



Evolutionary q -Gaussian Radial Basis Function Neural Network to determine the microbial growth/no growth interface of *Staphylococcus aureus*

Francisco Fernández-Navarro^{a,*}, César Hervás-Martínez^a, M. Cruz-Ramírez^a, Pedro Antonio Gutiérrez^a, Antonio Valero^b

^a Department of Computer Science and Numerical Analysis, University of Córdoba, Campus de Rabanales, Albert Einstein Building, 3rd Floor, 14071 Córdoba, Spain

^b Department of Food Science and Technology, University of Córdoba, Campus de Rabanales, Darwin Building, 14014 Córdoba, Spain

ARTICLE INFO

Article history:

Received 10 June 2010

Received in revised form 7 October 2010

Accepted 23 November 2010

Available online 9 December 2010

Keywords:

Imbalanced datasets

Synthetic Minority Over-Sampling

Technique (SMOTE)

q -Gaussian Radial Basis Function Neural Network

Predictive Microbiology

Memetic Algorithm

ABSTRACT

In this paper, q -Gaussian Radial Basis Functions are presented as an alternative to Gaussian Radial Basis Function. This model is based on q -Gaussian distribution, which parametrizes the Gaussian distribution by adding a new parameter q . The q -Gaussian Radial Basis Function allows different Radial Basis Functions to be represented by updating the new parameter q . For example, when the q -Gaussian function takes a value of $q \rightarrow 1$, it represents the standard Gaussian Radial Basis Function. The model parameters are optimized through a Memetic Algorithm that evolves both its structure and connections. To evaluate the effectiveness of the model, it is tested with a real problem of predictive microbiology. The problem consists of determining the growth boundaries of *Staphylococcus aureus*, a food borne pathogen responsible for several outbreaks. The data from the study of [1] belongs to growth/no growth conditions of *S. aureus* whose temperature, pH and water activity (a_w) has been divided into three categorical classes: growth (G), growth transition (GT) and no growth (NG). Due to the imbalanced nature of the problem, it has been necessary to apply an over-sampling algorithm. The over-sampling procedure selected was the Synthetic Minority Over-Sampling Technique (SMOTE) algorithm. This algorithm has been applied to the patterns in the minority class in order for the performance of the classifier in this class to be acceptable (the minority class in this problem is of vital interest).

© 2010 Elsevier B.V. All rights reserved.

1. Introduction

The use of Artificial Neural Networks (ANNs) as an alternative to other techniques in predictive microbiology has been significant due to their flexibility and high degree of accuracy in fitting to experimental data, all of which has been the object of several research studies [2,3]. Our study focuses on Radial Basis Function Neural Networks (RBFNNs), which have been successfully employed in a variety of pattern recognition problems such as the determination of the microbial growth/no growth interface [4]. Several common types of functions are used as transfer functions, for example, the standard Gaussian (SRBF), the Multiquadratic (MRBF), the Inverse Multiquadratic (IMRBF), and the Cauchy (CRBF).

This paper evaluates a novel RBF based on q -Gaussian distribution which parametrizes standard normal distribution by replacing

the exponential expressions with q -exponential expressions [5], and maximizing Tsallis entropy [6] under certain constraints [7]. q -Gaussian distributions are applicable to a variety of complex signals and systems, and have been applied in a broad range of fields [8] especially including thermodynamics, biology, economics, and quantum mechanics.

Vignat and Plastino [9] show that if the input data exhibit elliptical symmetry and the input data are normalized, the ensuing normalized input will always be a q -Gaussian probability law. The class of elliptically distributed random vectors plays an important role in statistics, and recently garnered a lot of attention in financial mathematics for being especially useful in risk management [8,10] because the normalization processing of data induces a change in its distribution, from a Gaussian to a q -Gaussian one.

This novel basis function incorporates a real q parameter (besides the centers and widths of the RBF) which can relax or contract the shape of the kernel. This basis function matches both the shape of the kernel and the distribution of the distances better, since the modification of the q parameter allows the representation of different basis functions, among others, Cauchy RBF (CRBF), the standard Gaussian RBF (SRBF), and the Inverse Multiquadratic

* Corresponding author at: Department of Computer Science and Numerical Analysis, University of Córdoba, 14014 Córdoba, Spain. Tel.: +34 957 21 83 49; fax: +34 957 21 83 60.

E-mail address: i22fenaf@uco.es (F. Fernández-Navarro).

RBF (IMRBF) functions. A Memetic Evolutionary Algorithm based on heuristics is employed to select the parameters of the q -Gaussian RBF model. Neural Network training using evolutionary algorithms is justified due to evolutionary algorithms have demonstrated their capability in designing a near optimum neural network architecture and simultaneously optimizing the corresponding weights [11], with several theoretical proposals [12,13] and practical applications [14–16].

Multi-classification patterns can often be applied to solve real problems when the output response is subjected to high inherent variability. In the case of food microbiology, predictive growth/no-growth models can estimate microbial behavior when environmental factors are confined to a limited range. Within this zone, the output result can lead us to an estimation of growth or no-growth. The general way to solve this problem is by performing two-class classification models; thus, theoretical studies have been focused almost entirely on learning binary functions. However, the result achieved by dividing the problem into two classes (growth and no-growth) does not correspond to reality given that, under the same conditions, simultaneous responses can be obtained. This reasoning supports the idea of creating a new class, called growth transition, which encompasses all the environmental conditions where growth and no-growth responses can occur. A multi-classification model would allow 100% accuracy in distinguishing between growth or no-growth conditions, and those where growth is observed a certain percentage of times. The problem for most of these algorithms is that the extension from two-class to the multi-class pattern classification problem is non-trivial, and often leads to unexpected complexity or weaker performances [17].

In this paper, the performance of the proposed methodology was evaluated in a real problem based on the study of [1]. This study was modified by providing a categorical classification of *Staphylococcus aureus* growth as a function of temperature, pH and water activity (a_w) in three different classes: “growth” (p_G), which included conditions in which the probability of growth was equal to 1; “no growth” which were conditions where the probability of growth was 0, and a class denominated “growth transition (GT)” that encompassed all conditions where the probability of growth was other than 0 and 1. In our approach, the output of the model was the probability of pertaining to one class instead of quantifying the probability of growth.

The main advantage of this approach in comparison to the logistic model published by Valero et al. [1], lies in the convergence of polynomial predictions to 0 and/or 1 in several cases. This fact was especially evident in the boundary conditions studied (i.e. conditions which led to a binary response of the microorganism). In these cases, the use of the methodology proposed can provide more accurate predictions and also give additional information regarding the variability of microbial responses under limiting conditions. This approach can help predictive modelers to better define the growth boundaries of microorganisms and to model the microbial variability associated with these conditions.

Due to the imbalanced nature of the problem (the *GT* class is clearly the minority class), it seems natural to increase the number of replicates per condition tested. Therefore, in the preprocessing stage, the minority class (the *GT* class) was doubled in order to improve classifier performance in this class [18].

This paper is organized as follows: the q -Gaussian Radial Basis Function Neural Networks are presented in Section 2; Section 3 describes the base classifier, the learning algorithm and the over-sampling procedure; Section 4 explains the experiment carried out; there is a discussion about the best model obtained by the learning algorithm and the results in Section 5; and finally, Section 6 summarizes the conclusions of our work.

2. q -Gaussian Radial Basis Function Neural Networks

We focus on RBFNNs which have been successfully employed in different pattern recognition problems in the last several years [19,20]. One advantage of RBFNNs when compared with MLPs is that the linearly weighted structure of RBFNNs, where parameters in the units of the hidden layer can often be pre-fixed, can be quickly and easily trained without involving nonlinear optimization. Another advantage of RBFNNs, compared to other basis function networks, is that each basis function in the hidden units is a nonlinear mapping that maps a multivariable input to a scalar value, and thus the total number of candidate basis functions involved in an RBFNN model is not very large and does not increase as the number of input variables increases.

Let the number of nodes in the input layer, in the hidden layer and in the output layer be K , M and J , respectively. For any sample $\mathbf{x} = [x_1, x_2, \dots, x_K]$, the output of the RBFNN is $\mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_J(\mathbf{x})]$. The model of an RBFNN can be described with the following equation:

$$f_j(\mathbf{x}) = \beta_{0j} + \sum_{i=1}^M \beta_{ij} \cdot \phi_i(d_i(\mathbf{x})), \quad j = 1, 2, \dots, J \quad (1)$$

where $\phi_i(d_i(\mathbf{x}))$ is a non-linear mapping from the input layer to the hidden layer, $\beta_j = [\beta_{1j}, \beta_{2j}, \dots, \beta_{Mj}]$, for $j = 1, 2, \dots, J$ is the connection weight between the hidden layer and the output layer, and β_{0j} is the bias value for the class j . The function $d_i(\mathbf{x})$ can be defined as:

$$d_i(\mathbf{x}) = \frac{\|\mathbf{x} - \mathbf{c}_i\|^2}{r_i^2} \quad (2)$$

where r_i is the scalar parameter that defines the width for the i th radial unit, $\|\cdot\|$ represents the Euclidean norm and $\mathbf{c}_i = [c_{i1}, c_{i2}, \dots, c_{iK}]$ are the centers of the RBFs. The standard RBF (SRBF) is the Gaussian function, which is given by:

$$\phi_i(d_i(\mathbf{x})) = e^{-d_i(\mathbf{x})} \quad (3)$$

SRBF present a very selective response, with high activation for patterns close to the centroid and very small activation for distant patterns. The RBFs $\phi_i(d_i(\mathbf{x}))$ can take different forms, including the Cauchy RBF (CRBF) defined by:

$$\phi_i(d_i(\mathbf{x})) = \frac{1}{1 + d_i(\mathbf{x})} \quad (4)$$

and the Inverse Multiquadratic RBF (IMRBF), given by:

$$\phi_i(d_i(\mathbf{x})) = \frac{1}{(1 + d_i(\mathbf{x}))^{1/2}} \quad (5)$$

The CRBF and the IMRBF have longer tails than the SRBF, i.e., their activations for patterns far from the centroid of the RBF is greater than the activation of the SRBF for those patterns. The CRBF has been successfully applied to image retrieval [21] and Computerized Tomography [22], whereas the IMRBF has been used in applications related to real-time signal-processing [23], among other scientific and engineering applications. As can be seen in Fig. 1a, for sufficiently large distance norms, the decay of the IMRBF and CRBF is very slow. In addition, the SRBF, CRBF and IMRBF functions do not fall asymptotically to zero.

This paper researches the use of the q -Gaussian RBF for multi classification problems because this family of functions considers, as already discussed in this section, different types of local functions, where the tail of the different functions plays a crucial role. The q -Gaussian RBF for the i th RBF can be defined as:

$$\phi_i(d_i(\mathbf{x})) = e_{q_i}^{-d_i(\mathbf{x})} \quad (6)$$

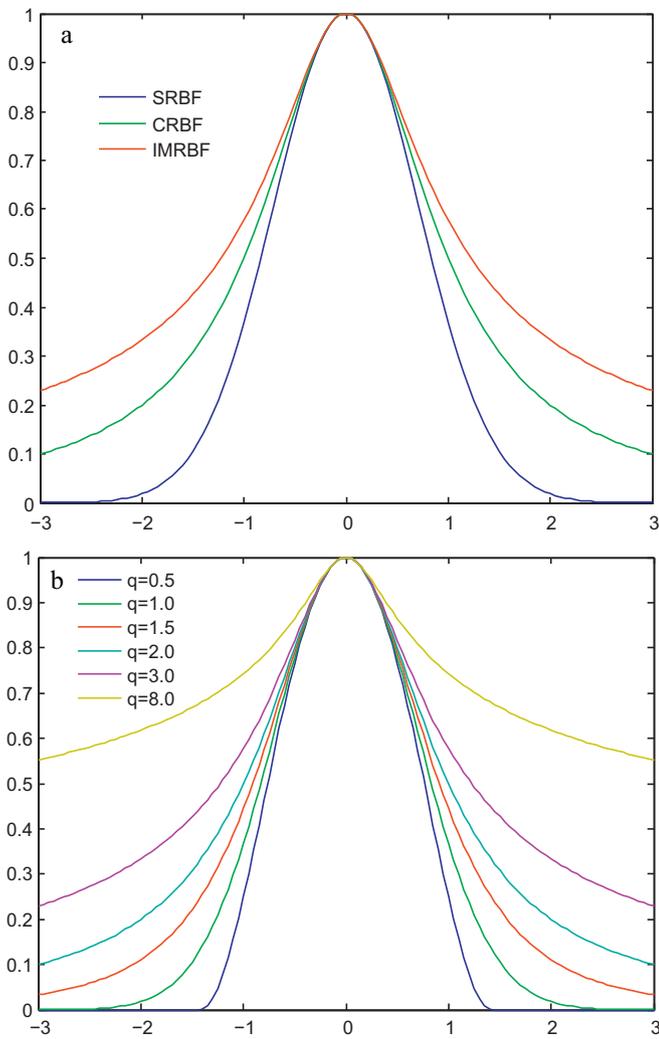


Fig. 1. Radial unit activation in one-dimensional space with $c=0$ and $r=1$ for different RBFs: (a) Gaussian, Cauchy and Inverse Multiquadratic and (b) q -Gaussian with different values of q .

where q_i is a real valued parameter and the q -exponential function of $-d_i(\mathbf{x})$ is given by:

$$\phi_i(d_i(\mathbf{x})) = \begin{cases} (1 - (1 - q)d_i(\mathbf{x}))^{1/(1-q)} & \text{if } (1 - (1 - q)d_i(\mathbf{x})) \geq 0; \\ 0 & \text{Otherwise} \end{cases} \quad (7)$$

The q -Gaussian RBF can reproduce different RBFs for different values of the real q . parameter As an example, when the q parameter is close to 2, the q -Gaussian is the CRBF, for $q=3$ we have the activation of a radial unit with an IMRBF for $d_i(\mathbf{x})$ is equal to the activation of a radial unit with a q -Gaussian RBF for $d_i(\mathbf{x})/2$ and, finally, when the value of q converges to 1, the q -Gaussian converges to the Gaussian function (SRBF). Fig. 1b presents the radial unit activation for the q -Gaussian RBF for different values of q .

3. Classification method

3.1. Probabilistic q -Gaussian RBFNN

In a classification problem, measurements $x_i, i=1, 2, \dots, K$, 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 $D = \{(\mathbf{x}_n, \mathbf{y}_n); n=1, 2, \dots, N\}$ is available, where $\mathbf{x}_n = (x_{1n}, \dots, x_{Kn})$ is the random vector of measurements taking

values in $\Omega \subset \mathbb{R}^K$, and \mathbf{y}_n is the class level of the n th individual, where the common technique of representing class levels using a “1-of- J ” encoding vector is adopted, $\mathbf{y} = (y^{(1)}, y^{(2)}, \dots, y^{(J)})$, 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 $I(\cdot)$ is the zero-one lost function. A good classifier tries to achieve the highest possible C in a given problem.

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

$$g_l(\mathbf{x}, \boldsymbol{\theta}_l) = \frac{\exp f_l(\mathbf{x}, \boldsymbol{\theta}_l)}{\sum_{j=1}^J \exp f_j(\mathbf{x}, \boldsymbol{\theta}_j)}, \quad l = 1, 2, \dots, J \quad (8)$$

where J is the number of classes in the problem, $f_j(\mathbf{x}, \boldsymbol{\theta}_l)$ is the output of the j th 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 $f_l(\mathbf{x}, \boldsymbol{\theta}_l)$ was defined in Eq. (1).

Using the softmax activation function presented in Eq. (8), the class predicted by the NN corresponds to the node in the output layer whose output value is the greatest. In this way, the optimum classification rule $C(\mathbf{x})$ is the following:

$$C(\mathbf{x}) = \hat{l}, \quad \text{where } \hat{l} = \operatorname{argmax}_l g_l(\mathbf{x}, \boldsymbol{\theta}_l), \quad \text{for } l = 1, 2, \dots, J \quad (9)$$

The function used to evaluate a q -Gaussian RBFNN is the function of cross-entropy error and it is given by the following expression:

$$l(\boldsymbol{\theta}) = -\frac{1}{N} \sum_{n=1}^N \sum_{l=1}^J y_n^{(l)} \log g_l(\mathbf{x}, \boldsymbol{\theta}_l) = \frac{1}{N} \sum_{n=1}^N \left[-\sum_{l=1}^J y_n^{(l)} f_l(\mathbf{x}_n, \boldsymbol{\theta}_l) + \log \sum_{l=1}^J \exp f_l(\mathbf{x}_n, \boldsymbol{\theta}_l) \right] \quad (10)$$

where $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_J)$. The proposed algorithm returns the best cross-entropy individuals as feasible solutions. Finally, because of the normalization condition:

$$\sum_{l=1}^J g_l(\mathbf{x}, \boldsymbol{\theta}_l) = 1 \quad (11)$$

and the probability for one of the classes does not need to be estimated. For that reason, the q -Gaussian RBFNN models evaluated have $J-1$ outputs nodes instead of J output nodes since the $g_j(\mathbf{x}, \boldsymbol{\theta}_j) = 1 - \sum_{i=0}^{j-1} g_i(\mathbf{x}, \boldsymbol{\theta}_i)$. This was used to reduce the number of output nodes in the q -Gaussian RBFNN and, consequently, the complexity of the model.

The error surface associated with the model is very convolved 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 an evolutionary algorithm. This kind of metaheuristics has been proven to be very effective when optimizing neural network models [24–26].

3.2. Memetic Algorithm for q -Gaussian Radial Basis Function (MQRBF)

The basic framework of the Evolutionary Algorithm (EA) is the following: the search begins with an initial population of q -Gaussian RBFNNs and, in each of the following iterations, a population-update algorithm is applied which evolves both its

MQRBF Algorithm:**Require:** Training dataset (D)**Ensure:** Best optimized q -Gaussian RBFNN (p^B)

```

1:  $P^I \leftarrow \{p_1^I, \dots, p_{5000}^I\}$   $\{p_i^I$  is a randomly generated RBFNN $\}$ 
2:  $\forall p_i^I \in P^I, f_i^I \leftarrow A(p_i^I)$   $\{\text{Evaluate fitness}\}$ 
3:  $P \leftarrow \{p_{(1)}, \dots, p_{(5000)}\}, (p_{(i)} \prec p_{(j)}) \iff (f_i^I > f_j^I)$   $\{\text{Sort individuals in } P^I \text{ by increasing } f_i^I\}$ 
4:  $P \leftarrow \{p_{(1)}, \dots, p_{(500)}\}$   $\{\text{Retain the best 500 RBFNNs}\}$ 
5:  $\forall p_{(i)} \in P, p_{(i)} \leftarrow k\text{-means}(p_{(i)})$   $\{\text{Improve individuals' centres}\}$ 
6: while not Stop Condition do
7:    $\forall p_i \in P, f_i \leftarrow A(p_i)$   $\{\text{Evaluate fitness}\}$ 
8:    $P \leftarrow \{p_{(1)}, \dots, p_{(500)}\}, (p_{(i)} \prec p_{(j)}) \iff (f_i > f_j)$   $\{\text{Sort individuals in } P \text{ by increasing } f_i\}$ 
9:    $p^B \leftarrow p_{(1)}$   $\{\text{Store Best Individual}\}$ 
10:   $P^P \leftarrow \{p_{(1)}, \dots, p_{(50)}\}$   $\{\text{Parametric mutation parents (best 10\% of individuals)}\}$ 
11:   $P^S \leftarrow \{p_{(1)}, \dots, p_{(449)}\}$   $\{\text{Structural mutation parents (best 90\% of individuals minus one)}\}$ 
12:   $P^R \leftarrow \{p_{(1)}, \dots, p_{(50)}\}$   $\{\text{Recombination parents (best 10\% of individuals)}\}$ 
13:   $\forall p_{(i)}^P \in P^P, p_{(i)}^P \leftarrow \text{parametricMutation}(p_{(i)}^P)$   $\{\text{Apply parametric mutation}\}$ 
14:   $\forall p_{(i)}^S \in P^S, p_{(i)}^S \leftarrow \text{structuralMutation}(p_{(i)}^S)$   $\{\text{Apply structural mutation}\}$ 
15:   $\forall p_{(i)}^R \in P^R, p_{(i)}^R \leftarrow \text{recombination}(p_{(i)}^R)$   $\{\text{Apply recombination}\}$ 
16:   $P \leftarrow P^P \cup P^S \cup P^R \cup \{p^B\}$   $\{\text{Offspring including the elite}\}$ 
17:  if isOptimizationEpoch then
18:     $P^{LS} \leftarrow \text{sensitivityClustering}(P)$   $\{\text{Apply sensitivity clustering and select closest individuals to centroids}\}$ 
19:     $P \leftarrow (P - P^{LS})$ 
20:     $\forall p_{(i)}^{LS} \in P^{LS}, p_{(i)}^{LS} \leftarrow \text{localSearch}(p_{(i)}^{LS})$   $\{\text{Apply } iRprop+ \text{ local search}\}$ 
21:     $P \leftarrow P \cup P^{LS}$   $\{\text{Include optimized individuals in the population}\}$ 
22:  end if
23: end while
24:  $\forall p_i \in P, f_i \leftarrow A(p_i)$   $\{\text{Evaluate fitness}\}$ 
25:  $P \leftarrow \{p_{(1)}, \dots, p_{(500)}\}, (p_{(i)} \prec p_{(j)}) \iff (f_i > f_j)$   $\{\text{Sort individuals in } P \text{ by increasing } f_i\}$ 
26:  $p^B \leftarrow p_{(1)}$ 
27: return  $p^B$ 

```

Fig. 2. MQRBF training algorithm framework.

structure and weights. The population is subject to the operations of replication, mutation and recombination.

The Memetic Algorithm for q -Gaussian Radial Basis Function (MQRBF) is detailed in Fig. 2, where p^B is the best optimized q -Gaussian RBFNN returned by the algorithm. $l(\theta)$ defined in (Eq. (10)) was considered as the error function of an individual g of the population. The fitness measure needed for evaluating the individuals is a strictly decreasing transformation of the error function $l(\theta)$ given by:

$$A(g) = \frac{1}{1 + l(\theta)}; \quad 0 < A(g) \leq 1 \quad (12)$$

The crossover operators considered are the binary and multipoint crossover operators. The severity of a mutation in an individual RBFNN model is dictated by the temperature $T(g)$ of the RBFNN model. $T(g)$ is related to $A(g)$ by means of the expression $T(g) = 1 - A(g)$, $0 \leq T(g) < 1$ and, for that reason, $T(g)$ is in decline throughout the evolutionary process, experiencing abrupt changes at the beginning (exploration) and slight changes at the end (exploitation). It is supposed that the $A(g)$ of the individuals in the

population must improve with each iteration of the evolutionary process.

Parametric mutation consists of a simulated annealing algorithm [27]. 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 proposed genetic operators can be seen in [28,29].

With regard to the mutation of the q parameter: if the structural mutator adds a new node in the q -Gaussian RBFNN, the q parameter is assigned to a γ value, where $\gamma \in [0.75, 1.25]$, because when $q \rightarrow 1$, the q -Gaussian RBF reproduces the SRBF. The q parameter is updated by adding a uniform ε value, where $\varepsilon \in [-0.25, 0.25]$, because the modification of the q -Gaussian RBFNN is very sensitive to q variation (as can be seen in Fig. 1b).

This Memetic Algorithm (MA) includes an optimization clustering process applied during specific stages of the evolutionary

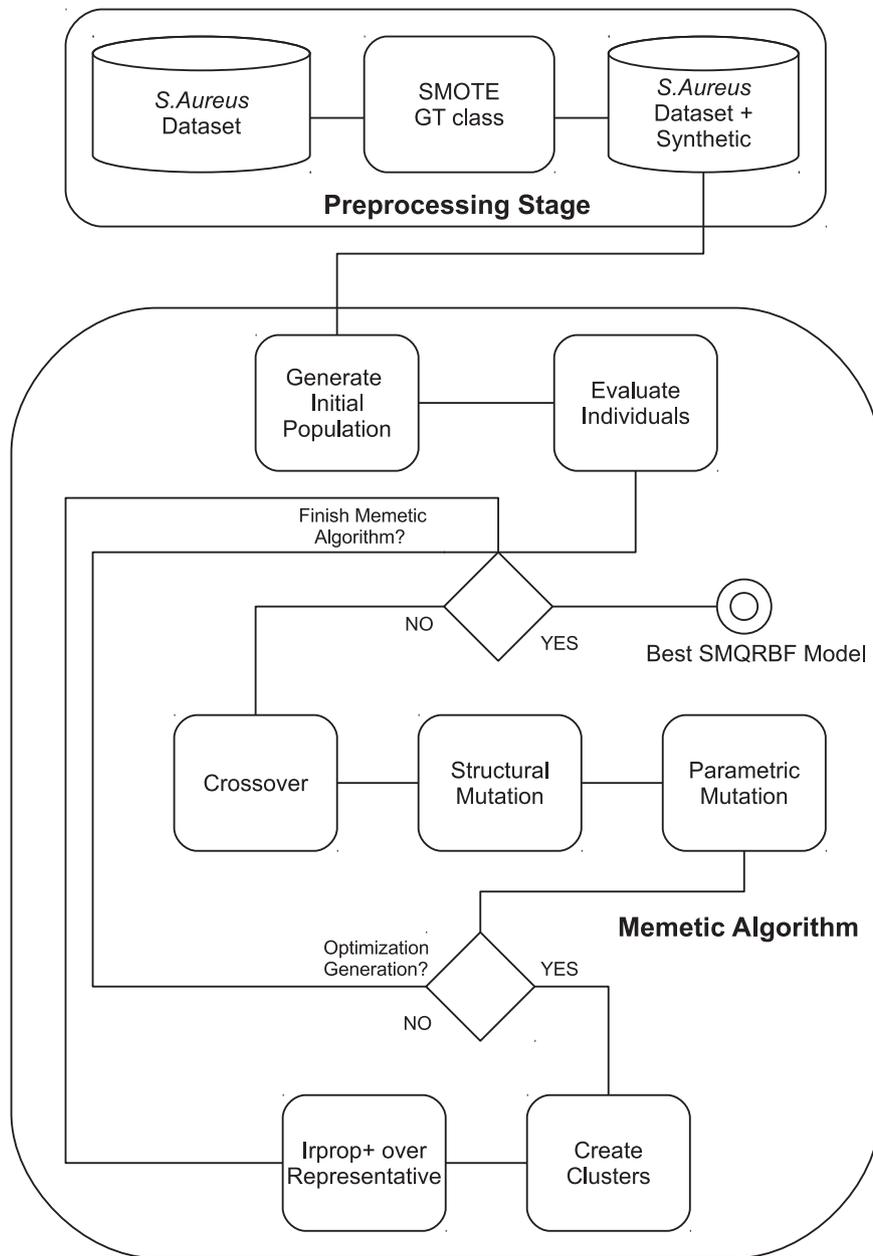


Fig. 3. Flow diagram of the SMQRBF method.

process. In this clustering process, each RBFNN model or individual is represented by the set of its accuracies per class. The clustering algorithm is able to obtain groups of individuals exhibiting similar behavior in different classes.

After that, *irprop+* algorithm [30] is applied to the individual closest to the centroid obtained in each cluster. It is important to note that each cluster has been determined by means of the standard *k*-means applied to the specific space previously mentioned. Finally, the optimized individuals are returned to the population with their fitness and values updated.

Finally, the Smote Memetic Algorithm for the *q*-Gaussian Radial Basis Function (SMQRBF) applies an over-sampling procedure to the minority class patterns (*GT* class) in the preprocessing stage, which is shown in detail in Fig. 3.

Synthetic examples were obtained by applying the Synthetic Minority Oversampling Technique (SMOTE) algorithm [31]. To determine the number of synthetic patterns that the SMOTE algorithm should generate, an experimental study was carried out by

multiplying the number of minority class patterns (*GT* class) by 1.5 and 2. We conclude that by doubling the number of *GT* patterns, the accuracy for the *GT* class is improved without drastically decreasing overall accuracy in the remaining classes. Therefore, the number of minority class patterns (*GT* patterns) was doubled.

The aim was to decrease the problem of an imbalanced rate by selecting the *GT* class to apply the re-sampling procedure to, since this class originally included half the number of patterns of the other classes (*G* and *NG*).

4. Experiments

4.1. Database description

Datasets for growth/no growth models in the predictive microbiology field normally follow a fractional factorial design, i.e. the selection of those conditions where are closed to the microbial interface. This requires a previous testing for calculating the envi-

ronmental factor range, where growth and no growth can occur. Factors usually selected for these purposes are temperature, pH and water activity, since they mostly influence on growth kinetics.

The original dataset was taken from [1] describing the growth/no growth boundaries of *S. aureus* as a function of temperature (T), pH and water activity (a_w) by an ordinary logistic regression model. Data were collected at 8, 10, 13, 16 and 19 °C at pH levels from 4.5 to 7.5 (0.5 intervals) and at 19 levels of a_w (from 0.856 to 0.999 at regular intervals). In this paper, the conditions in which *S. aureus* always grows have been labeled as Growth (G), those in which it never grows as No Growth (NG), and finally, those for which a binary response of the microorganism was observed (it grows between 1 and 29 times of the 30 replicates tested per condition), as Growth Transition (GT). For data processing, 146 out of 287 conditions performed, were selected for model training and 141 were chosen for model generalization. From the 146 conditions selected to train the model, 60 conditions were classified as G, 29 as GT and 57 as NG. For the conditions used to validate the performance of the model (141 conditions), 57 were classified as G, 28 as GT, and 56 as NG. More details can be found in [1].

4.2. Algorithms used for comparison purposes

The proposed method was compared to the following algorithms:

- The MQRBF method (detailed in Section 3.2). As our SQRBF approach applies an oversampling procedure in the preprocessing stage, it is necessary to compare its performance to the original MQRBF method.
- Multi-logistic regression methods [32]. Within the context of predictive microbiology, in most cases, models that describe the behavior of the pathogen are often logistic regression models [2,1]. Hence, we select two of the most popular logistic regression algorithms in order to compare the performance obtained by logistic regression models and the performance achieved by our proposed methodology:
 - MultiLogistic (MLogistic): It is an algorithm for building a multinomial logistic regression model with a ridge estimator to guard against overfitting by penalizing large coefficients, based on the work by le Cessie and van Houwelingen [33]. In order to find the coefficient vector, a Quasi-Newton Method is used.
 - SimpleLogistic (SLogistic): It is based on applying LogitBoost algorithm with simple regression functions and determining the optimum number of iterations by a five fold cross-validation. The data is equally splitted five times into training and test, LogitBoost is run on every training setup to a maximum number of iterations (500) and the classification error on the respective test set is logged. Afterwards, LogitBoost is run again on all data using the number of iterations that gave the smallest error on the test set averaged over the five folds. Further details about the algorithm can be found in [34].

We consider the pH, water activity (a_w) and temperature (T) as the initial co-variables (MLogistic(standard model) and SLogistic(standard model)). The model can be expressed as:

$$Y = b_0 + b_1 \cdot T + b_2 \cdot \text{pH} + b_3 \cdot a_w \quad (13)$$

where Y is the dependent variable, b_0 the intercept of model, and b_1, b_2, b_3 the partial regression coefficients. In order to allow a fair comparison between the new developed model and existing classic approaches, we applied the SMOTE algorithm in the patterns of the GT class in combination with the standard logistic regression models.

Furthermore, we compare our approach to logistic regression methods with square and cross products terms in the model (as

suggested in [2]): MLogistic ([2] model) and SLogistic ([2] model). This model is expressed as:

$$Y = b_0 + b_1 \cdot T + b_2 \cdot \text{pH} + b_3 \cdot a_w + b_4 \cdot T \cdot \text{pH} + b_5 \cdot T \cdot a_w + b_6 \cdot \text{pH} \cdot a_w + b_7 \cdot T \cdot \text{pH} \cdot a_w \quad (14)$$

As we did with the standard logistic regression models, the SMOTE algorithm was also applied to the pattern of the GT class in combination with the [2] approach.

- A Gaussian RBF Network (RBFN) [35], deriving the centers and width of hidden units using k -means and combining the outputs obtained from the hidden layer using logistic regression. k -means is applied separately to each class to derive k clusters for each class.
- The C-SVM algorithm [36] with RBF kernels (SVM). From a structural point of view, the SVMs are related to RBFNNs and they have become one of the most popular and developed methods nowadays. In order to face the multi-class case, a “1-against-1” approach has been considered, following the recommendations of Hsu and Lin [37].

We also compared our proposal to specific methods for imbalanced data: the OverSampling and SmoteOverSampling methods proposed in [18]. These methods have been selected due to their similarities to the model proposed. They use MLP neural networks as the base classifier, and the model is trained by the RProp algorithm. The main differences with our approach are the following: our model is trained by a MA and we used q -Gaussian RBFNN as the base classifier.

4.3. Experimental design

The parameter values used in the hybrid techniques proposed were the following: a simple linear rescaling of the input variables was performed in the interval $[-2, 2]$, X_i^* being the transformed variables. The connection between the hidden and output layer began in the $[-5, 5]$ interval. The initial value of the radii r_j was obtained in the interval $(0, d_{\max}]$, where d_{\max} is the maximum distance between two training input examples.

The size of the population was $N = 500$. For the structural mutation, the number of nodes that can be added or removed was within the $[1, 2]$ interval, and the number of connections to add or delete in the hidden and the output layer during structural mutations was within the $[1, 7]$ interval. The number of clusters was $k=6$ for the k -means algorithm. The $iRprop$ + local improvement procedure was performed every 50 generations, 8 times during the evolution. In this way, the algorithm stopped when 400 generations were completed. For the $iRprop$ + algorithm, a maximum of 75 cycles were considered.

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

Classifiers were evaluated by two measures derived from the confusion matrix: the Correct Classification Rate (CCR) and Minimum Sensitivity (MS) over the generalization dataset, as proposed by Fernández et al. [38], since this problem is of vital importance to know the overall accuracy of the best model and the accuracy of the most difficult class to classify (which in theory would be the GT class).

The contingency or confusion matrix $M(g)$ for a classification problem with J classes, N training or generalization patterns and g

as classifier is given by the following expression:

$$M = \left\{ n_{ij}; \sum_{i,j=1}^J n_{ij} = N \right\} \quad (15)$$

where n_{ij} represents the number of times the patterns are predicted by classifier g to be in class j when they really belong to class i . The diagonal corresponds to correctly classified patterns and the off-diagonal to mistakes in the classification task.

The CCR measure or accuracy is defined as:

$$CCR = \frac{1}{N} \sum_{j=1}^J n_{jj} \quad (16)$$

that is, the rate of all the correct predictions in the training set (CCR_T) or in the generalization set (CCR_G).

Let us denote the number of patterns associated with class i by $f_i = \sum_{j=1}^J n_{ij}$, $i = 1, \dots, J$. Let $S_i = n_{ii}/f_i$ be the number of patterns correctly predicted to be in class i with respect to the total number of patterns in class i (sensitivity for class i). The MS measure is defined as:

$$MS = \min \{S_i; i = 1, \dots, J\} \quad (17)$$

that is, the accuracy for the class that is the worst classified in the training set (MS_T) or in the generalization set (MS_G).

For the MQRBF, SMQRBF and the specific methods for imbalanced data, the procedures were run 30 times because they are stochastic methods and do not return the same result for each execution. For the other methods, the results were obtained by running them only once because all of them are deterministic methods.

The MQRBF algorithm was implemented in JAVA. For the SMQRBF method, the MQRBF algorithm was slightly modified, applying the oversampling procedure in the preprocessing stage. We also used “libsvm” [39] to obtain the results of the SVM method, WEKA [32] to obtain the results of the RBFN, MLogistic and SLogistic and the CSNN¹ software package to obtain the results of the OverSampling and SmoteOverSampling methods.

5. Results and discussion

5.1. Statistical analysis

A comparison of the SMQRBF method has been carried out with the well known classification techniques given in Section 4.2. Table 1 shows the results obtained with the different techniques tested. The SMQRBF method obtained the best result in terms of MS_G and CCR_G out of all the techniques compared.

To ascertain the statistical significance of the differences between the means (in CCR_G and MS_G for each stochastic methodology: OverSampling, Smote, MQRBF and SMQRBF), the Kolmogorov–Smirnov test (K–S test) was used with the significance level α , equal to 0.05 to evaluate if the CCR_G and MS_G values followed a normal distribution. As can be seen from the results in Table 2, a normal distribution can be assumed because the critical levels, p -values, were over 0.05 in all cases.

In order to determine the best methodology (in the sense of its influence on accuracy and on the Minimum Sensitivity in the generalization set, CCR_G and MS_G), an ANOVA statistical method test was carried out. The results of the ANOVA analysis for the CCR_G and MS_G values show that the effect of the methodology was statistically significant at a level of significance of 5% (see first row of Table 2).

Table 1

Comparison with other statistical and artificial intelligence methods: Correct Classification Rate and Minimum Sensitivity in the generalization set ($CCR_G(\%)$ and $MS_G(\%)$, respectively).

Method	$CCR_G(\%)$	$MS_G(\%)$
MLogistic (standard model)	76.60	39.29
SLogistic (standard model)	76.60	32.14
SMOTE + MLogistic (standard model)	71.63	50.00
SMOTE + SLogistic (standard model)	70.92	50.00
MLogistic ([2] model)	80.56	50.00
SLogistic ([2] model)	75.88	32.14
SMOTE + MLogistic ([2] model)	75.17	53.57
SMOTE + SLogistic ([2] model)	74.46	46.42
RBFN	75.18	39.29
SVM	80.98	42.86
OverSampling	78.58 ± 2.24	52.14 ± 8.11
SmoteOverSampling	75.60 ± 4.03	60.21 ± 13.29
MQRBF	80.31 ± 3.34	54.51 ± 6.59
SMQRBF	82.77 ± 1.90	78.52 ± 1.98

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

Once this test guaranteed that there were significant differences between the results of the different methods, a multiple comparison test was performed on the CCR_G and MS_G values in order to rank the different methods. First, a Levene test [40] was carried out to evaluate the equality of variances. Then, a Tamhane test [41] was performed because the variances are not equal (either for CCR_G or MS_G) in order to rank the different methods. Our aim was to find the methodology whose performance (in CCR_G and MS_G) was significantly better than that of the rest of the methodologies.

Table 2 shows the results obtained by the Tamhane test. On analyzing the average results for accuracy CCR_G , we can observe that the SMQRBF methodology obtained better results than those obtained with other methodologies. On the other hand, the results of the average MS_G show that the SMQRBF methodology obtained a significantly better performance giving a level of signification of 5%, more than the other methodologies. Therefore, SMQRBF is the classification methodology recommended in this paper for the problem analyzed.

Table 1 shows that the application of the SMOTE algorithm in combination with logistic regression techniques is not suitable because the final model was less accurate in correctly classifying generalization data. The application of the SMOTE algorithm in combination with logistic regression models improved the Minimum Sensitivity (MS_G) results but lowered accuracy results (CCR_G).

In general, these results show that the proposed approaches based on q -Gaussian RBFNNs are robust enough to tackle the multi-classification of the growth boundaries of *S. aureus*, and obtain better results than the majority of existing alternative methods.

5.2. Analysis of the best SMQRBF model

Table 3 shows the performance of the best SMQRBF model: Correct Classification Rate (CCR) on the training set considering the synthetic SMOTE data (CCR_{TS}) and not considering the synthetic SMOTE data (CCR_T), CCR on the generalization set (CCR_G), Minimum Sensitivity (MS) on the training set considering the synthetic SMOTE data (MS_{TS}) and not considering the synthetic SMOTE data (MS_T) and MS on the generalization set (MS_G), Confusion Matrix (CM) for the training set considering the synthetic SMOTE data (CM_{TS}), not considering the synthetic SMOTE data (CM_T) and CM for the generalization set (CM_G).

The outputs of this model are the values of probability that a pattern falls within each class: $G(p_G)$, $GT(p_{GT})$ and $NG(p_{NG})$. The

¹ <http://lamda.nju.edu.cn/datacode/CSNN.htm>.

Table 2

Statistical analysis: p -values of the Kolmogorov–Smirnov test for CCR_G and MS_G , p -values of the Snedecor's F ANOVA I test and ordered mean for the statistical multiple comparison Tamhane test.

Test variable	Kolmogorov–Smirnov test			
	OverSampling	SmoteOverSampling	MQRBF	SMQRBF
CCR_G	0.539	0.058	0.455	0.470
MS_G	0.679	0.448	0.671	0.359
	Snedecor's F ANOVA I and Tamhane test			
	CCR_G	MS_G		
F (p -values)	0.000 *	0.000 *		
Ranking of averages	$\mu_{SMQRBF} \geq \mu_{MQRBF} \geq \mu_{OS} > \mu_{SOS}$	$\mu_{SMQRBF} > \mu_{SOS} > \mu_{MQRBF} \geq \mu_{OS}$		

OS: OverSampling; SOS: SmoteOverSampling; $\mu_A \geq \mu_B$: methodology A yields better results than methodology B, but the difference is not significant; $\mu_A > \mu_B$: methodology A yields better results than methodology B with significant differences. The binary relation \geq is not transitive.

* Significant differences were found for $\alpha = 0.05$.

softmax activation function is considered in such a way that each pattern is associated to the class with the highest probability.

It should be highlighted that the robustness of the SMQRBF model is given by the number of replicates tested per condition ($n = 30$) which provided a more reliable classification in three classes. These results are described in the confusion matrices associated (Table 3). The classification accuracy of the SMQRBF model was high since more than 80% of the cases matched those in the classes observed. Regarding the training dataset, 86.30% of the cases were correctly classified (CCR_T), while in the generalization dataset, this percentage was slightly lower (84.39%, CCR_G). Misclassified cases were assigned when the estimated pattern was not associated with the class under observation. As three classes were considered, in this model the errors accounted from the classes G, GT and NG to the adjacent ones; and from the class G to NG and vice versa.

The best model provided by the Memetic Algorithm (Table 3) is composed of six basis functions. The first, q -Gaussian RBF, models an interaction between the three input variables (pH, a_w and T). The second, third and fifth q -Gaussian RBFs represent interactions between T and a_w . The fourth basis function models relations between the pH and a_w . Finally, the sixth basis function is only associated with the pH.

Table 3

Probability expression of the best SMQRBF model.

Best SMQRBF <i>S. aureus</i> multi-classification model		
$p_{NG}(\mathbf{x}, \theta) = \frac{e^{f_1(\mathbf{x}, \theta)}}{1 + \sum_{i=1}^2 e^{f_i(\mathbf{x}, \theta)}}$	$p_{GT}(\mathbf{x}, \theta) = \frac{e^{f_2(\mathbf{x}, \theta)}}{1 + \sum_{i=1}^2 e^{f_i(\mathbf{x}, \theta)}}$	$p_G(\mathbf{x}, \theta) = \frac{e^{f_3(\mathbf{x}, \theta)}}{1 + \sum_{i=1}^2 e^{f_i(\mathbf{x}, \theta)}}$
$f_1(\mathbf{x}, \theta) = -2.73 + 28.65RBF_1 - 24.44RBF_2 - 18.83RBF_3 + 7.38RBF_4 + 4.74RBF_6$ $f_2(\mathbf{x}, \theta) = -0.37 + 15.67RBF_1 - 10.37RBF_2 - 3.51RBF_3 - 569.43RBF_5 - 2.42RBF_6$ $f_3(\mathbf{x}, \theta) = 0$ $RBF_1 = (1 - (1 - 0.19)d_1)^{1/(1-0.19)}$; $RBF_2 = (1 - (1 - 0.78)d_2)^{1/(1-0.78)}$ $RBF_3 = (1 - (1 - 0.89)d_3)^{1/(1-0.89)}$; $RBF_4 = (1 - (1 - 1.1)d_4)^{1/(1-1.1)}$ $RBF_5 = (1 - (1 - 1.1)d_5)^{1/(1-1.1)}$; $RBF_6 = (1 - (1 - 2.23)d_6)^{1/(1-2.23)}$		
$d_1 = \left(\frac{\sqrt{(T^* - 0.93)^2 + (pH^* - 0.77)^2 + (a_w^* - 1.78)^2}}{3.57} \right)^2$; $d_2 = \left(\frac{\sqrt{(T^* + 2.53)^2 + (a_w^* + 0.19)^2}}{1.79} \right)^2$ $d_3 = \left(\frac{\sqrt{(T^* + 0.07)^2 + (a_w^* + 1.28)^2}}{1.98} \right)^2$; $d_4 = \left(\frac{\sqrt{(pH^* - 0.96)^2 + (a_w^* + 0.45)^2}}{1.18} \right)^2$ $d_5 = \left(\frac{\sqrt{(T^* + 0.16)^2 + (a_w^* + 1.09)^2}}{0.24} \right)^2$; $d_6 = \left(\frac{\sqrt{(pH^* + 1.69)^2}}{0.67} \right)^2$		
$T^*, pH^*, a_w^* \in [-2, 2]$; $(1 - (1 - q_i)d_i) \geq 0$		
$CCR_{TS} = 86.28\%$, $CCR_T = 86.30\%$, $CCR_G = 84.39\%$ $MS_{TS} = 82.75\%$, $MS_T = 79.31\%$, $MS_G = 78.57\%$		
$CM_{TS} = \begin{pmatrix} 52 & 6 & 2 \\ 6 & 48 & 4 \\ 0 & 6 & 51 \end{pmatrix}$; $CM_T = \begin{pmatrix} 52 & 6 & 2 \\ 3 & 23 & 3 \\ 0 & 6 & 51 \end{pmatrix}$; $CM_G = \begin{pmatrix} 47 & 9 & 1 \\ 3 & 22 & 3 \\ 1 & 5 & 50 \end{pmatrix}$		

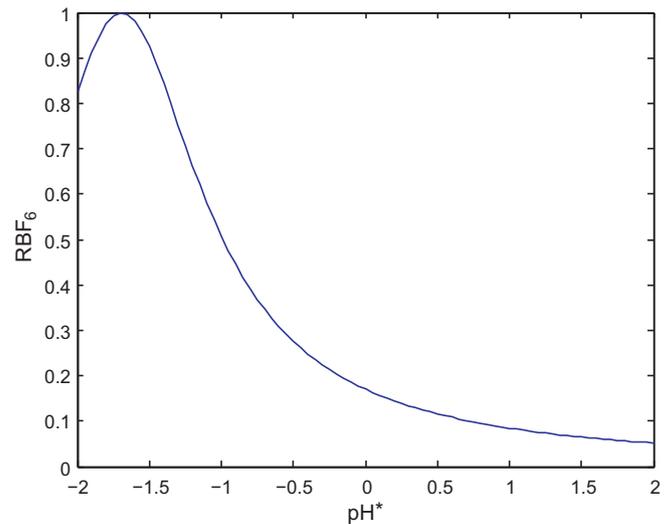


Fig. 4. Radial unit activation for the sixth q -Gaussian RBF of the best SMQRBF model with $c = -1.69$ and $r = 0.67$: X-axis represents the values of pH^* .

As discussed in previous sections, when the value of the q parameter of a q -Gaussian RBF tends to 1, it represents the standard Gaussian RBF. In the best SMQRBF model, four of the six basis functions have values of q very close to one (the basis functions 2, 3, 4 and 5). However, the first and the last basis function have q values other than one. In the case of the first basis function, its q value is close to 0, which means that the response of this basis function is very selective (even greater than the response that a standard Gaussian RBF provides) with high activation for patterns near the center (0.93, 0.74, 1.78) but too small for distant patterns. In contrast, the sixth basis function (Fig. 4), has a q value greater than one and close to two, which means that this basis function represents a Cauchy RBF, where the highest activation value is in extremely small values of pH.

Analyzed from a structural point of view, it could be said that the model generated is an ANN model with a hybrid hidden layer, since it has radial basis nodes of different types. First, four of the six hidden nodes correspond to the standard Gaussian RBF. The last hidden node represents the Cauchy RBF and the first hidden node corresponds to a RBF with a highly selective response because it has a tail even lower than the standard Gaussian RBF.

6. Conclusions

In this paper, we have proposed a new approach to determine optimized parameters for the q -Gaussian RBFNN. The use of q -

Gaussian RBFs made it possible to modify the shape of the RBF by changing the real q parameter and to have radial units with different RBF shapes in the same RBFNN. The q -Gaussian RBFNN proposed used the softmax function and the cross-entropy error function to interpret the output of the q -Gaussian RBFNN from the point of view of probability. The coefficients that minimized the cross-entropy error function were estimated by means of a Memetic Algorithm. The Memetic Algorithm was constructed specifically to take into account the characteristics of this kernel model.

The evaluation of the model and the algorithm for the real problem considered showed that the q -Gaussian RBFNN was the most accurate compared to the rest of the methods. The problem consists of determining the microbial growth/no-growth interface of *S. aureus*. A new class, named growth transition (GT) was included in order to encompass all conditions where the probability of growth was other than 0 and 1. Due to the imbalanced nature of the problem, the Memetic Algorithm was combined with an over-sampling procedure.

This new methodology provided accurate predictions by means of the generalization data regarding values of CCR (82.77%) and MS (78.52%) as previously shown. The existence of a new class, called GT, has been included in the model mainly due to the high number of replicates per condition (30) used in our study, which produced a smoother transition between growth and no-growth zones. This class is clearly justified since microbial responses are more variable in certain zones of the model domain and, therefore, a classification into G or NG cannot be accurate.

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 Andalucía” (Spain). The research of Francisco Fernández-Navarro has been funded by the “Junta de Andalucía” Predoctoral Program, grant reference P08-TIC-3745.

References

- [1] A. Valero, F. Prez-Rodríguez, E. Carrasco, J.M. Fuentes-Alventosa, R.M. García-Gimeno, G. Zurera, Modelling the growth boundaries of *Staphylococcus aureus*: effect of temperature, pH and water activity, *International Journal of Food Microbiology* 133 (1–2) (2009) 186–194.
- [2] Y. Le Marc, C. Pin, J. Baranyi, Methods to determine the growth domain in a multidimensional environmental space, *International Journal of Food Microbiology* 100 (1–3) (2005) 3–12.
- [3] M. Hajmeer, I. Basheer, A probabilistic neural network approach for modelling and classification of bacterial growth/no-growth data, *Journal of Microbiological Methods* (51) (2002) 217–226.
- [4] C.M. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, Oxford, UK, 1996.
- [5] R. Tinos, L. Murta, Selection of radial basis functions via genetic algorithms in pattern recognition problems, in: *Neural Networks, 2008. SBRN'08. 10th Brazilian Symposium on*, 2008, pp. 171–176.
- [6] C. Tsallis, Possible generalization of boltzmann-gibbs statistics, *Journal of Statistical Physics* 52 (1–2) (1988) 479–487.
- [7] C. Tsallis, R.S. Mendes, A.R. Plastino, The role of constraints within generalized nonextensive statistics, *Physica A: Statistical Mechanics and Its Applications* 261 (3–4) (1998) 534–554.
- [8] C. Tsallis, U. Tirnakli, Nonadditive entropy and nonextensive statistical mechanics – some central concepts and recent applications, *Journal of Physics: Conference Series* 201 (1) (2010) 1–15.
- [9] C. Vignat, A. Plastino, Why is the detection of q -Gaussian behavior such a common occurrence? *Physica A: Statistical Mechanics and Its Applications* 388 (5) (2009) 601–608.
- [10] Z. Landsman, A. Tsanakas, Stochastic ordering of bivariate elliptical distributions, *Statistics and Probability Letters* 76 (5) (2006) 488–494.
- [11] X. Yao, Evolving artificial neural networks, *Proceedings of the IEEE* 87 (9) (1999) 1423–1447.
- [12] P.A. Castillo, J.J. Merelo, M.G. Arenas, G. Romero, Comparing evolutionary hybrid systems for design and optimization of multilayer perceptron structure along training parameters, *Information Sciences* 177 (14) (2007) 2884–2905.
- [13] R. Panti, L.N. de Castro, Bio-inspired and gradient-based algorithms to train mlps: the influence of diversity, *Information Sciences* 179 (10) (2009) 1441–1453.
- [14] H. Abbass, An evolutionary artificial neural networks approach for breast cancer diagnosis, *Artificial Intelligence in Medicine* 25 (3) (2002) 265–281.
- [15] N. Roy, R. Ganguli, Filter design using radial basis function neural network and genetic algorithm for improved operational health monitoring, *Applied Soft Computing Journal* 6 (2) (2006) 154–169.
- [16] M. Bauer, O. Buchtala, T. Horeis, R. Kern, B. Sick, R. Wagner, Technical data mining with evolutionary radial basis function classifiers, *Applied Soft Computing Journal* 9 (2) (2009) 765–774.
- [17] E. Gelenbe, K.F. Hussain, Learning in the multiple class random neural network, *IEEE Transactions on Neural Networks* 13 (6) (2002) 1257–1267.
- [18] Z.-H. Zhou, X.-Y. Liu, Training cost-sensitive neural networks with methods addressing the class imbalance problem, *IEEE Transactions on Knowledge and Data Engineering* 18 (1) (2006) 63–77.
- [19] S.N. Qasem, S.M. Shamsuddin, Radial basis function network based on time variant multi-objective particle swarm optimization for medical diseases diagnosis, *Applied Soft Computing Journal Article* 11 (1) (2011) 1427–1438.
- [20] S.S. Panda, D. Chakraborty, S.K. Pal, Flank wear prediction in drilling using back propagation neural network and radial basis function network, *Applied Soft Computing Journal* 8 (2) (2008) 858–871.
- [21] K. Shkurko, X. Qi, A radial basis function and semantic learning space based composite learning approach to image retrieval, in: *Proceedings on ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 1, 2007, pp. 945–948.
- [22] J. Zhang, H. Li, A reconstruction approach to ct with cauchy rbfs network, in: *Lecture Notes in Computer Science*, 2004.
- [23] A. Saranlı, B. Baykal, Complexity reduction in radial basis function (rbf) networks by using radial b-spline functions, *Neurocomputing* 18 (1–3) (1998) 183–194.
- [24] N. Amjadi, F. Keynia, Application of a new hybrid neuro-evolutionary system for day-ahead price forecasting of electricity markets, *Applied Soft Computing Journal* 10 (3) (2010) 784–792.
- [25] S. Shrivastava, M.P. Singh, Performance evaluation of feed-forward neural network with soft computing techniques for hand written english alphabets, *Applied Soft Computing Journal Article* 11 (1) (2011) 1156–1182.
- [26] M.G. Epitropakis, V.P. Plagianakos, M.N. Vrahatis, Hardware-friendly higher-order neural network training using distributed evolutionary algorithms, *Applied Soft Computing Journal* 10 (2) (2010) 398–408.
- [27] 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.
- [28] P.A. Gutiérrez, C. Hervás-Martínez, M. Carbonero, J.C. Fernández, Combined projection and kernel basis functions for classification in evolutionary neural networks, *Neurocomputing* 72 (13–15) (2009) 2731–2742.
- [29] P.A. Gutiérrez, C. Hervás-Martínez, M. Lozano, Designing multilayer perceptrons using a guided saw-tooth evolutionary programming algorithm, *Soft Computing* 14 (6) (2010) 599–613.
- [30] C. Igel, M. Hsken, Empirical evaluation of the improved rprop learning algorithms, *Neurocomputing* 50 (6) (2003) 105–123.
- [31] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, Smote: synthetic minority over-sampling technique, *Journal of Artificial Intelligence Research* (2002) 321–357.
- [32] I.H. Witten, E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, Data Management Systems, 2nd edition, Morgan Kaufmann (Elsevier), 2005.
- [33] S. le Cessie, J. van Houwelingen, Ridge estimators in logistic regression, *Applied Statistics* 41 (1) (1992) 191–201.
- [34] N. Landwehr, M. Hall, E. Frank, Logistic model trees, *Machine Learning* 59 (1–2) (2005) 161–205.
- [35] I.T. Nabney, Efficient training of RBF networks for classification, *International Journal of Neural Systems* 14 (3) (2004) 201–208.
- [36] T. Hastie, R. Tibshirani, J.H. Friedman, *The Elements of Statistical Learning*, Springer, 2001.
- [37] C.-W. Hsu, C.-J. Lin, A comparison of methods for multi-class support vector machines, *IEEE Transaction on Neural Networks* 13 (2) (2002) 415–425.
- [38] J. Fernández, C. Hervás, F. Martínez-Estudillo, P. Gutiérrez, Memetic pareto evolutionary artificial neural networks to determine growth/no-growth in predictive microbiology, *Applied Soft Computing* 11 (1) (2011) 534–550.
- [39] C. Chang, C. Lin, *Libsvm: A Library for Support Vector Machines*, 2001.
- [40] R. Miller, *Beyond ANOVA, Basics of App. Statistics*, Chapman & Hall, London, 1996.
- [41] A.C. Tamhane, D.D. Dunlop, *Statistics and Data Analysis*, Prentice Hall, 2000.