Applied Intelligence 16, 223–234, 2002

Generalized Radial Basis Function Networks Trained with Instance Based Learning for Data Mining of Symbolic Data*

STERGIOS PAPADIMITRIOU, SEFERINA MAVROUDI, LIVIU VLADUTU
AND ANASTASIOS BEZERIANOS*
Department of Medical Physics, School of Medicine, University of Patras, 26500 Patras, Greece
stergios@heart.med.upatras.gr

Abstract. The application of the Radial Basis Function neural networks in domains involving prediction and classification of symbolic data requires a reconsideration and a careful definition of the concept of distance between patterns. This distance in addition to providing information about the proximity of patterns should also obey some mathematical criteria in order to be applicable. Traditional distances are inadequate to access the differences between symbolic patterns. This work proposes the utilization of a statistically extracted distance measure for Generalized Radial Basis Function (GRBF) networks. The main properties of these networks are retained in the new metric space. Especially, their regularization potential can be realized with this type of distance. However, the examples of the training set for applications involving symbolic patterns are not all of the same importance and reliability. Therefore, the construction of effective decision boundaries should consider the numerous exceptions to the general motifs of classification that are frequently encountered in data mining applications. The paper supports that heuristic Instance Based Learning (IBL) training approaches can uncover information within the uneven structure of the training set. This information is exploited for the estimation of an adequate subset of the training patterns serving as RBF centers and for the estimation of effective parameter settings for those centers. The IBL learning steps are applicable to both the traditional and the statistical distance metric spaces and improve significantly the performance in both cases. The obtained results with this two-level learning method are significantly better than the traditional nearest neighbour schemes in many data mining problems.

Keywords: neural network learning, data mining, symbolic data classification, radial basis functions, heuristic learning

1. Introduction

The emergence of neural network technology [1, 2] offers valuable insight to confront complicated data mining problems. Within this framework neural networks can be viewed as advanced mathematical models for recovering the complex dependencies between variables of a physical process from a set of perturbed observations. Actually, in modeling a complex physical process by a neural network we are building a nonlinear model of the physical process that generates the attribute sets and the corresponding outcomes [2–4]. However, the application of neural networks although proven to be very successful in problem domains such as signal processing and pattern recognition has not yielded adequate performances in the domain of data mining of symbolic data. Patterns arising both from commercial databases and from many engineering databases (as those that describe biosequences [5]) involve data defined over spaces that lack the fundamental properties of distance metric spaces. It is essential to be able to redefine these spaces in order to design neural algorithms for these domains.

The contribution of this paper is twofold. First, is the adaptation of a statistical distance metric for

*The implementation of the presented algorithms can be downloaded from http://www.heart.med.upatras.gr/∼stergios.

symbolic features, initially proposed for nearest neighbor schemes [6–8], within the context of the Radial Basis Function (RBF) networks. The geometric properties of this metric render it to be effective within the regularization formulation of the Generalized Radial Basis Function (GRBF) networks. Regularization techniques impose the learning of a smooth functional from the network [2–4, 9]. Hence, it is justifiable to expect from the network to be able of learning the underlying smooth dependence of the outcomes on the attributes, despite the presence of noise that induces perturbation. Although nearest neighbor schemes also exploit similar distance metric types, it is this potential of the GRBF networks to regularize their solution that theoretically justifies their improved performance.

The second contribution of the paper is the introduction of an Instance Based Learning (IBL) step for the estimation of proper RBF parameters in a two level learning process. The examples of the training set are not all of the same importance and reliability. Therefore, a learning mechanism is required for the estimation of the reliability and significance of each example. The RBF network is designed to exploit effectively the irregularity of the problem's state space with the selection of the proper training examples as RBF centers and the determination of parameters for each such RBF center. These parameters account for the significance and reliability of the corresponding example. We describe three different Instance Based Learning (IBL) algorithms for the implementation of this learning step.

The inclusion of irrelevant features inherent in the data results in increased computational demands and degradation in classification accuracy. In order to overcome this problem we endow the distance metric with feature weighting methods. We depict three basic weighting schemes to include domain-specific knowledge and adapt a class distribution weighting (CDW) that allows weights to vary at the class level.

The data mining techniques that the paper present can have a wide span of applications and can be adapted to heterogeneous data records with both symbolic and numeric data attributes. The Statistical Distance Metrics (SDM) adapted from the Modified Value Difference Metric (MVDM) [6, 8] demonstrate a clear performance advantage at the symbolic domain. However, for the numeric attributes the optimal policy for evaluating the distance is not obvious. In many cases, better generalization is obtained by handling the numeric values with a normalized numeric distance type, e.g. with an Euclidean distance metric normalized with the standard deviation. Some attribute domains though, are better handled with a discretized numeric distance metric augmented with an interpolation scheme that alleviates the discretization effects (e.g. the Interpolated Value Difference Metric [8]). Also, more elaborated schemes for treating numerical attributes by finding optimal multisplits [10] and by using a simulated annealing algorithm [11] for the effective discretization of continuous attributes can improve the results obtained by treating numeric attributes as discretized symbolic. These approaches exploit more effectively the information of the training set and therefore they usually improve the statistical distance metric space. Consequently, the performance of the proposed RBF designs that operate within that space is enhanced.

Regularization [3] and the application of Structural Risk Minimization [2] are principal approaches for improving the generalization performance. However, in order to exploit these techniques effectively in complex domains, hybrid learning schemes are required. An approach based on genetic algorithm optimization [12] derives an appropriate regularization parameter λ and a width of the RBF centers. However, this approach restricts the RBF centers to have the same width (i.e. the same spreading of the Gaussian envelopes). This limitation constitutes a major drawback because of the irregular domain that most data mining applications own.

The paper proceeds as follows: Section 2 reviews briefly the Generalized Radial Basis Function Networks (GRBFs). Section 3 defines the proposed Statistical Distance Metric (SDM) that has been proved quite effective. Section 4 fits the statistical distance within the context of GRBF networks. Section 5 presents the heuristic Instance Based Learning (IBL) phase that parses the training set in order to improve the GRBF design (i.e. the selection of centers and their spreads). Section 6 introduces feature weighting methods to account for irrelevant features. Section 7 discusses the results obtained by the application of the new algorithms on some data sets (most of which are standard UCI data sets) and demonstrates that the proposed GRBF approaches generally outperform the nearest neighborhood ones. Finally, the conclusions are presented along with directions for future work.

2. Generalized Radial Basis Function Networks

A standard RBF network has a feedforward structure that consists of two layers, a nonlinear hidden layer and

a linear output layer. The nodes, or basis functions, in the hidden layer operate on the distance between the input vector and the vector that corresponds to the centers of the RBF network. The response of each basis function is a nonlinear function of the distance and is radially symmetric about the center. The outputs are weighted linear combinations of the basis function responses. The standard RBF network formulation uses all the training examples as centers and performs a strict interpolation at the training set. Therefore, its generalization potential is usually poor.

The Generalized Radial Basis Function (GRBF) networks explore Tikhonov's regularization theory for obtaining generalization performance [4], thereby overcoming the main limitation of the standard RBFs. GRBFs attempt a tradeoff between a term that measures the fitness of the solution to the training set and one that evaluates the smoothness of the solution. Denoting by $\mathbf{x}_i$, d_i, $\mathbf{F}(\mathbf{x}_i)$ the input vectors, the desired responses and the corresponding realizations of the network respectively, this tradeoff can be formulated with a cost functional as [1, 3] :

$$C(F) = C_s(F) + \lambda C_r(F)$$

where,

$$C_s(F) = \frac{1}{2}\sum_{i=1}^{N}[d_i - \mathbf{F}(\mathbf{x}_i)]^2 \qquad C_r(F) = \frac{1}{2}\|\mathbf{DF}\|^2 \tag{1}$$

The $C_s(F)$is the standard error term that accounts for the fitting to the training set in the least squares error sense, λ is a positive real number called the *regularization parameter* and $C_r(F)$ is the regularized term that favors the smoothness of the solution. The later term is the most important regarding the generalization performance. The operator D is termed a *stabilizer* because it stabilizes the solution by providing smoothness. In turn, a smooth solution is significantly more robust to erroneous examples of the training set. Nevertheless, the design of adequate generalization performance for GRBF networks remains a difficult and complex issue that involves heuristic criteria. Many techniques for optimizing the performance of GRBF networks applications at diverse fields have been developed [12–15]. These designs however are based on the usual continuous numeric distance metrics (e.g. Euclidean, Manhattan) and consequently they cannot directly access the requirements of the symbolic data. The following section adapts a distance metric originated from nearest neighbour schemes within the context of GRBF networks. The mathematical optimisation that the GRBF learning algorithm performs exploits more cleverly the information provided by each example and outmatches the performance of the nearest neighbour approaches.

3. The Statistical Distance Metric (SDM)

A key problem for data mining applications involving symbolic features is the definition of the distance metric. In domains where features are numeric, it is straightforward to compute the distance between two points in the pattern space in terms of a geometric distance (e.g. Euclidean, Manhatan). Indeed, the traditional RBF learning algorithms have been formulated based on these distance metrics and operate effectively in numeric domains with such distances. However, when the features are symbolic (as usually happens in bioinformatics and in commercial data mining applications) the utilization of the traditional distance metrics yields inadequate performance. Two common approaches for handling symbolic information is the *overlap* method and the *orthogonal* representation [5, 6]. The overlap method simply counts the number of feature values that two instances have in common. This distance metric oversimplifies the pattern space, ignores all the information presented within the training set and yields (as expected) poor performance. The orthogonal representation encodes the symbolic features by binary vectors in such a way that the numerical distance between different features is the same. This method can be accessed as a reformulation of the overlap method for numerical implementation and therefore it suffers from the same inadequacies to extract any information embeded within the training set.

In order to be able to obtain an effective description of the distances between patterns with symbolic feature values we have adapted statistical distance measures within the context of the GRBF, which have a form initially developed for nearest neighbour schemes [6, 8]. The statistical distance measure computes the distance between two values for a feature by accounting for the overall similarity of classification over all the instances of the training set. This method extracts with a statistical approach a separate matrix for each feature from the training set. Each such matrix defines the distances between all possible values of a given feature. The distance measure for a specific feature f is defined

according to the following equation:

$$d_f(V_A, V_B) = \sum_{i=1}^{N_c} \left| \frac{C_{A_i}}{C_A} - \frac{C_{B_i}}{C_B} \right|^k \tag{2}$$

In the equation above, V_A and V_B denote two possible values for the feature f, e.g. for the DNA promoter data, they will be two nucleotides. The distance between the values is the sum over all the N_c classes. For example, for the DNA promoter example (discussed below) there are two classes, either the sequence is a promoter (i.e. a sequence that initiates a process called transcription) or not. The number of the training set patterns of class i with value $V_A(V_B)$ for their feature f, is denoted by $C_{A_i}(C_{B_i})$. Also, the total number of patterns that have value $V_A(V_B)$ for their feature f is denoted by $C_A(C_B)$, and k is a constant usually set to 1.

It becomes easily evident from (2) that the distance between feature values labeled with the same relative frequency, for all possible class labels, is zero. Also, the more correlated are the classifications of patterns pertaining to two feature values the smaller is their statistical distance computed with Eq. (2). Therefore, for feature values corresponding to training set patterns with similar classifications, a small statistical distance will be computed. Equation (2) accounts for the overall similarity of classification of all training instances for each possible class.

The distance $D(X, Y)$ between two patterns X, Y is obtained by a weighted sum of the distances between the values of the individual features of these patterns (obtained from Eq. (2)) and is given by:

$$D(X, Y) = \sum_{i=1}^{F} d\left(V_{X_i}, V_{Y_i}\right)^r \tag{3}$$

where F is the number of features, r is a parameter that controls how distances between individual features scale for the computation of the overall distance between patterns (usually $r = 1$ or 2). V_{X_f} and V_{Y_f} denote the values for the fth feature of X and Y respectively.

Up to now, the SDM is defined for symbolic attributes in a way that resembles the Modified Value Difference Metric (MVDM) that was used at the Parallel Exemplar Based Learning System (PEBLS) [16]. This system discretizes the numeric attributes and treats them consequently as symbolic. However, an important deviation from the PEBLS distance design is, that within the context of the GRBF designs for numeric features we have two major choices to extend the SDM definition:

a) To adopt a numerical distance measure (e.g. Euclidean) along these dimensions, therefore obtaining a distance metric of heterogeneous type [8].
b) To extend the functionality of the SDM to numeric features by discretizing the numerical ranges. A parameter of particular importance of this discretization is the number s of equal width intervals. Although, it is difficult to define general guidelines a heuristic rule of setting s to the larger of 5 or N_c, where N_c is the number of output classes of the problem domain, proves effective in practice.

We should note at this point that the distance metric is further extended in Section 6 to account also for feature weighting.

The Interpolated Value Difference Metric (IVDM) keeps the original numeric value y using it for interpolating between the proper discrete intervals [8]. This metric improves the performance of nearest neighbor schemes because it alleviates the discretization errors. Visibly, it also yields improved generalization results for the GRBF network case.

4. Generalized RBFs with the Statistical Distance Metric (SDM)

This section adapts the distance metric for patterns with symbolic attributes in order to fit a GRBF solution for discovering hidden dependencies in domains involving symbolic attributes. One prerequisite for the application of this statistical distance type, is to have enough training data for the accurate construction of the SDM space. Nevertheless, training sets of size large enough for providing the essential information for generalization, provide as well the necessary information for the computation of an effective distance matrix. Although we do not attempt to establish formal bounds for the sufficiency of the size of the training set, the results obtained from many experiments for which the GRBF with statistical distances produced excellent results, provide strong empirical support for the validity of this statement.

In contrast to examplar based nearest neighbor learning schemes, the GRBFs quest for good generalization ability already in their design. In order to ensure smoothness of their solution they learn a smooth functional that weights the contribution of each examplar subject to the requirements imposed by the regularizing term of (1). This provides some intuition for the superior performance of GRBF networks related to simple

Instance Based Learning (IBL) schemes. Even so, better generalization performance of the GRBF solution than of the nearest neighbourhood schemes, is attained only with a careful design of the GRBFs' parameters.

A parameter of particular importance is the spread parameter, which determines the region of influence of the Gaussian Radial Basis Function kernels. The determination of proper spread parameters complicates within the domain of statistical distances. The heuristic suggestions of [1] to compute the spread σ as $\sigma = d_{\max}/\sqrt{2m}$, where $d_{\max}$ is the maximum distance between patterns and m the number of RBF centers, although effective in numeric domains has not proven to be the same for symbolic ones. One basic reason for the inefficiency of the above formula is that values of different symbolic features can be completely dissimilar one to the other. Therefore, the spreading of the RBF centers along each single feature dimension should be computed by designing the corresponding distance metric for this feature *independently from the other features*. Thus, the proposed design of the GRBF network proceeds by computing different *feature spreading scaling factors* for each feature and different *weights* for the RBF centers.

The feature spreading scaling factors adjust the spread of the Gaussian kernels along the dimension of the corresponding feature. The weight parameter adjusts the region of influence of an RBF center that relates to the importance and the reliability of the example. In order to obtain an effective general setting for the computation of the feature spreading factors, a sensible approach is to obtain at a first step an estimate of the average distance $d_{av,f}$, of patterns within the space defined with the SDM independently for each feature f. The parameter $d_{av,f}$ is then used to compute the feature spreading scaling factor, since it is learned from the peculiarities of the particular feature and normalizes effectively the distances along the dimensions determined by the feature. The region of influence of the RBF kernels is designed by requiring that at a particular distance *Spread* from the RBF center expressed in units of $d_{av,f}$, the attenuation of influence is decreased by a. The meaning of these parameters is illustrated in Fig. 1. In particular, the figure illustrates the envelope of a Gaussian that corresponds to an evaluated parameter $d_{av,f} = 0.5$ and to an attenuation $a = 0.1$ at a distance three times the average distance $d_{av,f}$ (i.e. $Spread = 3$).

We should note, that with this method for the estimation of the spreads, the number of GRBF centers is considered only implicitly, since this number determines to a large extent the average distance parameter. Mathematically, the requirements for the rate of decaying of RBF kernels along each feature dimension can be transformed into the search of a parameter β_f such that:

$$\exp(-\beta_f \cdot d_{av,f} \cdot Spread) = a \tag{4}$$

According to this formula the required parameter β_f is derived as $\beta_f = -\log(a)/(Spread \cdot d_{av,f})$. Values of these parameters bringing up good results are those of the example presented in Fig. 1 i.e. $Spread = 3$ and $a = 0.1$. These parameters imply that at a distance from an RBF center 3 times larger than the average distance between patterns, the influence of the RBF function for the particular feature f attenuates with a factor of 0.1. For these parameters we obtain $\beta_f = -\log(a)/(Spread \cdot d_{av,f}) = 1.5351$, and the Gaussian envelope that is plotted at Fig. 1 corresponds to $\exp(-1.5351 \cdot x)$.

The evaluation of the network response proceeds by computing the response for each feature dimension independently and by summing up the individual responses. The network response for feature f at

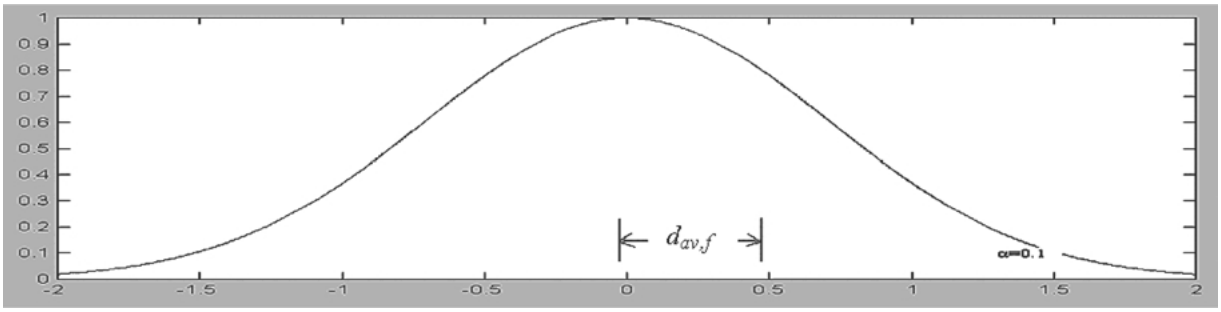

Figure 1. Illustration of the design of the Gaussian RBF spreading along each feature dimension.

a distance x, $RBF_f(x)$, is : $RBF_f(x) = \exp(-\beta_f \cdot x)$. Finally, the total response $RBF(x)$ is obtained by summing: $RBF(x) = \sum_f RBF_f(x)$. This scheme is easily augmented to account for feature weighting as: $RBF(x) = w_f \sum_f RBF_f(x)$.

As we describe in the next section we elaborate further on the design of the RBF network by exploiting the results of the Instance Based Learning (IBL) concerning the significance and reliability of each example. These results are accounted by further defining the paremeter that determines the spreading of RBF centers $\sigma_{f,c}$ as: $\sigma_{f,c} = \frac{\beta_f}{w_c}$, where w_c is the weight of the center c. The designed RBF centers own now an influence at a distance x from their center formulated by $\exp(-\sigma_{f,c} \cdot x)$ for each feature dimension. Clearly, the larger the values of these weight parameters, the smaller the parameter $\sigma_{f,c}$ and therefore the more "extended" is the region of influence of the corresponding Gaussian. In point of fact, this expresses that the example corresponding to the center c is a reliable classifier.

The above scheme with the spreads $\sigma_{f,c}$ dependent both on center c and feature f trains locally the spreads of RBF centers, accounting for the peculiarities and irregularities of the state space. The Instance Based Learning (IBL) step can estimate the relative importance of each RBF center summarized with the parameters $\sigma_{f,c}$ and therefore can improve the performance of the designed RBF solution.

We should note at this point that the value of each symbolic feature is a numeric code that represents the "coordinates" in the constructed statistical distance space. This value is not a real coordinate. In actual fact, it indexes a distance table computed according to the SDM metric. The interpattern distance is then derived with the distance information that this table provides. Therefore, the symbolic feature coordinates must remain unaltered by the RBF training algorithm (e.g. they are not normalized).

5. Instance Based Learning (IBL) Selection of RBF Centers and Determination of Their Parameters

Some examples of the training set are more reliable classifiers than others. It is highly desirable to detect these reliable examples and to exploit them as centers of the RBF network. Furthermore, the more reliable an example is, the larger should be its region of influence when the example is used as an RBF center. The extent of the region of influence is expressed by the spreading parameter σ of the RBF center. Instead of using clustering self-organizing techniques [3] or an approach that exploits the principle of structural risk minimization [15], a heuristic driven learning strategy is adopted for the determination of the examples that should be used as RBF centers and of their widths. The former approaches are well suited for the effective learning of smooth functionals, even if these lie in high dimensional spaces. Their mathematical application though in the statistical distance metric space, which is constructed from symbolic data, meets severe problems due to the irregularities of the space and the need to cope with many artefacted examples.

The proposed GRBF training approach consists of two phases. At the first phase, the potential of each example for serving as an RBF center (i.e. how representative and reliable the example is) is evaluated. This is accomplished by an *Instance Based Learning* (IBL) scheme that resembles the functionality of PEBLS [6]. The IBL learning step is of heuristic type and is implemented with nearest neighbor based schemes [7, 8]. It offers the potentiality to detect the noisy examples of the designed classification system at this initial approximate solution and therefore the feasibility to remove them from the second GRBF learning phase. The classification function of IBL viewed as an input-output mapping tends to have many class boundaries and discontinuous "islands" of misclassified regions placed near erroneously classified examples. The structure of the decision boundaries is smoothed and most of the artefacted regions are extracted to reject the influence of noisy examples. In other words, examples that do not yield satisfactory performance at the initial IBL learning step are detected and marked in order to avoid their use as RBF centers.

At the second learning step, the Radial Basis Function (RBF) step, the Green's matrix is constructed by using the reliable examples as RBF centers. The spreads of the Gaussian kernels are estimated at the heuristic IBL driven first step. In few words, during the first step, an empirical aproximation to the solution is constructed. This approximation is *regularized* [3] at the second step due to the potential of the GRBF to construct a solution as a functional that obeys smoothness constraints.

Three basic approaches have been exploited for the implementation of the first heuristic IBL learning pass: The one pass, the used correct and the increment method. The *one pass* approach is an exemplar weighting method that is used in combination with the nearest

neighbor parameter. The nearest neighbor parameter determines the number of neighbors of an instance and in this case it has to attain a value of larger than one. As the name implies, the learning for this approach is accomplished with only one pass through the training examples. At this pass, for each training instance, its k nearest neighbors are detected from among the remaining training set. If j neighbors have a matching class then a weight is assigned to the current instance according to the simple formula: $weight = 1 + k - j$. Therefore, the more the class of the exemplar is reinforced by its neighbors, the less the weight (i.e. the more reliable the exemplar is). Algorithmically, the *one pass* instance based learning algorithm takes the form:

for *each pattern P of the training set* ***do***
Detect the k nearest neighbors of P from the training set according to the Statistical Distance Metric;
Let j = number of nearest neighbors with the same class label as the actual class of P;
Set the weight parameter that quantifies the reliability of the exemplar P as weight $= 1 + k - j$;
endfor;

As a particular example, consider the case that the *nearest neighbor* parameter is set to six. The exemplar weights will range from one to seven, depending on the number of neighbors that have a matching class. A weight of one means that all the neighbors reinforce the class assignment of the example (i.e. $j = k$), therefore this example is reliable. Conversely, at the other extreme case, a weight of seven, implies with a high probability that the isolated exemplar is artefacted and therefore it should not account too much at the final solution.

The one pass technique succeeds in attenuating significantly the effect of artefacted exemplars. Evidently, it attenuates also the exemplars placed near class separation boundaries. Even so, since the strength of the boundary exemplars is symmetrically reduced, the algorithm is not biased towards favoring a particular class.

Used correct is an alternative IBL heuristic weighting approach for determining the influence of the RBF centers. In this approach the *performance history* related to each exemplar is evaluated by running a large number of classification trials. Usually, all the patterns of the training set are used at those trials. At each such trial a sample pattern P is picked up from the training set and its classification $class(P)$ is evaluated as the classification of its nearest neighbor P_n, i.e. $class(P) = class(P_n)$. The assigned classification $class(P)$ is compared with the actual class of the pattern. Denoting by $used(i)$ the number of times the exemplar i is used to classify (as the nearest neighbour example), and by $correct(i)$ the number of times it is used correctly, the formula for weighting becomes:

$$weight = used(i)/correct(i)$$

Exemplars that have been used successfully to classify their neighbors are assigned a small weight (correspondingly a large region of influence). To reword, the spreading of an RBF kernel is adjusted by considering its success ratio as the percentage of times that it was used to classify correctly. This percentage evaluates the effectiveness of an RBF center as a classifier. We should note here, that at the evaluation of the nearest neighbour, a weighting scheme for the exemplars is not used, i.e. all examples are treated equally at the distance computations.

The algorithm for the used correct approach is formulated as:

for *all patterns p of the training set* ***do***
  used[p] = correct[p] = 1;
endfor;
for *all patterns p of the training set* ***do***
  Cp = class(p); // actual class of pattern p
  Pnearest = nearest_neighbor(p); //nearest neighbour of p without using weighting schemes for the exemplars
  Cnearest = class(Pnearest); //class of the nearest neighbour pattern
  Used[Cnearest]++; // increment count of times the pattern Cnearest is used to perform classification
  if *Cnearest == Cp* ***then***
    correct[Cnearest]++; //pattern Cnearest has been used to classify correctly
  endfor;
  for *all patterns p of the training set* ***do***
    weight [p] = used[p]/correct[p];
  endfor;
endfor;

Finally, the *increment* method is another effective exemplar weighting method for the implementation of the IBL learning step. This method assigns initially to all the weights a value of 1.00. Then, the single neighbor of each training instance is determined. The distance

is computed by ignoring the weighting (equivalent to assuming that each instance has a weight of 1.00). Therefore, the exemplar's weight equals the number of times that the particular example it is used at the training process minus the number of times that is used correctly. This approach differs from the used correct approach only at the final step that assigns the weights differently, i.e.

for *all patterns p of the training set* ***do***
weight[p] = used[p] − correct[p];
end;

6. Feature Weighting

As data sets become more complex, the number of irrelevant features inherent in the data increases. A feature is irrelevant if it makes no meaningful contribution towards the classification task. At best, such features increase the dimensionality of the data set and the computational cost. For the RBF case the number of basis functions and the number of training examples which are necessary for accurate parameter estimation increase exponentially with the intrinsic dimensionality. In effect the curse of dimensionality is one of the severest problems concerning the application of RBF networks. Even more important the inclusion of such features often also results in degradation in classification accuracy. As described previously, the nearest neighbor approach determines the class label of an unclassified instance by comparing it to a set of stored, classified instances and identifying the class label of the nearest neighbor in the set. As the distance between the unclassified and the stored instance is determined from the values of each feature, this approach is especially susceptible to the presence of irrelevant features. As a result the overall accuracy of the proposed classification system would degrade whenever irrelevant features exist within the data set. To address this problem we apply a feature weighting scheme.

Individual features are assigned a weight as part of the value-distance metric metric. Accordingly, the formula of the distance metric is modified to account for feature weighting as:

$$D(X, Y) = \sum_{f=1}^{F} w_f d\left(V_{X_f}, V_{Y_f}\right)^r \qquad (5)$$

where w_f is the weight assigned to feature f reflecting its relavance.

It is obvious that weighting each feature dimension equally (i.e. for each feature $w_f = 1$) allows irrelevant features to influence the distance computation and enhances equally features with different degrees of relevance. Several approaches have been proposed in literature in order to find appropriate values for feature weights [17–19]. Researches have reported benefits from using domain-specific information to assign features weights [20]. In the present work we initially explore three basic approches for including domain-specific knowledge in the weigthing of features and another more sophisticated one.

The first method assigns weights using a predefined shape. For example, a *triangle* feature weight shape gives more weight to features closer to the center. This weighting scheme is appropriate for applications like the protein secondary structure prediction problem [16], where the objective is to predict the folding of the protein from its sequence constitution. At such cases it is common to assign a greater weight to features nearer to the central residue.

The second approach is to assign feature weights individually. This is the method of choice if the relative importance of each feature is known a priori and it provides complete flexibility over the feature weights. A drawback of this alternative is that it is less convenient than using a pre-defined shape.

An experimental strategy is to allow the system to set the feature weights. A *genetic* method is used so called because it uses a technique suggestive of a genetic algorithm. The idea is to tweak a random feature weight by a random amount in the range $-fl \ldots fl$, where fl is a predefined floating point value. A random training instance is then selected, and its nearest neighbor determined. If the neighbor's class is identical to the chosen training instance, then the adjustment is accepted. Otherwise, the adjustment is rejected. This procedure is repeated many times. This method is highly experimental and has been found to improve performance in some experiments.

Statistical properties of larger homogenous groups may also simplify the task of computing appropriate weights. Therefore, an alternative approach is to adapt a class distribution weighting (CDW) [21], that allows weights to vary at the class level. The CDW starts from the premise that the important features are those that tend to have different values associated with different classes. An ideal feature would take on a unique set of values for each class. If such a feature exists, it provides all the class information readily. In most applications

of course ideal features are not available, but the degree to which each feature approximates the ideal can be measured. The CDW algorithm weights each feature proportionally to this measurement of the amount of information it provides. In particular the weights for symbolic features for a class C_i are computed using:

$$w_f(C_i) = \sum_{V \in f} |r(f, V, \{\mathbf{x} \mid \mathbf{x} \in C_i\}) - r(f, V, \{\mathbf{x} \mid \mathbf{x} \notin C_i\})| \quad (6)$$

where V is a value of feature f and $r(f, V, X')$ is, for a training subset $\mathbf{X}'$, $P(x'_f = V \mid x' \in \mathbf{X}')$.

CDW measures the usefulnes of a feature for classification by comparing the distributions of the feature values across various subsets of the training set. The weights for a particular class on a given feature are based on a comparison between distribution of feature values for the instances in that class and the distribution of values for instances in all other classes. If the distributions are highly similar, the feature is considered not useful for distinguishing that class from others and is assigned a low weight. If the distributions are highly dissimilar the feature is considered useful and is assigned a high weight. In short, CDW sets a feature's weight for a class C_i according to the degree to which its distributions of values for C_i differ from its distribution of other classes.

For applications where feature relevance does not vary significantly per class, a variation of the CDW can be used [21, 22], which transforms the sample representation to include one feature for every feature-value pair.

Nevertheless, we should note that no single best feature weighting approach exists, but rather the appropriate weighting depends heavily on the particular application.

7. Results

We have applied the GRBF based system to many standard data sets from the UCI machine learning repository. The results are illustrated with Table 1. This table compares the PEBLS performance, presented in the 2nd column, with that obtained by the proposed GRBF solution. The 3rd column displays the average generalization performance of the GRBF system but without the IBL estimation of parameters. Each of the next three columns (4th to 6th) summarizes the corresponding average classification performance of the GRBF with one of the three different IBL approaches (i.e. the one-pass, the used-correct and the increment approach). The improvements achieved by the IBL learning step are evident. Moreover, the GRBF solutions outperform clearly the simple nearest neighbor schemes.

Below we describe in more detail one application from the field of bioinformatics and one from the domain of data mining of commercial databases. The application from bioinformatics concerns the prediction of promoter sequences [5]. This task involves predicting whether or not a given subsequence of a DNA sequence is a promoter, i.e. a sequence of genes that initiates a process called transcription. The data set contains 106 examples, 53 of which were positive examples (promoters) and the rest negative

Table 1. Illustration of the performance of the proposed GRBF + IBL data mining algorithms compared to the plain IBL as implemented with the PEBLS learning system. We can observe that the utilization of IBL within the framework of GRBFs improves the generalization results. However, we cannot easily conclude that a particular IBL learning approach is better.

Database	PEBLS	GRBF	GRBF + IBL *One-pass*	GRBF + IBL *Used correct*	GRBF + IBL *Increment*
Hypothyroid	*97.90*	*98.04*	*98.33*	**98.34**	98.29
Breast Cancer	94.23	95.8	96.01	**96.12**	96.08
Iris	94.62	95.2	**96.22**	96.2	95.09
Hepatitis	76.59	78.23	79.45	**84.31**	81.29
Liver disorders	63.45	62.98	65.9	72.5	**74.56**
Heart disease	81.90	82.34	82.28	83.20	**85**
Audiology	77.9	78.91	**81.06**	81.03	79.44

ones. A training pattern consists of a sequence of 57 nucleotides (features) from the alphabet a, c, g, t and a respective classification (promoter or not promoter). Since the available number of patterns was small the classification performance was tested with the leave-one-out methodology, i.e. repeatedly trials have been performed by training on 105 examples and testing on the remaining one. The computed performance was 2/106 (i.e. an average of 2 errors over 106 trials) versus 4/106 for a competitive experiment that used the KBANN neural network model [23].

For the other application related to data mining of commercial databases, a large training set is extracted from a database kept by a mobile telecommunication company. This set contains 62,000 records concerning attributes of customers (e.g. job, area, sex, and pay method) and their classification in terms of their quality as customers. The classification is in the form of six classes ranging from 0 to 5, depending on increasing order on the customer's quality. The objective for the learning system was to perform well at predicting the class of a new customer from its attributes and thus to guide the company's strategy. For this problem, again the presented GRBF system with the statistical distance has obtained the best classification performance, i.e. around 60% success at the prediction of the customer's class. In contrast, a nearest neighbourhood classification scheme based on the same distance obtains only around 30% and a Self-Organizing Map (SOM) operating with the traditional distance types with a numerical coding of feature values (it is not easy to adapt the statistical distance within the context of the SOM) obtains a performance around 42%.

8. Conclusions

Neural network algorithms designed for learning are very effective in domains in which all features have numeric values. In these domains, the examples are treated as points and distance metrics obey standard definitions. However, the usual domain of data mining applications is the symbolic domain. In this domain, the utilization of the traditional distance metrics usually results in incompetent results.

This work has adapted an initially for nearest neighbor schemes proposed Statistical Distance Metric (SDM), expressing the distance between values of symbolic features, within the context and the peculiarities of the Generalized Radial Basis Function (GRBF) networks. The potential of this distance metric to regularize the solution of the GRBF networks is the theoretical justification of the improved performance related to the simple nearest neighbor schemes.

Additional performance improvement has been obtained by exploiting the fact that the examples of the training set are not all of the same importance and reliability. An Instance Based Learning (IBL) mechanism is implemented for the estimation of the reliability and the significance of each example. The RBF network is then designed to exploit effectively the irregularity of the problem's state space through the selection of the proper training examples as RBF centers. Furthermore, the weight parameters for each RBF center are determined with the IBL learning pass. These weight parameters account for the significance and reliability of the corresponding example. We have described three different Instance Based Learning (IBL) algorithms for the implementation of this learning step. Additional feature weights are embedded within the distance metric, preventing from classification accuracy degradation due to the presence of irrelevant features.

The data mining techniques that the paper has presented have a wide span of possible applications and have been adapted to heterogeneous data records having both symbolic and numeric data attributes. For the symbolic attributes the Statistical Distance Metric has resulted the best performances. However, for the numeric attributes the optimal policy for evaluating the distance is not obvious. In many cases the generalization improvement is obtained by handling the numeric values with a normalized numeric distance type, e.g. with an Euclidean distance metric normalized with the standard deviation. In addition, some attribute domains are better handled with a discretized numeric distance metric augmented with an interpolation to alleviate the discretization effects (e.g. the Interpolated Value Difference Metric [8]).

Future work to upgrade further the proposed GRBF and IBL hybrid data mining algorithms can precede along many different directions. Specifically, more elaborated schemes for treating numerical attributes by finding optimal multisplits [10] can be incorporated within the context of the current work. Also, another approach for the effective discretization of continuous attributes using a simulated annealing algorithm [11] can improve the results obtained by treating numeric

attributes as discretized symbolic. Furthermore, a nonlinear optimization of the GRBF parameters with an approach like the Levenberg-Marquardt algorithm [24] can (at least theoretically) further enhance the performance but at the cost of a more complex (and perhaps more unstable) implementation.

Acknowledgment

The authors wish to thank the Greek State Scholarship Foundation (SSF) for the financial support for this research (contract No. 134, 1-11-1999).

References

1. S. Haykin, *Neural Networks*, 2nd edn., MacMillan College Publishing Company: London, 1999.
2. V.N. Vapnik, *Statistical Learning Theory*, Wiley: New York, 1998.
3. T. Poggio and F. Girosi, "Regularization algorithms for learning that are equivalent to multilayer perceptrons," *Science*, vol. 247, pp. 978–982, 1990.
4. T. Poggio and F. Girosi, "Networks for approximation and learning," *Proceedings of the IEEE*, vol. 78, pp. 1481–1497, 1990.
5. P. Baldi and S. Brunak, *Bioinformatics*, MIT Press: Cambridge, MA, 1998.
6. C. Stanfill and D. Waltz, "Toward memory-based reasoning," *Communications of the ACM*, vol. 29, no. 12, pp. 1213–1228, 1986.
7. S. Berchtold, D.A. Keim, H.-P. Kriegel, and T. Seidl, "Indexing the solution space: A new technique for nearest neighbor search in high-dimensional space," *IEEE Transactions on Knowledge and Data Engineering*, vol. 12, no. 1, Jan./Feb. 2000.
8. D.R. Wilson and T.R. Martinez, "Improved heterogenous distance functions," *Journal of Artificial Intelligence Research* vol. 6, pp. 1–34, 1997.
9. F. Girosi, "An equivalence between sparse approximation and support vector machines," *Neural Computation*, vol. 10, no. 6, pp. 1455–1480, 1998.
10. T. Elomaa and J. Rousu, "General and efficient multisplitting of numerical attributes," *Machine Learning*, vol. 36, pp. 201–244, 1999.
11. J.C.W. Debuse and V. Jayward-Smith, "Discretisation of continuous commercial database features for a simulated annealing data mining algorithm," *Applied Intelligence*, vol. 11, pp. 285–295, 1999.
12. S. Chen, Y. Wu, and B.L. Lu, "Combined genetic algorithm optimization and regularized orthogonal least squares learning for radial basis function networks," *IEEE Trans. on Neural Networks*, vol. 10, no. 5, Sept. 1999.
13. S. Papadimitriou, A. Bezerianos, and A. Bountis, "Radial basis function networks as chaotic generators for secure communication systems," *International Journal on Bifurcation and Chaos*, vol. 9, no. 1, pp. 221–232, 1999.
14. A. Bezerianos, S. Papadimitriou, and D. Alexopoulos, "Radial basis function neural networks for the characterization of heart rate variability dynamics," *Artificial Intelligence in Medicine*, vol. 15, pp. 215–234, 1999.
15. P. Bartlett and J. Shawe-Taylor, "Generalization performance of support vector machines and other pattern classifiers," in *Advances in Kernel Methods, Support Vector Learning*, MIT Press: Cambridge, MA, pp. 43–54, 1999.
16. S. Cos and S. Salzberg, "A weighted nearest neighbor algorithm for learning with symbolic features," *Machine Learning*, vol. 10, pp. 57–78, 1993.
17. D. Wettschereck, D.W. Aha, and T. Mohri, "A review and empirical comparison of feature weighting methods for a class of lazy learning algorithms," *AI Review*, vol. 11, pp. 273–314, 1997.
18. T. Hastie and R. Tibshirani, "Discriminant adaptive nearest neighbor classification," *IEEE Pattern Analysis and Machine Intelligence*, vol. 18, pp. 607–616, 1996.
19. X.C. Ling and H. Wang, "Towards optimal weights setting for the 1-nearest neighbour learning algorithm," *AI Review*, vol. 11, pp. 255–272, 1997.
20. T. Cain, M.J. Pazzani, and G. Silverstein, "Using domain knowledge to influence similarity judgement," in *Proceeding of a Case-Based Reasonong Workshop*, Washington, DC, Morgan Kaufmann: San Mateo, CA, 1991, pp 191–202.
21. N. Howe and C. Cardie, "Examining locally varying weights for nearest neighbor algorithms," in *Second International Conference on Case-Based Reasoning*, edited by D. Leake and E. Plaza, Lecture Notes in Artificial Intelligence, Springer: Berlin, 1997, pp. 445–466.
22. D.W. Aha, "Feature weighting for lazy learning algorithms," in *Feature Extraction, Construction and Selection: A Data Mining Perspective*, edited by H. Liu and H. Motoda, Kluwer: Norwell, MA, 1998.
23. G. Towell, J. Shavlik, and M. Noordewier, "Refinement of aproximate domain theories by knowledge-based neural networks," in *Proceedings Eight National Conference on Artificial Intelligence*, AAAI Press: Menlo Park, CA, pp. 861–866, 1990.
24. C.M. Bishop, *Neural Networks for Pattern Recognition*, Clarendon Press: Oxford, 1996.

Stergios Papadimitriou received the Dipl. Eng. degreee and the Ph.D. degree from the Computer Engineering and Informatics Department, University of Patras, Greece, in 1990 and 1996, respectively. He worked for five years as a Research Assistant at the Institute of Computer Technology, Patras, Greece. His main research interests are neuro-fuzzy computing, chaotic dynamics, chaotic encryption, computational and artificial intelligence, support vector

learning and recurrent neural-network architectures. He is now a Senior Researcher at the Biomedical Signal Processing Laboratory of the Medical Physics Department and at the Artificial Intelligence Laboratory of the Computer Engineering Department of the University of Patras.

Seferina Mavroudi received the Dipl. Eng. degree in electronical engineering from the Department of Electronical and Computer Engineering of the Aristotle University of Thessaloniki, Greece, in 1998. She participated in an inter-departmental post-graduate program, where she received the Master's degree in biomedical engineering from the Medical School of the University of Patras, the Department of Electrical and Computer Engineering and the Department of Mechanical Engineering of the National Technical University of Athens, Greece in 2000. She is now working as a Ph.D. Fellow at the Biomedical Signal Processing Laboratory of the Medical Physics Department of the University of Patras. Her main research interests include neural networks, neuro-fuzzy architectures, artificial intelligence and nonlinear dynamics, mainly for biomedical applications.

Liviu Vladutu (S'01) received the B.S. degree (Hons.) in automation and computers science in 1987 from Craiova University, Romania, and the M.S. degree in biomedical engineering from the University of Patras, Greece, in 1999. He is currently working as a Ph.D. Fellow at the Department of Medical Physics, University of Patras, Greece. His current work involves application of computational intelligence methods for biomedical signal processing. His other interests include statistical learning theory and multiresolution analysis.

Anastasios Bezerianos (M'97) was born in Patras, Greece, in 1953. He received B.Sc. in physics from Patras University in 1976, the M.Sc. degree in telecommunications and electronics from Athens University, and the Ph.D. degree in Medical Physics from Patras University. He is currently Associate Professor in Medical School of Patras University. His main research interests are concentrated in biomedical signal processing and medical image processing, as follows: 1) data acquisition and on line processing using digital signal processors, 2) nonlinear time series analysis of elec-trocardiogram, 3) wavelet analysis of high resolution ECG, 4) modeling of heart muscle and heart rate variability, and 5) wavelet analysis of medical images.