T.C.
Bahçeşehir University

IMPROVED ALGORITHMS for LINEAR DISCRIMINANT ANALYSIS

Master Thesis

Caner GÜLLÜOĞLU

Istanbul, 2010

T.C.

Bahçeşehir University

The Graduate School of Natural and Applied Sciences

Master of Science in Computer Engineering

IMPROVED ALGORITHMS for LINEAR DISCRIMINANT ANALYSIS

Master Thesis

Caner GÜLLÜOĞLU

Advisor: Assist.Prof. Turgay TEMEL

Istanbul, 2010

T.C
BAHÇEŞEHİR UNIVERSITY
The Graduate School of Natural and Applied Sciences
Master of Science in Computer Engineering

Title of the Master's Thesis : Improved algorithms for linear discriminant analysis
Name/Last Name of the Student : Caner Güllüoğlu
Date of Thesis Defence : 04 June 2010.

The thesis has been approved by the Graduate School of Natural and Applied Sciences.

Signature

Assist.Prof. Tunç BOZBURA

This is to certify that we have read this thesis and that we find it fully adequate in scope, quality and content, as a thesis for the degree of Master of Science.

Examining Committee Members:

Assist. Prof. Turgay TEMEL(Supervisor) :

Assoc. Prof. Taşkın KOÇAK :

Assist. Prof. Olcay KURŞUN :

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my thesis supervisor Asist.Prof. Turgay Temel. His endless support during my research and the time he has dedicated for this thesis is unforgettable. Without his contribution and motivation, this study would not have been completed. I also would like to thank him and Prof. Dr. Bekir Karlık for providing real data which was used in their research.

For his endless and precious understanding, I would like to thank my manager and undergraduate instructor Eşref Seğmen.It is impossible for me to complete this study without his support.

I would like to thank my thesis committee members, Assoc.Prof.Taşkın Koçak and Assist.Prof. Olcay Kurşun for devoting time and energy to evaluate this research.

I want to thank my undergraduate advisor Prof. Yavuz Akpınar at Boğaziçi University for changing my vision and his guidance for academic life.

I also would like to thank members of Bahçeşehir University, Assist.Prof. Çağrı Güngör and Murat Yağcı, for their academic motivation and support on me.

This thesis is also dedicated to my precious family and my beloved friends. This work, among many other things, would be lack of meaning without their existence.

ABSTRACT

IMPROVED ALGORITHMS for LINEAR DISCRIMINANT ANALYSIS

GÜLLÜOĞLU, Caner

Master of Science in Computer Engineering

Supervisor: Assist.Prof. Turgay TEMEL

June 2010

Data recognition and classification are key research topics in machine learning. Although there are algorithms such as multi-layered perceptron neural networks which are able to discriminate even highly complex data, it is difficult to suggest a direct methodology to determine their respective configuration, i.e. type of feedback, number of hidden layers etc. An important aspect which determines the efficiency and generalization capability of a classification algorithm is how data spread in raw sample space. Most classification algorithms can be brought in improved generalization capability by providing them with loosely scattered or less overlapped classes of data without reducing the information content. By doing so, it is possible to avoid the need of redundantly formed high-dimensional representation of data. Resulting classifier is expected to leverage in classification performance as well as remedial to problem of 'curse of dimensionality'. A widely adopted method for better scattering in sample space is to employ a pre-processing algorithm before introducing data into classifier. Resulting simpler classifier is expected to exhibit improved generalization capabilities. An important outcome to be attained with simplicity is real-time processing, i.e. recognition of the input.

As per the statements about pre-processing for loosely scattered data, discriminate analysis has been well known. Despite some modifications such as nonlinear discriminate analysis based on kernels which satisfy certain criteria, the simplicity in

formulation and direct consequence onto neural classifiers, linear discriminant analysis (LDA) has been regarded for numerous classifier-based machine learning applications. Due to its simplicity, LDA has considerable benefit advantages compared to other spectral methods such as principal component analysis (PCA), or singular value decomposition (SVD).

In this thesis, a new pre-processing algorithm toward improved data scatter properties as an LDA algorithm is introduced. It is experimented with real odor data utilized in a well-known pattern recognition algorithms. The performance comparison is evaluated to those which do not employ LDA in terms of the number of training samples to achieve a desired generalization capability and the number of iterations needed to get the algorithm to converge the associated learning algorithm.

Keywords : Linear Discriminant Analysis, data scattering, data pre-processing

ÖZET

DOĞRUSAL DİSKRİMİNANT ANALİZİ İÇİN İYİLEŞTİRME ALGORİTMALARI

GÜLLÜOĞLU, Caner

Bilgisayar Mühendisliği Yüksek Lisans Programı

Danışman: Yard.Doç.Dr. Turgay TEMEL

Haziran 2010

Örüntü tanımlama ve sınıflandırma, makine öğreniminde önemli araştırma alanlarındandır. Bu alanlar için önerilmiş pek çok algoritma olmasına rağmen,örneğin çok katmanlı perceptron yapay sinir ağları çok karmaşık verileri ayrıştırabilme özelliğine sahiptir, verinin özelliklerini göz önüne alınarak; örneğin geri besleme yöntemi, gizli katmanların sayısı vs, doğrudan uygulanabilecek genel bir yöntem önermek çok güçtür. Sınıflandırma algoritmalarının genelleyebilme kapasitesini ve etkinliğini belirleyen önemli özelliklerden biri de, işlenmemiş verinin örnek uzayda nasıl bir şekilde dağılmış olduğudur. Seyrek dağılmış ve ya az çakışan veri sınıfları yardımı ile pek çok sınıflandırma algoritmasının genelleyebilme kapasitesi, bilgi içeriğini kaybetmeden, daha iyi bir duruma gelebilir. Böylece, çok boyutlu verinin gereksiz yere kullanımı engellenebilir. Elde edilen ayrıştırıcı fonsiyonun, sınıflandırma performansını yükseltmesi beklendiği gibi, ayrıca 'boyut sorunu' na da çözüm getirmesi beklenir. Veriyi, ayrıştırıcı fonksiyonu ile işlemeden önce, bir ön-işleme algoritmasına tabi tutma yolu ile örnek uzayda daha iyi dağılımlar elde etmek sıkça uygulanan bir modeldir. Buna göre elde edilen daha basit ayrıştırıcı fonksiyonun daha iyi genelleyebilme kapasitesi göstermesi beklenir.Ayrıştırıcı fonksiyonun basitleştirilmesi gerçek-zamanlı işleme yapılabilmesi açısından önemlidir, ör: girilen verinin tanımlanması vs.

Seyrek dağılmış veriyi ön-işleme tabi tutma ihtiyacı doğduğundan beri, diskriminant analizi kullanımı yaygındır. Doğrusal olmayan diskriminant analizinin kernel durumu gerektirdiği gibi bazı özel durumlar için değişiklik ihtiyacı olmasına rağmen, formülasyonundaki basitlikten ve nöral ayrıştırıcı fonksiyonlar için doğrudan sonuç vermesinden dolayı, doğrusal diskriminant analizi(LDA) ayrıştırıcı fonksiyon bazlı makine öğrenimi uygulamalarında önemli bir yer tutmaktadır.

Bu tez içerisinde, doğrusal diskriminant analizi öncesinde uygulanabilecek ve daha iyi veri dağılımı özellikleri ortaya çıkarabilecek yeni bir algoritma sunulmuştur.Algoritma, gerçek koku verileri ile çok tanınmış bazı örüntü tanımlama algoritmaları kullanılarak test edilmiştir.İstenilen genelleyebilme kapasitesine ulaşabilmek için gereken alıştırma örneklerinin sayısı ve istenilen öğrenme algoritmasına yakınsama için gereken döngü sayısı baz alınarak, doğrusal diskriminant analizi kullanmayan algoritmalar ile bir performans karşılaştırılması yapılmıştır.

Anahtar Kelimeler : Doğrusal diskriminant analizi, veri dağılımı, veri ön-işleme

TABLE of CONTENTS

LIST of TABLES

LIST of FIGURES

LIST OF ABBREVIATIONS

Principal Component Analysis	:	PCA
Linear Discriminant Analysis	:	LDA
Singular Value Decomposition	:	SVD
Discriminant Analysis	:	DA
Learning Vector Quantization	:	LVQ
Three Dimensional	:	3D
Two Dimensional	:	2D
Feed-Forward, Multilayer Perceptron	:	FFMLP
Nearest Neighbour	:	NN
Probability Density Functions	:	PDF
Expectation-Maximization	:	EM
Self-Organized Mapping	:	SOM
Independent Component Analysis	:	ICA
Karhunen Loeve' Expansion	:	KHE
Kernel Fisher Discriminant	:	KFD
Generalized Discriminant Analysis	:	GDA

LIST of SYMBOLS

Between-class scatter matrix	:	S_B
Within-class scatter matrix	:	S_W
Error probability	:	P_e
Class label k	:	C_k
Covariance matrix	:	Σ
Mean vector	:	μ
Learning rate	:	η
Eigenvector	:	λ

1. INTRODUCTION

As the science and technology progress rapidly, the data collection, storage units and processing tools have also developed. As a result, very large amount of data can be collected and processed for some of the research areas. A project of NASA, called SETI (Search of Extra-Terrestrial Intelligence) can be an example for large amount of data collection (Satorius and Brady 1988). Some satellites in deep space, such as Hubble, send huge amount of visual data for the project. Although, making more observations is needed to get healthy conclusions about the sources to be identified and investigated; the number of variables (features) have also increased for devising optimum models, which imposes cumbersome mathematical challenges on processing the datasets of interest.

A major area in pattern recognition is to deploy a robustly generic model to the problem for which data was collected so that respective source is identified and modelled for anticipating its behavior. Generally, resulting algorithms elucidate hidden statistical information within sample attributes. The objective is to assign a given unlabeled sample is assigned or identified to a class label of a source or object, which is also termed classification. The label association is usually performed based on biologically-motivated neural information processing paradigm in terms of cognitive plasticity, and memory formation (Kung and Mao 1991). However, since mostly it is not well known in advance what features are extracted and structured, and how in terms of biological processing, available feature cues are determined by developer's intellectuality and underlying background on the problem (Zhang, Zhao and Fen 2009). The process of developing and designing classification algorithms also involves thorough understanding of the problem at hand and the classifier itself. For example, considering speech recognition problem(Nadas 1985), the number of features which represent the voice characteristics will determine the structure of the classifier, which implies that the classifier with varying number of input features will also vary. However, the same classifier will not be applied to the same problem if the features are changed. Classifier design will also involve the clear determination of how the source information will be handled. If the process is to be real-time, overall classifier should be as simple as possible while maintaining efficiency. It is well-known that, e.g. XOR problem, simple

classifiers are not able to distinguish complex data and they need to be modified with augmented capabilities.

Another major issue which has impact on the classifier structure is how information is seen in the data hyperspace. The classes which are separated or spread loosely and do not overlap will be identified with simple classification architectures. Even if classes do overlap and an effective solution to be applied to raw samples is devised to spread them away each other which is a pre-processing scheme (Tattersall, Chichlowski and Limb 1992), the classifier which follows will be able to operate in real-time.

The problem of scattering dataset appropriately per se is in fact a transformation of the individual classes with respect to remaining dataset (Wang and Wong 1978). The topic has been examined in detailed treatment by large number of researchers since Ronald Fisher's(1936) contributions of statistics, which have yielded elaborated algorithms to be employed for classification.

The key problem is, although there are many features within the dataset, only a few of them are meaningful in the domain of the research, most of the features are irrelevant. This famous dilemma is known as "the curse of dimensionality" which is a term proposed by Bellman(1961). The term refers to exponential growth of the hyper volume as a function of dimensionality (Bellman 1961). The high dimensional data may be hard to cope with for several reasons: redundant features increase error rate and poor classification, inefficient use of storage while reducing the noise immunity; increased mathematical complexity in treatments involves complicated computations which usually makes it difficult to perform in real-time.

The problem for the curse of dimensionality can be solved by dimension reduction algorithms. Such algorithms are needed to optimize the classification performance and to increase the efficiency of classification. However, there are no generalized algorithm proposed which can be applied for any case, each of the existing algorithms can be applied to a specific problem depends on the dataset, corresponding to mean and variance of the data. Moreover, determination of the number of useful features is not easy since it may vary from problem to problem. This dependency is a major constraint in spectral-decomposition based pre-processing schemes (Kwak and Choi 2002).

However, in most problems where large number of features may be needed, e.g. image processing, recognition etc. feature reduction techniques are favoured (Bigun 1992). For example Principal Component Analysis (PCA) is one the most prominent feature reduction algorithms in the field. It is a non-parametric algorithm to extract relevant information from a large dataset (Martinez and Kak 2001), which makes the algorithm suitable to be referred to as an unsupervised method. The core idea of the PCA is to project samples onto a data subspace of some of the largest-variance dimensions. The projection is done with use of the associated eigenvectors of the covariance matrix, which match with the largest eigenvalues. Given p-dimensional vector, PCA tends to find another s-dimensional vector ($s<p$) according to maximum variance direction.

The output of the feature reduction can be further processed or directly employed as the input of a classification algorithm. However, feature reduction methods do not give information how far the classes are located from each other. Since classification is based on prescribed discriminative surfaces which discriminate classes in the data hyperspace and each class is identified by a group of vectors, investigating the class locations will allow to transform them more appropriately. If it is possible to do so, then relocating or mapping them uniquely for better scattering characteristics will be a much more convenient way to utilize simpler classifiers such as nearest neighbor (NN) with improved generalized capability. In fact it can be shown that relocated locations have close resemblance to feed-forward, multilayer perceptron (FFMLP) neural networks (Temel, 2010).

Discriminant analysis methods have been taken up by many researchers since Fisher's prominent study (1936) . Defining a between-class scatter matrix (S_B) and within-class scatter matrix (S_W) LDA tries to find the best linear hyper-plane as a classifier vector which discriminates the labelled classes after a training phase. LDA can be simply considered as a maximization of ratio S_B/S_W. However, it should be noted that the optimization objective is achieved by considering saliency features of the classes within the data hyper-space. As will be shown in the next chapters optimization will be subject to spectral decomposition and associated transformation given these matrices. Derivation of resulting expressions for a transformation which yields optimum scatter properties has been thoroughly studied by many researchers (Koutsougeras and Srikanth

1993; Jimenez, Arzuaga and Velez 2007). It has been observed that contriving data-oriented transformation suitable to perform in a simple manner is not easy and it becomes almost obsolete even for Gaussian densities.

Considering as such described concerning pattern recognition system as a classifier topology which can operate in real-time can be depicted as follows:

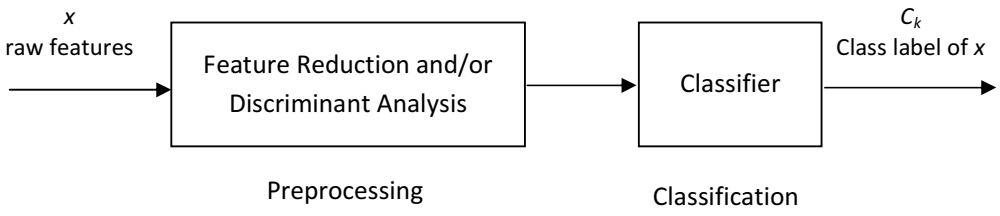

Figure 1.1 : A typical classifier with feature reduction and discriminant analysis operations.

In this study, a new iterative algorithm to express LDA-transformation matrix is presented as a pre-processing. The algorithm is similar to that proposed Sammon-Foley in (1975) without regarding orthogonal projection of features. A main theme of the algorithm consists in sample-based information-theoretic entropy description outlined in (Temel 2010). The resultant pre-processing algorithm is tested and exemplified with use of learning vector quantization classifier for synthetic multi-variate normal densities of various dimensions. In order to reflect the suitability of the algorithm for natural applications, it is also employed in identifying real-data odor class labels. The performance of classifiers with LDA is compared to that which does not employ a pre-processing. Their strength and weaknesses are pointed out for further studies.

The thesis is organized as follows: In remaining sections of this chapter, the notion of classification is reviewed with literature background. Important concepts concerning for classification performance is introduced. New study is presented in chapter IV along with experimental results. Prospective topics concerning pre-processing with LDA and iterative techniques are summarized in chapter V.

2. CLASSIFICATION

The purpose of the classification is to assign a given (test or unlabelled) data sample to one of M different classes expressed in terms of stochastic ensemble quantities (Temel 2010). In general terms, classification envisages a decision plane which yields the class label k, i.e., C_k for each sample x with assurance of some statistical optimality rules. In this section we will review some of the well-known classification algorithms: Probabilistic Bayesian, Nearest-neighbor (NN) and Learning Vector Quantization (LVQ) neural network.

2.1 NON-PARAMETRIC CLASSIFICATION

As an example of non-parametric classification methods, Nearest-neighbor classifiers will be reviewed.

2.1.1 Nearest-Neighbor (NN) Classifier

NN classifier (Shakhnarovich, Darrell and Indyk 2006), is a well-known non-parametric classifier. The classification of the sample (feature vector) x is performed based on the similarity/proximity measure between training samples and the sample is assigned to the class to which the closest sample belongs. The nearest neighbor classifier is formulated as

$$k = \underset{j}{\operatorname{argmin}} \left\| x - x_i \right\|_{\forall x_i \in C_j} \tag{2.1}$$

If sample x is picked from multivariate distributions, then the similarity measure can be given in the form of a normalization, such as Mahalanobis(1936) which is defined as

$$\left\| x - x_i \right\|^2_{x_i \in C_j} = (x - x_i)^T \Sigma_j^{-1} (x - x_i) \tag{2.2}$$

with $(.)^T$ denoting the transpose where covariance matrix is estimated.

Due to comparison for calculating the closest sample in whole dataset, despite the simplicity, nearest-neighborhood classifiers generally require longer computation time than most parametric models. Moreover their generalization capability is poorer compared to parametric models.

2.2 PARAMETRIC CLASSIFICATION

Various parametric models have been known for long. Some of them are summarized as follows:

2.2.1 Probabilistic Classifiers

Probabilistic classifier is a parametric model. Possibly the most known probabilistic classifier is the one which resorts on statistical models and associated parameters which need to be estimated with training samples in probabilistic sense. The best known optimality condition is expressed as the minimum error probability P_e amongst M different classes, (Temel 2010), which is known the Bayesian decision rule, (Zhou, Wu and Liu 1998) According to Bayesian decision rule, the maximum a posteriori (MAP) probability determines the class label to which the sample x is assigned as

$$k = \arg\max_{j} P(C_j \mid x) \tag{2.3}$$

The a posteriori probability $P(C_j|\mathbf{x})$ is written in terms of likelihood or class-conditional probability density functions (pdf) $p(\mathbf{x}|C_k)$ and a priori probabilities $P(C_k)$ as

$$P(C_k \mid x) = \frac{p(x \mid C_k)P(C_k)}{\sum_{j=1}^{M} p(x \mid C_j)P(C_j)} \tag{2.4}$$

From Eqn. (2.4), since denominator is the same as for all the classes, the decision is mainly seen to be determined by the respective class-conditional pdfs $p(\mathbf{x}|C_k)$. Therefore, for a lower error rate in decision the pdf $p(\mathbf{x}|C_j)$ of each class C_j needs to be estimated as reliably as possible, from the training set. Estimation of the class pdf, $p(\mathbf{x}|C_j)$, in fact is a model development.

Here we describe general Gaussian mixture models are described where the class conditional-pdf of a class C_j is expressed as a linear combination of M_j Gaussian pdfs corresponding to component c_is as

$$p(x \mid C_j) = \sum_{i=1}^{M_j} P(c_i \mid C_j)\, p(x \mid c_i, C_j) \tag{2.5}$$

where multivariate Gaussian density for component c_i of class C_j given d-dimensional feature (column) vector x is

$$p(x \mid c_i, C_j) = \frac{1}{(2\pi)^{d/2} \mid \Sigma_{i,j} \mid^{1/2}} \exp[-\frac{1}{2}(x-\mu_{i,j})^T \Sigma_{i,j}^{-1}(x-\mu_{i,j})] \tag{2.6}$$

with the constraint $\sum_{i=1}^{M_j} P(c_i \mid C_j) = 1$. Above, |.| stands for determinant of its argument. The covariance matrix $\Sigma_{i,j}$ is a model parameter that can be computed by using maximum-likelihood estimation method with respect to the training samples belonging to class $c_i \subset C_j$ as

$$\hat{\Sigma}_{i,j} = \frac{1}{N_j - 1} \sum_{\forall x_i \in c_i, C_j} (x_i - \hat{\mu}_{i,j})(x_i - \hat{\mu}_{i,j})^T \tag{2.7}$$

where $\mu_{i,j}$ is the sample mean of the class C_j having N_j member samples and it is defined as

$$\hat{\mu}_{i,j} = \frac{1}{N_j} \sum_{\forall x_i \in c_i, C_j} x_i \tag{2.8}$$

Each component c_i can be initialized and formed by using either nearest-neighborhood, which is to be described next, or K-Means algorithm (Selim and Ismail 1984). Once the components have been obtained as such, raw a priori probabilities for each component can be calculated as $P^{(0)}(c_i \mid C_j) = \zeta_i^{(0)} = n_i / N_j$ where n_i is the number of samples contained by the component c_i with constraint $N_j = \sum_{i=1}^{M_j} n_i$.

The model parameters of each component conditional-pdf $p(x|c_i,C_j)$ were estimated in the maximum-likelihood sense. However, the bias in component pdf parameters can be remedied while they are being optimized by using the expectation-maximization (EM) algorithm proposed by Dempster, Laird and Rubin (1977). The EM algorithm is executed until the overall class likelihood function reaches a (local) um or a predefined number of iterations have been used. EM description of the i-th component conditional-pdf model parameters at the (m+1)-st iteration with $\zeta_i^{(m+1)} = P^{(m+1)}(c_i \mid C_j)$ and $P^{(m)}(i \mid x) = P^{(m)}(c_i \mid x)$ is as follows:

$$\zeta_i^{(m+1)} = \frac{1}{N_j} \sum_{\forall x \in c_i, C_j} P^{(m)}(i \mid x)$$

$$\mu_i^{(m+1)} = \frac{\sum_{\forall x \in c_i, C_j} x.P^{(m)}(i \mid x)}{N_j.\zeta_i^{(m+1)}} \tag{2.9}$$

$$\Sigma_i^{(m+1)} = \frac{\sum_{\forall x \in c_i, C_j} P^{(m)}(i \mid \mathrm{x}).(x - \mu_i^{(m+1)}).(x - \mu_i^{(m+1)})^T}{N_j.\zeta_i^{(m+1)}}$$

Although it is possible to utilize likelihood fitting procedures, such as Akaike's Information Criterion, AIC,(1974), the major problem for the EM algorithm lies in difficulty in choosing number of components for each class (Fessler and Hero 1994).

Non-convergence with small training sets and relatively long training time are other disadvantages of EM algorithm.

2.2.2 Neural Network Based Classifiers

Neural network classifiers benefit from the functional structure of human nervous system in learning for memory formation and reasoning (Temel 2010). There have been various neural networks structures which have been successfully applied in very diverse fields such as speech recognition,(Xiaoming and Baoyu 1998), image processing and coding etc (Dunstone 1994). Since this thesis is mainly concerned with Learning-vector quantization (LVQ), which is attributed to self-organized mapping (SOM) proposed and further developed by Kohonen, (1982, 1990, 1993), here we will review SOM foundations.

The main motivation for SOM is the biological plausibility in which brain is organized into regions that respond to different sensory excitation to reflect in localized dependency. Hence it is simulates biological systems' ability to learn and extract common attributes found in the retinal cortex, which can be represented as aggregated competing cluster centers (Kohonen, 1982). Moreover, SOM classifiers are regarded unsupervised since they are so arranged as to track (ir)regularities within input without supervision which makes them possibly the most commonly used neural network topology.

A SOM-based neural network consists of fully connected input and output layers. The output layer is also known the Kohonen layer. Figure 2.1 illustrates simply arranged one and two-dimensional SOM neural networks, respectively.

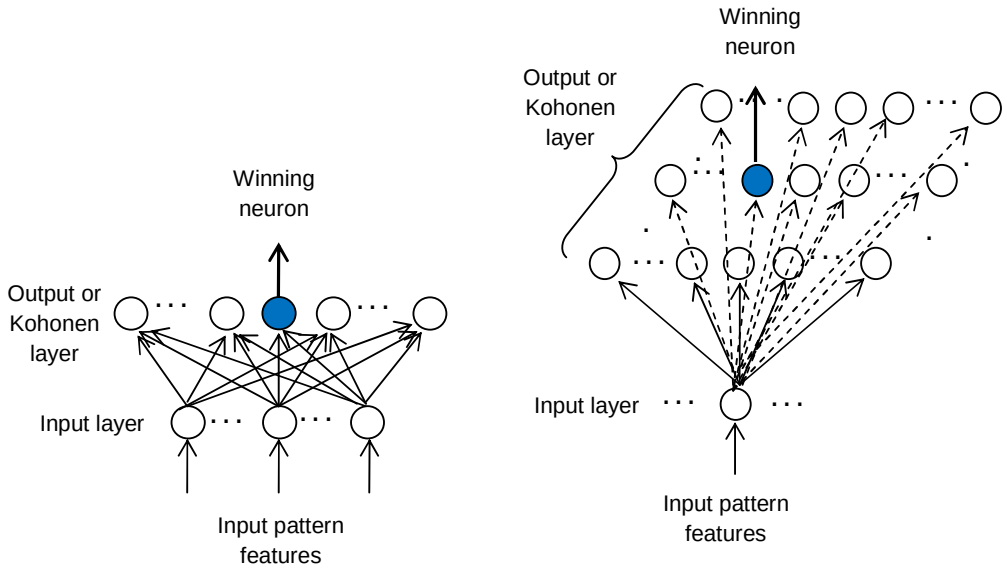

Figure 2.1 Architecture of the Kohonen networks, for one-dimensional, two-dimensional cases. In two-dimensional case the weight connections are depicted for a particular input layer neuron and other input layer neurons are similarly connected to output layer neurons, (Temel, 2010).

The connective value between input neurons and a particular output neuron i is represented as a vector $\mathbf{w}_i$ in an n-dimensional hyperspace. SOM networks operate and are structured in the form of competitive learning (Kohonen,1990). In competitive learning only the output layer neuron which resembles most or closest to the input stimulus gain precedence to respond/fire as the winner. Due to this nature SOM is a "the winner-take-all" paradigm.

SOM networks also consider the excitatory or inhibitory interaction between output neurons, which called the lateral-feedback. Such interaction in neuromorphic engineering and neuroscience is denoted as weight. However, there is a distinction between an ordinary neural connection weight and a lateral connection weight: lateral

feedback preserves topological arrangement of output neurons in localized dependency. Some lateral-feedback connections between output layer neurons for one-dimensional case are shown in Figure 2.2 in dotted lines.

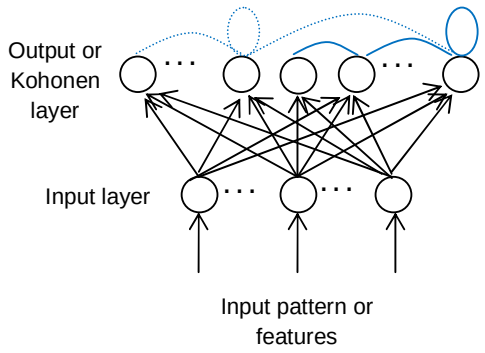

Figure 2.2 Some of the possible lateral feedback connections (dotted lines) in one-dimensional Kohonen layer.

Lateral-feedback weights are usually taken to vary in the form of a function which is expressed by the so-called Mexican hat function, h(.), shown in Figure 2.3.

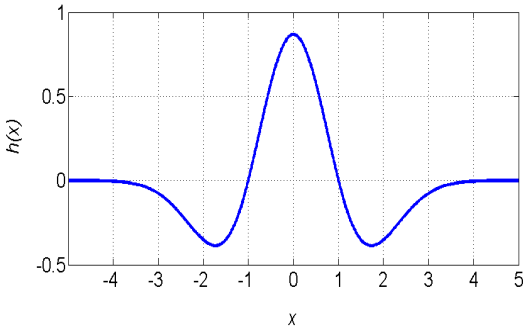

Figure 2.3 Mexican hat function, h(x). Note the regions of positive and negative reinforcement.

LVQ classifiers are formed on the basis of deterministic similarity measure between a group of weight vectors, **w**, and the training samples. Although the training seems to be deterministically executed it should be noted for each class, the weights are assumed to be picked from distinct stochastic random process (Bishop 2006).

As a well-known SOM algorithm, LVQ identifies categories which are known a priori group of classes for input patterns. Training phase of LVQ is an unsupervised process which is followed by a regulatory supervised phase. In training phase the weight also called codebook vectors are constructed while in the supervised phase each output neuron is then assigned a respective class label. It should be noted that in a simple LVQ realization each output neuron represents a category of a single class. The supervised stage is executed for iteratively readjusting codebook vectors under known labels by using the rule

$$w_j(k+1) = w_j(k) \pm \eta[x(k) - w_j(k)] \tag{2.10}$$

where $\mathbf{w}_j(k)$ is the weight vector between input and the output winner neuron at the k-th iteration, i.e. k-th input sample. Above rule is applied until convergence. The sign of η called the learning rate is taken to vary with iteration number and its sign is '+' if the input sample $\mathbf{x}(k)$ is correctly classified, i.e. reward otherwise '-', i.e. punishment. The training phase of LVQ is given below (Kasabov 1998):

1- Initialize the weight vectors, e.g. randomly, and choose an adequate value for the learning rate.
2- For input vector $\mathbf{x}(k)$ in the training set, find the winning neuron $\mathbf{w}_j$ with $d(\mathbf{x}(k),\mathbf{w}_j)<d(\mathbf{x}(k),\mathbf{w}_i)$ for all i and update it according to Equation (37) while other neurons remain unchanged.
3- Adjust the learning rate, e.g. reduce it as a function of iteration.
4- Terminate if $\mathbf{w}(k+1)= \mathbf{w}(k)$ for all weight vectors otherwise go to (2).

LVQ algorithm depicted above updates the winner only without modifying others. This property has been observed to cause poor topographic mapping. To remedy this shortcoming various versions of LVQ named have been proposed (Kasabov 1998).

3. DATA PRE-PROCESSING

Modern data analysis algorithms in machine learning need generalized information from samples. For a chosen learning model or algorithm, it is known that there has to be enough sample data available. If the dimension is 1-D and observation number is 2 for a learning model, then the same models needs 4 observations for 2-D, and 8 observations for 3-D (Verleysen and François 2005). This exponential increase in the number of features needed by the learning algorithm is referred to as 'curse of dimensionality'. It was proposed by Richard Bellman in (1961). Curse of dimensionality causes problems on models processing on high-dimensional data because there are more combinations of values of the features than can possibly be observed in a dataset. It leads learning algorithms to give unexpected results over a high -dimensional datasets (Verleysen and François 2005). In order to alleviate this shortcoming, feature reduction algorithms are deployed. In this chapter we will briefly review a major feature reduction algorithm called Principal Component Analysis (PCA).

While feature reduction algorithms serve as a tool to point out the saliency and cumulative characteristics of classes, classes may still need to be further processed for improved discrimination. Such a class separation process may involve linear and/or nonlinear relocation of dataset onto new feature coordinate axes. There have been numerous works for expressing optimality conditions toward applicability of dataset relocation (Duchene and Leclercq 1988; Baggenstoss 2004).As far as mathematical treatments are concerned, optimal relocation of dataset can be considered in terms of spectral decomposition which is the basis of feature reduction algorithms. However, since we aim to introduce methods for better class separation without reducing the number features in sample space we will focus on them in particular.

3.1 FEATURE REDUCTION

Major feature reduction methods include Independent Component Analysis (ICA), (Comon,1994), Karhunen Loeve' Expansion (KHE),(Matevosyan 1995) and Principal

Component Analysis (PCA) (Jolliffe 2002). Although these methods have advantages and disadvantages, PCA has been regarded to share some commonalities with unsupervised methods and successfully applied to numerous diverse complex problems, (Hu 2006; Jaruszewicz and Mandziuk 2002) where the size of data attributes leads to complicated classifier structures. PCA is in fact close relationship to singular value decomposition which is presented in Appendix A.

In its own theoretical foundations, PCA is a simple, non-parametric algorithm of extracting relevant information and reducing dimensions from a high-dimensional dataset (Jolliffe 2002) The mathematical definition of PCA can be given as an orthogonal linear transformation of the data which maps it into a new coordinate system. The first greatest variance of the data lies into first coordinate, the second greatest variance lies to the second coordinate and so on.

Assume that there is a set of m-dimensional observation data (column) vectors x_1, x_2, … , x_n. PCA algorithm is summarized as follows:

First step: The first step is to subtract the mean of data (**μ**) from each data vector to yield zero-mean vectors

$$\mathbf{\Phi}_i = \mathbf{x}_i - \boldsymbol{\mu} \tag{3.1}$$

and form a matrix n x m dimensional $\mathbf{A} = [\mathbf{\Phi}_1, \mathbf{\Phi}_2, \ldots \mathbf{\Phi}_n]$

Second step: Compute the covariance matrix C of the zero-mean data vectors **Φ** as

$$C = \frac{1}{N} \sum_{i,j=1}^{N} \mathbf{\Phi}_i \mathbf{\Phi}_j^T \tag{3.2}$$

Third step: Find the eigenvectors and eigenvalues of covariance matrix C. Then sort the eigenvalues in decreasing order and form the similarity matrix by using the corresponding eigenvectors

Eigenvalues of C $=\lambda_1 > \lambda_2 > \ldots > \lambda_m$

Eigenvectors of C = $[\mathbf{\upsilon}_1\ \mathbf{\upsilon}_2\ \ldots,\ \mathbf{\upsilon}_m]$

Forth step: Find a basis for transformation: Covariance matrix, C, is symmetric, hence its columns form a basis for transformation since any vector $\mathbf{\Phi}_i$ can be written as the linear combination of the eigenvectors as

$$\mathbf{\Phi_i} = \sum_{j=1}^{m} \omega_j \mathbf{\upsilon}_j \tag{3.3}$$

Fifth step: Select a value for the reduced dimension $\kappa << m$ and retain only κ largest eigenvalues. The selection of κ can be made according to a predefined threshold T as:

$$T < \frac{\sum_{i=1}^{\kappa} \lambda_i}{\sum_{i=1}^{m} \lambda_i} \tag{3.4}$$

3.2 DISCRIMINANT ANALYSIS FOR CLASS SCATTER

In conventional machine learning, discriminant analysis (DA) refers to determination of a group of functional hyperplanes which separate classes (Fukunaga 1990). For example, consider two normal multivariate distributions, $f_1(\mathbf{x}) : \aleph(\boldsymbol{\mu}_1, \Delta_1)$ and

$f_2(\mathbf{x}) : \aleph(\boldsymbol{\mu}_2, \Delta_2)$ which are depicted in Figure 3.1 for two-dimensional case. Given a sample data vector $\mathbf{x}=[x_1, x_2]$ the decision hyperline L can be expressed as a $\mathbf{w}\mathbf{x}^T+w_0$ where $\mathbf{w}$ and w_0 are to be determined with Bayesian decision rule, (Zhou, Wu and Liu 1998), in terms of means and covariances.

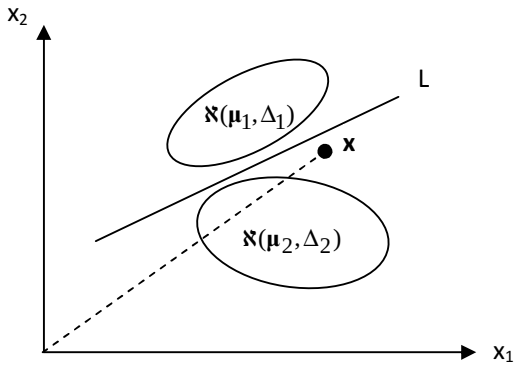

Figure 3.1 Two normal multivariate distributions, for two-dimensional case.

It should be noted that if the means of the class distributions are same or very close to each other, the line L above will not be determined uniquely or no such line will be available. Devijver and Kittler (1982) showed that the discriminatory information may lie in the variance of the data and LDA will fail to separate the classes. Specifically the worst case for LDA is the coincidence of class means.

Although it is a powerful classification algorithm, LDA is not always guaranteed to find the best discriminant directions efficiently(Zhu and Hastie 2003). The computation of eigen-decomposition can be very costly in case of high dimensional data. Moreover, if the number of the features is larger than the number of the training samples, singularity occurs since in such case the covariance matrices turn out singular, hence non-invertible. In such cases, SVD or PCA can be applied as a pre - process to overcome the singularity issue. However, these algorithms further increase the time complexity of the overall classification of LDA (Belhumeur, Hepanha and Kriegman 1997), which will be introduced next section as a separate section.

Since LDA can only classify with linear features, it is infeasible to apply it for a dataset of non-linear features. However, using kernel functions, the data can be projected into a linear space and then LDA is applied. Mika et al.(1999) proposed a Kernel Fisher Discriminant (KFD) method for two classes with non-linear features. Baudat et al (2001), investigated for the case of multi-class kernels, which has been coined as generalized discriminant analysis (GDA) since then.

No matter whether or not a feature reduction algorithm has been applied improved classification will be subject to how it represents the general scatter properties of individual classes. Especially the classes which overlap are difficult to generalize even with diverse training dataset. Therefore, it can be suggested that if the classes are separated from each other such that even linear discriminant functions can be utilized then it is possible to keep the complexity at the minimum.

Beside the notion of DA in determining the shape or behaviour of discriminatory data hyper plane, it can also be referred to in scatter properties of individual classes. Particularly the LDA, which is also named Fisher's discriminant analysis, has been applied to attain a mapping which augments the class separation in optimality terms. Within this prospective, the LDA algorithm itself can be considered as the classifier. As per, the scattering properties of a dataset along with LDA will be reviewed in the following section.

3.3 LINEAR DISCRIMINANT ANALYSIS (LDA)

The discriminant analysis method can be viewed as a general form of determining data hyperplane which separates classes. However, if the density profile of individual classes are known a priori, the hyperplane can be determined in terms of simple Bayesian decision rule (Zhou, Wu and Liu 1998). It should be noted that if samples are assumed to be independent then a simple assumption concerning the density estimation is to use generic Gaussian characteristics. It should be noted that such formalism only supposes that there are discriminative functions separating classes. However, if class densities are unknown and they overlap, then a need arises to devise a method to separate classes

enough so that suitable discriminate functions can be applied with approximate densities (Jieping etal 2004) A methodic approach which makes use of class separation with internal class condensation was proposed by Fisher in (1936).

Fisher's discriminant analysis method seeks an optimal linear separation of classes in data space. In order to describe the LDA algorithm consider a transformation

$$\mathbf{y=Ax} \tag{3.5}$$

where **x** is an input vector where matrix **A** is chosen such that in each class samples belonging to it are come closer to each other while the classes are better separated from each other. Thus, it is inferred that LDA aims at finding the best projection on data by minimizing the distance among the data points of same class and by maximizing the distance among the data points of different classes as seen in Figure 3.1. The problem of computation of the best projection on the training data can be fulfilled by applying an eigen-decomposition on the scatter matrices of data, which will be explained next.

The optimum projection matrix A is calculated using following equation by eigen-decomposition of scatter matrices.

$$S_B W = \lambda S_W W \tag{3.6}$$

Assuming that S_W is invertible(non-singular), then the equation above becomes :

$$S_W^{-1} S_B W = \lambda W \tag{3.7}$$

The rank of S_B is bounded with the number of the classes and can be at most $C-1$ .So there are at most $C-1$ non-zero eigenvectors according to non zero eigenvalues. Since data is transformed or mapped into a different feature set, the mean vector of each density profile will also be transformed. It can be shown that the class separation under

such conditions can be formulated by the following two optimization constraints (Fukunaga, 1990).

$$W = \arg\max_{A} \frac{\left|A^t S_b A\right|}{\left|A^t S_t A\right|} \tag{3.8}$$

or

$$W = \arg\max_{A} \frac{tr(\mathbf{A}^t S_b A)}{tr(A^t S_t A)} \tag{3.9}$$

where tr(.) and |.| are the trace and determinant of the matrix argument, respectively. Above, $\mathbf{S}_b$, and $\mathbf{S}_t$ refer to the total between-class scatter, and the total covariance matrices, respectively. Assuming N data samples represented as row vectors, **x**, with mean vector $\boldsymbol{\mu}$ coming from c classes they are defined as

$$\begin{aligned} S_b &= \frac{1}{N}\sum_{i=1}^{c} N_i(\boldsymbol{\mu}_i - \boldsymbol{\mu})^t(\boldsymbol{\mu}_i - \boldsymbol{\mu}) = \frac{1}{N}\sum_{i=1}^{c} N_i \boldsymbol{\mu}_i{}^t \boldsymbol{\mu}_i - \boldsymbol{\mu}^t\boldsymbol{\mu} \\ S_t &= \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}_i - \boldsymbol{\mu})^t(\mathbf{x}_i - \boldsymbol{\mu}) = S_b + S_w \end{aligned} \tag{3.10}$$

where $\boldsymbol{\mu}_i$ is the mean vector of the class label i, i.e. C_i, having N_i member patterns. The term $\mathbf{S}_w$ in Equation (3.10) is the total within-class scatter covariance matrix defined as

$$S_w = \frac{1}{N}\sum_{i=1}^{c}\left(\sum_{x_i \in C_i}^{N_i}(\mathbf{x}_i - \boldsymbol{\mu}_i)^t((\mathbf{x}_i - \boldsymbol{\mu}_i))\right) \tag{3.11}$$

an analytical solution to Equation (3.10) or (3.11) is obtained by exploiting the spectral decomposition, (Chen, Shan, and de Haan, 2009), in terms of statistical attributes such as relevant covariance matrices. However, as the dimension of input increases the spectral algorithms with eigen-decomposition methods become unattractive. For example, processing such data as vision and genomes, or networking which processes

large number of instantaneous sensory inputs in real-time may involve an alternative and straightforward, even albeit restrictive, method. Therefore, an appropriate approach needs to be developed.

3.4 SUMMARY

In this section we described major machine learning issues concerning classification and pre-processing are described in overview. Well-known parametric and non-parametric classification algorithms are reviewed. As a non-parametric classifier, nearest neighbor algorithm is reformulated while parametric classifiers are reviewed under the subcategories of probabilistic and deterministic neural classifiers. It should be noted that various categorization schemes are possible depending on the context. In order to improve the performance of the classification algorithm of interest with complicated data it is useful and most of the time mandatory to devise a pre-processing scheme if it is not possible to sacrifice the information content of the raw samples. Of major pre-processing schemes, feature reduction algorithms such as PCA as a particular form of singular value decomposition methods as a major tool which has been successfully applied in reducing the number of features in various fields, which also simplifies the complexity of the classifier. Another important pre-processing method which can be adopted in case the classes overlap and/or has shape with convexity and feature reduction algorithms do not contribute much, classes may need to be separated further. The section reviewed the relevant theory for such an objective and introduced the fundamental aspects of Fisher’s LDA algorithm.

4. SiStLDA ALGORITHM and ITS APPLICATION to CLASSIFICATION

4.1 INTRODUCTION

Despite considerable research efforts, which were discussed previously, to develop an expression for a linear transformation toward optimum class scatter properties, to our knowledge, no satisfactory progress has been known in literature. This is mainly due to difficulties in mathematical treatments involved. Most studies exploit certain ensemble characteristics such as multivariate normal densities for simplification and maximum likelihood principles to associate class statistics to distributions in analytical form (Miyamoto, Sato and Umayahara 1998). However, even for simple cases, such as two classes the treatment becomes rather involved. In case density profiles diverge from idealistic assumptions, it becomes impossible to yield appropriate transformation. It seems that possible scenarios to obtain generalized solution for a transformation which satisfies optimization turn out to be obscure. However, it is possible to benefit from the well-known covariance matrix properties of the whole dataset.

An important observation on available methodologies for expressing a transformation with optimality conditions is that even if such a transformation were found, it would not represent a particular class with respect to others. This pitfall is mainly due to eagerness to attain a global solution for whole dataset. However, global solution may deteriorate scatter characteristics of exceptional classes in some particular cases. Therefore, an efficient solution should be able to emphasize sample scatter of a given class relative to others. Considering globally analytic solutions above, a transformation which takes into account the individual class covariance is expected to yield better discriminative properties. Even if mathematically difficult and cumbersome, it can be intuitively claimed that it is possible to interpret Eqn. (3.8) in that instead of globally defined between- and/or within- class covariance characteristics, it would be more convenient to include individual class behavior into optimization rule.

In this section we will describe a new algorithm as an implicative solution to above shortcomings albeit ad-hoc based on individual class covariance matrices with respect

to overall dataset. It is shown that it operates highly efficiently in terms of class and dataset scatter behavior. The algorithm is tested for synthetic multivariate normal and real data. The resultant scheme is employed with previously described NN, EM-probabilistic and LVQ classifiers for both of data groups while FFMLP classifier performance for real odor data is also presented. The experimental results indicate that the proposed method provides the classifier being used with a much better generalization capability as well as suitability to real-time pattern recognition applications.

4.2 SiStLDA ALGORITHM

Considering class saliency relationship between overall dataset and individual classes demonstrated in (Härdle and Simar 2003), it may be inferred from the mathematical treatment concerning the saliency of feature vectors in individual classes with respect to overall dataset. In (Fukunaga, 1990), various forms of optimization criteria which correspond to different mappings were presented where individual class covariance matrices play a salient operation contrary to the overall dataset covariance matrix. Combining with treatment developed in (Härdle and Simar 2003, Temel 2010) proposed a new LDA mapping, called SiStLDA (individual class covariance matrix with respect to overall covariance matrix) which elaborates both quantity as a suitable transformation as :

$$\mathbf{y} = \mathbf{S}_i \mathbf{S}_t^{-1} \mathbf{x} \tag{4.1}$$

which is a modified version of the algorithm proposed in (Temel and Karlik 2007) where the transformation was given as the inverse of Eqn. (4.1). The advantage of the above transformation is that the matrix inverse operation applies only once, i.e. to the overall dataset, although being composed of at least one class, which reduces the operational overhead in case of large number of classes. Therefore new method will speed up pattern recognition task compared to that proposed in (Temel and Karlik 2007).

The SiStLDA algorithm was originally investigated with feed-forward multi-perceptron (FFMLP) neural network classifier for real odor data in (Temel, 2010), which will also be considered in this study. The labeled sample **x** belongs to class C_i with covariance matrix $\mathbf{S}_i$, i=1, 2, …, K. It should be noted that the pre-processing needs to be applied at both training and post-training phases while the algorithm needs matrix multiplication $\mathbf{s}_i\mathbf{s}_t^{-1}$ to be stored for all the classes given an input sample. These matrices need to be modified as new classes enter the pre-processing stage. It should be noted that the rank of the transformation is equal to the rank of its entries since all covariance matrices are of the same rank. Therefore, no information loss occurs and every input data is uniquely mapped to a respective feature vector.

4.3 APPLICATION of SiStLDA to CLASSIFICATION

In this section use of the LDA algorithm in Eqn. (4.1) is described with NN, EM probabilistic-Bayesian and LVQ classifiers in order to validate the performance improvement in classification tasks. Moreover, classification performance with real odor data is to be provided for FFMLP neural classifier without and with use of new method. Two groups of 100 distinct experiments were carried out with synthetic multivariate and real sensory odor data. For each group of experiments, two classifiers were designed, i.e. one with raw samples and one another with pre-processed samples. In each group of experiments the learning rate η is varied $\eta(k) = \eta_0 / k$ where the effect of initial learning rate η_0 is also studied for the values of η_0=0.05, 0.1, 0.15, 0.2. The classifier performance was assessed as the correctly classified test patterns over total test patterns with randomly initialized weights.

It should be noted that NN and EM- probabilistic classifiers are straightforwardly built in single-step, hence convergence is only used for depicting behavior of covariance matrix and learning parameter is needed in training phase. Their training is performed half the size of the class data. EM-probabilistic classifier was designed for single and two subclasses, respectively.

4.3.1 Classification of synthetic data with SiStLDA algorithm

In the first group of 100 experiments, two-dimensional three multivariate distribution each representing 100 samples were used where in each experiment, training samples were picked randomly from each class 100-sample reservoir. The distributions with mean vector **μ** and covariance matrix **Δ** are as follows:

Table 4.1: The distributions with mean vector **μ** and covariance matrix **Δ** for synthetic data.

	N_1	N_2	N_3
Mean (**μ**) :	[1 0]	[0 0]	[0 1]
Covariance matrix (**Δ**) :	$\begin{bmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{bmatrix}$	$\begin{bmatrix} 0.9 & 0 \\ 0 & 0.9 \end{bmatrix}$	$\begin{bmatrix} 0.9 & -0.1 \\ -0.1 & 0.9 \end{bmatrix}$

Figure 4.1 illustrates the scattering properties of above classes with raw and pre-processed data where x_i/y_i denotes the i-th coordinate for them, respectively.

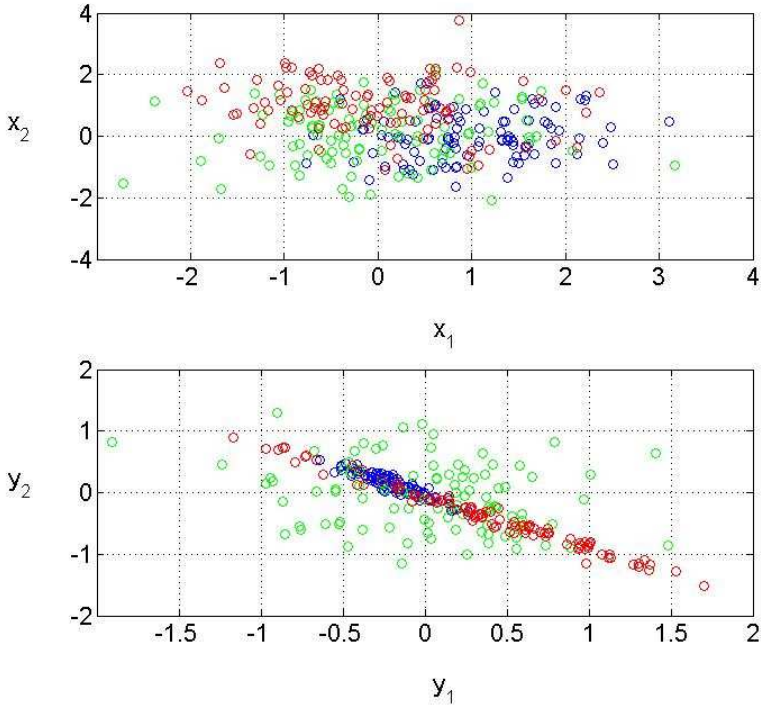

Figure 4.1: Scatter characteristics of $N_{1,2,3}$ (a) without, (b) with application of the SiStLDA pre-processing algorithm proposed.

The effect of proposed SiStLDA pre-processing algorithm on classification with the classifiers previously described, is also investigated. In each experiment, the number of 40 training samples and 60 testing samples were taken. Given the values of the initial learning rate parameter, η_0, Table 4.2 shows the mean/standard deviation of successful classification/the number of iterations for the training to converge with classification methods.

Table 4.2 : Classification performance with raw and pre-processed samples picked from 2D multivariate densities N1, N2 and N3.

	Mean. of success/Std. of Success/Number of iterations to converge			
Classifier	η_0=0.05	η_0=0.1	η_0=0.15	η_0=0.2
LVQ without pre-processing	46.1/4.6/20.9	43.9/5.2/17.1	40.1/6.3/16.4	38.3/6.1/18.6
LVQ with SiStLDA	76.3/3.2/8.6	78.2/3.0/8.0	71.0/4.1/8.9	72.2/3.9/4.4
NN without pre-processing	29.1/5.8/-			
NN with SiStLDA	69.3/4.2/-			
EM-probabilistic without pre-processing	Number of subclass=1 40.5/5.6/-		Number of subclass=2 46.1/5.0/-	
EM-probabilistic with SiStLDA	Number of subclass=1 70.2/4.4/-		Number of subclass=2 72.1/5.3/-	

As can be seen from the above table, pre-processing overwhelmingly improves the classification performance for the synthetic data chosen in terms of generalization. The algorithmic complexity of the training phase with SiStLDA is much less than that with raw data. The new algorithm also brings in robustness against the choice of initial learning rate value.

4.3.2 Classification of real odor with SiStLDA algorithm

The second group of experiments was carried out to assess the performance change of the classifiers with the SiStLDA algorithm for real odor data. In this group of experiments 32 samples collected from 20 different odorant perfumes were used.

Sampling was performed with two chemical sensors operating in real-time sampling mode. Figure 4.2 reveals the scattering characteristics raw and pre-processed dataset to be classified with respective classifiers where x_i / y_i refers to raw/pre-processed entry from sensor i=1, 2.

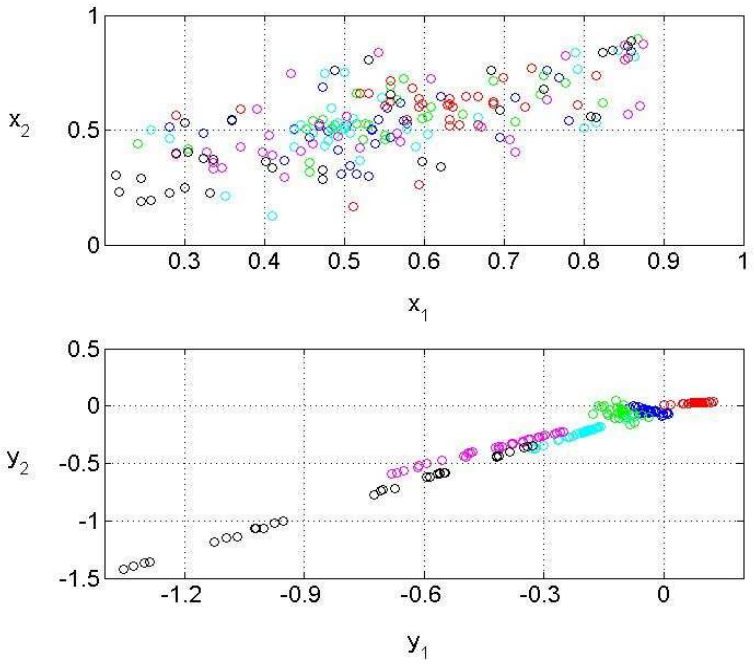

Figure 4.2 Scatter characteristics of five real odor classes as raw features (x_1, x_2), and pre-processed features (y_1, y_2) with application of the SiStLDA algorithm.

In order to evaluate overall odor recognition performance in statistical terms, raw samples were populated by using the boot-strapping, (Gong, 1986), to yield 10 times larger dataset for classifiers. Each classifier was trained and tested with datasets of populated raw and their pre-processed counterparts in 100 distinct experiments. For each experiment conducted, the classifiers were cross-validated using 9 labelled training subgroups and one unlabeled testing subgroup. Given predetermined values of η_0, Table 4.3 shows the mean/standard deviation of successful classification/the number of iterations for the training phase to converge with LVQ and FFMLP classifiers where data concerning the latter is provided by (Temel, 2010). For FFMLP classifier, the momentum term was taken 0.1. Similar to classification of synthetic data, Table also presents classification performance of non-parametric and EM-probabilistic classifiers.

Table 4.3: Classification performance with raw and pre-processed 2D samples picked from 20 odor classes.

Classifier	**Mean. of success/Std. of Success/Number of iterations to converge**			
	η_0=0.05	η_0=0.1	η_0=0.15	η_0=0.2
LVQ without pre-processing	41.2/5.6/21.6	38.1/6.2/24.2	38.4/6.3/23.9	34.3/7.0/24.6
LVQ With SiStLDA	73.8/3.9/8.1	75.9/4.0/9.2	69.2/3.8/11.1	70.9/3.3/10.2
FFMLP without pre-processing	50.6/6.2/12.5	53.5/6.6/14.2	48.3/8.1/15.4	50.4/7.6/13.4
FFMLP with SiStLDA	88.5/5.3/6/2	88.9/4.9/7.9	83.1/6.9/7.3	86.7/5.1/7.1
NN without pre-processing	30.6/6.3/-			
NN with SiStLDA	68.1/5.5/-			
EM-probabilistic without pre-processing	Number of subclass=1 39.3/5.9/-		Number of subclass=2 43.2/5.5/-	
EM-probabilistic with SiStLDA	Number of subclass=1 74.1/4.9/-		Number of subclass=2 73.8/4.7/-	

Similar to classification of synthetic data, classification with SiStLDA algorithm outperforms its counterpart without pre-processing scheme for real odor dataset. Considering high-level overlap between odor classes, the algorithm speeds up classification twice as fast as that without pre-processing for all classifiers used. Robustness against the variation is also preserved similar to synthetic dataset.

5. CONCLUSION

In this thesis, a recently proposed discriminant analysis method is studied as a pre-processing algorithm which can be used in real-time pattern recognition schemes. The algorithm is in the form of a class-dependent mapping/transformation. It has the advantage that it is class-adaptable and it does not involve spectral decomposition as opposed to theoretical development of conventional methods. Since for each class the transformation is solely determined by individual class statistical characteristics, i.e. respective class covariance, and due associative relation to overall dataset class covariance, the scheme is guaranteed to be invertible and unique for inputs of interest.

Considering the subjective parameter dependency of spectral decomposition methods they are not feasible most of the time, albeit theoretically optimal and hence loosely applied in real-time problems. Loss of information due to threshold assignment may be severe in conventional spectral methods. Although it seems ad-hoc the new algorithm alleviates this shortcoming. This advantage makes the algorithm suitable for generic application even the problem domain changes. The only issue which needs cautious is, the storage requirement for storing class covariances (or their inverses). In case a new class is added to the dataset it can be shown that modified dataset can be adapted easily.

The algorithm is validated for classification of synthetic and real data classification with 3 type of classifiers, which differs from each other with classification rules. The outcome of the classification demonstrates that the algorithm leads much better results. As expected thanks to more scarcely distributed data obtained from application of the SiStLDA algorithm, outcome is much more improved in all performance and implementation parameters in class identification compared to that without pre-processing. It is seen that if the problem domain is divisible into subcomponents, the algorithm also can be modified through algebraic manipulations so more complicated patterns can be identified. Moreover, the increase observed in the speed of the classifier with proposed algorithm makes it possible to implement real-time pattern recognition applications.

REFERENCES

1. Akaike, H., 1974. A new look at the statistical model identification. *Automatic Control,*19, pp. 716 - 723.

2. Baggenstoss, P.M., 2004. Class-specific classifier: avoiding the curse of dimensionality.*Aerospace and Electronic Systems Magazine,* 19, pp. 37 - 52

3. Baudat, G. & Anouar, F., 2001. Kernel-based methods and function approximation. *Neural Networks*, 2, pp. 1244 - 1249.

4. Belhumeur P. N., Hepanha J. P., & Kriegman D. J., 1997. Eigenfaces vs.fisherfaces: recognition using class specific linear projection. *Pattern Analysis and Machine Intelligence,*19, pp. 711-720.

5. Bellman, R., 1961. *Adaptive Control Processes: A Guided Tour*. Princeton University Press.

6. Bigun, J., 1992 . Unsupervised feature reduction in image segmentation by local Karhunen-Loeve transform. *11th IAPR International Conference on Pattern Recognition Methodology and Systems,*2, pp. 79 - 83.

7. Bishop C.M.,2006. *Pattern Recognition and Machine Learning*. Springer.

8. Chen W., Shan C. & Gerard de H., 2009. Optimal Regularization Parameter Estimation for Spectral Regression Discriminant Analysis. *Circuits And Systems For Video Technology,*19.

9. Comon P., 1994. Independent Component Analysis: a new concept?. *Signal Processing, Elsevier*, 36, pp. 287-314.

10. David C., Wang C., Andrew K. & Wong C., 1978. Classification of discrete data with feature space transformation. *Decision and Control including the 17th Symposium on Adaptive Processes,* 17 , pp. 774 - 778

11. Dempster, A.P., Laird N.M., & Rubin D.B., 1977. Maximum likelihood from incomplete data via the EM algorithm. *J. Royal Statiscal Soc., Ser. R*, pp. 1-38

12. Dequan Z., Liguang W. & Guosui L., 1998. Bayesian classifier based on discretized continuous feature space. *Fourth International Conference on Signal Processing Proceedings*, 2 , pp. 1225 - 1228.

13. Duchene, J. & Leclercq, S., 1988. An optimal transformation for discriminant and principal component analysis. *Pattern Analysis and Machine Intelligence*, 10, pp. 978 - 983.

14. Dunstone, E.S., 1994. Image processing using an image approximation neural network. *Image Processing*, 3, pp. 912 - 916.

15. Zhang F., Zhao Y.J. & Jun F. 2009. Unsupervised feature selection based on feature relevance. *International Conference on Machine Learning and Cybernetics*,1, pp.487 - 492.

16. Fessler, J.A. & Hero, A.O., 1994. Space-alternating generalized expectation-maximization algorithm. *Signal Processing*, 42 , pp. 2664 - 2677.

17. Foley, D.H. & Sammon, J.W., Jr., 1975. An Optimal Set of Discriminant Vectors. Computers,C-24, pp. 281 - 289.

18. Fukunaga K., 1990.*Introduction to statistical pattern recognition 2nd ed.* Academic Press Professional.

19. Gong, G. (1986). Cross-validation, the jackknife, and the bootstrap: Excess error estimation in forward logistic regression. *J. Amer. Statist. Assoc*., 81, pp. 108-113.

20. Shakhnarovich G. , Darrell T. & Indyk P. 2006.Nearest-Neighbor Methods in Learning and Vision Theory and Practice**.** *MIT Press.*

21. Härdle, W. & Simar, L., 2003. *Applied multivariate statistical analysis*. Springer Verlag, NewYork.

22. Jaruszewicz, M. & Mandziuk, J., 2002. Application of PCA method to weather prediction task. *Neural Information Processing*,5, pp. 2359 - 2363.

23. Jie Hu, 2006. Application of PCA Method on Pest Information Detection of Electronic Nose. *Information Acquisition*, pp. 1465 - 1468

24. Jieping Y., Tao L., Tao X. & Janardan, R., 2004. Using uncorrelated discriminant analysis for tissue classification with gene expression data. *Computational Biology and Bioinformatics*, 1,pp. 181 - 190.

25. Jimenez-Rodriguez, L. O., Arzuaga-Cruz, E. & Velez-Reyes, M., 2007. Unsupervised Linear Feature-Extraction Methods and Their Effects in the Classification of High-Dimensional Data. *Geoscience and Remote Sensing*, 45, pp. 469 - 483.

26. Jolliffe, I.T., 2002. *Principal Component Analysis* 2nd ed.. Springer Series in Statistics

27. Kasabov, N. K., 1998. *Foundations of Neural Networks, Fuzzy Systems, and Knowledge Engineering*, (2nd ed.) . Cambridge, MA: MIT Press. Philapdelphia: SIAM.

28. Kittler, J. & Devijver, P. A., 1982. Statistical Properties of Error Estimators in Performance Assessment of Recognition Systems. *Pattern Analysis and Machine Intelligence*, PAMI-4, pp. 215 - 220.

29. Kohonen, T., 1986. Learning Vector Quantization for Pattem Recognition. *Technical Report No. TKK-F-A601, Helsinki University of Technology*.

30. Kohonen, T., 1982. Self-organized Formation of Topographically Correct Feature Maps. *Biological Cybernetics*, 43, pp. 59-69.

31. Kohonen, T., 1990. The Self-organizing Map. *Proceedings of IEEE*, 78,pp. 1464-1480.

32. Koutsougeras, C. & Srikanth, R., 1993. Data transformation for learning in feedforward neural nets. *Fifth International Conference on Tools with Artificial Intelligence*, pp. 22 - 29.

33. Kung, S.Y. & Mao, W.D., 1991 . Competition-based supervised learning algorithm for nonlinear discriminant functions. *Acoustics, Speech, and Signal Processing*, pp. 1073 - 1076.

34. Kwak, N. & Chong-Ho C. 2002. Input feature selection for classification problems. *Neural Networks*, 13 , pp. 143 - 159

35. Mahalanobis, P. C., 1936. On the generalised distance in statistics. *Proceedings of the National Institute of Sciences of India*, 2, pp.49–55.

36. Martinez, A.M.& Kak, A.C., 2001. PCA versus LDA. *Pattern Analysis and Machine Intelligence*, 23, pp. 228 - 233

37. Matevosyan, A.K., 1995. Karhunen-Loeve-like expansions. *Computer-Based Medical Systems*, pp. 325 - 327.

38. Mika, S., Ratsch, G., Weston, J., Scholkopf, B. & Mullers, K.R., 1999. Fisher discriminant analysis with kernels. *Neural Networks for Signal Processing*, pp. 41 - 48.

39. Miyamoto, S., Sato, M. & Umayahara, K., 1998. Generalization of discriminant analysis for possibility distributions. *Knowledge-Based Intelligent Electronic Systems*,3, pp. 177 - 182.

40. Nadas, A., 1985. Optimal solution of a training problem in speech recognition. *Acoustics, Speech and Signal Processing*, 33, pp. 326 - 329.

41. Fisher R. A., 1936. The use of multiple measurements in taxonomic problems. *Annals Eugen*,7 ,pp. 179-188.

42. Satorius E. & Brady R., 1988. SETI Signal Processing. *Signals, Systems and Computers Twenty-Second Asilomar Conference*, pp. 194 - 198.

43. Selim, S. Z. & Ismail, M. A., 1984. K-Means-Type Algorithms: A Generalized Convergence Theorem and Characterization of Local Optimality. *Pattern Analysis and Machine Intelligence*, PAMI-6, pp. 81 - 87.

44. Tattersall, G.D., Chichlowski, K. & Limb, R., 1992. Pre-processing and visualisation of decision support data for enhanced machine classification. *First International Conference on Intelligent Systems Engineering*, pp. 275 - 280.

45. Temel T., 2010. System And Circuit Desıgn For Biologically-Inspired Learning. *To Be Publıshed In September 2010 By Igi-Global Publisher*.

46. Temel, T., & Karlik, B. 2007. An improved Odor Recognition System Using Learning Vector Quantization with a New Discriminant Analysis. *Neural Network World*, 4(7), pp. 287-294.

47. Verleysen M. & Damien F., 2005. The curse of dimensionality in data mining and time series prediction. *Computational Intelligence and Bioinspired System*.

48. Xiaoming W. & Zheng B., 1998 . A new neural network oriented speech recognition. *International Conference on Communication Technology Proceedings*, 2 .

49. Zhu, M. & Hastie, T. J., 2003. Feature extraction for nonparametric discriminant analysis. *Journal of Computational & Graphical Statistics*.

APPENDIX A - SINGULAR VALUE DECOMPOSITION

SVD can be evaluated from three different points of view. First of all, it is a method for transforming correlated variables into a set of uncorrelated variables to reveal any other relationships between the original data items. It is also a method for exposing and ordering the dimensions along the direction of the data points at most variation. And finally SVD can be used as feature reduction method by using the best approximation of the data points using fewer dimensions.

Singular value decomposition is a theorem of linear algebra which evaluates a rectangular $n \ x \ m$ matrix as a dot product of three different matrices.

Let us consider an $n \ x \ m$ matrix is **A** :

$$A_{n \ x \ m} = \ U \ S \ Z^T$$

where

U is a $m_x m$ orthogonal matrix those columns are orthonormal eigenvectors of AA^T .

Z is a $n_x n$ orthogonal matrix those columns are orthonormal eigenvectors of $A^T A$.

S is a diagonal matrix those columns are the square roots of eigenvalues from U or Z in descending order.