Nonlinear Cluster Transformations for Increasing Pattern Separability

R. Polikar, L. Udpa, S. S. Udpa
Materials Assessment Research Group
Dept. of Electrical and Computer Engineering
Iowa State University, Ames, Iowa, USA 50011

Abstract

The objective of classification is to generate a nonlinear multidimensional decision boundary that partitions the pattern space into prescribed classes. However, these algorithms are successful only when the data is well distributed in their domain. In practice, patterns from different classes can be closely packed with significant overlap. Prior to classification, the data is generally preprocessed so that the intercluster to intracluster distance ratio is maximized. This paper discusses limitations of conventional approaches for preprocessing based on Fisher's linear discriminant, and proposes an intuitive nonlinear cluster transformation (NCT) that can be used for increasing the intercluster distances within a set of data points. A generalized regression neural network (GRNN) is used to learn the functional mapping between original clusters and transformed clusters. The performance of this proposed method was tested on a benchmark database and then on a real world database of patterns generated for odor identification. Initial results using NCT have been very promising.

1. Introduction

Classification algorithms are finding increasing application as tools for consistent and accurate interpretation of data. Many techniques of classification have been developed over the years, such as neural networks, genetic algorithms, discriminant analysis, Bayes classifiers, etc. However, these techniques perform well only when the clusters are separable. In general, classification algorithms share the basic requirement of adequately large intercluster distances between patterns of different classes, and small intracluster distances between patterns of the same class.

The required separability is often obtained by using an appropriate feature extraction algorithm as a preprocessing step to classification. The fundamental objective of feature extraction is to reduce the dimensionality of pattern vectors without loosing discriminatory information. The general problem of feature extraction can be formulated as one of determining a mapping of the form $\mathbf{y} = f(\mathbf{x})$, or $\mathbf{y} = \mathbf{W}\mathbf{x}$, that transforms pattern vectors onto a lower dimensional feature space in which the corresponding feature vectors are separable. The Fisher Linear Discriminant (FLD) was one of the first methods proposed to achieve dimensionality reduction based on maximizing a criterion function, ratio of intercluster to intracluster distance. The FLD algorithm projects the data onto a lower dimensional space where this criterion function is maximized. Consequently, FLD is a feature reduction algorithm that ensures maximum separability of patterns in the transformed space. However, FLD has its limitations, depending on the number of patterns in the training database, number of classes, and the dimensionality of patterns.

The primary purpose of this paper is to develop a simple, yet effective procedure specifically for increasing class separability of patterns measured quantitatively by the ratio of intercluster to intracluster distances used in the FLD algorithm. A brief discussion of the FLD algorithm is given in the next section and its limitations with respect to the problem of increasing class separability are discussed. The motivation for developing a general-purpose method that could be used for increasing intercluster distances is then presented. A nonlinear cluster transformation (NCT) method is introduced in Section 3 as a candidate procedure. The proposed procedure is explained in detail in this section. The performance of the proposed method is presented on synthetic and real world data in Section 4. A comparison of classifier performance with and without NCT as a preprocessing step is also presented. Finally, concluding remarks and directions for future work are discussed in Section 5.

2. Background

The Fisher linear discriminant (FLD) technique has enjoyed much attention and success largely as a technique for reducing the dimensionality of a classification problem by

0-7803-5529-6/99/$10.00 ©1999 IEEE

projecting the data instances onto a new space of lower dimension.

Consider a multi-class classification problem and let C be the number of classes. For the i^{th} class, let $\{\mathbf{X}_i\}$ be the set of patterns in this class, $\mathbf{m}_i$ be the mean of vectors $\mathbf{x} \in \{\mathbf{X}_i\}$, n_i be the number of patterns in $\{\mathbf{X}_i\}$. Let $\mathbf{m}$ be the mean of all patterns in all C classes. Then the within class scatter matrix $\mathbf{S}_W$, and between class scatter matrix $\mathbf{S}_B$ are defined as follows:

$$\mathbf{S}_W = \sum_{i=1}^{C} \sum_{\mathbf{x} \in X_i} (\mathbf{x} - \mathbf{m}_i) \cdot (\mathbf{x} - \mathbf{m}_i)^T$$
$$\mathbf{S}_B = \sum_{i=1}^{C} n_i (\mathbf{m} - \mathbf{m}_i) \cdot (\mathbf{m} - \mathbf{m}_i)^T \quad (1)$$

The transformation onto a lower dimensional feature space, can be expressed as

$$\mathbf{y} = \mathbf{W} \cdot \mathbf{x} \quad (2)$$

where the column vector $\mathbf{y}$ is the feature vector in the projected space corresponding to pattern $\mathbf{x}$. The optimum matrix $\mathbf{W}$ is obtained by maximizing criterion function

$$J(\mathbf{W}) = \mathbf{S}_{BF} \Big/ \mathbf{S}_{WF} \quad (3)$$

where $\mathbf{S}_{BF}$ and $\mathbf{S}_{WF}$ are the corresponding scatter matrices in the (feature) projection space. It can be shown that $\mathbf{S}_{BF}$ and $\mathbf{S}_{WF}$ can be written as [1]

$$\mathbf{S}_{BF} = \mathbf{W}^T \mathbf{S}_B \mathbf{W}$$
$$\mathbf{S}_{WF} = \mathbf{W}^T \mathbf{S}_W \mathbf{W} \quad (4)$$

Therefore, the criterion function can be represented in terms of the scatter matrices of the original patterns.

$$J(\mathbf{W}) = \frac{\mathbf{W}^T \mathbf{S}_B \mathbf{W}}{\mathbf{W}^T \mathbf{S}_W \mathbf{W}} \quad (5)$$

$J(\mathbf{W})$ is a vector valued function, and the determinant of this function can be used as a scalar measure of $J(\mathbf{W})$. The columns of $\mathbf{W}$ that maximizes the determinant of $J(\mathbf{W})$ are then the eigenvectors that correspond to the largest eigenvalues in the generalized eigenvalue equation [1]

$$\mathbf{S}_B \mathbf{w}_i = \lambda_i \mathbf{S}_W \mathbf{w}_i \quad (6)$$

For nonsingular $\mathbf{S}_W$, Equation 6 can be written as

$$\mathbf{S}_W^{-1} \mathbf{S}_B \mathbf{w}_i = \lambda \mathbf{w}_i \quad (7)$$

From Equation 7, we can directly compute the eigenvalues λ_i and the eigenvectors $\mathbf{w}_i$, which then constitutes the columns of the $\mathbf{W}$ matrix.

The limitation of the FLD method can be explained with the help of two theorems. The proofs of these theorems are straightforward [1].

Theorem 1. Regardless of the dimension of the original pattern, the FLD transforms a pattern vector onto a feature vector, whose dimension can be at most $C-1$, where C is the number of classes.

Theorem 2. The matrix $\mathbf{S}_W$ is nonsingular if and only if $N-C < d$, where N is the number of training data and d is the dimension of the pattern vector.

Conditions identified in these theorems limit the application of FLD to a constrained set of problems and in such cases alternate techniques are required for increasing the intercluster distances.

The next section describes the proposed nonlinear cluster transformation method, for addressing the problem of overlapping clusters. The quantitative measure of effectiveness of the method in achieving this goal is measured using the same criterion function of the FLD algorithm given in Equation 3.

3. Nonlinear Cluster Transformation (NCT)

Nonlinear cluster transformation is a three-step supervised procedure that attempts to increase the intercluster distances and reduce the intracluster distances, while preserving the dimensionality of the pattern vectors. NCT has no limitations in terms of dimensionality, number of classes, or the total number of patterns in the database.

In the first step, reduction of intracluster distances is achieved by eliminating the outliers. In the second step, the desired cluster separation is obtained by a translation of each cluster along an optimal direction. In the last step, the data generated in step two is used to train a generalized regression neural network (GRNN) to approximate the function mapping between original clusters and the translated clusters. The performance of this algorithm is evaluated by computing the FLD criterion function in the pattern space and feature space. The feature vectors are the input to a classifier of choice. The details of these steps are explained in the following paragraphs.

Outlier Removal

The patterns in each class i in the training database are first normalized according to

$$\mathbf{x} = \frac{\mathbf{x}}{\sqrt{\sum_{k=1}^{d} (x^k)^2}} \quad (8)$$

where $\mathbf{x}^k$ is the k^{th} element of pattern $\mathbf{x}$, and d is the dimensionality of the patterns.

Outlier removal is performed next, based on the Mahalanobis distances of patterns from the cluster centers. For each cluster i, the Mahalanobis distance of pattern $\mathbf{x}$ in class i is computed as

$$M_D = (\mathbf{x}-\mathbf{m}_i)^T \mathbf{C}_i^{-1}(\mathbf{x}-\mathbf{m}_i) \qquad \mathbf{x}\in\{\mathbf{X}_i\} \qquad (9)$$

where $\mathbf{C}_i$ is the covariance matrix of the pattern population of the i^{th} class, and $\mathbf{m}_i$ is the mean of this population. M_D can be used as a measure of dispersion within the cluster. Note that the Mahalanobis distance is a better distance criterion then the Euclidean distance. The Euclidean distance simply measures distance from the cluster center. In contrast, Mahalanobis measures distances to the cluster as a whole, and therefore it is more suited for outlier detection.

Cluster Translation

This step addresses the problem of closely packed and possibly overlapping clusters. The underlying idea is to translate the clusters appropriately in order to physically separate them. Conceptually, all clusters are thought of as like charged particles, and the magnitude and direction of the translation vector are then derived using the concept of a repulsive force exerted by each cluster i on all other clusters. The procedure is first explained for a two-class problem. The natural extension to the multi-class case is then derived.

Consider a two-class problem with (possibly) overlapping clusters, whose centers are located at $\mathbf{m}_1$ and $\mathbf{m}_2$. The distance between these two clusters can be increased if the class I patterns are translated along a vector $\mathbf{S}_1 = -(\mathbf{m}_2-\mathbf{m}_1)$, and class II patterns are translated along $\mathbf{S}_2 = -\mathbf{S}_1 = -(\mathbf{m}_1-\mathbf{m}_2)$. This idea can be extended to multi-class problems of arbitrary dimensionality, where patterns of class C_i can be translated along an optimal direction $\mathbf{S}_i$ computed as follows:

Let $\mathbf{m}_j-\mathbf{m}_i$ be the vector directed from cluster i to cluster j. The resultant vector due to all clusters is

$$\mathbf{M}_i = \left(\sum_{j\neq 1}(\mathbf{m}_j-\mathbf{m}_i)\right) \qquad (10)$$

The optimal translation vector for cluster i is then given by

$$\mathbf{S}_i = -\sum_{j\neq i}^{C}(\mathbf{m}_j-\mathbf{m}_i) \qquad (11)$$

All patterns $\mathbf{x}_i$ are translated according to

$$\mathbf{x}_{Si} = \mathbf{x}_i + (\mathbf{S}_i/\|\mathbf{S}_i\|)\cdot dist_i \qquad (12)$$

where $dist_i = 1/|\mathbf{m}-\mathbf{m}_i|$ is a normalizing constant that controls the magnitude of translation, and $\mathbf{x}_{Si}$ is the new location of the pattern $\mathbf{x}_i$.

It is straightforward to show mathematically that these translation directions maximize intercluster distances. For a C class problem, we first define the overall intercluster distance D as

$$D = \sum_{i,j=1}^{C} D_{ij} = \sum_{i,j=1}^{C}(\mathbf{m}_i-\mathbf{m}_j)^T(\mathbf{m}_i-\mathbf{m}_j) \qquad (13)$$

After translating cluster i according to Equation 10, the new overall intercluster distance is

$$D^{new} = \sum_{j=1}^{C}(\mathbf{m}_i+\mathbf{M}_i-\mathbf{m}_j)^T(\mathbf{m}_i+\mathbf{M}_i-\mathbf{m}_j) + \sum_{\substack{j,k=1\\ j,k\neq i}}^{C}(\mathbf{m}_k-\mathbf{m}_j)^T(\mathbf{m}_k-\mathbf{m}_j) \qquad (14)$$

The vector $\mathbf{M}_i$ that *minimizes* D^{new} is obtained by

$$\frac{\partial D^{new}}{\partial \mathbf{M}_i} = \sum_{j=1}^{C} 2(\mathbf{m}_i+\mathbf{M}_i-\mathbf{m}_j) = 0 \Rightarrow \mathbf{M}_i = \frac{1}{C}\sum_{j=1}^{C}(\mathbf{m}_j-\mathbf{m}_i) \qquad (15)$$

since

$$\frac{\partial^2 D^{new}}{\partial(\mathbf{M}_i)^2} = 2C > 0 \qquad (16)$$

We therefore deduce that the optimal direction of translation, $\mathbf{S}_i$, to maximize the new overall intercluster distance is

$$\mathbf{S}_i = -\mathbf{M}_i = -\frac{1}{C}\sum_{j=1}^{C}(\mathbf{m}_j-\mathbf{m}_i) \qquad (17)$$

Note from Equation (17) that $\mathbf{S}_i$ points away from all other clusters in the opposite direction of the resultant vector.

The cluster transformation described here can also be expressed in a matrix form. Let $i=1,2,...,C$, where C is the number of classes, $n=1,2,...,N_i$ where N_i is the number of patterns in class i, X_n^i be the n^{th} pattern of the i^{th} class, and Y_n^i be the corresponding pattern after translation. Then,

$$\begin{pmatrix} Y_1^i \\ Y_2^i \\ . \\ . \\ Y_{Ni}^i \end{pmatrix} = -dist_i \cdot \begin{pmatrix} 1 & \cdots & 1 \\ \vdots & & \\ & \vdots & \\ & & \vdots \\ 1 & \cdots & 1 \end{pmatrix} \cdot \begin{pmatrix} \mathbf{m}_1-\mathbf{m}_i \\ \mathbf{m}_2-\mathbf{m}_i \\ . \\ . \\ \mathbf{m}_C-\mathbf{m}_i \end{pmatrix} + \begin{pmatrix} X_1^i \\ X_1^i \\ . \\ . \\ X_{Ni}^i \end{pmatrix} \qquad (18)$$

This equation is implemented on the training data sets to generate a second dataset that can be used to train a GRNN to learn the overall transformation function.

Function Mapping

In order to preprocess the test data, we need to *learn* how to translate patterns without knowing the class information. Hence, we have a function approximation problem, where the function to be approximated maps the d dimensional original patterns to their new locations. A GRNN

was used to accomplish this function approximation. GRNNs, developed by Specht [2] can be thought of as a special case of radial basis function neural networks (RBFNN), and they have been used with significant success in multidimensional function approximation. GRNN is based on the theory of nonlinear regression analysis, and it does not require an iterative training. GRNN is commonly used as a statistical function estimation scheme. The problem of nonlinear regression analysis estimate the most likely value of the dependent variable y given a set of training data of independent variable $\mathbf{x}$, and the corresponding values of y. The expected value of y can be computed as [3]

$$E[y \mid \mathbf{x}] = \int_{-\infty}^{\infty} y\, f(\mathbf{x}, y)dy \Big/ \int f(\mathbf{x}, y)dy \qquad (19)$$

where y is the output of the estimator, $\mathbf{x}$ is the input vector for which the corresponding output is to be estimated and $f(\mathbf{x},y)$ is the joint probability density function of $\mathbf{x}$ and y. Specht showed that Equation (19) can be optimally approximated as

$$y_j = \sum_{i=1}^{N} R_i w_{ij} \Big/ \sum_{i=1}^{N} R_i \qquad (20)$$

where

$$R_i = e^{\|\mathbf{x}-\mathbf{u}\|/2\sigma^2} \qquad (21)$$

is output of the i^{th} receptive field (hidden neuron), w_{ij} is the weight that connects the i^{th} hidden neuron to the j^{th} output neuron, σ is a spread constant that controls the ranges of the receptive regions, and finally $\mathbf{u}$ are the training vectors and they are the centers of the receptive fields.

4. Experimental Results

The proposed algorithm was tested on various databases, only, two of which are discussed here due to space limitations. The first database is the double spiral database, which consists of two interleaved spirals, and the second database is derived from a highly challenging real world application related to odor classification.

Double Spiral (DS) Database

The DS database is a popular database used extensively for evaluating neural network architectures. This database consists of two distinct spirals in the x-y plane, each of which coils three times around one another. The advantage of this database is that, it uses a two dimensional feature space, thus allowing easy visualization of patterns. This dataset represents classes with poor separability, since it is known to be a challenging dataset for MLP and related networks.

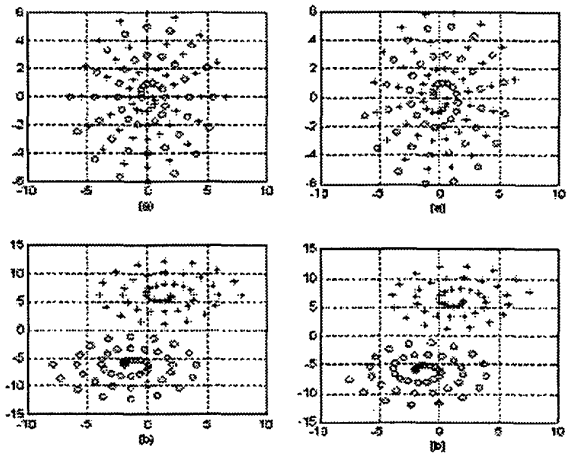


Figure1 (a)T_{DS} dataset (b) T_{DS} after translation

Figure 2. (a) E_{DS} dataset (b) E_{DS} after NCT

The database was divided into a training dataset, T_{DS}, and an evaluation dataset, E_{DS}. The separation was obtained by putting every other instance to T_{DS}, and the remaining into E_{DS}. Figure 2(a) illustrates the T_{DS} database, and Figure 2(b) shows the result of cluster translation on T_{DS}. These patterns were then used to train a GRNN, which was evaluated on the E_{DS}. Figures 3 (a) and (b) illustrate the E_{DS} dataset before and after NCT operation. Note that the two spirals are now linearly separable and can be classified using a simple linear decision function.

Gas Sensing Database

The classification of patterns generated in odor sensing procedure was the driving force for the development of the NCT algorithm. Identification and quantification of volatile organic compounds (VOCs) are of crucial importance for environmental monitoring as well as for many industries, including nondestructive evaluation for detecting gas leaks. Piezoelectric acoustic wave sensors, which comprise a versatile class of chemical sensors, are used for the detection of VOCs. Addition or subtraction of gas molecules from the surface or bulk of an acoustic wave sensor results in a change of its resonant frequency. The frequency change, Δf, caused by a deposited mass Δm can be described by $\Delta f = -2.3\times10^{6} \cdot f^{2}.\frac{\Delta m}{A}$ where f is the fundamental resonant frequency of the bare crystal, and A is the active surface area. For sensing applications, a sensitive polymer film is cast on the surface of the QCM. This layer can bind a VOC of interest, altering the resonant frequency of the device, in proportion to the added mass.

One of the most challenging issues is the identification of VOCs in mixtures. The database generated for this study contains 24 binary mixtures of VOCs. Each mixture con-

tains two of the following VOCs: Octane (OC), acetonitrile (ACN) xylene (XL), ethanol (ET), toluene (TL), trichloroethane (TCA), trichloroethylene (TCE), methylethylketone (MEK), and hexane (HX). Six sensors, each coated with a different polymer, were exposed to these mixtures at sixteen different concentration levels for 24 binary mixtures. The polymers used were Apiezon (APZ), Polyisobutelene (PIB), Poly(diethyleneglycoladipate) (DEGA), Solgel (SG), Siloxane (OV275), and Poly(diphenoxylphoshate) (PDPP). Figure 3 shows typical signals obtained from four mixtures of XL. The vertical axis shows relative frequency change. Each bar represents the response of one sensor coated with a different polymer. Note the similarity of the patterns for different mixtures. The main reason of this similarity is the dominance of the XL gas compared to the others.

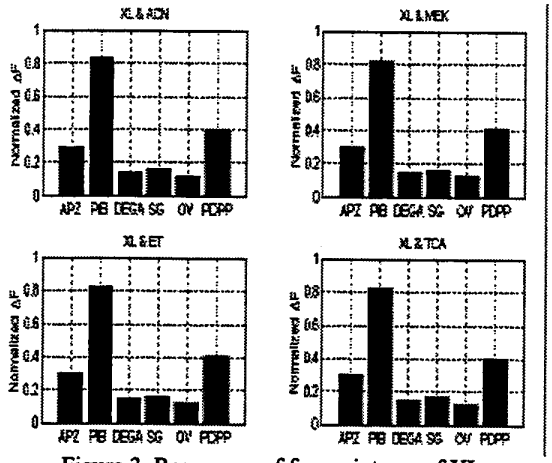
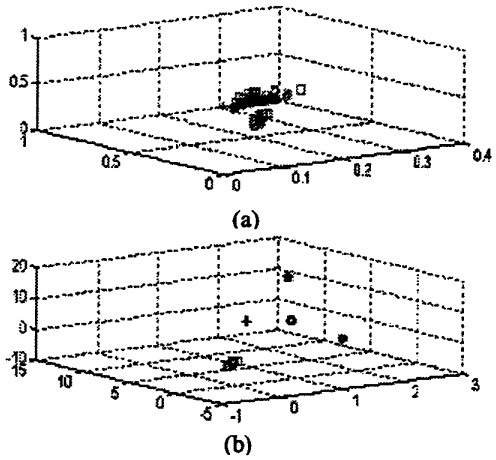


Figure 3. Responses of four mixtures of XL.

When a dominant VOC is present in the mixture, the responses of sensors to other VOCs become partially, or completely, masked by the response to the dominant VOC. This phenomenon causes patterns of different classes to be extremely tightly clustered or overlapped. Among the VOCs listed, XL, TL, ET, TCE and OC are known to cause a higher response of the sensors, and therefore the main goal was to identify these dominant VOCs. Traditional MLP neural networks were unable to distinguish these signals according to their dominant VOCs (though once the dominant VOC was identified, a two-hidden layer MLP was able to identify the secondary VOCs).

NCT of Gas Sensing Data

The sample patterns shown in Figure 3 represent typical patterns encountered and illustrate the difficulty of this database. The entire database consisted of 384 six-dimensional patterns. For the identification of the dominant VOC problem, there were five classes, namely, OC, ET, XL, TL and TCE. For the sake of visualizing the data, and the effect of NCT, only three attributes were used in the following figures. The computations, however, used all six attributes. The database was partitioned into two parts, T_{VOC} for training and E_{VOC} for evaluation. Each partition had 192 instances. Figure 4 illustrate T_{VOC} with its first three attributes, and the result of cluster translation on T_{VOC}. The training database T_{VOC} and its translated version were used to train the GRNN to learn the NCT for this database. GRNN was then evaluated on E_{VOC}. Figure 5(a) illustrates the evaluation database E_{VOC}, and Figure 5(b) shows the output of the GRNN for this dataset. As we can see from Figure 5(b) the transformed patterns do not look quite as well separated (at least in 3-D) as the training data in Figure 4(b). However, the five clusters in Figure 5(b) are more separable after the NCT compared to unprocessed patterns of Figure 5(a).



(a)

(b)

Figure 4. T_{VOC} (a) before and (b) after transformation

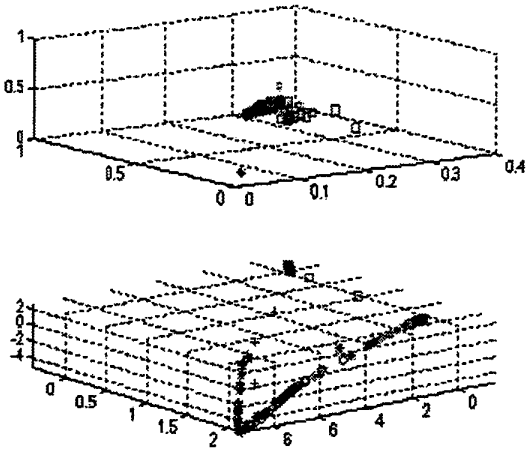


Figure 5. E_{VOC} (a) before and (b) after NCT

When the unprocessed data was used, the network was not even able converge. FLD was also tried on this database, however, FLD was unable to project the data to a 4-D space where patterns were more separable. However, using the transformed patterns of E_{VOC}, a three-layer MLP was easily trained. Repeating over twenty trials, a correct classification performance of 80% - 95% was obtained over the entire E_{VOC}.

The J(W) criterion function, defined earlier in Equations (3) and (5) was evaluated before and after NCT. For the E_{VOC} dataset, J(W) showed an increase of seven orders of magnitude after NCT. This demonstrates the effectiveness of NCT on the evaluation database.

5. Conclusions and Future Work

NCT, an intuitively simple, yet considerably successful scheme has been introduced for increasing intercluster distances while reducing intracluster distances. The procedure was originally inspired by the FLD. Preprocessing with NCT allowed improved performances of subsequent classification algorithms, and in fact, it made training possible for double spiral and VOC databases.

Training for NCT is a one step procedure, which does not require an iterative learning, however, NCT requires considerably larger memory than iterative learning techniques, particularly for databases with large number of data instances.

Work is currently underway for testing the NCT algorithm on other databases of higher dimensionality. An improved translation criterion that considers variances of the clusters as well as the distance between clusters is being developed. Finally, the problem of co-linear distribution of clusters is also being investigated.

References

[1]. Duda R.O. and Hart P.E., *Pattern Classification and Scene Analysis*, John Wiley and Sons, New York, 1973.
[2]. Specht D.F., A general regression neural network, *IEEE Transactions on Neural Networks,* Vol. 2, No.6, p.568-576, 1991.
[3]. Wasserman P.D., *Advanced Methods in Neural Computing*, Van Nostrand Reinhold, New York, 1993.