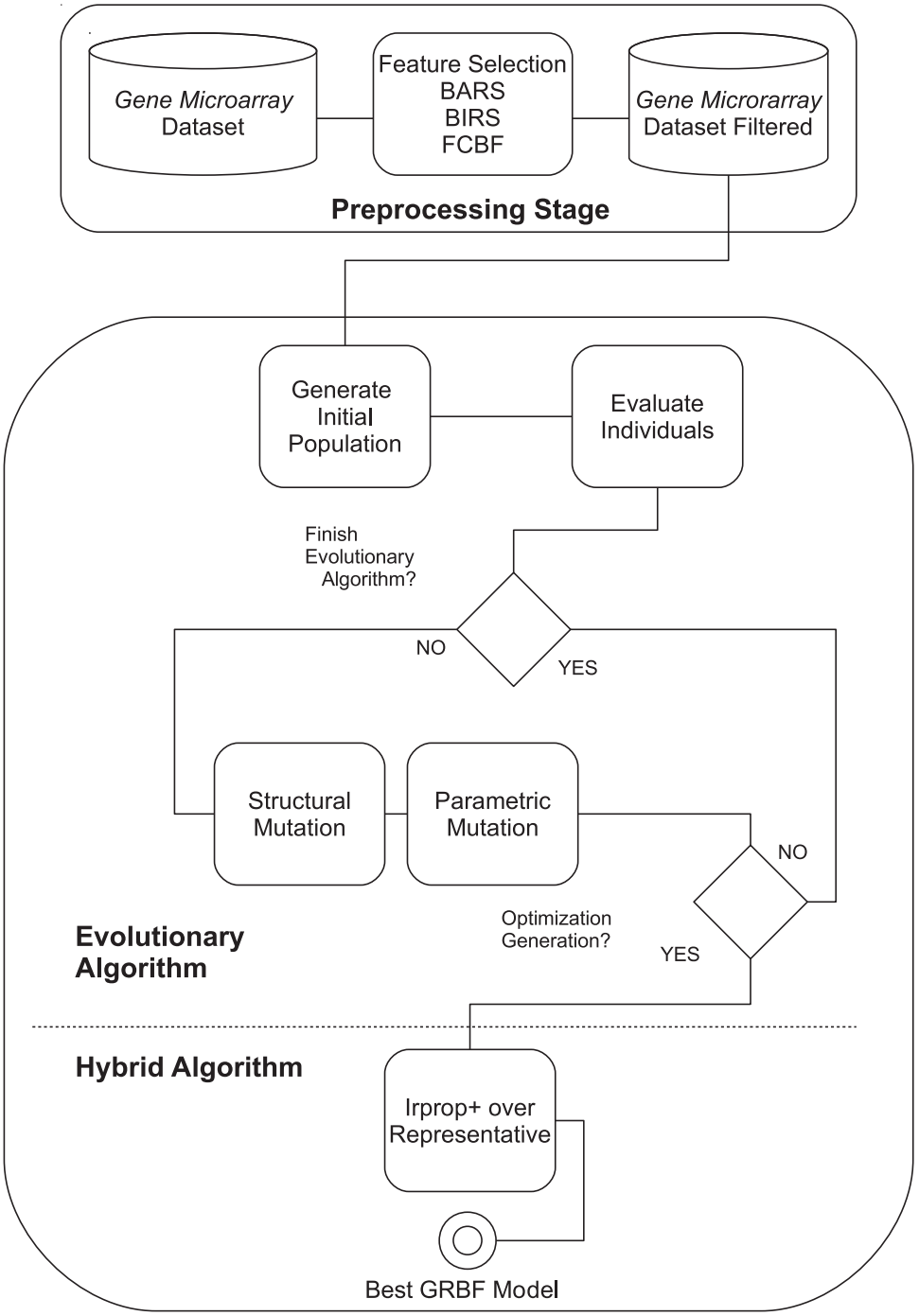

Fig. 5. Flow diagram of the GRBF method.

F and a feature subset **S** (**S** $\subseteq$ **F**), the feature F_i is incrementally useful to L with respect to **S** if the evaluation of the hypothesis that L produces using the group of features $\{F_i\} \cup \mathbf{S}$ is better than the evaluation achieved using just the subset of features **S**. i.e., if F_i is not incrementally useful to L with respect to **S**, then the evaluation value given the subset **S** is equal or better than the subset evaluation result known $\{F_i\} \cup \mathbf{S}$. It suggests that F_i gives no information beyond what is already in **S**, therefore, F_i could be removed safely, or in this case, F_i would not be added to **S**. However, since the computational complexity to determine all possible interactions among features is very high (mainly in high-dimensional domains), BIRS considers using a guided search in preference to an ordered list of attributes.

BIRS deals with incremental ranked usefulness in order to devise an approach to explicitly identify relevant features and not take into account redundant features. The idea is to choose the F_i feature from a ranked list one by one in the following way: firstly, the features are ranked according to some evaluation measure ($SU_{i,c}$); secondly, BIRS deals with the list of features once, crossing the ranking from the beginning to the last ranked feature. The evaluation results using CFS with the first feature in the list are obtained and it is marked as selected. The result is obtained again with the first and second features. The second will be marked as selected depending on whether the evaluation obtained is statistically significantly better. The process is repeated until the last feature on the ranked list is reached. Finally, the algorithm returns the best subset found, and

1: **FCBF Algorithm**:
2: Let R a feature ranking by SU, $R = [F_1, F_2, \ldots, F_n]$
3: Initialize FS with the all features in R that surpass a threshold
4: Let F_i the first feature in R
5: **repeat**
6: Remove of FS all features F_j such as $SU_{i,c} \geqslant SU_{j,c} \wedge SU_{i,j} \geqslant SU_{j,c}$
7: Set F_i the next feature in R and
8: **until** the end of the ranking
9: FS is the solution

Fig. 6. FCBF algorithm.

it will not contain irrelevant or redundant features. Fig. 7 describes the BIRS method.

3.4. Best Agglomerative Ranked Subset (BARS)

BARS is called agglomerative due to the way it constructs the final subset of selected features. The method begins by generating a ranking. Then, pairs of features are obtained with the ranking's first features, in combination with each one of the remaining features on the list. The pairs of features are ranked according to the value of the evaluation, and the process is repeated, that is, the subsets made up by the first sets on the new list are compared with the rest of the sets.

The process continues until the final subset obtained. Step one generates a feature ranking ranging from best to worst according to a correlation measure (CFS). Next, a list of solutions is generated, in such a way that a solution for each individual feature is created and the same ranking order is maintained. The agglomerative search consists of making a subset of relevant features by joining subsets with a lower number of features. With every iteration a new list of solutions from the previous structure is generated. Each candidate set, made by joining two sets from the previous list of solutions, will become part of the next list of solutions if, when the subset evaluator (CFS) is applied to it, it gives back a higher measure value than the one obtained with the best (or first) subset from the previous list of solutions. To prevent the algorithm from becoming prohibitively time consuming, new sets of features are generated by joining the first sets to the remaining previous list of solutions. That is, the first set on the list is joined to the second set, next the first set is joined to the third set, and so on until the end of the list. Next, the second set of the list is joined to the third set, the second set and the fourth set, and so on until the last set on the list. This process of combining a set of features with the rest of the sets on the list is carried out with the best *k* feature sets from the previous list of solutions. The process ends when only one feature subset is left, or when combining the subsets no longer causes an improvement. At the end, the algorithm returns the best positioned feature subset of all the subsets evaluated. Fig. 8 describes a sequence of numbered steps of the BARS method.

4. Experiments

This section presents the experimental results and analysis of GRBF models on 6 public microarray datasets with high dimensionality/small sample size. At the beginning, the datasets and several machine learning algorithms used in this analysis are briefly described. Subsequently, experimental results are given and discussed with respect to various aspects.

4.1. Microarray data

To validate the effectiveness of our method, a series of experiments were performed on 6 publicly available gene microarray datasets (Table 1). These datasets were taken from bioinformatic and biomedical domains. They are often used to validate the performance of the classifier and gene selector. Due to high dimensionality and small sample size, gene selection is an essential prerequisite for further data analysis. Brief descriptions of them are given in continuation.

Breast consists of 97 samples collected from breast cancer patients. 46 of them are from patients labelled as *relapse*, the rest of the 51 samples are from patients who remain healthy from the disease and are regarded as *non-relapse*. Each sample is described by 24,481 genes.

CNS (Central Nervous System) is derived from patient samples in embryonal tumours of the central nervous system. The total number of genes to be tested is 7129 and the number of samples is 60. There are two types of samples in the dataset, where 21 are survivors (who survive the treatment) and 39 are failures (who succumbed to the disease).

Colon uses Affymetrix oligonucleotide arrays to monitor expression levels of over 6500 human genes from 40 tumour and 22 normal colon tissue samples. The 2000 genes with the highest minimal intensity across the 62 tissues were used in this analysis.

Leukaemia refers to the primary disorders of bone marrow. This dataset contains 72 samples with malignant neoplasms of haematopoietic stem cells, of which 47 are *acute lymphobastic leukaemia* (ALL) and 25 *acute myeloid leukaemia* (AML). The total number of genes to be tested is 7129.

Lung has 12,600 genes in 203 samples. The 203 samples consist of 139 lung adenocarcinomas (AD), 21 squamous (SQ) cell carcinoma cases, 20 pulmonary carcinoid (COID) tumours and 6 small cell lung cancer cases (SCLC), as well as 17 normal lung (NL) samples.

GCM contains 190 samples. These samples are divided into 14 varieties of tumour. The expression levels of 16,063 genes are reported.

1: **BIRS Algorithm**:
2: Let R a feature ranking by SU, $R = [F_1, F_2, \ldots, F_n]$
3: Initialize FS with the first feature in R
4: **repeat**
5: Add the following feature F_i in R to FS and compare the CFS evaluator before and after. If adding the feature F_i the results don"t improve significantly, remove F_i of FS
6: **until** the end of the ranking
7: FS is the solution

Fig. 7. BIRS algorithm.

Table 1
Characteristics of the six datasets used for the experiments: feature selection type (*FS*), number of instances (Size), number of Real (R), Binary (B) and Nominal (N) input variables, total number of inputs (# In), number of classes (# Out), per-class distribution of the instances (Distribution), minimum and maximum number of hidden nodes used for each dataset ($[M_{min}, M_{max}]$) and the number of generations (# Gen).

Dataset	Source	FS	Size	R	B	N	# In	# Out	Distribution	$[M_{min}, M_{max}]$	*Gen*
Breast	Van't Veer et al. [40]	BARS	97	183	–	–	183	2	(46,51)	[1, 3]	100
		BIRS		261	–	–	261				
		FCBF		493	–	–	493				
CNS	Pomeroy et al. [41]	BARS	60	187	–	–	187	2	(21,39)	[1, 3]	10
		BIRS		206	–	–	206				
		FCBF		170	–	–	170				
Colon	Alon et al. [42]	BARS	62	58	–	–	58	2	(40,22)	[1, 3]	10
		BIRS		93	–	–	93				
		FCBF		59	–	–	59				
Leukaemia	Golub et al. [43]	BARS	72	225	–	–	225	2	(42,25)	[1, 3]	50
		BIRS		240	–	–	240				
		FCBF		203	–	–	203				
Lung	Bhattacharjee et al. [44]	BARS	203	237	–	–	237	5	(139,17,6,21,20)	[5, 8]	100
		BIRS		263	–	–	263				
		FCBF		250	–	–	250				
GCM	Ramaswamy et al. [45]	BARS	253	311	–	–	311	14	(11,10,11,11,22,	[25, 28]	400
		BIRS		288	–	–	288		11,10,10,30,11,		
		FCBF		264	–	–	264		11,11,11,20)		

In these 6 microarray datasets, all gene expression values are numeric. For convenience sake, they were standardized before our experiments, that is, for each gene represented, the mean and standard deviation were zero and one, respectively, after the standardized operation had been performed.

4.2. Alternative statistical and artificial intelligence methods used for comparison purposes

Different state-of-the-art statistical and artificial intelligence algorithms have been implemented for comparison purposes. Specifically, the results of the following algorithms have been compared to the GRBF method presented in this paper:

1. A Gaussian Radial Basis Function Network (RBFN) [28], deriving the centres and width of hidden units using *k*-means, and combining the outputs obtained from the hidden layer using logistic regression.
2. The MultiLogistic (MLogistic) algorithm. It is a method for building a multinomial logistic regression model with a ridge estimator to guard against overfitting by penalizing large coefficients [29].
3. The SimpleLogistic (SLogistic) algorithm. It is based on applying LogitBoost algorithm with simple regression functions and determining the optimum number of iterations by a five fold cross-validation [30].
4. The C4.5 classification tree inducer [31].
5. The Naive Bayes standard learning algorithm (NaiveBayes) [28].
6. The Logistic Model Tree (LMT) [30] classifier.
7. The IB1 classifier [32]. It uses a simple distance measure to find the training instance closest to the given test instance, and predicts the same class as this training instance.
8. The Support Vector Machine (SVM) classifier [33] with RBF kernels.

These algorithms have been selected for comparison since they are some of the best performing algorithms in recent literature on classification problems. Many of these approaches have also been tested before in the classification problem on microarray gene expression. The detailed description and some previous results of these methods can be found in [34,28,30].

1: **BARS Algorithm**:
2: Let R be a feature ranking by SU, $R = [F_1, F_2, \ldots, F_n]$, and a constant k where $k \ll n$
3: Let $L = [L_1, L_2, \ldots L_n]$ be a list of subset features, initially L has n subset, each one with only an ordered feature, i.e. $L_i = \{F_i\}$
4: **repeat**
5: Set $FS = L_1$ and set $T = CFS(L_1)$, i.e. FS is the most relevant subset at the moment and T is the result of its evaluation by CFS
6: Build M new feature subset by aggregating elements of L, $M = L_i U L_j$ with $i = 1..k$ and $j = i + 1..n$
7: If $CFS(M) > T$ add M to new list NL
8: Set $L =$ feature subset of NL in decreasing order by CFS
9: **until** L is empty
10: FS is the solution

Fig. 8. BARS algorithm.

Table 2
Comparison of the proposed method to other probabilistic methods: mean and standard deviation (SD) of the accuracy results (C_G(%)) from 30 executions, mean accuracy ($\overline{C}_G$(%)) and mean ranking ($\overline{R}$).

Dataset	FS	Method (C_G(%))								
		RBFN	MLogistic	SLogistic	C4.5	NaiveBayes	LMT	IB1	SVM	GRBF
		Result	Result	Result	Result	Result	Result	Result	Result	*Mean* ± *SD*
Breast	BARS	*80.00*	*80.00*	72.00	72.00	52.00	72.00	76.00	*80.00*	**83.06** ± **2.07**
	BIRS	72.00	76.00	64.00	64.00	76.00	64.00	80.00	**84.00**	*82.26* ± *5.32*
	FCBF	80.00	*84.00*	*84.00*	64.00	80.00	*84.00*	76.00	76.00	**86.66** ± **3.53**
CNS	BARS	**80.00**	73.33	66.66	66.66	66.66	66.66	73.33	66.67	*76.88* ± *3.38*
	BIRS	80.00	**93.33**	60.00	73.33	80.00	60.00	*86.66*	66.67	82.00 ± 3.97
	FCBF	86.66	**100.00**	80.00	60.00	86.66	80.00	93.33	66.67	*96.44* ± *5.46*
Colon	BARS	**93.75**	93.75	**100.00**	81.25	*93.75*	**100.00**	87.75	62.50	**100.00** ± **0.00**
	BIRS	81.25	68.75	67.50	**87.50**	81.25	**87.50**	81.25	62.50	*85.83* ± *4.32*
	FCBF	**87.50**	75.00	81.25	75.00	81.25	75.00	81.25	62.50	*86.04* ± *3.91*
Leukaemia	BARS	*94.44*	**100.00**	88.89	83.33	**100.00**	88.89	*94.44*	66.67	**100.00** ± **0.00**
	BIRS	*94.44*	**100.00**	*94.44*	83.33	**100.00**	*94.44*	**100.00**	66.67	**100.00** ± **0.00**
	FCBF	*94.44*	*94.44*	83.33	83.33	**100.00**	83.33	**100.00**	66.67	**100.00** ± **0.00**
Lung	BARS	96.07	96.07	*98.03*	94.11	96.07	*98.03*	*98.03*	*98.03*	**99.64** ± **0.98**
	BIRS	94.11	90.19	*98.03*	94.11	*98.03*	*98.03*	*98.03*	*98.03*	**98.45** ± **0.67**
	FCBF	94.11	94.11	**98.03**	74.50	94.11	**98.03**	94.11	94.11	*97.50* ± *0.98*
GCM	BARS	75.00	73.07	63.49	57.69	75.00	71.15	59.61	75.00	**82.32** ± **4.30**
	BIRS	76.92	78.84	75.00	57.69	78.84	75.00	75.00	76.92	**79.87** ± **3.45**
	FCBF	82.00	80.76	71.15	48.07	71.15	67.30	69.23	80.76	79.75 ± 4.90
$\overline{C}_G$(%)		85.70	*86.20*	80.32	73.32	83.93	81.29	84.66	75.02	**89.81**
$\overline{R}_{C_G}$		4.41	*4.22*	5.69	7.69	4.66	5.58	4.69	6.05	**1.97**

The best result is in bold face and the second best result in italics.

4.3. Experimental design

The evaluation of the different models has been performed using two different measures: Correctly Classified Rate (*CCR*) or accuracy and Root Mean Square Error (*RMSE*). *CCR* represents threshold metrics and *RMSE* a rank metric. *RMSE* is a metric corresponding to the expected value of the squared error loss or quadratic loss. *RMSE* is a frequently used measurement of the differences between values predicted by a model or an estimator, and the values actually observed in what is being modelled or estimated.

All the parameters used in the HA (Section 2.3) except the maximum and minimum number of RBFs in the hidden layer ([M_{min}, M_{max}]) and the number of generations (# *Gen*) have the same values in all problems analysed below (Table 1). The connections between hidden and output layer are initialized in the [−5, 5] interval (i.e. [−*I*, *I*] = [−5, 5]). The size of the population is *N* = 500. For the structural mutation, the number of nodes that can be added or removed is within the [1,2] interval, and the number of connections to add or delete in the hidden and the output layers during structural mutations is within the [1,7] interval.

For the selection of the SVM hyperparameters (regularization parameter, *C*, and width of the Gaussian functions, γ), a grid search algorithm has been applied with a ten-fold cross-validation, using the following ranges: $C \in \{2^{-5}, 2^{-3}, \ldots, 2^{15}\}$ and $\gamma \in \{2^{-15}, 2^{-13}, \ldots, 2^{3}\}$.

The experimental design was conducted using a holdout cross validation procedure with $3n/4$ instances for the training dataset and $n/4$ instances for the generalization dataset. In order to evaluate the stability of the methods, the evolutionary algorithm is run 30 times.

The HA and the model proposed was implemented in JAVA. We also used "libsvm" [35] to obtain the results of the SVM method, and WEKA[1] to obtain the results of the remaining methods.

[1] http://www.cs.waikato.ac.nz/ml/weka/

4.4. Comparison of the GRBF model with other classifiers

In this subsection, the GRBF model is compared to other base line classifiers. The purpose of this section is to show the improvement in accuracy in the classification problem of the microarray gene expression. Tables 2 and 3 show the mean and the standard deviation of the correct classification rate (C_G) and the Root Mean Square Error ($RMSE_G$) in the generalization set for each dataset and the RBFN, MLogistic, SLogistic, C4.5, NaiveBayes, LMT, IB1, SVM and GRBF methods. Based on the mean C_G and $RMSE_G$, the ranking of each method in each dataset ($R = 1$ for the best performing method and $R = 9$ for the worst one) is obtained and the mean accuracy and RMSE ($\overline{C}_G$ and $\overline{RMSE}_G$) and the mean ranking ($\overline{R}_{C_G}$ and $\overline{R}_{RMSE_G}$) are also included in Tables 2 and 3.

From the analysis of the results, it can be concluded, from a purely descriptive point of view, that the GRBF method obtained the best results for ten datasets in C_G and for fourteen datasets in $RMSE_G$. Furthermore, the GRBF method yield the best mean ($\overline{C}_G = 89.81$ %) and ranking ($\overline{R}_{C_G} = 1.97$) in C_G, and, taking $RMSE_G$ into account, the GRBF got the best performance in both measures ($\overline{RMSE}_G = 0.18$, $\overline{R}_{RMSE_G} = 1.75$).

To determine the statistical significance of the rank differences observed for each method in the different datasets, we have carried out a non-parametric Friedman test [36] with the ranking of C_G and $RMSE_G$ of the best models as the test variables (since a previous evaluation of the C_G and $RMSE_G$ values results in rejecting the normality and the equality of the variances' hypothesis). The test shows that the effect of the method used for classification is statistically significant at a significance level of 5%, as the confidence interval is $C_0 = (0, F_{0.05} = 2.00)$ and the *F*-distribution statistical values are $F^* = 8.19 \notin C_0$ for C_G and $F^* = 9.19 \notin C_0$ for $RMSE_G$. Consequently, we reject the null-hypothesis stating that all algorithms perform equally in mean ranking.

Based on this rejection, the Nemenyi post hoc test is used to compare all the classifiers to each other. This test considers that the performance of any two classifiers is deemed significantly

Table 3
Comparison of the proposed method to other probabilistic methods: Mean and Standard Deviation (SD) of the RMSE results (RMS_G) from 30 executions, mean RMSE ($\overline{RMSE}_G$) and mean ranking ($\overline{R}_{RMSE_G}$).

Dataset	FS	Method ($RMSE_G$)								
		RBFN	MLogistic	SLogistic	C4.5	NaiveBayes	LMT	IB1	SVM	GRBF
		Result	Result	Result	Result	Result	Result	Result	Result	*Mean* ± *SD*
Breast	BARS	0.42	0.44	0.41	0.53	0.69	0.41	0.48	**0.34**	*0.38* ± *0.01*
	BIRS	0.52	0.46	0.53	0.58	0.48	0.53	0.44	*0.38*	**0.36** ± **0.02**
	FCBF	0.41	0.37	*0.34*	0.57	0.43	*0.34*	0.48	0.42	**0.33** ± **0.04**
CNS	BARS	**0.43**	0.50	0.57	0.54	0.57	0.57	0.51	0.47	*0.44* ± *0.02*
	BIRS	0.43	**0.25**	0.51	0.50	0.44	0.51	0.36	0.47	*0.34* ± *0.04*
	FCBF	0.37	**0.00**	0.46	0.61	0.36	0.46	0.25	0.47	*0.18* ± *0.06*
Colon	BARS	*0.13*	0.25	0.18	0.43	0.24	0.18	0.35	0.49	**0.09** ± **0.06**
	BIRS	0.40	0.56	**0.37**	0.35	0.43	**0.37**	0.43	0.49	*0.38* ± *0.02*
	FCBF	*0.33*	0.50	0.36	0.48	0.43	0.39	0.43	0.49	**0.16** ± **0.01**
Leukaemia	BARS	0.23	**0.00**	0.32	0.40	**0.00**	0.32	*0.23*	0.47	**0.00** ± **0.00**
	BIRS	0.33	**0.00**	*0.20*	0.40	**0.00**	*0.20*	**0.00**	0.47	**0.00** ± **0.00**
	FCBF	*0.23*	*0.23*	0.40	0.39	**0.00**	0.40	**0.00**	0.47	**0.00** ± **0.00**
Lung	BARS	0.12	0.11	0.08	0.15	0.12	*0.08*	0.08	0.20	**0.05** ± **0.01**
	BIRS	0.12	0.19	0.08	0.13	*0.08*	*0.08*	*0.08*	0.14	**0.07** ± **0.05**
	FCBF	0.15	0.13	**0.08**	0.31	0.15	**0.08**	0.15	0.17	*0.11* ± *0.04*
GCM	BARS	*0.18*	0.19	0.22	0.24	*0.18*	0.20	0.24	0.22	**0.17** ± **0.07**
	BIRS	0.18	*0.16*	0.17	0.23	0.17	0.17	0.18	0.20	**0.15** ± **0.04**
	FCBF	**0.15**	**0.15**	*0.18*	0.26	0.20	0.21	0.20	*0.18*	**0.15** ± **0.08**
$\overline{RMSE}_G$		0.28	*0.24*	0.30	0.39	0.27	0.30	0.27	0.36	**0.18**
$\overline{R}_{RMSE_G}$		4.47	*4.38*	4.88	7.58	5.02	5.05	5.05	6.77	**1.75**

The best result is in bold face and the second best result in italics.

different if their mean ranks differ by at least the critical difference (CD):

$$CD = q\sqrt{\frac{K(K+1)}{6D}} \tag{8}$$

where K and D are the number of classifiers and datasets, and the q value is derived from the studentized range statistic divided by $\sqrt{2}$ [37,38]. However, it has been noted that the approach of comparing all classifiers to each other in a post hoc test is not as sensitive as the approach of comparing all classifiers to a given classifier (a control method). One approach to this latter type of comparison is the Bonferroni–Dunn test. This test can be computed using Eq. (8) with appropriate adjusted values of q [38].

The results of the Bonferroni–Dunn and Nemenyi tests for $\alpha = 0.10$ and $\alpha = 0.05$ can be seen in Tables 4 and 5, using the corresponding critical values (and also in the Bonferroni critical difference diagrams of Fig. 9). From the results of this test, it can be concluded that GRBF obtains a significantly higher $RMSE_G$ ranking when compared to all methods for $\alpha = 0.05$ and a significantly better C_G ranking when compared to all methods except MLogistic for $\alpha = 0.10$, which justifies the proposal.

4.5. Analysis of performance: GRBF versus SVM

The low accuracy provided by the SVM classifier is especially noteworthy. The reason for this has already been analysed by Klement [39]. As noted in Section 1, the proposed classifier was evaluated in gene microarray datasets. Such datasets have small sample size and high dimensionality and a well-known effect; if dimensionality is increased towards infinity, a finite set of points will lose more and more of its spatial topology. At the limit, the points will be located on the vertices of a regular simplex, i.e. all samples have nearly the same distances to the origin as well as from each other, and they are pair-wise orthogonal. Klement showed that even comparatively low dimensional data will behave as if infinitely dimensional. So, especially for low sample size data, infinity is rather small.

The main findings of this study about the performance of SVM in high dimensional and small sample size datasets were:

- Klement showed that the leave-one-out CV error for hard-margin SVMs will approach 1 in high-dimensional feature spaces for equal-sized classes drawn from the same distribution – despite the expected error rate of 0.5, which would be the outcome for the same setting in low dimensions. Moreover this observation was generalized to two classes drawn from different distributions.
- Due to the counterintuitive geometric properties of only a few samples in high-dimensional space and the asymmetries of a resampling scheme such as leave-one-out crossvalidation, the soft-margin approach did not increase the generalization performance of the hard-margin SVM.

It should be emphasized that dealing especially with high-dimensional but small sample size data leads to various counterintuitive and unfamiliar side effects which can have significant impact on training and validation.

4.6. Analysis of the best pair (classifier/feature selection algorithm) for the classification problem of microarray gene expression

The purpose of this last section is to determine which pair classifier/feature selection algorithm is the best methodology for the classification of microarray genes. In Table 6, the mean and standard deviation of the correct classification rate and the root mean square error in the generalization set (C_G and $RMSE_G$) are shown for each family of feature selectors (*BARS*, *BIRS* and *FCBF*).

From the analysis of the results, it can be concluded, from a purely descriptive point of view, that the GRBF model and the *FCBF* feature selection algorithm obtained the best mean result both in C_G as well as in $RMSE_G$. For this reason, the GRBF model and the *FCBF* feature selection algorithm are recommended to improve the

Table 4
Comparison of the GRBF method with other approaches: Critical Difference (CD) values and differences of rankings of the Nemenyi and Bonferroni–Dunn tests, using GRBF as the control method and C_G as the test variable.

Nemenyi test

Method (i)	Method (j)								
	RBFN	MLogistic	SLogistic	C4.5	NaiveBayes	LMT	IB1	SVM	GRBF
RBFN	–	0.19	1.27	$3.27^{\bullet}$	0.25	1.16	0.27	1.63	2.44
MLogistic	–	–	1.47	$3.47^{\bullet}$	0.44	1.36	0.47	1.83	2.25
SLogistic	–	–	–	2.00	1.02	0.11	1.00	0.36	$3.72^{+}_{\bullet}$
C4.5	–	–	–	–	$3.02^{+}_{\bullet}$	2.11	$3.00^{+}_{\bullet}$	1.63	$5.72^{+}_{\bullet}$
NaiveBayes	–	–	–	–	–	0.91	0.02	1.38	$2.69^{+}_{\circ}$
LMT	–	–	–	–	–	–	0.88	0.47	$3.61^{+}_{\bullet}$
IB1	–	–	–	–	–	–	–	1.36	$2.72^{+}_{\circ}$
SVM	–	–	–	–	–	–	–	–	$4.08^{+}_{\bullet}$

$CD_{\alpha=0.1} = 2.60$, $CD_{\alpha=0.05} = 2.83$

Bonferroni–Dunn test

Control Method	Compared Method								
	RBFN	MLogistic	SLogistic	C4.5	NaiveBayes	LMT	IB1	SVM	GRBF
GRBF	$2.44^{+}_{\circ}$	2.25	$3.72^{+}_{\bullet}$	$5.72^{+}_{\bullet}$	$2.69^{+}_{\bullet}$	$3.61^{+}_{\bullet}$	$2.72^{+}_{\bullet}$	$4.08^{+}_{\bullet}$	–

$CD_{\alpha=0.1} = 2.28$, $CD_{\alpha=0.05} = 2.48$

•, ○: Statistically difference with $\alpha = 0.05$ (•) and $\alpha = 0.1$ (○).
+: The difference is in favour of Method (j) (Nemenyi test) or Control Method (Bonferroni–Dunn test).

Table 5
Comparison of the GRBF method with other approaches: Critical Difference (CD) values and differences of rankings of the Nemenyi and Bonferroni–Dunn tests, using GRBF as the control method and $RMSE_G$ as the test variable.

Nemenyi test

Method (i)	Method (j)								
	RBFN	MLogistic	SLogistic	C4.5	NaiveBayes	LMT	IB1	SVM	GRBF
RBFN	–	0.08	0.41	$3.11^{\bullet}$	0.55	0.58	0.58	2.30	$2.72^{+}_{\circ}$
MLogistic	–	–	0.50	$3.19^{\bullet}$	0.63	0.66	0.66	2.38	$2.63^{+}_{\circ}$
SLogistic	–	–	–	$2.69^{+}_{\circ}$	0.13	0.16	0.16	1.88	$3.13^{+}_{\bullet}$
C4.5	–	–	–	–	2.55	2.52	2.52	0.80	$5.83^{+}_{\bullet}$
NaiveBayes	–	–	–	–	–	0.02	0.02	1.75	$3.27^{+}_{\bullet}$
LMT	–	–	–	–	–	–	0.00	1.72	$3.30^{+}_{\bullet}$
IB1	–	–	–	–	–	–	–	1.72	$3.30^{+}_{\bullet}$
SVM	–	–	–	–	–	–	–	–	$5.02^{+}_{\bullet}$

$CD_{\alpha=0.1} = 2.60$, $CD_{\alpha=0.05} = 2.83$

Bonferroni–Dunn test

Control Method	Compared Method								
	RBFN	MLogistic	SLogistic	C4.5	NaiveBayes	LMT	IB1	SVM	GRBF
GRBF	$2.72^{+}_{\bullet}$	$2.63^{+}_{\bullet}$	$3.13^{+}_{\bullet}$	$5.83^{+}_{\bullet}$	$3.27^{+}_{\bullet}$	$3.30^{+}_{\bullet}$	$3.30^{+}_{\bullet}$	$5.02^{+}_{\bullet}$	-

$CD_{\alpha=0.1} = 2.28$, $CD_{\alpha=0.05} = 2.48$

•, ○: Statistically difference with $\alpha = 0.05$ (•) and $\alpha = 0.1$ (○).
+: The difference is in favour of Method (j) (Nemenyi test) or Control Method (Bonferroni–Dunn test).

Table 6
Comparison of the proposed method to other baseline classifiers: Mean and Standard Deviation (SD) of the accuracy results ($C_G(\%)$) and RMSE results ($RMSE_G$) for each feature selection algorithm.

FS	RBFN	MLogistic	SLogistic	C4.5	NaiveBayes	LMT	IB1	SVM	GRBF
	Mean ± *SD*	*Mean* ± *SD*	*Mean* ± *SD*	*Mean* ± *SD*	*Mean* ± *SD*	*Mean* ± *SD*	*Mean* ± *SD*	*Mean* ± *SD*	*Mean* ± *SD*
Method ($C_G(\%)$)									
BARS	86.54 ± 9.20	86.03 ± 12.00	81.51 ± 16.15	75.84 ± 13.02	80.58 ± 19.14	82.78 ± 14.68	81.52 ± 14.53	74.81 ± 13.03	*90.31 ± 10.69*
BIRS	83.12 ± 9.21	84.51 ± 11.85	76.49 ± 16.10	76.66 ± 14.12	85.68 ± 10.48	79.82 ± 15.94	86.82 ± 10.16	75.79 ± 13.46	88.06 ± 8.86
FCBF	87.45 ± 5.98	88.05 ± 9.59	82.96 ± 8.71	67.48 ± 12.66	85.52 ± 10.39	81.27 ± 10.28	85.65 ± 11.98	74.45 ± 11.76	**91.06 ± 8.03**
Method ($RMSE_G$)									
BARS	0.25 ± 0.13	0.24 ± 0.19	0.29 ± 0.17	0.38 ± 0.15	0.30 ± 0.27	0.29 ± 0.17	0.31 ± 0.16	0.36 ± 0.13	*0.19 ± 0.18*
BIRS	0.33 ± 0.15	0.27 ± 0.20	0.31 ± 0.18	0.36 ± 0.16	0.26 ± 0.20	0.31 ± 0.18	0.24 ± 0.18	0.35 ± 0.15	0.22 ± 0.15
FCBF	0.27 ± 0.11	0.23 ± 0.17	0.30 ± 0.14	0.43 ± 0.14	0.26 ± 0.17	0.31 ± 0.14	0.25 ± 0.17	0.36 ± 0.15	**0.16 ± 0.10**

The best result is in bold face and the second best result in italics.

Table 7
Probability expression of the best GRBF model for the Colon dataset and using BARS as the feature selection algorithm. Performance of this model: Correct Classification Rate (CCR) on the training set (CCR_T), CCR on the generalization set (CCR_G), and Root Mean Square Error (RMSE) on the training set ($RMSE_T$), RMSE on the generalization set ($RMSE_G$). Confusion Matrix (CM) for the training set (CM_T) and CM for the generalization set (CM_G).

Best GRBF *Colon-BARS* Multi-Classification Model
$p_1(\mathbf{x},\boldsymbol{\theta}) = \frac{e^{f_1(\mathbf{x},\boldsymbol{\theta})}}{1+e^{f_1(\mathbf{x},\boldsymbol{\theta})}}; p_2(\mathbf{x},\boldsymbol{\theta}) = \frac{1}{1+e^{f_1(\mathbf{x},\boldsymbol{\theta})}}$
$f_1(\mathbf{x},\boldsymbol{\theta}) = -3.91 + 13.32 GRBF_1$
$f_2(\mathbf{x},\boldsymbol{\theta}) = 0$
$GRBF_1 = e^{-\left(\frac{\sqrt{(x_9^\star+2.20)^2+(x_{11}^\star+1.96)^2+(x_{14}^\star-0.17)^2+(x_{27}^\star+0.59)^2+(x_{30}^\star+1.14)^2+(x_{35}^\star+1.26)^2+(x_{37}^\star+1.22)^2}+}{5.38}\right)^{4.58}}$
$\frac{\sqrt{(x_{39}^\star-0.58)^2+(x_{44}^\star-0.12)^2+(x_{45}^\star-1.48)^2+(x_{47}^\star-0.26)^2+(x_{53}^\star-0.08)^2+(x_{54}^\star+0.09)^2+(x_{56}^\star-0.07)^2}}{5.38}$
$x_i^\star \in N[0,1], i = 1,\ldots,58$
$CCR_T = 91.30\%, CCR_G = 100.00\%$
$RMSE_T = 0.23, RMSE_G = 0.18$
$CM_T = \begin{pmatrix} 28 & 2 \\ 2 & 14 \end{pmatrix}; CM_G = \begin{pmatrix} 10 & 0 \\ 0 & 6 \end{pmatrix}$

accuracy value and root mean square error in the classification of microarray gene expression.

4.7. Analysis of the best GRBF model obtained for the Colon dataset using the BARS as the feature selection algorithm

As discussed above, the proposed method reduces the number of features using a two-step procedure. In the first stage, a hybrid method of feature selection reduces the number of genes from thousands to hundreds. In the second one, the HA reduces the number of genes from hundreds to tens by pruning connections and removing nodes in the hidden layer. This can result in a better interpretability of the model, which is especially important when dealing with real problems. In order to analyse the importance of this feature reduction in the genome data sets, in this section, we present an example of this reduction using the Colon datasets and the BARS feature selector.

Thus, Table 7 includes the best predictor functions of the GRBF model obtained for the Colon problem using BARS as the feature selection algorithm. As discussed in Section 4, the dataset includes 58 input variables and the observations are to be classified in two classes. From these predictor functions, the probability that each pattern x has of belonging to each class can be easily derived by using softmax functions.

As we can see in Table 7, the final GRBF model includes only 14 input variables, since feature selection has been performed in two stages: first, the BARS algorithm reduces the input space of 2000 input variables to 58; secondly, the HA dynamically eliminates variables by pruning connections.

As stated in subsection 4.1 mean and standard deviation of datasets were zero and one, respectively, after performing the standardized operation. Based on this information, it is possible to rank the most influential genes according to their discriminatory ability. Thus, the most relevant genes of the model in Table 7 (Colon dataset) are those whose kernel average is close to zero, since there is a greater likelihood of the kernel representing the values obtained in this direction of the input space. The eight most influential genes are shown in decreasing order in the first column of Table 8.

Besides reporting the gene accession number (Genbank) and giving a brief description of the gene in Table 8, we point out its ranking in works about gene selection published with respect to the same dataset, although an asterisk indicates the presence of the gene in the respective publication when no ranking was

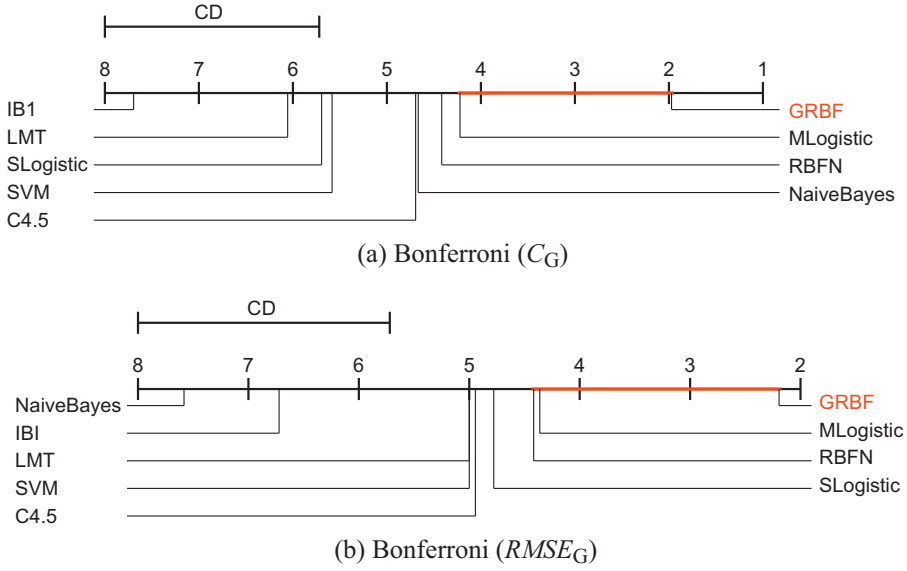

Fig. 9. Comparison of the GRBF method with other approaches: graphic of performance comparison ($\alpha = 0.10$). (a) Bonferroni (C_G), (b) Bonferroni ($RMSE_G$).

Table 8
The most relevant selected genes used by the final GRBF model in colon cancer data: Gene accession number (Genbank), description and [] denotes the rank in papers published on the same dataset.

Genbank	Description	[46]	[47]	[48]	[49]	[50]	[51]	[52]	[53]	[54]
T41204	P14780 92 KD TYPE V COLLAGENASE PRECURSOR	26		21		16				
J05032	Human aspartyl-tRNA synthetase alpha-2 subunit mRNA, complete cds.		3		6		*			5
H08393	COLLAGEN ALPHA 2(XI) CHAIN (Homo sapiens)	2	6	3	5	1		5	2	1
R08021	INORGANIC PYROPHOSPHATASE (Bos taurus)							11	5	
D14812	Homo sapiens KIAA0026 mRNA, complete cds.	3		18						10
M88108	Human p62 mRNA, complete cds.						*			
M34344	Human platelet Glycoprotein IIb (GPIIb) gene, exon 30.									
R84411	SMALL NUCLEAR RIBONUCLEOPROTEIN ASSOCIATED PROTEINS B AND B' (HUMAN)		74		38	4				
	Number of genes:	(26)	(77)	(80)	(90)	(46)	(–)	(15)	(5)	(50)

performed. The end of each column also shows the number of genes selected in the corresponding reference.

As we can observe, in all cases, except for M34344, these genes were among the high-ranked genes obtained by other methods. It is noteworthy that the H08393 gene, one of the three most relevant genes in our model, appears in the first positions in all the papers in question but one. Therefore, our method not only is able to get good accuracy values, but also proves that it can supply additional valuable information with regard to feature influence.

5. Conclusions

In this paper, we analyse the performance of a novel Radial Basis Function classifier, called Generalized Radial Basis Function (GRBF), which is based on Generalized Gaussian distribution, in DNA microarray classification. The coefficients of the neural network classifier proposed are estimated by a Hybrid Algorithm (HA). The HA proposed uses an Evolutionary Algorithm (EA) to locate the GRBF near an optimal point (global). Then, the *iRprop+* algorithm (local search) is applied to the best GRBF obtained in the EA to reach the optimal point. Due to the enormously high dimensionality of the DNA microarray dataset, three algorithm filters were applied to reduce noise and to improve accuracy classification.

The proposed methodology (composed of two stages) for microarray gene classification allows thousands of features (24,481 in Breast) to be reduced to tens (14 in Colon). This reduction of features is obtained by applying the feature selector that reported the best results in terms of accuracy in the preprocessing stage, the FCBF method, and by means of performing operations that removed connections and hidden nodes incorporated by the Hybrid Algorithm (HA). The average results for the 6 datasets using the FCBF feature selector and the GRBF classifier show values over 91% in accuracy and under 0.17 in RMSE.

Finally, because so few features obtained their best models, it is possible to interpret them and then analyse the causal relationship between gene characteristics and the probability of belonging to each class. An example of the interpretation of the best model has been discussed in the previous section on Colon/BARS datasets, which was observed to correctly classify 100% of the patterns in both classes of the generalization set.

Acknowledgements

This work has been partially subsidized by the TIN 2008-06681-C06-03 project of the Spanish Inter-Ministerial Commission of Science and Technology (MICYT), FEDER funds, and the P08-TIC-3745 project of the "Junta de AndalucA-a" (Spain). The research of Francisco Fernández-Navarro has been funded by the "Junta de Andalucia" Predoctoral Program, grant reference 390015-P08-TIC-3745.

Appendix A. Adaptation of the *iRprop+* algorithm to the softmax function

We have carried out the adaptation of the *iRprop+* local improvement procedure to the softmax activation function (Eq. (5)) and the cross-entropy error function (Eq. (7)). In this case, the gradient vector is given by the following equation:

$$\nabla l(\boldsymbol{\beta_l}, \mathbf{c}, \boldsymbol{r}, \mathbf{q}) = \left(\frac{\partial l}{\partial \boldsymbol{\beta_l}}, \frac{\partial l}{\partial \mathbf{c}}, \frac{\partial l}{\partial \boldsymbol{r}}, \frac{\partial l}{\partial \boldsymbol{\tau}} \right) \tag{9}$$

Let η_l be any of the parameters of $\boldsymbol{\beta_l}$, $\mathbf{c}$, $\boldsymbol{r}$ and $\boldsymbol{\tau}$, being therefore:

$$\frac{\partial l}{\partial \eta_l} = \frac{1}{N} \sum_{n=1}^{N} \sum_{l=1}^{J-1} y_n^{(l)} \frac{1}{g_l(\mathbf{x}_n)} \frac{\partial g_l(\mathbf{x}_n, \theta_l)}{\partial \eta_l}$$

$$\frac{\partial g_l(\mathbf{x}, \theta_l)}{\partial \eta_l} = \frac{1}{\left(1 + \sum_{l=1}^{J-1} e^{f_l}\right)^2} \left(e^{f_l} \frac{\partial f_l}{\partial \eta_l} \left(1 + \sum_{l=1}^{J-1} e^{f_l}\right) - e^{f_l} \sum_{l=1}^{J-1} e^{f_l} \frac{\partial f_l}{\partial \eta_l} \right)$$

$$\frac{\partial g_l(\mathbf{x}, \theta_l)}{\partial \eta_l} = g_l \frac{\partial f_l}{\partial \eta_l} - g_l^2 e^{-f_l} \sum_{l=1}^{J-1} e^{f_l} \frac{\partial f_l}{\partial \eta_l}$$

Finally, we have the following expressions for the output layer

$$\frac{\partial f_l}{\partial \beta_0^k} = \begin{cases} 0 & k \neq l \\ 1 & k = l \end{cases}$$

$$\frac{\partial f_l}{\partial \beta_s^k} = \begin{cases} 0 & k \neq l; \\ \phi_s(d_s(\mathbf{x})) & k = l \end{cases}$$

and for the hidden layer:

$$\frac{\partial f_l}{\partial c_{ts}} = \beta_s^l \frac{\phi_s(\mathbf{x}) \tau_s (x_{st} - c_{st}) (-\ln \phi_s(\mathbf{x}))^{\frac{\tau_s - 2}{\tau_s}}}{r_s^2},$$

$$\frac{\partial f_l}{\partial r_s} = \beta_s^l \frac{\phi_s(\mathbf{x}) \tau_s (-\ln \phi_s(\mathbf{x}))}{r_s},$$

$$\frac{\partial f_l}{\partial \tau_s} = \beta_s^l \frac{\phi_s(\mathbf{x}) \ln(\phi_s(\mathbf{x})) \ln(-\ln \phi_s(\mathbf{x}))}{\tau_s}$$

where $s = 1, 2, \ldots, M$, $l = 1, 2, \ldots, J-1$ and $t = 1, 2, \ldots, K$

References

[1] J.A.S. Freeman, D. Saad, Learning and generalization in radial basis function networks, Neural Computation 7 (5) (1995) 1000–1020.

[2] F. Fernández-Navarro, A. Valero, C. Hervás-Martínez, P.A. Gutiérrez, R.M. García-Gimeno, G. Zurera-Cosano, Development of a multi-classification neural network model to determine the microbial growth/no growth interface, International Journal of Food Microbiology 141 (3) (2010) 203–212.

[3] C. Lee, P. Chung, J. Tsai, C. Chang, Robust radial basis function neural networks, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 29 (6) (1999) 674–685.
[4] M. Verleysen, D. François, G. Simon, V. Wertz, On the effects of dimensionality on data analysis with neural networks, in: J.A.e.J. Mira (Ed.), Artificial Neural Nets Problem solving methods, Lecture Notes in Computer Science 2687, Springer-Verlag, 2003, 105–112(2).
[5] F. Fernández-Navarro, C. Hervás-Martínez, J. Sánchez-Monedero, P.A. Gutierrez, MELM-GRBF: a modified version of the extreme learning machine for generalized radial basis function neural networks, Neurocomputing 74 (16) (2011) 2502–2510.
[6] D. Fisch, A. Hofmann, B. Sick, On the versatility of radial basis function neural networks: a case study in the field of intrusion detection, Information Sciences 180 (12) (2010) 2421–2439.
[7] F. Fernández-Navarro, C. Hervás-Martínez, M. Cruz, P.A. Gutierrez, A. Valero, Evolutionary q-Gaussian radial basis function neural network to determine the microbial growth/no growth interface of *Staphylococcus aureus*, Applied Soft Computing 11 (3) (2011) 3012–3020.
[8] F. Fernández-Navarro, C. Hervás-Martínez, P.A. Gutierrez, M. Carboreno, Evolutionary q-Gaussian radial basis functions neural networks for multi-classification, Neural Networks 24 (7) (2011) 779–784, URL http://dx.doi.org/10.1016/j.neunet.2011.03.014.
[9] F. Fernández-Navarro, C. Hervás-Martínez, P.A. Gutierrez, J. Peña, F. López-Granados, Parameter estimation of q-Gaussian radial basis functions neural networks with a hybrid algorithm for binary classification, Neurocomputing 75 (2012) 123–134.
[10] S. Nadarajah, A generalized normal distribution, Journal of Applied Statistics 32 (7) (2005) 685–694.
[11] B. Geng, X. Zhou, Y.S. Hung, Growing enzyme gene networks by integration of gene expression, motif sequence, and metabolic information, Pattern Recognition 42 (4) (2009) 557–561.
[12] S. Wang, X. Li, S. Zhang, J. Gui, D. Huang, Tumor classification by combining PNN classifier ensemble with neighborhood rough set based gene reduction, Computers in Biology and Medicine 40 (2) (2009) 179–189.
[13] M. Takahashi, H. Hayashi, Y. Watanabe, K. Sawamura, N. Fukui, J. Watanabe, T. Kitajima, Y. Yamanouchi, N. Iwata, K. Mizukami, T. Hori, K. Shimoda, H. Ujike, N. Ozaki, K. Iijima, K. Takemura, H. Aoshima, T. Someya, Diagnostic classification of schizophrenia by neural network analysis of blood-based gene expression signatures, Schizophrenia research 119 (1) (2010) 210–218.
[14] R. Bhattacharyya, Cohesion: a concept and framework for confident association discovery with potential application in microarray mining, Applied Soft Computing 11 (1) (2011) 592–604.
[15] U. Maulik, Analysis of gene microarray data in a soft computing framework, Applied Soft Computing 11 (6) (2011) 4152–4160.
[16] C.-P. Lee, Y. Leu, A novel hybrid feature selection method for microarray data analysis, Applied Soft Computing 11 (1) (2011) 208–213.
[17] A. Blum, P. Langley, Selection of relevant features and examples in machine learning, Artificial Intelligence 97 (1–2) (1997) 245–271.
[18] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, Journal of Machine Learning Research 3 (2003) 1157–1182.
[19] L. Yu, H. Liu, Efficient feature selection via analysis of relevance and redundancy, Journal of Machine Learning Research 5 (2004) 1205–1224.
[20] R. Ruiz, J. Riquelme, J. Aguilar-Ruiz, Incremental wrapper-based gene selection from microarray expression data for cancer classification, Pattern Recognition 39 (2006) 2383–2392.
[21] R. Ruiz, J. Aguilar-Ruiz, J. Riquelme, Best agglomerative ranked subset for feature selection, JMLR Workshop and Conference Proceedings 4 (2008) 146–160.
[22] F.J. Martínez-Estudillo, C. Hervás-Martínez, P.A. Gutiérrez, A.C. Martínez-Estudillo, Evolutionary product-unit neural networks classifiers, Neurocomputing 72 (1-2) (2008) 548–561.
[23] C. Hervás-Martínez, F. Martínez-Estudillo, Logistic regression using covariates obtained by product-unit neural network models, Pattern Recognition 40 (1) (2007) 52–64.
[24] C. Igel, M. Hüsken, Empirical evaluation of the improved rprop learning algorithms, Neurocomputing 50 (6) (2003) 105–123.
[25] D. Bell, H. Wang, A formalism for relevance and its application in feature subset selection, Machine Learning 41 (2) (2000) 175–195.
[26] W. Press, B. Flannery, S.A. Teukolski, W. Vetterling, Numerical Recipes in C, Cambridge University Press, London, GB, 1988.
[27] M. Hall, Correlation-based feature selection for discrete and numeric class machine learning, in: Proceedings of the 17th International Conference on Machine Learning, Morgan Kaufmann, San Francisco, CA, 2000, pp. 359–366.
[28] I.H. Witten, E. Frank, Data Mining: Practical Machine Learning Tools and Techniques. 2nd edition. Data Management Systems, Morgan Kaufmann (Elsevier), 2005.
[29] S. le Cessie, J. van Houwelingen, Ridge estimators in logistic regression, Applied Statistics 41 (1) (1992) 191–201.
[30] N. Landwehr, M. Hall, E. Frank, Logistic model trees, Machine Learning 59 (1-2) (2005) 161–205.
[31] J.R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann, 1993.
[32] D.W. Aha, D. Kibler, M.K. Albert, Instance-based learning algorithms, Machine Learning 6 (1) (1991) 37–66.
[33] V.N. Vapnik, The Nature of Statistical Learning Theory, Springer, 1999.
[34] T. Hastie, R. Tibshirani, J.H. Friedman, The Elements of Statistical Learning, Springer, 2001.
[35] C. Chang, C. Lin, Libsvm: A Library for Support Vector Machines, 2001.
[36] M. Friedman, A comparison of alternative tests of significance for the problem of *m* rankings, Annals of Mathematical Statistics 11 (1) (1940) 86–92.
[37] O.J. Dunn, Multiple comparisons among means, Journal of the American Statistical Association 56 (1961) 52–56.
[38] Y. Hochberg, A. Tamhane, Multiple Comparison Procedures, John Wiley & Sons, 1987.
[39] S. Klement, A. Madany Mamlouk, T. Martinetz, Reliability of cross-validation for SVMs in high-dimensional, low sample size scenarios, in: Proceedings of the 18th International Conference on Artificial Neural Networks, Part I, ICANN '08, 2008, pp. 41–50.
[40] L.J. Van't Veer, H. Dai, M.J. Van de Vijver, Y.D. He, A.A.M. Hart, M. Mao, H.L. Peterse, K. Van Der Kooy, M.J. Marton, A.T. Witteveen, G.J. Schreiber, R.M. Kerkhoven, C. Roberts, P.S. Linsley, R. Bernards, S.H. Friend, Gene expression profiling predicts clinical outcome of breast cancer, Nature 415 (6871) (2002) 530–536.
[41] S.L. Pomeroy, P. Tamayo, M. Gaasenbeek, L.M. Sturla, M. Angelo, M.E. McLaughlin, J.Y.H. Kim, L.C. Goumnerova, P.M. Black, C. Lau, J.C. Allen, D. Zagzag, J.M. Olson, T. Curran, C. Wetmore, J.A. Biegel, T. Poggio, S. Mukherjee, R. Rifkin, A. Califano, G. Stolovitzky, D.N. Louis, J.P. Mesirov, E.S. Lander, T.R. Golub, Prediction of central nervous system embryonal tumour outcome based on gene expression, Nature 415 (6870) (2002) 436–442.
[42] U. Alon, N. Barka, D.A. Notterman, K. Gish, S. Ybarra, D. Mack, A.J. Levine, Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays, Proceedings of the National Academy of Sciences of the United States of America 96 (12) (1999) 6745–6750.
[43] T.R. Golub, D.K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J.P. Mesirov, H. Coller, M.L. Loh, J.R. Downing, M.A. Caligiuri, C.D. Bloomfield, E.S. Lander, Molecular classification of cancer: class discovery and class prediction by gene expression monitoring, Science 286 (5439) (1999) 527–531.
[44] A. Bhattacharjee, W. Richards, J. Staunton, C. Li, S. Monti, P. Vasa, C. Ladd, J. Beheshti, R. Bueno, M. Gillette, M. Loda, G. Weber, E. Mark, E. Lander, W. Wong, B. Johnson, T. Golub, D. Sugarbaker, M. Meyerson, Classification of human lung carcinomas by mrna expression profiling reveals distinct adenocarcinoma subclasses., Proceedings of the National Academy of Sciences of the United States of America 98 (24) (2001) 13790–13795.
[45] S. Ramaswamy, P. Tamayo, R. Rifkin, S. Mukherjee, C. Yeang, M. Angelo, C. Ladd, M. Reich, E. Latulippe, J.P. Mesirov, T. Poggio, W. Gerald, M. Loda, E.S. Lander, T.R. Golub, Multiclass cancer diagnosis using tumor gene expression signatures, Proceedings of the National Academy of Sciences of the United States of America 98 (26) (2001) 15149–15154.
[46] W. Chu, Z. Ghahramani, F. Falciani, D.L. Wild, Biomarker discovery in microarray gene expression data with Gaussian processes, Bioinformatics 21 (16) (2005) 3385–3393.
[47] R. Maglietta, A. D'Addabbo, A. Piepoli, F. Perri, S. Liuni, G. Pesole, N. Ancona, Selection of relevant genes in cancer diagnosis based on their prediction accuracy, Artificial Intelligence in Medicine 40 (1) (2007) 29–44.
[48] C. Furlanello, M. Serafini, S. Merler, G. Jurman, Entropy-Based Gene Ranking without Selection Bias for the Predictive Classification of Microarray Data, BMC Bioinformatics (4) (2003) 54.
[49] A. Ben-Dor, L. Bruhn, A. Laboratories, N. Friedman, M. Schummer, I. Nachman, U. Washington, U. Washington, Z. Yakhini, Tissue classification with gene expression profiles, Journal of Computational Biology 7 (2000) 559–584.
[50] R.J.S. Hu, Statistical redundancy testing for improved gene selection in cancer classification using microarray data, Cancer Informatics 3 (2007) 29–41.
[51] M.E. Gerritsen, R. Soriano, S. Yang, G. Ingle, C. Zlot, K. Toy, J. Winer, A. Draksharapu, F. Peale, T.D. Wu, P.M. Williams, In silico data filtering to identify new angiogenesis targets from a large in vitro gene profiling data set, Physiological Genomics 10 (1) (2002) 13–20.
[52] P. Mahata, K. Mahata, Selecting differentially expressed genes using minimum probability of classification error, Journal of Biomedical Informatics 40 (6) (2007) 775–786.
[53] T. Kawamura, H. Takahashi, H. Honda, Proposal of new gene filtering method, bagpart, for gene expression analysis with small sample, Journal of Bioscience and Bioengineering 105 (1) (2008) 81–84.
[54] T. Hellem, I. Jonassen, New feature subset selection procedures for classification of expression profiles, Genome Biology 3 (4) (2002), 0017.1–0017.11.