Feature Selection Using Radial Basis Function Networks

J. Basak and S. Mitra

Machine Intelligence Unit, Indian Statistical Institute, Calcutta, India

A new method of feature selection using a Radial Basis Function network is described. The parameters of the radial basis function network, in general, form a compact description of class structures. The intraclass and interclass distances are expressed in terms of the parameters of the trained network, and two different feature evaluation indices are derived from these distances. The effectiveness of the algorithm is demonstrated on Iris and speech data, and a comparative study is provided with several existing techniques.

Keywords: Feature evaluation index; Feature selection; Radial Basis Function network

1. Introduction

Feature selection is a task where the optimum salient characteristics necessary for the recognition process are retained, and hence the dimensionality of the measurement space is reduced. Various classical/fuzzy set theoretic methods for feature selection are reported in the literature [1–5].

Artificial Neural Networks (ANNs) have the capability of fault tolerance, adaptivity/generalisation, and scope for massive parallelism. Often, they are employed for dealing with various optimisation tasks. Selecting the optimal subset from a given set of features is one such optimisation problem. Ruck et al. [6] developed a multi-layer perceptron-based (*MLP*-based) algorithm for feature ranking, where the sensitivity of output of the network to its input is used to rank the input features. The methods based on multilayer feedforward networks include the determination of saliency (usefulness) of input features [7], development of Sammon's Nonlinear Discriminant Analysis (NDA) network, the Linear Discriminant Analysis (LDA) network [8], and a neuro-fuzzy approach for evaluating the effect of suppressing certain feature(s) [9]. Investigations have also been made for the development of neuro-fuzzy approaches for supervised feature selection [10,11] and unsupervised feature selection [12]. Those based on self-organising networks include the development of nonlinear projection (NP-SOM) based on Kohonen's self-organising feature map [8], distortion tolerant Gabor transformations followed by minimum distortion clustering by multilayer self-organising maps [13], a non-linear projection method based on Kohonen's topology preserving maps [14].

In this article, a new method is presented for selecting the optimal set of features by examining the parameters of a trained Radial Basis Function (RBF) network [15,16]. The parameters of a RBF network form a compact description of the class structures from the given data set used to train the network. These parameters are used to formulate interclass and intraclass distances. Two different feature evaluation indices are, in turn, computed from these distances. The importance of a set of feature(s) is evaluated by examining the effect of its absence on the evaluation indices.

Section 2 provides a brief description of the RBF network. The feature selection algorithm, including the new evaluation indices, is presented in Section 3. The effectiveness of the algorithm on real-life data is experimentally demonstrated and the results are compared with some of the existing methods in Section 4. Finally, Section 5 concludes the article.

Correspondence and offprint requests to: J. Basak, Machine Intelligence Unit, Indian Statistical Institute, 203 B.T. Road, Calcutta 700 035, India. Email: jayanta@isical.ac.in

2. Radial Basis Function Network

A Radial Basis Function (RBF) network [15,16] consists of two layers, as shown in Fig. 1. The connection weight vectors of the input and output layers are denoted as $\vec{\mu}$ and $\vec{\mathrm{W}}$, respectively. The basis (or kernel) functions in the hidden layer produce a localised response to the input stimulus. The output nodes form a weighted linear combination of the basis functions computed by the hidden nodes.

The input and output nodes correspond to the input features and output classes, while the hidden nodes represent the number of clusters (specified by the user) that partition the input space. Let $\vec{x} = (x_1, \ldots, x_i, \ldots, x_n) \in R^n$ *and* $\vec{y} = (y_1, \ldots, y_i, \ldots, y_l) \in R^l$ be the input and output, respectively, and m the number of hidden nodes.

The output u_j of the jth hidden node, using the *Gaussian kernel* function as a basis, is given by

$$u_j = \exp\left[-\frac{(\vec{x}-\vec{\mu}_j)^T(\vec{x}-\vec{\mu}_j)}{2\sigma_j^2}\right], \quad j = 1, 2, \ldots, m$$

where $\vec{x}$ is the input pattern, $\vec{\mu}_j$ is its input weight vector (i.e. the centre of the Gaussian for node j) and σ_j^2 is the normalisation parameter, such that $0 \leq u_j \leq 1$ (the closer the input is to the centre of the Gaussian, the larger the response of the node).

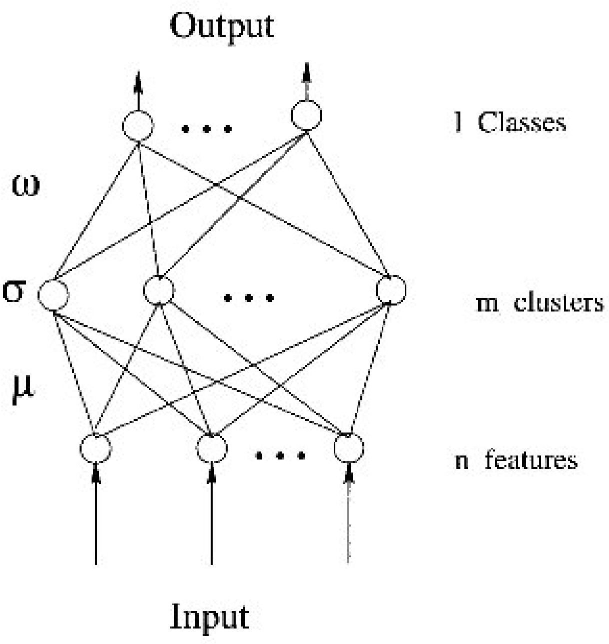

Fig. 1. A Radial Basis Function network consisting of n input nodes representing features, m hidden nodes representing cluster centres, and l output nodes corresponding to classes. Cluster centres (μ) are stored in the links from the input to hidden layer. σ is the normalisation parameter vector of the hidden node activation functions. ω represents the weights of links from the hidden to output layer.

The output y_j of the jth output node is

$$y_j = \vec{W}_j^T \vec{u}, \quad j = 1, 2, \ldots, l \tag{2}$$

where $\vec{W}_j$ is the weight vector for this node, and $\vec{u}$ is the vector of outputs from the hidden layer. The network performs a linear combination of the non linear basis functions of Eq. (1).

The problem is to minimise the error

$$E = \frac{1}{2}\sum_{p=1}^{N}\sum_{j=1}^{l}(y_j^p - {}^*y_j^p)^2 \tag{3}$$

where ${}^*y_j^p$ and y_j^p are desired and computed output at the jth node for the pth pattern, N is the size of the data set, and l is the number of output nodes. In the sequel, the superscript p is omitted for the sake of representation.

Learning in RBF networks can, in general, be performed by two different strategies [17]. A fixed set of cluster centres is first formed by clustering algorithm (e.g. c-means algorithm [1]). The associations of the cluster centres with the output are then learned by squared error minimisation (i.e. minimisation of E (Eq. (3)). Alternatively, the cluster centres can also be learned along with the weights from the hidden layer to the output layer by gradient descent technique. However, learning the centres along with weights may lead to some locally fixed points in the error space, thus leading to a deviation from the desired result.

Here a fixed set of cluster centres is formed by the c-means algorithm [1]. Let the cluster centres, so determined, be denoted as $\vec{\mu}_j$, $j = 1, \ldots, m$. The normalisation parameter σ_j represents a measure of the spread of data associated with each node.

Learning in the output layer is performed after the parameters of the basis functions have been determined. The weights are typically trained using the Least Mean Squares algorithm given by

$$\Delta W_j = -\eta e_j \vec{u} \tag{4}$$

where $e_j = y_j - {}^*y_j$ and η is the learning rate.

3. Feature Selection

In this section, we describe how the RBF network can be used for feature selection. An evaluation index is formulated for representing the importance of different features and their combinations. Based on this evaluation index, the features/sets of features can be selected. It is assumed that the cluster centres ($\vec{\mu}$) are the most representative points of clusters identified by the c-means algorithm, and are used in the subsequent training of RBF network. The intercluster distances can therefore be directly computed from these cluster centres.

During training of the RBF network, each class is viewed as a collection of clusters. The contribution of a cluster to any class is modulated by the weights of the links from the hidden layer to the output layer. The magnitude of these second layer weights provide a measure of the importance of a cluster with respect to a class. Thus, the interclass and intraclass distances can be logically computed from the identified cluster centres and the weights of the corresponding links from the hidden to the output layer. To represent the relative importance of the clusters with respect to classes, we have used normalised absolute values of the second layer weights given as

$$\omega_{ck} = \frac{|W_{ck}| - W_{min}}{W_{max} - W_{min}} \tag{5}$$

where

$$W_{max} = \max_{\forall c,k} \{|W_{ck}|\}$$

and

$$W_{min} = \min_{\forall c,k} \{|W_{ck}|\}.$$

The cluster and class indices are represented by c and k, respectively. The greater the value of ω_{ck}, the greater is the importance of cluster c with respect to class k. Thus, larger values of ω's should contribute more to the interclass and intraclass distances. Let us now provide the feature evaluation indices using normalised absolute link weights and variances (σ) stored in the hidden nodes.

3.1. Evaluation Index I

The variance in the input data (σ) stored in each hidden node measures the sparseness of the cluster represented by the hidden node. It is assumed that the sparser a cluster is, the less should it contribute to the class distances computed from the cluster centres. Therefore, the cluster distances are weighted by $1/\sigma$ in order to obtain the class distances.

The feature evaluation index for a feature subset $\mathscr{F}(FEI)$ is given by

$$FEI = \sum_{k} \frac{d_k}{D_k} \tag{6}$$

where d_k represents the compactness of class k and D_k provides a measure of the distance of class k from all other classes, ignoring the feature subset $\mathscr{F}$. These are mathematically expressed as

$$d_k = \sum_{c} \sum_{c^{'}>c} \sum_{\forall j \notin \mathscr{F}} \frac{(\mu_{jc} - \mu_{jc^{'}})^2}{\sigma_c \,.\, \sigma_{c^{'}}} \,.\, \omega_{ck}.\omega_{c^{'}k} \tag{7}$$

and

$$D_k = \sum_{k^{'} \neq k} \sum_{c} \sum_{c^{'}>c} \sum_{\forall j \notin \mathscr{F}} \frac{(\mu_{jc} - \mu_{jc^{'}})^2}{\sigma_c . \sigma_{c^{'}}} \,.\, \omega_{ck}.\omega_{c^{'}k^{'}} \tag{8}$$

The feature evaluation index measures how the ratio of the intraclass and interclass distances gets affected when a particular feature is ignored. If the classes are well separated and compact, then the ratio of intraclass and interclass distances will be small. If the exclusion of some feature subset deteriorates the compactness of the classes (i.e. increases d_k) and/or decreases the separation between classes (i.e. decreases D_k), then it should be treated as an important feature. This will be reflected in the index FEI, because in that case the ratio will increase.

3.2. Evaluation Index II

Here we provide an alternate measure of evaluation index for the features. Let a pattern p in cluster c_1 and a pattern q in cluster c_2 be denoted by $\vec{x_1^p} = x_{11}^p, x_{12}^p, \ldots, x_{1n}^p$ and $\vec{x_2^q} = x_{21}^q, x_{22}^q, \ldots, x_{2n}^q$, respectively. Let the distance $\mathscr{D}(c_1, c_2)$ between two clusters c_1 and c_2 be the summation of the pairwise distances of the points in these two clusters, i.e.

$$\mathscr{D}(c_1, c_2) = \sum_{p} \sum_{q} d(\vec{x}_1^p, \vec{x}_2^q) \tag{9}$$

where $d(.)$ is the physical distance between two points in the pattern space. If $d(.)$ corresponds to the squared Euclidean distance, then

$$\mathscr{D}(c_1, c_2) = \sum_{p} \sum_{q} \sum_{i} (x_{1i}^p - x_{2i}^q)^2 \tag{10}$$

After simplification, the distance between two clusters can be written as

$$\mathscr{D}(c_1, c_2) = N_1 N_2 \,(\sigma_{c_1}^2 + \sigma_{c_2}^2 + d(\vec{\mu}_{c_1}, \vec{\mu}_{c_2})) \tag{11}$$

where N_1 and N_2 are the number of patterns present in the clusters c_1 and c_2, respectively.

Now, if the number of points in some cluster is greater, the weight of the link connected to the hidden node corresponding to the cluster will be iterated for a larger number of trials during training of the network, provided all the classes are equally likely. In that case, it is expected that the value of

ω (Eq. (5)) corresponding to that link will be more. In other words, the greater the number of points present in a cluster, the larger will be the importance of the link connected to the hidden node corresponding to the cluster, and *vice versa*. Based on this concept, and Eq. (11), an alternative measure for intraclass and interclass distances is given as

$$d_k^{'} = \sum_{c}\sum_{c^{'}>c}\left(\sum_{\forall_j \notin \mathcal{F}} (\mu_{jc} - \mu_{jc^{'}})^2 + \sigma_c^2 + \sigma_{c^{'}}^2\right) \omega_{ck}.\omega_{c^{'}k} \tag{12}$$

and

$$D_k^{'} = \sum_{k^{'}\neq k}\sum_{c}\sum_{c^{'}>c}\left(\sum_{\forall_j \notin \mathcal{F}} (\mu_{jc} - \mu_{jc^{'}})^2 + \sigma_c^2 + \sigma_{c^{'}}^2\right) \omega_{ck}.\omega_{c^{'}k^{'}} \tag{13}$$

The altemative measure for the feature evaluation index is given as

$$FEI^{'} = \sum_{k} \frac{d_k^{'}}{D_k^{'}} \tag{14}$$

3.3. Method of Feature Selection

The evaluation indices are computed based on the weighted distance between the clusters represented by the hidden nodes. The number of clusters should be such that the classification performance of the network reaches the optimum value. At the same time, a very large number of clusters may lead to undesirable redundancy in the network. Here we consider RBF networks with a minimum number of hidden nodes, such that the classification performance does not deteriorate.

It is assumed that the required number of clusters (m) cannot be less than the number of classes (l). This is because if these are equal, then ideally an individual cluster will correspond to an individual class. Therefore, to find out the RBF network with a minimum number of hidden nodes, we start with a network having $m = l$. Then m is increased until there is no significant increase in the classification performance. *FEI* and *FEI'* are computed with this network, and the feature subsets are ranked accordingly. This method is algorithmically described below.

Step 1: Select an RBF network with a number of hidden nodes equal to the number of classes.

Step 2: Train the network and test the classification performance.

Step 3: Increase the number of hidden nodes, and repeat step 2 until the classification performance does not improve significantly.

Step 4: Compute normalised absolute values of link weights of the trained RBF obtained in step 3.

Step 5: Compute *FEI* and *FEI'* for each feature subset according to Eqs (6) and (14).

Step 6: Rank the feature subsets according to the values of the evaluation indices.

4. Experimental Results

The algorithm was implemented on real data, *viz. Iris* and *Vowel*. The results obtained by the proposed algorithm (model BM) are compared with some of the existing techniques, considered as benchmarks in this study. These are (i) the statistical method of Devijver and Kittler [2] (model DK), (ii) the fuzzy entropy-based method of Pal and Chakraborty [3] (model PC), (iii) the neural network-based method of Ruck et al. (model R*), and (iv) that of Ishibuchi [9] (model IM).

The *Iris* data [18] consists of 150 pattern points with four input features corresponding to measurements of *sepal length, sepal width, petal length, petal width* on 50 flowers from each of three species *setosa, versicolor, virginica* represented by the three output classes. The RBF network used to learn this data set, therefore, consists of four input and three output nodes. The choice of eight hidden nodes was found to yield the best result after several trials.

The speech data *Vowel* [19] deals with 871 Indian Telugu vowel sounds. These were uttered in a Consonant-Vowel-Consonant context by three male speakers in the age group of 30 to 35 years. It has three features corresponding to the first, second and third vowel formant frequencies obtained through spectrum analysis of the speech data. The data contains six vowel classes – ∂, *a*, *i*, *u*, *e*, *o*. Figure 2 shows the data in the first and second feature

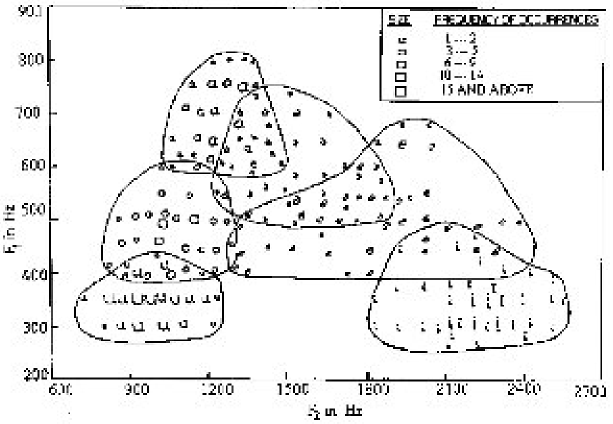

Fig. 2. Vowel data.

Table 1. Ranking for *Iris* data.

Feature set	*FEI*	Rank	Overall rank	*FEI′*	Rank	Overall rank
1	0.618	4	11	0.702	4	11
2	0.649	3	9	0.722	3	9
3	0.777	1	4	0.908	1	4
4	0.667	2	7	0.744	2	8
1, 2	0.604	6	13	0.669	6	13
1, 3	0.689	3	6	0.879	2	5
1, 4	0.621	5	10	0.692	5	12
2, 3	0.763	2	5	0.868	3	6
2, 4	0.657	4	8	0.717	4	10
3, 4	0.905	1	3	1.01	1	2
1, 2, 3	0.598	4	14	0.758	3	7
1, 2, 4	0.606	3	12	0.65	4	14
1, 3, 4	1.793	1	1	1.254	1	1
2, 3, 4	0.916	2	2	0.981	2	3

Table 2. Ranking for *Vowel* data.

Feature set	*FEI*	Rank	Overall rank	*FEI′*	Rank	Overall rank
1	0.891	2	4	0.932	2	4
2	1.029	1	2	1.08	1	2
3	0.847	3	5	0.903	3	5
1, 2	1.096	1	1	1.097	1	1
1, 3	0.839	3	6	0.878	3	6
2, 3	0.94	2	3	1.068	2	3

plane, for ease of depiction. The RBF network in this case consists of three input, six output and twelve hidden nodes.

Table 1 illustrates the ranking of features, based on the feature evaluation indices *FEI* (Eq. (6)) and *FEI′* (Eq. (14)). It is observed that feature 3 is most important, followed by feature 4, when considered individually, using both the evaluation indices. Combining pairs of features, it is found that the first three significant pairs are (3, 4), (2, 3) and (1, 3) when *FEI* is used. When *FEI′* is used, the ranking becomes (3, 4), (1, 3), (2, 3). Considering triplets, the set (1, 3, 4) is found to be most important using both *FEI* and *FEI′*.

Considering all possible subsets, (1, 3, 4) is found to be most important using both the evaluation indices. The next three in overall ranking are (2, 3, 4), (3, 4) and (3). It is observed from Table 1 that in many cases an individual feature has higher ranking than its supersets. For example, feature 3 is found to be more important than the subsets (1, 3), (2, 3) and (1, 2, 3). This indicates that the addition of extra (redundant) features may deteriorate the performance of a classifier.

Table 2 provides a similar study for *Vowel* data. The individual and pairwise ranking of the features are found to be the same with *FEI* and *FEI′*. Feature 2 is found to be most significant, followed by feature 1. Considering feature pairs, the combination (1, 2) is observed to be most important. It is also found, from the overall ranking, that (1, 2) is most important among all possible subsets, followed by (2) and (2, 3).

Table 3 demonstrates a comparative study of the individual feature orderings generated by different algorithms for the two data sets. Since the individual feature ordering is the same for both the evaluation indices (*FEI* and *FEI′*), they are referred in a single row of model *BM*. As *Iris* data is typically studied by researchers (in the pattern recognition field), an extensive comparison has been provided for this data. The overall study shows that the results tally with each other. The features 4 and 3 are found to be more important than the features 1 and 2 for

Table 3. Comparative study.

Data	Algorithm	Feature ordering
Iris	BM	3, 4, 2, 1
	DK	3, 4, 1, 2
	PC	4, 3, 1, 2
	IM	3, 4, 2, 1
	R*	3, 4, 2, 1
Vowel	BM	2, 1, 3
	PC	2, 1, 3

classifying *Iris* data. The class structures in the *Vowel* data, on the other hand, are highly overlapping and ambiguous (as observed from Fig. 2) and not so typical (like *Iris* data). Hence its comparison has been made only with model PC. It is observed that features 2 and 1 are most important for classifying *Vowel* data. This information tallies with the experts' opinion [19].

5. Conclusions

Two new evaluation indices for selecting optimal set of features are described. The evaluation indices for a feature subset are derived from the effect (deterioration) on the separation between and/or compactness of the classes due to the absence of the feature set. The interclass and intraclass distances, representing the separation between and compactness of classes, respectively, are computed directly from a trained radial basis function network with minimum redundancy. The method of feature selection is independent of the testing phase of the network, because once the network is trained, the indices are computed directly from the parameters of the network. The effectiveness of the evaluation indices is tested on a couple of real-life data. The performance is also compared with the existing methods and found to provide desired feature rankings.

Acknowledgements. The authors are grateful to Mr R.K. De for helpful suggestions. The technical and programming help rendered by MTech. (computer science) students is highly acknowledged.

References

1. Tou JT, Gonzalez RC. Pattern Recognition Principles. Addison-Wesley, London, 1974
2. Devijver PA, Kittler J. Pattern Recognition, A Statistical Approach. Prentice-Hall, London, 1982
3. Pal SK, Chakraborty B. Fuzzy set theoretic measure for automatic feature evaluation. IEEE Trans Systems, Man and Cybernetics 1986; 16: 754–760
4. Pal SK. Fuzzy set theoretic measures for automatic feature evaluation: II. Information Sciences, 1992; 64: 165–179
5. Bezdek JC, Castelaz PF. Prototype classification and feature selection with fuzzy sets. IEEE Trans Systems, Man and Cybernetics 1977; 7: 87–92
6. Ruck DW, Rogers SK, Kabrisky M. Feature selection using a multilayer perceptron. Neural Network Computing 1990; 20: 40–48
7. Priddy KL, Rogers SK, Ruck DW, Tarr GL, Kabrisky M. Bayesian selection of important features for feedforward neural networks. Neurocomputing 1993; 5: 91–103
8. Mao J, Jain AK. Artificial neural networks for feature extraction and multivariate data projection. IEEE Trans Neural Networks 1995; 6: 296–317
9. Ishibuchi H, Miyazaki A. Determination of inspection order for clasifying new samples by neural networks. Proceedings of IEEE International Conference on Neural Networks, Orlando, USA, 1994; 2907–2910
10. Pal SK, Basak J, De RK. Fuzzy feature evaluation index and connectionist realisation. Information Sciences 1998; 105: 173–188
11. Basak J, De RK, Pal SK. Fuzzy feature evaluation index and connectionist realisation – II: Theoretical analysis. Information Sciences 1998; 111: 1–17
12. Basak J, De RK, Pal SK. Unsupervised feature selection using neurofuzzy approach. Pattern Recognition Letters 1998; 19: 997–1006
13. Lampinen J, Oja E. Distortion tolerant pattern recognition based on self-organizing feature extraction. IEEE Trans Neural Networks 1995; 6: 539–547
14. Kraaijveld MA, Mao J, Jain AK. A non-linear projection method based on Kohonen's topology preserving maps. IEEE Trans Neural Networks 1995; 6: 548–559
15. Moody J, Darken CJ. Fast learning in networks of locally-tuned processing units. Neural Computation 1989; 1: 281–294
16. Hush DR, Horne BG. Progress in supervised neural networks. IEEE Signal Processing Magazine, January 1993; 8–39
17. Haykin S. Neural Networks: A Comprehensive Founation. Macmillan, New York, 1994
18. Fisher RA. The use of multiple measurements in taxonomic problem. Annals of Eugenics 1936; 7: 179–188
19. Pal SK, Dutta Majumder D. Fuzzy sets and decision making approaches in vowel and speaker recognition. IEEE Trans Systems, Man and Cybernetics 1977; 7: 625–629