DATA CLASSIFICATION WITH NEURAL CLASSIFIER USING RADIAL BASIS FUNCTION WITH DATA REDUCTION USING HIERARCHICAL CLUSTERING

M. Safish Mary\(^1\) and V. Joseph Raj\(^2\)

\(^1\)Department of Computer Science, St. Xavier’s College (Autonomous), India
E-mail: safish@rediffmail.com

\(^2\)Department of Computer Science, Kamaraj College, India
E-mail: v.jose08@gmail.com

Abstract
Classification of large amount of data is a time consuming process but crucial for analysis and decision making. Radial Basis Function networks are widely used for classification and regression analysis. In this paper, we have studied the performance of RBF neural networks to classify the sales of cars based on the demand, using kernel density estimation algorithm which produces classification accuracy comparable to data classification accuracy provided by support vector machines. In this paper, we have proposed a new instance based data selection method where redundant instances are removed with help of a threshold thus improving the time complexity with improved classification accuracy. The instance based selection of the data set will help reduce the number of clusters formed thereby reduces the number of centers considered for building the RBF network. Further the efficiency of the training is improved by applying a hierarchical clustering technique to reduce the number of clusters formed at every step. The paper explains the algorithm used for classification and for conditioning the data. It also explains the complexities involved in classification of sales data for analysis and decision-making.

Keywords:
Radial Basis Function Neural Network, Gradient Descent, Spherical Gaussian Function, Feature Extraction, Instance-based Data Selection

1. INTRODUCTION

In finance and business, analysts are faced with the problem of classifying large volumes of data. Though analysis of huge volumes of data is a very complex and time consuming process, it is an important task to be completed for effective decision-making. Neural Networks are a proven, widely used technology for solving such complex classification problems. Neural networks are interconnected networks of independent processors termed as neurons that, by changing their connection weights (i.e., training), provide solution to a problem.

Radial Basis Function (RBF) neural networks emerged as a variant of artificial neural network in late 80’s. RBF networks have become one of the most used feedforward classifier for regression, classification and function approximation applications [1, 2]. RBF’s are embedded in a two layer neural network, where each hidden unit implements a radial activated function. The output units implement a weighted sum of hidden unit outputs. The input into an RBF network is nonlinear while the output is linear. Their excellent approximation capabilities have been studied in [3, 4]. Due to their nonlinear approximation properties, RBF networks are able to model complex mappings, which the perceptron neural networks can only model by means of multiple intermediary layers.

The RBF’s are characterized by their localization and Gaussian activation function using supervised (gradient-based) procedures to obtain the expected result [5]. In a supervised application, the network is provided with a set of data samples called training set for which the corresponding outputs are known. After training the network, to produce expected result, it is tested with another set of data samples called test set, to check whether the classifier has learnt to classify the given data effectively.

In this paper, we have used Radial Basis Function network to classify the sales data obtained from an automobile store. The samples are grouped into two sets: one for training the network and the other set for testing the learning capability of the network. The data is classified into three classes as high sales products, moderate sales products and poor sales products. This classification will help the business people to make a decision in purchasing and stocking the items in the store.

This paper is structured as follows: in section 2 we explain the network topology, in section 3 we explain the training algorithm used for classification. In section 4 we provide some experimental study of the application of RBF network to classify sales data and the proposed method for instance reduction to improve classification accuracy and the conclusions of this study are given in section 5.

2. NETWORK TOPOLOGY

Radial basis functions are embedded into a two-layer feedforward neural network. Such a network is characterized by a set of inputs and a set of outputs. In between the inputs and outputs there is a layer of processing units called hidden units. Each of them implements a radial basis function. In this study the inputs represent the feature entries and the output corresponds to a class as shown in Fig.1. The hidden units correspond to subclasses in the neural network. The number of hidden units determines the classification accuracy of the network. The determination of number of hidden units is usually done using k-means clustering method.

In the given architecture, there are \(p\) input vectors, \(k\) hidden layer units and \(q\) output units each unit corresponding to a class. The input of each RBF hidden unit is the linear combination of the input vector \(X = [x_1, x_2, \ldots, x_p]^T\) and the scalar weights between an input layer and the hidden layer which is usually a unitary value. In the hidden layer, each hidden unit computes the activation of the weight vector \(c_j\) associated with the \(j^{th}\) hidden unit (represented by the \(j^{th}\) column of a weight matrix \(C\)) and applies a radial symmetric output function \(f\) (typically a Gaussian function) to \(X_j\),
3. PROPOSED LEARNING ALGORITHM

3.1 PROPOSED LEARNING ALGORITHM

3.1.1 Initialization:

The number of centers determines the number of hidden layer nodes. The training samples are analyzed mathematically to determine the probability that a training sample lies in a class- is determined and the neighboring class- samples are evenly spaced by a distance which is the average distance between two adjacent class- training samples with respect to the sample .

3.1.2 Learning:

Select one input vector and the output class . Then each node in the hidden layer computes the activation of the input vector .

The activation function used at the hidden node is known as spherical Gaussian function approximation and is as follows:

\[
\hat{f}_j(v) = \sum_{s_i \in s_j} w_i \exp\left[-\frac{(v - s_i)^2}{2\sigma_i^2}\right]
\]

where,

\[
\hat{f}_j(v)\text{ is the spherical Gaussian function approximator for class- training samples}
\]

\[
v \text{ is the input vector}
\]

\[
s_j \text{ is the set of class- training samples}
\]

\[
\|v - s_i\| \text{ is the distance between vectors } v \text{ and } s_i
\]

\[
w_i \text{ and } \sigma_i \text{ are parameters to be set by the learning algorithm}
\]

Compute,

\[
\sigma_i = \beta \frac{\overline{R}(s_i)\sqrt{\pi}}{m(k_i + 1)\Gamma\left(\frac{m}{2}\right) + 1}
\]

where,

\[
\overline{R}(s_i) \text{ is the maximum distance between the sample } s_i \text{ and its } k_i \text{ nearest training samples of the same class as } s_i
\]

\[
\beta = \frac{\sigma_i}{\delta_i} \text{ is the smoothing parameter used to determine the bound of the class}
\]

Different values of \(\beta\) will give different smoothing effects

\[
w_i = \left(\frac{k_i + 1}{\lambda^m} \right) \frac{1}{\Gamma\left(\frac{m}{2}\right) + 1} \frac{m}{\overline{R}(s_i)^m \pi^{\frac{m}{2}}}
\]

where,
\[ \lambda = \sum_{h=0}^{\infty} \exp \left[ -\frac{h^2}{2\beta^2} \right] \]

where, \( h = 1, 2, \ldots, k_1 \).

This method is capable of matching or exceeding the performance of neural networks with back-propagation algorithm, but gives training comparable with those of sigmoid type of feedforward neural networks [12].

The classification accuracy can further be improved by applying data reduction techniques. In literature several training dataset condensation algorithms have been worked out to reduce the training time thus increasing the efficiency of the classifier. This includes methods like instance-based [13], lazy [14], memory-based [15] and case-based learners [16]. Recently, Jingnian et al [17] proposed an instance selection method called FINE to reduce the training dataset. Data reduction using Nearest Neighborhood rule proposed by Angiulli et al [18] shows faster condensation of training dataset. Before dividing the data set into training and test sets, preprocessing techniques must be applied to fill missing data values, noisy data and irrelevant data values. For classification tasks, the data set must contain numerical values. If any attribute contains nominal data, convert them into numerical values. Then normalize the attribute values into the range [0, 1]. Replace missing value filters can be used to fill the missing values with the mean of the available values for that attribute. Noise is removed to some extent when we normalize the data. Feature extraction techniques can be applied to remove irrelevant attributes. F-correlation technique is applied to find the correlation between the attributes and C-correlation technique is applied to find the correlation between the class and the attribute. Now the data is ready for classification.

The preprocessed data is divided into two groups: training data set and test data set. The training data set should have samples for all the class labels. The Spherical Gaussian Function (SGF) network model is trained with the samples in the training set and classification accuracy is tested with the samples in the testing data set.

The classification accuracy can be further improved by removing redundant vectors in the training sample. Instead of removing all duplicates, we propose a method of building the training set using cluster analysis. A cluster is defined to be a subset of objects whose degree of dissimilarity within a cluster is less compared to the degree of dissimilarity of objects in two different clusters. Given the data set, apply k-means clustering algorithm to form groups. Randomly select a sample in a group and identify instances similar to it in all clusters. Find the Manhattan distance between the selected instances and specification uncertainty measure. Then apply a hierarchical clustering method to further minimize the number of instances within a cluster. At each step, the distance between the instances in a cluster is calculated using agglomerative hierarchical clustering method. If the distance measure is below a specified threshold \( \theta \), then the instances are merged. This method has been found to be very effective and efficient because it deals with the homogenization of the classes. Fixation of this threshold depends on the amount of data set available and the number of clusters formed for a particular study. If the number of instances is very large, then threshold can be set to a value greater than 0.5 so that the resulting instances in the training set is reduced and hence time for learning is reduced. If the number of instances available is small then threshold is set to a value lesser than 0.5, so that the network model is trained well and classification accuracy is improved.

The training time can be reduced there by increasing the efficiency of the classifier by concentrating on the instances on the boundaries of the clusters. The distance measure between the instance in the boundary of a cluster and all the other nearby clusters are calculated. If this distance is significantly greater than the distance between the instances within the cluster then the instance is termed a negative instance and placed in a nearby cluster. If the distance measure is negligible then it is removed from the cluster. This method is called instance-based data reduction technique. The number of clusters thus formed is reduced in the beginning itself, hence the number of hidden nodes is reduced and there is a great reduction in the time complexity. After the data reduction, the correlation of the instances within a cluster is studied using discriminant analysis. This helps in studying the characteristics of the data that are more relevant to the classification of the data into a given class. This shows increased classification accuracy.

4. EXPERIMENTAL STUDY

For this study, sales data of 1200 samples were used for studying the effectiveness of radial basis function neural networks as a classifier. The sales records were classified into “high sales cars”, “moderate sales cars” and “low sales cars”. The RBF neural network architecture considered for this application was a single hidden layer with Spherical Gaussian RBF. Gaussian-type RBF was chosen here due to its similarity with the Euclidean distance and also since it gives better smoothing and interpolation properties.

Out of the 1200 samples, 720 samples (60%) were used for training the neural network and the remaining 480 samples (40%) were used as test data. Feature selection is a critical process in any classification task because it determines the number of hidden layer units to be used. In our study the sales records were preprocessed to remove noise and unwanted attributes, attributes with wide range of unique values. Finally records with three attributes: the car model, fuel type and the price were used as features. Among the 720 samples taken, 40% samples were of high sales items, 30% of moderate sales and 30% were of low sales. The RBF neural network performed at its best and the percentage of correct classification was above 91%. The network was trained with different centers and also by varying the number of iterations.

Table.1 shows the classification accuracy of the kernel density estimation algorithm (KDE), and the proposed kernel density estimation algorithm with data reduction (KDEDR) technique. In the first method the removal of redundant data has less improvement compared to the classification accuracy that results after redundant data removal using threshold in the second method.
Table 1. Comparison of classification accuracy between KDE with naive data reduction and KDE with data reduction using threshold

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Classification Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>KDE</td>
</tr>
<tr>
<td>Full Data</td>
<td>85.6</td>
</tr>
<tr>
<td>Preprocessed Data</td>
<td>90</td>
</tr>
<tr>
<td>Reduced Data</td>
<td>92.1</td>
</tr>
</tbody>
</table>

4.1 TIME COMPLEXITY

The CPU time for classification also reduces when feature extraction and data reduction techniques are applied and the following graph shows the time complexity when KDE with naive data reduction and KDE with data reduction using threshold technique is applied. The naïve data reduction method applied in KDE simply removes the redundant instances in each cluster and accordingly the hidden units are decided. In data reduction with threshold technique applied to KDED, the clusters are reduced resulting in reduced number of hidden nodes and hence a great reduction in CPU execution time. Still the effect of the time reduction is to be studied in comparison with other universally approved data sets.

![Time Complexity Comparison](image)

Fig. 2. Comparison of execution times in seconds before applying hierarchical clustering technique. Fig. 3 shows the reduction in processing time when hierarchical clustering method is applied to dynamically reduce the instances in each class to retain only the instances that are very relevant to the class to which they belong and hence the classification accuracy is greatly improved.

The various measures of classification accuracy are tested with 10 fold cross validation and it is found to be substantially higher in case of the proposed data reduction technique. The actual execution time depends on the numbers of training samples that are left over after applying data reduction.

5. CONCLUSION

In this study, the proposed method of data reduction shows good improvement in terms of classification accuracy and speed. The number of attributes considered for this experiment is three. The method can be further analyzed by increasing the number of attributes considered and a method can be evolved to fix the threshold value so that the method becomes more effective in removing redundant instances in large data sets. The classification performance can be analyzed by adding fuzzy logic which will be the proposed study of this continuous research.

REFERENCES


