Data Dimensionality Reduction Using RBF Neural Networks

Xiuju Fu and *Lipo Wang**

School of Electrical and Electronic Engineering
Nanyang Technological University
Block S2, Nanyang Avenue
Singapore 639798
Email: {p146793114,elpwang}@ntu.edu.sg
http://www.ntu.edu.sg/home/elpwang
*Corresponding author

Abstract

In most cases, there are redundant or irrelevant attributes in data sets. Hence, it is desirable to remove the redundant or irrelevant attributes from the data sets, which can facilitate practical applications in improving speed and relieving memory constraints. In this paper, a novel method, a separability-correlation measure (SCM), is proposed to rank the importance of attributes. An RBF classifier is applied to evaluate the best subset of attributes which should be retained. Different subset of attributes are input to the RBF classifier. Those attributes that increase the validation error are deemed unimportant or irrelevant and are deleted. The complexity of the RBF classifier is thus reduced and the classification performance improved.

1 Introduction

Data dimensionality reduction (DDR) is the process to reduce the number of attributes while maintaining the concept of data sets. DDR has become an important aspect of pattern recognition, since operators and automated controllers are able to make better use of lower-dimensional data compared to higher-dimensional ones. Computation burden can be reduced in automated processes by DDR, for example, when constructing a radial basis function (RBF) neural network to classify data. Reduced data dimensionality leads to less complicated network structure and thus increased efficiency in processing data.

As an important task of pattern recognition, DDR maps high-dimensional patterns onto lower-dimensional patterns. Many techniques have been proposed for reducing data dimensionality. In general, techniques for DDR may be classified into two categories: feature extraction and feature selection. Feature extraction creates a number of new features through a transformation of the raw features. Linear Discriminant Analysis (LDA) [7] and Principal Components Analysis (PCA) [5] are two popular techniques for feature extraction. It is very difficult to prevent by-products from affecting detrimentally the original concepts in the data, though the transformations are designed to maintain concepts in the data.

To some extent, feature selection is more desirable than feature extraction since it does not generate new features or unwanted by-products. Feature selection techniques try to select the best subset of features out of the original set. The attributes which are considered to be important for maintaining the concepts in the original data are sifted from the entire attribute set. Thus, the importance level of each attribute becomes the key to feature selection. Mutual information based feature selection (MIFS) [1][7] 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 level of features are ranked based on their ability to maximizing the evaluation formula. However, in MIFS, the number of features to be selected need to be predefined.

In feature selection algorithms, there are two basic categories. The first is the filter approach [3][16] which

sieves a suitable feature subset based on a fitness criterion, such as the inconsistency between the feature subset with class labels. The second is the wrapper approach [9][10]. In the wrapper approach, feature selection is wrapped in the induction algorithm. The feature subset is selected during the reasoning process of an induction algorithm. In [10], the importance factor of each input feature of a multi-layer perceptron (MLP) neural network is determined by the weighted connections between the input and the second layer of the MLP during training. The features with importance factors below a certain level are eliminated.

The difference of the two algorithms lies in whether or not the feature selection is carried out independently of induction algorithms. Sometimes, the filter approach can not efficiently remove the irrelevant features because it totally ignores the effect of the selected feature subset on the performance of induction algorithms. The wrapper approach can be time consuming especially for those induction algorithms that are computationally intensive, such as neural networks. [16] combines the filter and the wrapper approaches to reduce time complexity and improve classification accuracy.

In this paper, a separability-correlation measure (SCM) is proposed for determining the importance of the original attributes. The SCM is composed of the intraclass distance to interclass distance ratio and an attribute-class correlation measure. The magnitude SCM corresponding to a certain attribute gives the importance of the attribute. An RBF classifier is generated to remove unimportant attributes using the ranking result from SCM.

We focus on reducing the dimensionality of data and simplifying the structure of RBF neural networks in this paper. The SCM measure for ranking the importance of attributes is proposed in Section 2. Section 3 introduces how to construct a RBF neural network classifier efficiently. Experimental results on obtaining a simpler architecture of the RBF classifier is shown in Section 4. Finally, we conclude the paper in Section 5.

2 Separability-Correlation Measure

Our attribute importance ranking is based on the class separability and the correlation between attributes and class labels. Class separability may be measured by the intraclass distance S_w and the interclass distance S_b [4]. The greater S_b is and the smaller S_w is, the better the separability of the data set is. The ratio of S_w and S_b is calculated and is used to measure the separability of the classes: the smaller the ratio, the better the separability. If omitting 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, attribute k_1 is more important for classification of the data set than attribute k_2, and vice versa. Hence the importance of the attributes can be ranked by computing the intraclass-to-interclass distance ratio with each attribute omitted in turn.

The correlation between the changes in attributes and their corresponding changes in class labels is considered while ranking the importance of attributes. We propose the following correlation between the k-th attribute and the class labels in the data set :

$$C_k = \sum_{i \neq j} |x_{ik} - x_{jk}| \cdot \text{sign}|y_i - y_j| \quad , \qquad (1)$$

where for any y, sign$|y| = 1$ if $|y| > 0$ and sign$|y| = 0$ if $|y| = 0$. x_{ik} and x_{jk} are the k-th attribute 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. A great magnitude of C_k shows that there are close correlation between class labels and the k-th attribute, which indicates great importance of attribute k in classifying the patterns, and vice versa.

We propose a separability-correlation measure (SCM) R_k as the sum of the class separability measure S_{wk}/S_{bk} and the correlation measure C_k (k refers to the k-th attribute), where S_{wk} and S_{bk} are intraclass and interclass distances calculated with the k-th attribute omitted from each pattern, respectively.

The importance level of attributes is ranked through the value 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 4.

3 A Simplified RBF Classifier

RBF neural networks are popular in various fields due to its simple architecture. Usually, there are three layers in the RBF neural network, the input layer, i.e., the hidden layer with Gaussian activation functions and the output layer. In this paper, we use the RBF network for classification. If there is M classes in the dataset, we write the m-th output of the network as follows:

$$y_m(\mathbf{X}) = \sum_{j=1}^{K} w_{mj} \emptyset_j(\mathbf{X}) + w_{m0} b_m \quad . \qquad (2)$$

Here $\mathbf{X}$ is the input pattern vector (n-dimension). m=1, 2, ..., 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. $\phi_j(\mathbf{X})$ is the activation function of the j-th hidden unit:

$$\phi_j(\mathbf{X}) = \mathrm{e}^{\frac{||\mathbf{X}-\mathbf{C_j}||^2}{2\sigma_\mathbf{j}^2}} \quad , \tag{3}$$

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

In the RBF neural network, the output of a hidden unit is based on 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 calculated by the linear least square (LLS) method [2], which is fast and free of local minima, in contrast to the multilayer perceptron neural network.

Once the centers, widths, and the weights are determined, the architecture of an RBF network is fixed. Both the dimensionality and the distribution of the input patterns affect the number of the hidden units. Dimensionality reduction will lead to the reduction in the number of hidden units.

[8] shows that overlapped receptive fields of different clusters can improve the performance of the RBF classifier in rejecting noise when tackling with noisy data. In [13] and [6], overlapping Gaussian kernel functions are created to map out the territory of each cluster with a less number of Gaussians. Small overlaps between the Gaussians for different classes are measured by the ratio between the number of the in-class patterns and the number of the out-class patterns in this cluster. A pre-defined value θ is set, i.e., the ratio should be not less than θ to guarantee a small classification error rate for the training data set (this is θ-criterion). When small overlaps among clusters for *different classes* are permitted, small overlaps among clusters for the *same class* may exist.

In this paper, by allowing for large overlaps between clusters for *the same class*, we can further reduce the number of clusters substantially. This will lead to a simplified RBF neural network and will be demonstrated by computer simulations in the next section.

Through the algorithm described in Section 2, the importance level of each attribute is obtained. The subset of attributes which includes k most important attributes is input to the RBF classifier, and for $k = 1, 2, ..., N$, classification error rate is calculated for each k. For small k, classification error decreases as k increases until all important attributes are included.

Table 1: Reduction in the number of hidden units for Iris data set

Comparisons	Small overlap	Large overlap
Error rate in classification	0.0373	0.0467
Number of hidden units	5.2	4

Table 2: Comparison of the number of hidden-units before and after the irrelevant attributes are removed for Iris data set

Input attributes	Before removal	4,3,1,2
	After removal	4,3
Number of hidden units	Before removal	4
	After removal	3
Classification error rate	Before removal	0.0467
	After removal	0.0333

If for some k_1, the classification rate for $k = k_1 + 1$ is greater than that for $k = k_1$, then attributes $(k_1 + 1)$, $(k_1 + 2), ..., N$ are considered irrelevant.

4 Computer Simulations

Iris data set is used for testing our method. There are 4 attributes in Iris data set. The data set is divided into 3 parts, i.e., training, validation, and test sets. 150 patterns of Iris data set is divided into 50 patterns for each set. We set $\alpha = 0.1$ and $\theta = 7$ in our experiments. The experiment is repeated 5 times with different initial conditions and the average results are recorded.

Table 1 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.

The rank of importance of the attributes according to our SCM is: 4, 3, 1, 2 for Iris data set. Table 2 shows the classification error rate of the RBF classifier for various subsets of attributes in the order of importance. We see from Table 2 that as the number of attributes used increases the validation error first decreases, reaches minimum when 2 attributes are used, and then increases. Hence in the Iris data set,

attributes 1 and 2 are irrelevant attributes for classification and are then removed. This improves the classification performance and decreases the number of inputs and the number of hidden units of the RBF neural network. Table 2 summarizes the advantages of removing irrelevant attributes.

5 Conclusions

In this paper, a SCM is proposed to rank the importance of attributes. Unimportant attributes are removed from the inputs to the RBF classifier according to the ranking results from SCM. Iris data set is used to test the method. Experimental results show that the method proposed is effective in reducing the size of data sets and reducing the structural complexity of the RBF neural network. The RBF neural network combined with the SCM saves time for DDR for that only a limited number of candidate subsets need to be checked. We also proposed a useful modification to train the RBF network by allowing for large overlaps among clusters of the same class, which further reduces the number of hidden units while retaining the classification performance.

References

[1] Battiti, R., "Using mutual information for selecting features in supervised neural net learning", IEEE Transactions on Neural Networks, Vol.54, pp.537-550, July 1994.

[2] Bishop, C.M. , *Neural network for pattern recognition*, Oxford University Press Inc., New York, 1995.

[3] Chaikla, N. and Qi, Y.L., "Genetic algorithms in feature selection", 1999 IEEE International Conference on Systems, Man, and Cybernetics, 1999, Vol.5, pp.538-540.

[4] Devijver, P. A. and Kittler, J., *Pattern recognition: a statistical approach*, Prentice-Hall International, Inc. London, 1982.

[5] Kambhatla, N. and Leen, T. K., "Fast non-linear dimension reduction", IEEE International Conference on Neural Network, Vol.3, pp.1213-1218, 1993.

[6] Kaylani, T. and Dasgupta, S., "A new method for initializing radial basis function classifiers systems", IEEE International Conference on Man, and Cybernetics, Vol.3, pp.2584-2587, 1994.

[7] Kurt, B.D. and Ghosh, J., "Mutual information feature extractors for neural classifiers ", IEEE International Conference on Neural Networks, Vol.3, pp.1528-1533, 1996.

[8] Maffezzoni, P. and Gubian, P., "Approximate radial basis function neural networks(RBFNN) to learn smooth relations from noisy data", Proceedings of the 37th Midwest Symposium on Circuits and Systems, Vol.1, pp.553 -556, 1994.

[9] Mao, J.C., Mohiuddin, K. and Jain, A. K., "Parsimonious network design and feature selection through node pruning", Proceedings of the 12th IAPR International. Conference on Pattern Recognition, Vol.2, pp.622-624, 1994.

[10] Matecki, U. and Sperschneider, V., "Automated feature selection for MLP networks in SAR image classification", Sixth International Conference on Image Processing and Its Applications, Vol.2, pp.676 - 679, 1997.

[11] Moody, J. and Darken, C. J., "Fast learning in network of locally-tuned processing units", Neural computation, Vol.1, pp.281-294, 1989.

[12] Poechmueller, W., Hagamuge, S. K., Glesner, M., Schweikert, P. and Pfeffermann, A., "RBF and CBF neural network learning procedures", 1994 IEEE World Congress on Computational Intelligence, Vol.1, pp.407-412, 1994.

[13] Roy, A., Govil, S., and Miranda, R., "An algorithm to generate radial basis function (RBF)-like nets for classification problems", Neural networks, Vol.8, No.2, pp.179-201, 1995.

[14] Roy, A., Govil, S., and Miranda R., "A neural-network learning theory and a polynomial time RBF algorithm", IEEE Transactions on neural network, Vol.8, No.6, pp.1301-1313, November 1997.

[15] Saha, A. and Wu, C.L., "Approximation, dimension reduction, and nonconvex optimization using linear superpositions of gaussians", IEEE Transactions On Computers, Vol.42, No.10, pp.1222-1233, OCT 1993.

[16] Yuan, H., Tseng, S.S., Wu, G.S. and Zhang, F.Y., "A two-phase feature selection method using both filter and wrapper", IEEE International Conference on Systems, Man, and Cybernetics, 1999, vol. 2, pp.132-136.