Data Dimensionality Reduction With Application to Simplifying RBF Network Structure and Improving Classification Performance

Xiuju Fu and Lipo Wang, *Senior Member, IEEE*

***Abstract*—For high dimensional data, if no preprocessing is carried out before inputting patterns to classifiers, the computation required may be too heavy. For example, the number of hidden units of a radial basis function (RBF) neural network can be too large. This is not suitable for some practical applications due to speed and memory constraints. In many cases, some attributes are not relevant to concepts in the data at all. In this paper, we propose a novel separability-correlation measure (SCM) to rank the importance of attributes. According to the attribute ranking results, different attribute subsets are used as inputs to a classifier, such as an RBF neural network. Those attributes that increase the validation error are deemed irrelevant and are deleted. The complexity of the classifier can thus be reduced and its classification performance improved. Computer simulations show that our method for attribute importance ranking leads to smaller attribute subsets with higher accuracies compared with the existing SUD and Relief-F methods. We also propose a modified method for efficient construction of an RBF classifier. In this method we allow for large overlaps between clusters corresponding to the same class label. Our approach significantly reduces the structural complexity of the RBF network and improves the classification performance.**

***Index Terms*—Classifier, data dimensionality reduction, overlaps, RBF neural networks, SCM.**

I. Introduction

THE amount of data stored in organizations, both in terms of the number of patterns (samples) and the number of attributes (variables), is increasing rapidly. Data dimensionality reduction (DDR) aims at reducing the number of attributes under consideration. DDR has become an important aspect of data mining, since human experts and corporate managers are able to make better use of lower-dimensional data than higher-dimensional data. In addition, DDR can decrease the computational burden for various automated processes, for example, when a radial basis function (RBF) neural network is used to classify data. Reduced data dimensionality leads to less complicated network structure and thus increases efficiency in processing data. Removal of irrelevant data can significantly improve the accuracy of a classifier.

DDR is to map high-dimensional patterns onto lower-dimensional patterns. Techniques for DDR may be classified into two categories: feature (attribute) extraction and feature selection. Feature extraction creates a number of new features through a transformation of the raw features. Linear discriminant analysis (LDA) [3], [7], [11], [13], [14] and principal components analysis (PCA) [4], [6] are two popular techniques for feature extraction. Although these types of transformations are designed to maintain concepts in the data, it is difficult to prevent artifacts from affecting the original concepts in the data detrimentally.

Feature selection techniques select the best subset of features out of the original set. The attributes that are important to maintain the concepts in the original data are selected from the entire attribute set. How to determine the importance level of attributes is the key to feature selection technique. The idea of ranking feature importance for further selecting features out of the original feature set derives from the fact that different features contribute to the classification task differently, i.e., irrelevant features degrade the performance of classification both in the classification accuracy and the complexity of representing classification results. Mutual information based feature selection (MIFS) [3], [1] is a common method of feature selection, in which "the information content" of each attribute (feature) is evaluated with regard to class labels and other attributes. By calculating mutual information, the importance levels of features are ranked based on their ability to maximize the evaluation formula. However, in MIFS, the number of features to be selected needs to be pre-defined. Dash *et al.* [10] use an entropy measure (SUD) to evaluate the relative importance of attributes. The entropy measure is based on the similarities of different instances without considering the class labels. Attribute subsets were then selected based on attribute importance ranking and classification accuracy with C4.5. Kononenko [9] proposed Relief-F to rank attribute importance. In the Relief-F method, for a given instance, M nearest neighbors are searched from M different classes. A pair of instances is formed including the given instance and one of its nearest neighbors. Together with the nearest neighbor from the same class, there are $M + 1$ pairs of instances. The difference in an attribute in each pair of instances is calculated. The probabilities of these differences are used to evaluate the importance of the attribute.

In this paper, we propose a novel separability-correlation measure (SCM) for determining the importance of the original attributes. The SCM includes two parts, the intra-class distance to inter-class distance ratio and an attribute-class correlation measure. The attribute-class correlation measure is used to evaluate the power of each attribute affecting the class label for each pattern. The larger the correlation factor is, the more important the attribute is for determining the class labels of

Manuscript received May 28, 2000; revised April 21, 2002. This paper was recommended by Associate Editor P. K. Willett.

The authors are with the School of Electrical and Electronic Engineering, Nanyang Technological University (NTU), Singapore (e-mail: elpwang@ntu.edu.sg).

Digital Object Identifier 10.1109/TSMCB.2003.810911

patterns. The relative importance of a feature is given by its relative magnitude of the SCM.

Once attribute importance ranking is obtained using the SCM, a classifier needs to be used to select the attribute subset that leads to the lowest classification error. In this paper, the RBF neural network is used as a classifier.

We propose a simplified RBF classifier by allowing for large overlaps between patterns of the same class. The RBF neural network has attracted much attention in recent years due to its simple architecture and its ability to escape from the local minima from which multilayer perceptrons suffer. Usually, its kernel function is Gaussian. Each Gaussian kernel function is a cluster serving mainly a certain class. The boundary of the receptive field of the kernel function is a hyper-sphere. The Euclidean distance between a pattern and the center of the cluster measures the probability that a pattern belongs to a class.

The paper is organized as follows. The SCM measure for ranking the importance of attributes is proposed in Section II. Section III introduces how to construct the modified RBF neural network classifier efficiently. Experimental results on reducing data dimensionality and obtaining a simpler architecture of the RBF classifier are shown in Section IV. Finally, we conclude the paper in Section V.

II. Separability-Correlation Measure for Feature Importance Ranking

A. Class Separability Measure

The probability of correct classification is large, when the distances between different classes are large. Therefore, to identify a subset of features that can maximize the separability between classes is a desirable objective of feature selection.

Class Separability may be measured by the intraclass distance (the distance of patterns within class) S_w and the interclass distance (the distance between patterns of different classes) S_b [5]

$$S_w = \sum_{i=1}^{C} \frac{P_i}{n_i} \sum_{k=1}^{n_i} [(\bar{\vec{X}}_{ik} - \vec{m}_i)(\bar{\vec{X}}_{ik} - \vec{m}_i)^T]^{\frac{1}{2}} \tag{1}$$

and

$$S_b = \sum_{i=1}^{C} P_i [(\vec{m}_i - \vec{m})(\vec{m}_i - \vec{m})^T]^{\frac{1}{2}}. \tag{2}$$

Here C is the number of classes in the data set. n_i is the number of patterns in the i-th class. P_i is the probability of the i-th class. $\bar{\vec{X}}_{ik}$ is the normalized data vector, whose j-th attribute, $X_{ik}(j)$ is normalized as

$$\bar{X}_{ik}(j) = \frac{X_{ik}(j)}{\mathrm{Max}(x_j) - \mathrm{Min}(x_j)} \tag{3}$$

where $\mathrm{Max}(x_j)$ and $\mathrm{Min}(x_j)$ are the maximum and minimum of the j-th attribute in the data set respectively. $j = 1, 2, \ldots, n$. n is the number of attributes. $X_{ik}(j)$ is the original (unnormalized) data. $\vec{m}_i$ is the mean vector of the i-th class

$$\vec{m}_i = \frac{\sum_{k=1}^{n_i} \bar{\vec{X}}_{ik}}{n_i}. \tag{4}$$

$\vec{m}$ is the mean of all patterns in the data set

$$\vec{m} = \frac{\sum_{i=1}^{c} \sum_{k=1}^{n_i} \bar{\vec{X}}_{ik}}{n}. \tag{5}$$

N is the total number of patterns in the data set, i.e., $N = n_1 + n_2 + \cdots + n_c$.

The greater S_b is and the smaller S_w is, the better the separability of the data set is. Therefore, the ratio of S_w and S_b can be used to measure the distinction of the classes: the smaller the ratio, the better the separability [5].

If removing attribute k_1 from the data set leads to less class separability, i.e., a greater S_w/S_b, compared to the case where attribute k_2 is removed, one may consider attribute k_1 more important for classification of the data set than attribute k_2 is, and vice versa. Hence we may rank the importance of the attributes by calculating the intraclass-to-interclass distance ratio with each attribute omitted in turn.

However, the ratio S_w/S_b does not always work well as a class separability measure. For example, consider two classes, with one class surrounding the other, but are completely *separable*. Since $\vec{m}_1, \vec{m}_2$ and $\vec{m}$ defined in (4) and (5) are equal, $S_b \to 0$, which indicate total *inseparability*. Therefore there is a need to have other importance measures.

B. Attribute-Class Correlation Measure

In addition to the separability of classes in the data set, the correlation between the changes in attributes and their corresponding changes in class labels should be taken into account when ranking the importance of attributes. This correlation directly links features with class labels. To two different patterns, if their class labels are different, the variations of attributes in the two patterns are considered to be the affecting factor for the variation of class labels and should be weighted positively; if the class labels are the same, the variations in the attributes are irrelevant in deciding the classes and should be weighted negatively. The correlation measure can be a useful factor by combining together with our class separability measure.

We propose the following correlation between the k-th attribute and the class labels in the data set

$$C_k = \sum_{i \neq j} |\bar{\vec{X}}_{ik} - \bar{\vec{X}}_{jk}| \cdot \mathrm{magn}(y_i - y_j) \tag{6}$$

where $\bar{\vec{X}}_{ik}$ and $\bar{\vec{X}}_{jk}$ are the k-th attributes of the i-th pattern and the j-th pattern, respectively. y_i and y_j are the class labels of the i-th pattern and the j-th pattern, respectively. For any $y, \mathrm{magn}(y) = 1$ if $|y| > 0$ and $\mathrm{magn}(y) = -t$ if $|y| = 0$. A great magnitude of C_k shows that there is a close correlation between class labels and the k-th attribute, which indicates the great importance of attribute k in classifying the patterns, and vice versa.

C. Separability-Correlation Measure for Attribute Importance Ranking

We propose the following separability-correlation measure (SCM) to evaluate the importance levels of attributes by combining the above two measures

$$R_k = \chi \overline{S_k} + (1 - \chi) \overline{C_k} \tag{7}$$

where $S_k = (S_{wk}/S_{bk}), \overline{S_k} = (S_k - \mathrm{Min}(S_k))/(\mathrm{Max}(S_k) - \mathrm{Min}(S_k))$ is the normalization of S_k. $\mathrm{Max}(S_k)$ and $\mathrm{Min}(S_k)$ are the maximum and minimum of all S_k, respectively. $k = 1, 2, \ldots, n$. n is the number of attributes. S_{wk} and S_{bk} are intraclass and interclass distances calculated with the k-th attribute omitted from each pattern, respectively. For example, the i-th pattern $\vec{X}_i = \{x_{i1}, x_{i2}, \ldots x_{ik}, x_{ik+1}, \ldots x_{in}\}$ becomes $\vec{X}_i' = \{x_{i1}, x_{i2}, \ldots, x_{ik-1}, x_{ik+1}, \ldots, x_{in}\}$ when R_k is calculated. $\overline{C_k} = (C_k - \mathrm{Min}(C_k))/(\mathrm{Max}(C_k) - \mathrm{Min}(C_k))$ is the normalization of C_k. χ is the parameter to weight the two items for the final measure. Here $0 \leq \chi \leq 1$ and χ is determined empirically: the best choice of χ should lead to a subset of attributes which results in the highest classification accuracy.

The importance levels of attributes are ranked using the values of R_k. The greater the magnitude of R_k, the more important the k-th attribute. We will demonstrate the use of our SCM method in Section IV.

We use a combination of two measures, i.e., class separability and attribute class correlation, because either of them alone does not work well, as shown in our experimental results presented later in the paper.

D. Bottom-Up, Top-Down, and Exhaustive Search for Ranking Attributes

Either bottom-up search or top-down search can be used for ranking attribute importance. In a bottom-up search, we begin from an empty set. The SCM is used for evaluating each attribute by omitting this attribute from each pattern, i.e., the attribute is considered to be more important if its corresponding SCM magnitude is larger than others. The selected attribute is included to the empty attribute subset. This operation is continued until n attributes are included. The order of attributes entering the attribute set indicates the importance order of attributes. In a bottom-up search, the number of attribute combinations in the SCM calculation for determining attribute importance equals to n (n is the number of attributes).

In a top-down search, we start from the complete attribute set. Each attribute is removed from the attribute set temporarily for calculating its SCM. Then the least important attribute, whose corresponding value in SCM is the smallest, is eliminated from the current attribute set. The steps are iterated until only one attribute is left in the attribute set. The number of attribute combinations in the SCM calculation is $n + (n-1) + \cdots + 1 = n(n+1)/2$. In Section IV, the differences of the two searches will be shown.

In an exhaustive search, all the possible attribute subsets are examined. The number of attribute combinations needed to be checked is $C_n^1 + C_n^2 + \cdots + C_n^{(n-1)} + C_n^n = 2^n - 1$. In the branch and bound algorithm, some attribute sets need not to be examined, which leads to the saving in computation; however, the number of calculations still grows exponentially with n. In addition, the computational saving is achieved by assuming the feature selection evaluation functions is monotonic [5].

Due to the computational burden of optimal search methods, one has to resort to suboptimal feature selection methods. In classification tasks, since the goal is to obtain better classification accuracy with less complicated construction of classifiers, the strategy of using the classification accuracy as evaluation for selecting features is used widely. We use suboptimal search and RBF classifiers as evaluators in this paper.

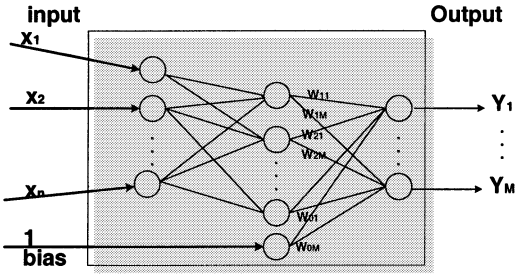

Fig. 1. Architecture of an RBF neural network.

III. Modified Method for Constructing an Efficient RBF Classifier

RBF neural networks [15], [17], [18] are widely used for function approximation, pattern classification and so on. In this paper, we use an RBF classifier together with the SCM to select the best subsets of attributes. In the RBF neural network, the activation of a hidden unit is determined by the distance between the input vector and the center vector of the hidden unit. The weights connecting the hidden layer and the output layer can be determined by a linear least square (LLS) method [2], which is fast and free of local minima, in contrast to the multilayer perceptron neural network.

There are three layers in the RBF neural network, i.e., the input layer, the hidden layer with Gaussian activation functions, and the output layer. The architecture of the RBF neural network is shown in Fig. 1. In this paper, we use the RBF network for classification. If there are M classes in the data set, we write the m-th output of the network as

$$y_m(\mathbf{X}) = \sum_{j=1}^{K} w_{mj} \emptyset_j(\mathbf{X}) + w_{m0} b_m. \tag{8}$$

Here, $\mathbf{X}$ is the n-dimensional input pattern vector, $m = 1, 2, \ldots, M, K$ is the number of hidden units. M is the number of output. w_{mj} is the weight connecting the j-th hidden unit to the m-th output node. b_m is the bias. w_{m0} is the weight connecting the bias and the m-th output node. $\emptyset_j(\mathbf{X})$ is the activation function of the j-th hidden unit

$$\emptyset_j(\mathbf{X}) = e^{\frac{\|\mathbf{X} - \mathbf{C_j}\|^2}{2\sigma_j^2}} \tag{9}$$

where $\mathbf{C_j}$ and σ_j are the center and the width for the j-th hidden unit, respectively, which are adjusted during learning.

Finding the centers, widths, and the weights connecting hidden nodes to the output is the key for constructing and training the RBF classifier. Both the dimensionality and the distribution of the input patterns affect the number of the hidden units. If the dimensionality is reduced, the number of hidden units will also be decreased.

Overlapped receptive fields of different clusters can improve the performance of the RBF classifier when dealing with noisy data [12]. In [8] and [19], overlapping Gaussian kernel functions are created to map out the territory of each cluster with a smaller number of Gaussians. In those previous methods, the

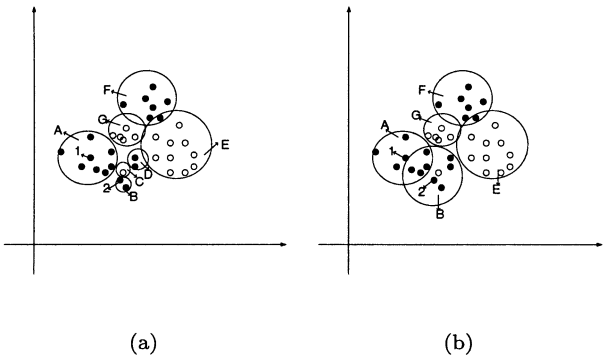

Fig. 2. Comparison between (a) existing algorithms: small overlaps between clusters, and (b) the modified algorithm with reduced number of clusters: small overlaps between clusters of different classes, but large overlaps between clusters of the same class.

clusters are formed as follows. A pattern is randomly selected from the data set V as the initial center of a cluster. The radius of this cluster is chosen in such a way that the ratio between the number of patterns of a certain class (in-class patterns) and the total number of patterns in the cluster is not less than a pre-defined value θ. Once this cluster is formed, all patterns inside this cluster are "removed" from the data set and do not participate in the formation of other clusters. The value of θ is determined empirically and is related to an acceptable classification error rate. Since θ determines the radii of the clusters, it also indirectly determines the degree of overlaps between different clusters. Generally, a large θ leads to small radii of clusters, thus it leads to small overlaps between the Gaussians for different clusters and a small classification error rate for the training data set. Since a small classification error is desired, there usually exist small overlaps between the Gaussians representing the clusters.

Let us consider a simple example. Suppose $\theta = 0.8$, i.e., there must have at least 80% in-class patterns in each cluster. In Fig. 2(a), suppose cluster A has been formed and its members "removed" from the data set V. Suppose pattern 2 is subsequently selected as the initial center of a new cluster and cluster B is thus formed. Clusters C through G are then formed similarly in sequence. We see that clusters B, C, and D are quite small and therefore the effectiveness of the above clustering algorithm needs to be improved.

In this paper, we propose an algorithm to reduce the number of clusters as follows. We first make a copy V_c of the original data set V. When a qualified cluster (the ratio of in-class patterns should be higher than θ), e.g., cluster A in Fig. 2(b) [same as in Fig. 2(a)], is generated, the members in this cluster are "removed" from the copy data set V_c, but the patterns in the original data set V remain unchanged. Subsequently, the initial center of the next cluster is selected from the *copy data set* V_c, but the candidate members of this cluster are patterns in the *original data set* V, and thus include the patterns in the cluster A. Subsequently when pattern 2 is selected as an initial cluster center, a much larger cluster B, which combines clusters B, C, and D in Fig. 2(a), can still meet the θ-criterion and can therefore be created. By allowing for large overlaps between clusters for *the same class*, we can further reduce the number of clusters substantially. This will lead to more efficient construction of RBF networks, and will be demonstrated by computer simulations in the next section.

We therefore use the following algorithm to construct an efficient RBF classifier, incorporating the above modification to the existing algorithms [19], [20].

1) Initialize for training.
   a) We divide the data set into three parts, the training data set, the validation data set, and the test data set.
   b) In order to derive the widths of the kernel functions, a general scale of neighborhood δ_0 is obtained by calculating the standard deviation of the data set [19].
2) Set stage $L = 1$ (L indicates the steps in one training procedure); $\delta(L) = \delta_0, \delta = \alpha \cdot \delta_0$, where $\delta(L)$ is the initial radius of clusters at training stage L (training stage indicates the status of the training procedure, in which initial radius of clusters are affected), the and δ is the increment step for the radius. α is the change rate of radius.
3) Generate V_c, a copy of the original training data set V.
4) Forming clusters.
   a) Count the number of patterns in classes in V_c. If the number of patterns in a class is fewer than a predefined number (we use four in this paper), the patterns in the class will not be selected.
   b) Set sub-stage $L_s = 1$ (L_s indicates the shrinking degree in the radius of cluster in a sub-stage training procedure), $\delta(L_s) = \delta(L)$.
   c) Select randomly a pattern from V_c as an initial cluster center, search in V all the patterns within $\delta(L)$-neighborhood of the center pattern. Thus large overlaps are permitted among clusters of the same class, as proposed above.
   d) Check whether the ratio between in-class patterns and the total patterns in the subset is greater than a pre-defined value θ. If the ratio is less than θ, set $L_s = L_s + 1$, and $\delta(L_s) = \delta(L_s) - \delta$. Search the patterns within $\delta(L_s)$-neighborhood of the selected pattern. Stop only if the ratio criterion is met or if $L_s \geq (1/(2\alpha))$. Count the number of epoch N_e, if $N_e \geq D \cdot i$ (i is the number of patterns in V_c, D is an integer. Empirically, $D = 3$.), $\theta = 0.9\theta$. Repeat 4 until the training set V_c is empty.
5) Calculate the center and width of each cluster: the center is the mean pattern of all patterns in the cluster and the width is the standard deviation of these patterns.
6) Obtain weights by the LLS method [2].
7) Calculate $E_{\rm tr}$ (the classification error of the training set) and E_v (the classification error of the validation set). Stop if both of $E_{\rm tr}$ and E_v are smaller than a pre-specified value E_0. In this paper, $E_0 = E_{\rm Pre}$ ($E_{\rm Pre}$ is a predefined value of classification error rate. $E_{\rm Pre} = 2\%$ in this paper). Else:
   If $E_v(L) < E_v(L-1)$, set $L = L + 1$ and $\delta(L) = \delta(L) - \delta/2$. Go to 3.
   If $E_{\rm tr}(L) > E_{\rm tr}(L-1)$ and $E_v(L) > E_v(L-1), L = L + 1, \delta(L) = \delta(L) - \delta$, go to 3.
   If $L > (1/\alpha)$ or $E_v > E_0$ go to 2.

Compared to Roy *et al.*'s original algorithm [19], [20], we have made following changes.

1) In Step 4, large overlaps among clusters of the same class are allowed in order to reduce the number of hidden units without reducing the classification accuracy.
2) In Step 6, the LLS method [2] is used to obtain the weights between the hidden layer and the output layer.
3) In [19], six training stages were used corresponding to $\theta = \{50\%, 60\%, \ldots, 100\%\}$, respectively. In each stage, θ is unchanged and all clusters of this stage should meet the θ-criterion. A classification result is obtained for this θ. The classification error rate from a stage is compared to that in the previous stage. Whether to continue to the next training stage is determined by a comparison in the classification results. If the classification accuracy is better than the previous stage, i.e., it is possible to obtain a higher accuracy if θ is increased. Thus, the data is trained again using a larger θ. Otherwise, the training is stopped. Although a larger θ leads to a higher accuracy in the training data set, it may lead to poorer generalization in the testing data set. Hence, in our algorithm (Step 2), θ is automatically adjusted according to the training condition, i.e., how many patterns of the class concerned are left. With the decreasing number of patterns, θ is decreased by a certain factor, say 0.9 in our simulations.

The aim of the modification proposed above, i.e., allowing for large overlaps among clusters of the same class, is to decrease the number of Gaussian hidden units without reducing the classification accuracy. Overlaps between clusters of *different* classes affect the classification error rate, i.e., the larger the overlaps between clusters of *different* classes, often the larger the classification error rate. However, in our paper, large overlaps between clusters of the *same* class are allowed, which is not expected to degrade the classification accuracy. Since large overlaps between clusters of the same class can help combine small clusters into larger ones, and noise may thus be suppressed by these combinations, the accuracy of classification may even be improved.

The cost of the modification is increased training time. Assume the number of patterns in the data set is N. The number of hidden units is M_1 when allowing for large overlaps among clusters of the same class, and the number of hidden units is M_2 without allowing for large overlaps. For a certain hidden unit k, the number of patterns in this cluster is N_k. The processing time for one pattern is T in the training procedure. Assume that both algorithms (with the modification and without the modification) have the same initial conditions. The initial center of each candidate cluster is selected randomly. Thus, the average number of trials in two algorithms for searching qualified clusters may be assumed to be the same, say M_0. With the modification, all N patterns in the data set will be checked when searching for a qualified cluster in each trial. The time required for one trial is thus NT. For M_0 trials, the total time required is $M_0 \cdot NT$. Without the modification, if a cluster is considered to be qualified, the patterns included in this cluster will be removed from the data set. Thus, the total time required for classification without the modification is $\sum_{m=1}^{M_0} T(N - \sum_{k=1}^{m-1} N_k) = (M_0 N - \sum_{m=1}^{M_0} \sum_{k=1}^{m} N_k)T < M_0 NT$. Thus, more time is needed when applying the modification. However, the cost is worthwhile in that less complicated classifiers with higher accuracy are found in most cases.

TABLE I
REDUCTION IN THE NUMBER OF HIDDEN UNITS IN THE RBF NETWORK WHEN LARGE OVERLAPS ARE ALLOWED BETWEEN CLUSTERS FOR THE SAME CLASS. ALL ATTRIBUTES IN EACH DATA SET ARE USED AS INPUTS

Data set		Iris	Monk3	Thyroid	Breast
classification	small overlap	0.0373	0.0591	0.0465	0.0292
error rate	large overlap	0.0467	0.0688	0.0465	0.0146
number of	small overlap	5.2	33.2	17	34
hidden units	large overlap	4	19.6	8	11

Based on the attribute importance ranking, we further propose to reduce the structural complexity and to improve the performance of the RBF network as follows. According to the rank of importance level obtained by the algorithm described in Section II, J most important attributes are used for classification with the RBF neural network for $J = 1, 2, \ldots, N-1, N$. The classification error rate is calculated for each J. Thus, N classification error rates are calculated corresponding to N subsets of attributes. For small J, classification error rate decreases as J increases until all relevant attributes are included. As J increases further, the classification error rate may remain unchanged or even increase because redundant or irrelevant attributes are included. The best subset of attributes is the one with the smallest classification error rate.

IV. EXPERIMENTAL RESULTS

Iris, Monk3, Thyroid, and Breast Cancer data sets from the UCI Repository of Machine Learning Databases [16] are used in this paper to test our algorithms for ranking attribute importance and constructing a simplified RBF network. Each data set is divided into three parts, i.e., training, validation, and test sets. We set $\alpha = 0.1$ and $\theta = 100\%$ in our experiments. Each experiment is repeated five times with different initial conditions and the average results are recorded.

Table I shows that when large overlaps among clusters of the same class are permitted, the number of hidden units is decreased while nearly the same classification error rate is maintained.

Attribute importance rankings using the SCM with different χ's [see (7)] are shown in Table II, which shows that χ affects the order of attribute importance ranking. 5χ's are used, i.e., $\chi = 0.0, 0.4, 0.5, 0.7$, and 1.0. In order to determine which order is better, different subsets of attributes are input to the RBF classifier for each order, so as to find the best subset for that order. If there are n original attributes in the data set, there are n candidate subsets of attributes as discussed in Subsection II.D. The classification results are used to evaluate the attribute subsets. We select the subset of attributes corresponding to the lowest classification error rate for each data set and each ranking order. According to the experimental results (details for each data set are given below), when $\chi = 0.4$, the importance ranking results for the four data sets lead to the lowest or nearly the lowest validation error rates with the smallest attribute subsets.

TABLE II
ATTRIBUTE IMPORTANCE RANKING USING THE SCM WITH DIFFERENT χ OBTAINED BY BOTTOM-UP SEARCH

χ	Iris	Monk3	Thyroid	Breast
$\chi = 0.0$	4,3,1,2	5,4,2,1,6,3	2,3,5,1,4	7,2,4,3,8, 9,5,6,1
$\chi = 0.4$	4,3,1,2	5,2,4,1,6,3	2,3,5,4,1	2,7,3,4,9, 5,8,6,1
$\chi = 0.5$	4,1,3,2	5,2,4,1,6,3	2,3,5,4,1	2,7,3,4,9, 5,1,8,6
$\chi = 0.7$	1,4,2,3	5,2,4,1,6,3	2,5,3,4,1	2,7,1,3,4, 9,5,8,6
$\chi = 1.0$	1,2,4,3	5,2,3,6,4,1	2,5,3,4,1	1,2,7,3,4, 9,5,8,6

TABLE III
CLASSIFICATION ERROR RATES FOR IRIS DATA SET WITH DIFFERENT SUBSETS OF ATTRIBUTES ACCORDING TO THE IMPORTANCE RANKING SHOWN IN TABLE II WHEN $\chi = 0.0$ AND $\chi = 0.4$. ATTRIBUTE SUBSET WITH THE LOWEST VALIDATION ERROR IS HIGHLIGHTED IN BOLD

Attributes used	Error Rate		
	Training set	Validation set	Test set
4	0.1222	0.0667	0.1333
4,3	**0.0333**	**0.0000**	**0.0333**
4,3,1	0.0556	0.0333	0.1000
4,3,1,2	0.0889	0.1000	0.1000

TABLE IV
SAME AS TABLE III, WHEN $\chi = 0.5$

Attributes used	Error Rate		
	Training set	Validation set	Test set
1	0.3333	0.2333	0.4333
1,4	0.0778	0.0000	0.1000
1,4,2	**0.0556**	**0**	**0.0333**
1,4,2,3	0.0556	0	0.0333

TABLE V
SAME AS TABLE III, WHEN $\chi = 0.7$

Attributes used	Error Rate		
	Training set	Validation set	Test set
4	0.3333	0.3667	0.4667
4,1	0.1111	0.2000	0.1000
4,1,3	0.3333	0.2333	0.4333
4,1,3,2	**0.0778**	**0.1000**	**0.0333**

TABLE VI
SAME AS TABLE III, WHEN $\chi = 1.0$

Attributes used	Error Rate		
	Training set	Validation set	Test set
1	0.4556	0.5667	0.3667
1,2	0.1889	0.3333	0.2333
1,2,4	0.0778	0.0667	0.1667
1,2,4,3	**0.0444**	**0.0667**	**0.0333**

TABLE VII
CLASSIFICATION ERROR RATES FOR Monk3 DATA SET WITH DIFFERENT SUBSETS OF ATTRIBUTES ACCORDING TO THE IMPORTANCE RANKING SHOWN IN TABLE II WHEN WHEN $\chi = 0.0$. THE ATTRIBUTE SUBSET WITH THE LOWEST VALIDATION ERROR IS HIGHLIGHTED IN BOLD

Attributes used	Error Rate		
	Training set	Validation set	Test set
5	0.2705	0.2328	0.2100
5,4	0.2541	0.3060	0.2450
5,4,2	**0.0902**	**0.0991**	**0.0650**
5,4,2,1	0.1967	0.2371	0.2050
5,4,2,1,6	0.1148	0.0948	0.1000
5,4,2,1,6,3	0.1885	0.2112	0.2600

TABLE VIII
SAME AS TABLE VII, WHEN $\chi = 0.4, \chi = 0.5$ AND $\chi = 0.7$

Attributes used	Error Rate		
	Training set	Validation set	Test set
5	0.1880	0.3000	0.2870
5,2	0.1780	0.2830	0.2690
5,2,4	**0.0242**	**0.0585**	**0.067**
5,2,4,1	0.0899	0.3360	0.1830
5,2,4,1,6	0.0498	0.1897	0.1320
5,2,4,1,6,3	0.0328	0.2030	0.1240

A. Iris Data Set

There are four attributes in Iris data set. 150 patterns of Iris data set are divided into three sets, i.e., 90 patterns for training, 30 for validation, and 30 for testing.

In Tables III–VI, classification error rates are shown for all attribute subsets corresponding to different attribute importance ranking results based on different χ's. $\chi = 0.4$ is selected for that it leads to the smallest attribute subset $\{3, 4\}$ with the nearly lowest classification error rate.

B. Monk3 Data Set

There are six attributes in Monk3 data set. Monk3 data has a training set with 122 patterns and a test set with 421 patterns. We divide the test set into 200 patterns for validation and 221 patterns for testing.

In Tables VII to IX, classification error rates are shown for all attribute subsets corresponding to different attribute importance ranking results based on different χ's. $\chi = 0.4$ is selected for that it leads to the smallest attribute subset $\{2, 4, 5\}$ with the lowest classification error rates.

TABLE IX
SAME AS TABLE VII, WHEN $\chi = 1.0$

Attributes used	Error Rate		
	Training set	Validation set	Test set
5	0.2705	0.2328	0.2100
5,2	0.2213	0.1853	0.2050
5,2,3	0.1967	0.1638	0.1400
5,2,3,6	**0.1066**	**0.0690**	**0.0700**
5,2,3,6,4	0.2131	0.1767	0.1700
5,2,3,6,4,1	0.1230	0.1552	0.1600

TABLE X
CLASSIFICATION ERROR RATES FOR THYROID DATA SET WITH DIFFERENT SUBSETS OF ATTRIBUTES ACCORDING TO THE IMPORTANCE RANKING SHOWN IN TABLE II WHEN $\chi = 0.0$. THE ATTRIBUTE SUBSET WITH THE LOWEST VALIDATION ERROR IS HIGHLIGHTED IN BOLD

Attributes used	Error Rate		
	Training set	Validation set	Test set
2	0.1860	0.1628	0.2093
2,3	0.0698	0.0698	0.2093
2,3,5	**0.0543**	**0.0465**	**0.0930**
2,3,5,1	0.0543	0.0465	0.1163
2,3,5,1,4	0.0388	0.0465	0.1395

TABLE XI
SAME AS TABLE X, WHEN $\chi = 0.4, \chi = 0.5, \chi = 0.7$ AND $\chi = 1.0$

Attributes used	Error Rate		
	Training set	Validation set	Test set
2	0.0930	0.0930	0.0930
2,3	0.0698	0.0465	0.0698
2,3,5	**0.0543**	**0.0233**	**0.0233**
2,3,5,4	0.0543	0.0233	0.0465
2,3,5,4,1	0.0388	0.0233	0.0233

C. Thyroid Data Set

There five attributes in Thyroid data set. There are 215 patterns in Thyroid data set. 115 patterns for training, 50 for validation, and 50 for testing.

In Tables X and XI, classification error rates for all attribute subsets corresponding to different attribute importance ranking results based on different χ's are shown. $\chi = 0.4$ is selected for that it leads to the smallest attribute subset $\{2, 3, 5\}$ with the lowest classification error rates.

D. Breast Cancer Data Set

There are and nine attributes in Breast cancer data set. There are 699 patterns in Breast cancer data set. 16 patterns with losing attribute are removed. Of the 683 patterns left, 444 were benign, and the rest were malign. In 683 patterns, 274 patterns for training, 204 for validation, 205 for testing.

TABLE XII
CLASSIFICATION ERROR RATES FOR BREAST CANCER DATA SET WITH DIFFERENT SUBSETS OF ATTRIBUTES ACCORDING TO THE IMPORTANCE RANKING SHOWN IN TABLE II WHEN $\chi = 0.0$. THE ATTRIBUTE SUBSET WITH THE LOWEST VALIDATION ERROR IS HIGHLIGHTED IN BOLD

Attributes used	Error Rate		
	Training set	Validation set	Test set
7	0.1100	0.0803	0.1022
7,2	0.0954	0.0803	0.0949
7,2,4	0.0391	0.0511	0.0219
7,2,4,3	**0.0318**	**0.0438**	**0.0073**
7,2,4,3,8	0.0367	0.0365	0.0219
7,2,4,3,8,9	0.0244	0.0365	0.0146
7,2,4,3,8,9,5	0.0318	0.0438	0.0146
7,2,4,3,8,9,5,6	0.0342	0.0365	0.0146
7,2,4,3,8,9,5,6,1	0.0342	0.0511	0.0219

TABLE XIII
SAME AS TABLE XII, WHEN $\chi = 0.4$

Attributes used	Error Rate		
	Training set	Validation set	Test set
2	0.1100	0.0803	0.1022
2,7	0.0709	0.0657	0.0876
2,7,3	**0.0269**	**0.0365**	**0.0073**
2,7,3,4	0.0391	0.0438	0.0365
2,7,3,4,9	0.0269	0.0365	0.0219
2,7,3,4,9,5	0.0342	0.0365	0.0146
2,7,3,4,9,5,8	0.0293	0.0438	0.0073
2,7,3,4,9,5,8,6	0.0269	0.0438	0.0146
2,7,3,4,9,5,8,6,1	0.0342	0.0365	0.0146

For 5χ's, there are five different attribute importance ranking results. In Tables XII–XVI, classification error rates for all attribute subsets corresponding to different attribute importance ranking results based on different χ's are shown. $\chi = 0.4$ is selected for that it leads to the smallest attribute subset $\{2, 3, 7\}$ with the lowest classification error rates.

E. Comparisons Between Top-Down and Bottom-Up Search and With Other Methods

In this subsection, we compare results obtained from the SCM using bottom-up and top-down search for $\chi = 0.4$. We also compare with results derived from the attribute importance ranking by Relief-F [9] and SUD [10].

Fig. 3(a) shows the classification error rates of the RBF classifier for different subsets of Iris attributes according to the importance ranking obtained with SCM using *bottom-up*

TABLE XIV
SAME AS TABLE XII, WHEN $\chi = 0.5$

Attributes used	Error Rate		
	Training set	Validation set	Test set
2	0.1443	0.1314	0.1387
2,7	0.0513	0.0511	0.0365
2,7,3	**0.0269**	**0.0292**	**0.0073**
2,7,3,4	0.0318	0.0511	0.0146
2,7,3,4,9	0.0293	0.0438	0.0219
2,7,3,4,9,5	0.0269	0.0365	0.0292
2,7,3,4,9,5,1	0.0244	0.0438	0.0073
2,7,3,4,9,5,1,8	0.0318	0.0365	0.0146
2,7,3,4,9,5,1,8,6	0.0318	0.0365	0.0146

TABLE XV
SAME AS TABLE XII, WHEN $\chi = 0.7$

Attributes used	Error Rate		
	Training set	Validation set	Test set
2	0.1443	0.1314	0.1387
2,7	0.0538	0.0584	0.0438
2,7,1	0.0685	0.0876	0.0730
2,7,1,3	**0.0342**	**0.0365**	**0.0146**
2,7,1,3,4	0.0367	0.0365	0.0219
2,7,1,3,4,9	0.0269	0.0438	0.0292
2,7,1,3,4,9,5	0.0318	0.0511	0.0146
2,7,1,3,4,9,5,8	0.0293	0.0365	0.0292
2,7,1,3,4,9,5,8,6	0.0391	0.0511	0.0292

TABLE XVI
SAME AS TABLE XII, WHEN $\chi = 1.0$

Attributes used	Error Rate		
	Training set	Validation set	Test set
1	0.3423	0.3869	0.3358
1,2	0.1467	0.1314	0.1460
1,2,7	0.0660	0.0730	0.0511
1,2,7,3	**0.0489**	**0.0292**	**0.0146**
1,2,7,3,4	0.0416	0.0365	0.0146
1,2,7,3,4,9	0.0367	0.0438	0.0292
1,2,7,3,4,9,5	0.0269	0.0365	0.0146
1,2,7,3,4,9,5,8	0.0318	0.0365	0.0219
1,2,7,3,4,9,5,8,6	0.0318	0.0365	0.0292

search when $\chi = 0.4$. We obtained the same attribute ranking results and hence the same attribute subsets from the SCM using bottom-up and top-down search (Table XVII) for Iris. It

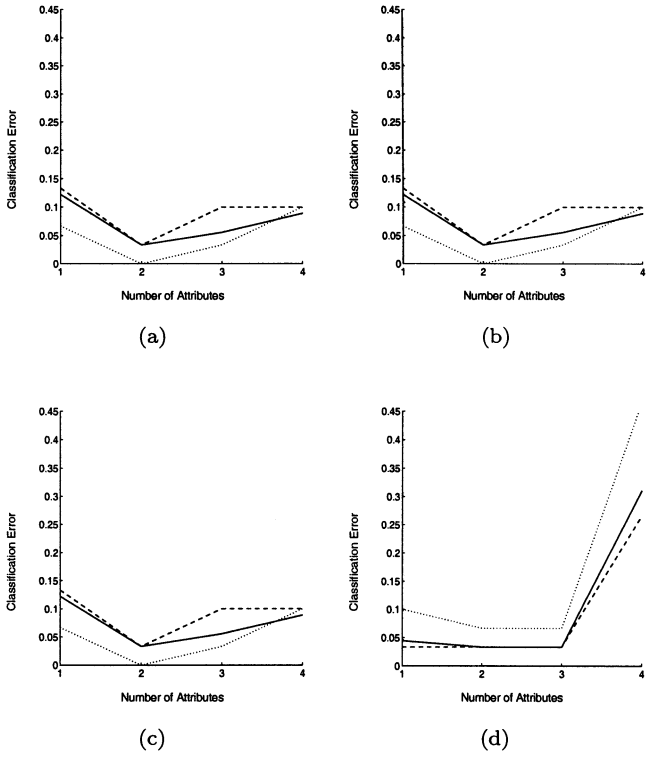

Fig. 3. Classification error rates of Iris data set for different numbers of attributes used according to attribute ranking results obtained from: (a) SCM with bottom-up search, (b) SCM with top-down search, (c) relief-F, and (d) SUD. Solid line: training data set; dotted line: validation data set; dashed line: test data set.

is seen from Fig. 3(a)–(b) that as the number of attributes used increases, the test error first decreases, reaches a minimum when the first 2 attributes in the attribute ranking queue are used, and then increases. Hence in the Iris data set, attributes 3 and 4 are relevant attributes for classification and are then selected, which improves the classification performance and decreases the number of inputs and the number of hidden units of the RBF neural network. The classification error rate is reduced from 0.0467 to 0.0333, and the number of Gaussian hidden units is reduced from 4 to 3. Table XVIII summarizes results for different data sets.

Further, we carry out classification by inputting different attribute sets to the RBF classifiers based on the attribute importance ranking results obtained by SUD [10] and Relief-F [9] methods (reproduced in Table XVII). We compare the classification error rates on test data sets corresponding to the selected attribute subsets obtained using the SCM, SUD, and Relief-F methods (Table XIX).

In Fig. 3(c) and (d), we show that attributes 3 and 4 lead to the lowest error rates (Table XIX) in both SUD and Relief-F methods. Hence, the selected attribute subset for Iris data set is $\{3, 4\}$ according to SUD and Relief-F, which is the same as the result based on our SCM method.

We obtained the same attribute ranking results and hence the same attribute subsets using our SCM with both bottom-up and top-down search for Monk3 data set (Table XVII). Fig. 4(a)–(b) show that attributes 2, 4, and 5 should be selected for Monk3 data set, which decreases the classification error rate from

TABLE XVII
COMPARISON BETWEEN IMPORTANCE RANKING RESULTS OBTAINED BY OUR SCM USING BOTTOM-UP AND TOP-DOWN SEARCH WHEN $\chi = 0.4$, THE SUD AND RELIEF-F METHODS

Data set	Decreasing order of importance			
	SCM (bottom-up)	SCM (top-down)	SUD	Relief-F
Iris	4,3,1,2	4,3,1,2	3,4,1,2	4,3,1,2
Monk3	5,2,4,1,6,3	5,2,4,1,6,3	5,2,4,1,6,3	2,5,4,3,6,1
Thyroid	2,3,5,4,1	2,5,3,4,1	4,5,3,2,1	4,3,1,2,5
Breast	2,7,3,4,9,5,1,8,6	7,2,3,4,9,5,8,6,1	1,7,3,2,5,6,4,8,9	6,2,3,7,5,1,4,8,9

TABLE XVIII
COMPARISON OF THE NUMBERS OF HIDDEN-UNITS AND CLASSIFICATION ERRORS BEFORE AND AFTER IRRELEVANT ATTRIBUTES ARE REMOVED ACCORDING TO THE SCM RANKING METHOD. (B) BEFORE REMOVAL, (A) AFTER REMOVAL

Comparison		Data set			
		Iris	Monk3	Thyroid	Breast
Input attributes	B	1,2,3,4	1,2,3,4,5,6	1,2,3,4,5	1,2,3,4,5,6,7,8,9
	A	4,3	5,2,4	2,3,5	2,7,3
Number of hidden units	B	4	19.6	8	11
	A	3	11.6	5	5
Classification error rate	B	0.0467	0.0688	0.0465	0.0146
	A	0.0333	0.067	0.0233	0.0073

TABLE XIX
COMPARISON BETWEEN CLASSIFICATION ERROR RATES ON TESTING DATA SETS WITH THE BEST ATTRIBUTE SUBSETS OBTAINED BY OUR SCM, SUD AND RELIEF-F METHODS

Data set	classification error rates		
	SCM	SUD	Relief-F
Iris	0.0333	0.0333	0.0333
Monk3	0.067	0.0.067	0.09
Thyroid	0.0233	0.093	0.1163
Breast	0.0073	0.0146	0.0073

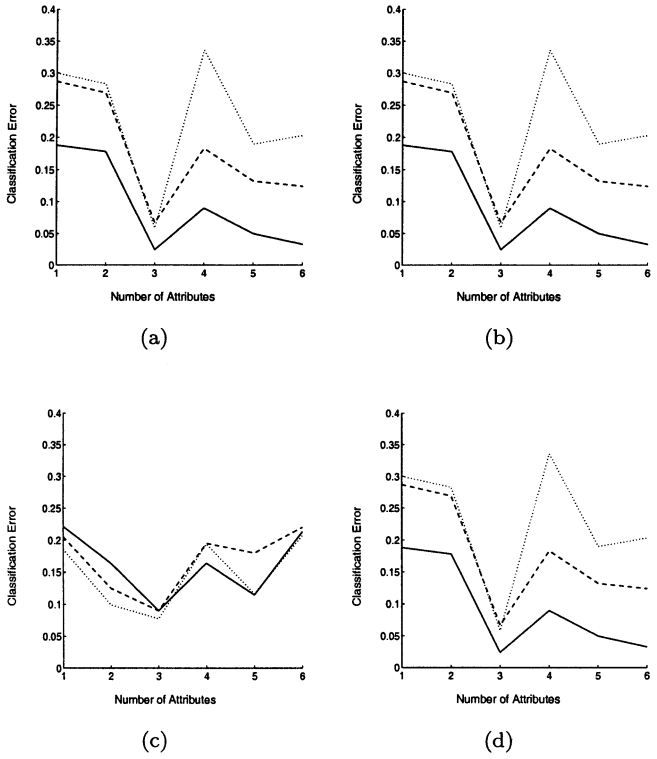

Fig. 4. Same as Fig. 3, Monk3 data set.

0.0688 to 0.067, the number of inputs from 6 to 3, and the number of Gaussian hidden units from 19 to 11 (Table XVIII). In Fig. 4(c) and (d), attributes 2, 4, and 5 should be selected according to both SUD and Relief-F methods which is the same as the attribute subset obtained based on our SCM method.

For Thyroid data set, the attribute ranking queue corresponding to our SCM with bottom-up search is $\{2, 3, 5, 4, 1\}$ and is $\{2, 5, 3, 4, 1\}$ with top-down search. Figs. 5(a)–(b) show that, in both bottom-up and top-down search, attributes 2, 3, and 5 are considered to be relevant for classification and are selected, which decreases the classification error rate from 0.0465 to 0.0233, the number of inputs from 5 to 3, and the number of Gaussian hidden units from 8 to 5 (Table XVIII). It is shown in Fig. 5(c) that attributes 4, 5, 3, and 2 are selected based on the ranking result of the Relief-F method. Fig. 5(d) shows that attributes 4, 3, 1 and 2 should be selected according to SUD. The classification error rates on the test data set when the respective selected attribute subsets are used as inputs for RBF classifiers are 0.0233, 0.093, and 0.1163 for our SCM method, SUD and Relief-F, respectively (Table XIX). Hence the attribute subset based on our SCM method is smaller with higher accuracy compared to the SUD and Relief-F methods.

For Breast cancer data set, the attribute ranking queue corresponding to our SCM with bottom-up search is

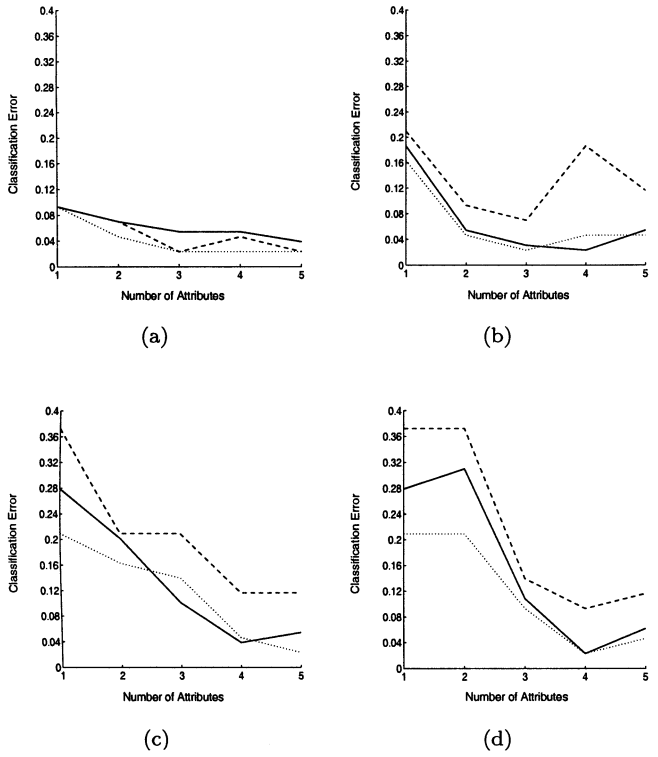

Fig. 5. Same as Fig. 3, Thyroid data set.

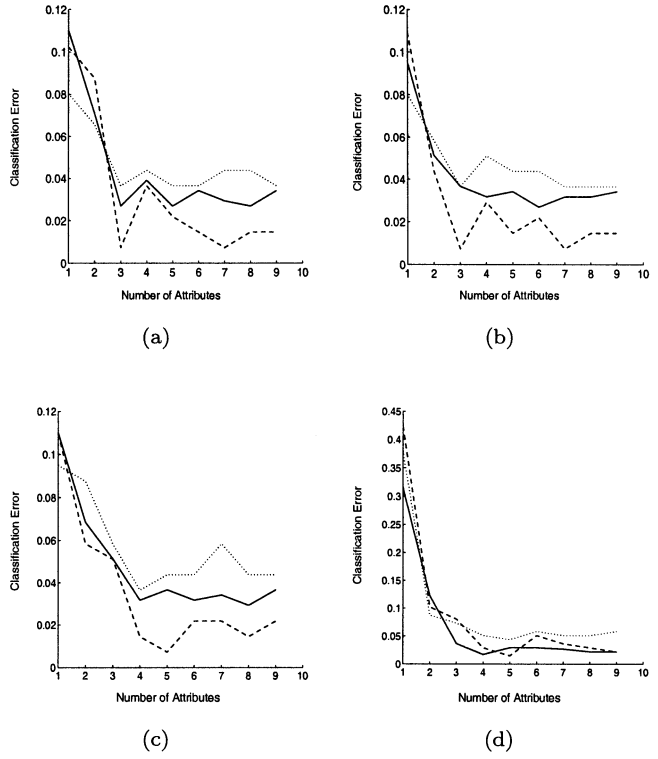

Fig. 6. Same as Fig. 3, Breast cancer data set (note that the scale for classification error rates in (d) is different from those in (a)–(c)).

$\{2, 7, 3, 4, 9, 5, 1, 8, 6\}$ and is $\{7, 2, 3, 4, 9, 5, 8, 6, 1\}$ with top-down search. It is shown in Fig. 6(a)–(b) that, in both bottom-up and top-down search, attributes 2, 3, and 7 are considered to be important for classification and are then selected, which decreases the classification error rate from 0.0146 to 0.0073, the number of inputs from 9 to 3, and the number of hidden units of the RBF neural network from 11 to 5 (Table XVIII). Fig. 6(c) shows that the attribute subset including the first five attributes (attributes 6, 2, 3, 7, and 5) in the Relief-F attribute ranking queue leads to the lowest classification error rates. According to the classification results shown in Fig. 6(d) based on the ranking result of the SUD method, attributes 1, 7, 3, 2, and 5 should be selected because the subset leads to the lowest error rates. The classification error rates on test data set when the selected attribute subsets are used as inputs for RBF classifiers are 0.0073, 0.0146, 0.0073 for our SCM, SUD and Relief-F, respectively (Table XIX). Hence the attribute subset based on our SCM method is the smallest with the highest classification accuracy.

V. Conclusion

In this paper, data dimensionality reduction is carried out in order to improve classification performance and to reduce the number of attributes as well as the complexity of an RBF neural network. The novel SCM is proposed to rank the importance of attributes. According to the ranking results, different attribute subsets are used as inputs to RBF classifiers. The attribute subsets with the lowest classification error rates and the least numbers of attributes are selected. Although attribute importance rankings obtained from the SCM with bottom-up and top-down search may sometimes differ slightly, the same selected attribute subsets are eventually obtained in the four benchmark data sets tested. Compared to existing attribute importance ranking methods, such as SUD [10] and Relief-F [9] methods, the SCM leads to smaller attribute subsets and higher classification accuracies in simulations. We have also proposed a useful modification for the construction and training of the RBF network by allowing for large overlaps among clusters of the same class, which further reduces the number of hidden units while maintaining the classification accuracy. Experimental results show that the proposed methods are effective in reducing the attribute size, the structural complexity of the RBF neural network, and the classification error rates. In future work, we will extract simple and efficient rules from data sets using the simplified RBF neural network classifiers.

Acknowledgment

The authors wish to thank the Associate Editor and the three reviewers, for reading the manuscript and offering numerous constructive comments which helped improve the paper significantly.

References

[1] R. Battiti, "Using mutual information for selecting features in supervised neural net learning," *IEEE Trans. Neural Networks*, vol. 5, pp. 537–550, July 1994.

[2] C. M. Bishop, *Neural Network for Pattern Recognition*. New York: Oxford University Press, 1995.

[3] K. D. Bollacker and J. Ghosh, "Mutual information feature extractors for neural classifiers," in *Proc. IEEE Int. Conf. Neural Networks*, vol. 3, 1996, pp. 1528–1533.

[4] L. H. Chen and S. Chang, "An adaptive learning algorithm for principal component analysis," in *IEEE Trans. Neural Networks*, vol. 6, Sept. 1995, pp. 1255–1263.

[5] P. A. Devijver and J. Kittler, *Pattern Recognition: A Statistical Approach*. London, U.K.: Prentice-Hall, 1982.
[6] N. Kambhatla and T. K. Leen, "Fast nonlinear dimension reduction," in *Proc. IEEE Int. Conf. Neural Network*, vol. 3, 1993, pp. 1213–1218.
[7] T. Kawatani and H. Shimizu, "Handwritten kanji recognition with the LDA method," in *Proc. 14th Int. Conf. Pattern Recognition*, vol. 2, 1998, pp. 1301–1305.
[8] T. Kaylani and S. Dasgupta, "A new method for initializing radial basis function classifiers systems," in *Proc. IEEE Int. Conf. Man, Cybern.*, vol. 3, 1994, pp. 2584–2587.
[9] I. Kononenko, "Estimating attributes: Analysis and extension of RELIEF," in *Proc. Eur. Conf. Machine Learning*, 1994, pp. 171–182.
[10] M. Dash, H. Liu, and J. Yao, "Dimensionality reduction of unsupervised data," in *Proc. 9th IEEE Int. Conf. Tools Artificial Intell.*, 1997, pp. 532–539.
[11] C. J. Liu and H. Wechsler, "Enhanced Fisher linear discriminant models for face recognition," in *Proc. 14th Int. Conf. Pattern Recognition*, vol. 2, 1998, pp. 1368–1372.
[12] P. Maffezzoni and P. Gubian, "Approximate radial basis function neural networks (RBFNN) to learn smooth relations from noisy data," in *Proc. 37th Midwest Symp. Circuits Syst.*, vol. 1, 1994, pp. 553–556.
[13] W. Malina, "Two-parameter Fisher criterion," *IEEE Trans. Syst., Man Cybern. B*, vol. 31, pp. 629–636, Aug. 2001.
[14] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, and K. R. Mullers, "Fisher discriminant analysis with kernels," *Neural Networks Signal Processing IX, 1999*, pp. 41–48, 1999.
[15] J. Moody and C. J. Darken, "Fast learning in network of locally-tuned processing units," *Neural Computat.*, vol. 1, pp. 281–294, 1989.
[16] P. M. Murphy and D. W. Aha. (1994) *UCI Repository of Machine Learning Databases* [Online]. Available: http://www.ics.uci.edu/mlearn/MLRepository.html
[17] I. T. Nabney, "Efficient training of RBF networks for classification," in *Proc. 9th Int. Conf. Artificial Neural Net.*, vol. 1, (Conf. Publ. no. 470), 1999, pp. 210–215.
[18] W. Poechmueller, S. K. Hagamuge, M. Glesner, P. Schweikert, and A. Pfeffermann, "RBF and CBF neural network learning procedures," in *Proc. 1994 IEEE World Congr. Computat. Intell.*, vol. 1, 1994, pp. 407–412.
[19] A. Roy, S. Govil, and R. Miranda, "An algorithm to generate radial basis function (RBF)-like nets for classification problems," *Neural Networks*, vol. 8, no. 2, pp. 179–201, 1995.
[20] ——, "A neural-network learning theory and a polynomial time RBF algorithm," *IEEE Trans. Neural Net.*, vol. 8, pp. 1301–1313, Nov. 1997.

Xiuju Fu received the B.S. degree and the M.S. degree, from Beijing Institute of Technology, Beijing, China, in 1995 and 1999, respectively, and pursued the Ph.D. degree at Nanyang Technological University, Singapore from 1999 to 2002.

She is currently a Research Fellow with the Institute of High Performance Computing, Singapore. She has co-authored more than ten papers in conference proceedings and journals. Her current research areas include: neural networks, genetic algorithms, data mining, classification, data dimensionality reduction, and rule extraction.

Ms. Fu received the Singapore Government Scholarship from 1999 to 2002.

Lipo Wang (SM'98) received the B.S. degree from National University of Defense Technology, China, and the Ph.D. degree from Louisiana State University, Baton Rouge.

Currently, he is Associate Professor at the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. Previously, he was a tenured faculty member in computing at Deakin University, Australia. His research interests are in theory of computational intelligence and its applications to optimization and data mining. He is author or co-author of over 50 journal publications and 60 conference presentations, holds a U.S. patent, and is co-editor of two books and five conference proceedings. He is an Associate Editor/editorial board member of *Soft Computing* and *Knowledge and Information Systems*.

Dr. Wang is an Associate Editor/editorial board member of IEEE TRANSACTIONS ON NEURAL NETWORKS. He has been/is General Chair for four international conferences and has served/is serving on numerous conference committees. He is President of the Asia-Pacific Neural Network Assembly.