TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

LEI XU

EXPLOITING SALIENCY INFORMATION IN DISCRIMINANT SUBSPACE LEARNING

Master of Science Thesis

Examiner: Alexandros Iosifidis, Moncef Gabbouj
Examiner and topic approved by the Faculty of Computing and Electrical Engineering

on 20 April 2017

ABSTRACT

TAMPERE UNIVERSITY OF TECHNOLOGY
Master's Degree Programme in Signal Processing
XU, LEI: Exploiting saliency information in discriminant subspace learning
Master of Science Thesis, 60 pages
June 2017
Major: Signal Processing
Minor: Signal Processing
Examiner: Alexandros Iosifidis, Moncef Gabbouj
Keywords: Linear Discriminant Analysis, Weighted LDA, Saliency Estimation

The objective of this thesis is to investigate a new linear discriminant analysis method, which could overcome underlying drawbacks of traditional Linear Discriminant Analysis (LDA) and other LDA variants targeting problems involving imbalanced classes. Traditional LDA sets assumptions related to (Gaussian) class distribution and neglects influence of outlier classes or sample structure inside class, that might affect performance. We exploit intuitions coming from a probabilistic interpretation of saliency in order to redefine the between-class and within-class scatters in a more robust, with respect to outliers and class cardinality invariant, manner. The proposed method is named as Saliency-based weighted LDA ($SwLDA$).

We propose several associated $SwLDA$ variants and evaluate them on six publicly available facial image and three imbalanced datasets. Comparing to traditional LDA and other weighted LDA variants, the proposed $SwLDA$ shows certain improvements on facial image classification and class-imbalanced classification problems. The best improvement for our approaches is 11.14% on BU dataset, comparing to traditional LDA.

PREFACE

This thesis was completed at the Multimedia Research Group, in the Laboratory of Signal Processing of Tampere University of Technology. The duration of this project is from January to May 2017.

Several persons have supported me to accomplish this thesis. I would therefore firstly like to express my deepest thanks to my two supervisors, Prof. Moncef Gabbouj and Dr. Alexandros Iosifidis, for their excellent guidance and patience during this process. They gave me the opportunity and support to explore academic work in machine learning area. Furthermore, I would also like to thank all young and smart friends I have met in this school during my studies. Their diligence and passionate attitude towards academic work have been inspiring me for my further studying.

I would like to thank Prof. Quanli Liu from Dalian University of Technology, Prof. Zhengqi Zheng and my tutor Chuanyu Jin from East China Normal University for recommending me for studies in Tampere University of Technology.

Finally, I would like to thank my wonderful and super smart husband, who always is full of curiosity about knowledge, always encourages me to realize my dreams and always supports me unconditionally.

Lei Xu
June 2017

CONTENTS

ABBREVIATIONS

LDA	Linear Discriminant Analysis
SwLDA	Saliency-based weighted Linear Discriminant Analysis
SW	Within-class scatter matrix of traditional LDA method
SB	Between-class scatter matrix of traditional LDA method
ST	Total scatter matrix of traditional LDA method
Sw	Within-class scatter matrix of LDA variants
Sb	Between-class scatter matrix of LDA variants
St	Total scatter matrix of LDA variants
PSE	Probabilistic Saliency Estimation

1. INTRODUCTION

Pattern Recognition is a sub-field of Computer Science aiming at the process and analysis of various kinds of information (found in, e.g. images, videos and audios, etc), in order to describe, distinguish, classify and interpret phenomena or patterns. It is a significant part of information science, artificial intelligence and machine learning. Pattern recognition methods can be categorized in syntactic and statistical ones. As pattern recognition has been playing an increasingly important role for machine learning techniques during the last decades, statistical methods have been proved to be superior to syntactic methods in various notable applications, such as automatic translation [1] and object/face recognition [2].

Usually, information is collected by analog sensors. In order to exploit it by computers, we need to convert it into a digital form and describe it using vectorial representations. For instance, we can use a D-dimensional ($D = m \times n$) vector $\mathbf{x} = (x_1, ..., x_D)^T$ to present an image with resolution of $m \times n$ pixels, where $x_i, i \in \{1, 2, ..., D\}$ corresponds to the intensity value of the i-th pixel of this image. Furthermore, a dataset formed by N images can be represented by a matrix $\mathbf{X}_{D\times N} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N]$.

A (simplified) pattern recognition work-flow usually involves of three parts [3]: (a) data collection (using a sensor), (b) feature extraction and selection and (c) data analysis for classification, regression or clustering. A sensor converts analog signals into digital data [4]. We not only need to convert analog signals into digital data, but also to discard irrelevant and redundant information, due to the high dimension of raw data. Thus, data should be processed using transformation techniques to extract intrinsic dimensions or select subsets of a given set of variables [3]. Then, we can employ learning algorithms taking such processed data as input to obtain an estimated result, which can possibly be considered as reference to adjust previous steps of optimization.

There are various learning algorithms in pattern recognition for classification tasks, such as support vector machines [5], decision trees [6] and k-nearest neighbors classifiers [7]. Linear Discriminant Analysis (LDA) [8], as a traditional statistical machine learning technique, has been employed for several classification tasks, such as facial image analysis for identification and expression recognition [9], human action recognition [10; 11], and person identification [12], due to its effectiveness in

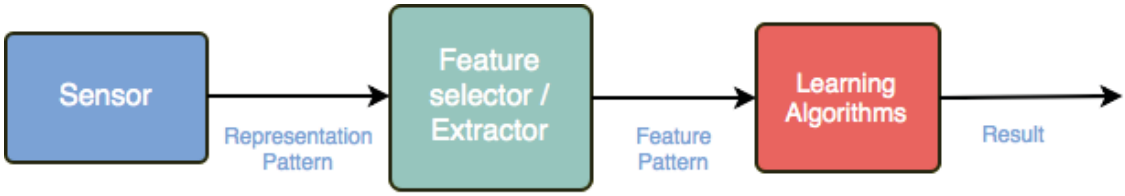


Figure 1.1: Pattern recognition work-flow

reducing dimensions and extracting discriminative features.

LDA is used to define an optimal linear projection, by means of Fisher's discriminant criterion optimization. Once the optimal projection is obtained, high dimensional data can be mapped into the discriminant subspace for classification. Despite the widespread application of traditional LDA, its performance is affected by several issues related to its underlying assumptions. Traditional LDA represents each class with the corresponding class mean in order to define the class compactness and discriminates between classes based on the scatter of these classes representations with respect to the total data mean. Such a class discrimination definition may cause large overlaps of neighboring classes [13], and receive a sub-optimal result, since an outlier class being far from the others dominates the solution [13].

Furthermore, in traditional LDA all classes equally contribute to the within-class scatter definition [14] based on the assumption of the same Gaussian distribution for all classes. This assumption overemphasizes well-separated outlier classes, which should have lower contribution in the overall within-class scatter definition. A method that automatically determines optimized class representations for LDA-based projections was proposed in [15; 16]; however, it also suffers from the class imbalanced problems discussed above. In order to overcome aforementioned drawbacks of traditional LDA, extensions imposing weighting strategies for the definition of the within-class and between-class scatters have been proposed in [17; 18; 19; 20]. In these methods, the weighting factors incorporated to the scatter matrices definitions are based on class statistics, e.g. class cardinality, and class representation is still assumed to be the class mean.

In order to improve the performance of existing LDA variants, a novel extension of LDA that exploits intuitions from saliency [21] is proposed in this thesis. A probabilistic criterion is formulated in order to express the *saliency of a sample within its original class* following a probabilistic saliency estimation framework [22]. Such a saliency definition is naturally expressed by graph notation, in which several types of graphs can be exploited. Both fully connected and k-Nearest Neighbor (k-NN) graphs are considered. After defining the probability of each class sample to belong to the corresponding class, this information is used in order to define class represen-

tations, as well as new within-class and between-class scatters. Compared to traditional LDA and its weighted variants, the proposed Saliency-based weighted LDA ($SwLDA$) has shown enhanced performance on facial image classification problems and class-imbalanced classification problems.

The remainder of this thesis is organized as follow. Chapter 2 presents the theoretical background of this work. This chapter provides a detailed literature review on related LDA methods and description on theories related to $SwLDA$. Chapter 3 illustrates the derivation process of the proposed $SwLDA$. Chapter 4 provides the experimental evaluation of the related methods. Finally, chapter 5 concludes this thesis and discusses topics of further study.

2. THEORETICAL BACKGROUND

This section describes theoretical background of *SwLDA*. As mentioned in the Introduction of this thesis, *SwLDA* is inspired by weighted LDA and saliency estimation methods. Thus, it is imperative to demonstrate related theories in detail in advance. Firstly, section 2.1 describes the general concept of supervised learning. Then, similarity measure and data normalization are demonstrated in sections 2.2 and 2.3, respectively. The principles and shortcomings of LDA are presented in section 2.4. Weighted LDA variants are described in section 2.5 in detail. Section 2.6 describes one-class classification. Since, as will be described in section 2.7, it can be one of the interpretations of probability-based saliency estimation.

2.1 Supervised Learning

Learning algorithms in Pattern Recognition can be categorized in supervised and unsupervised ones, according to whether training labels are exploited. Generally, in a supervised learning task, we assume a training dataset $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$ is available, where N is the cardinality of this set. Each sample (or instance) $\mathbf{x}_i = (x_{i1}; x_{i2}; ...; x_{iD})$ in this set is a vector living in a D-dimensional feature space $\mathbb{R}^D$, i.e. $\mathbf{x}_i \in \mathbb{R}^D$. $(\mathbf{x}_i, y_i)$ presents the input-output pairs of i-th sample, where $y_i \in \mathcal{Y}$ is the label of sample $\mathbf{x}_i$ and $\mathcal{Y}$ is the label set.

Supervised learning models can be categorized in classification and regression ones. Regression produces a continuous result using continuous function; while, classification maps the inputs into discrete labels. Fig. 2.1 shows an overview of the work-flow followed in a classification task.

Generally, a mapping algorithm $f : \mathcal{X} \mapsto \mathcal{Y}$ is established from input space $\mathbb{R}^D$ to the label set $\mathcal{Y}$ by means of learning the mapping on the training dataset $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), ..., (\mathbf{x}_N, y_N)\}$. Classification task is named as binary classification, when it involves only two classes. Multi-class classification involves multiple classes. We usually present the label set $\mathcal{Y}$ as $\{-1, +1\}$ or $\{0, 1\}$ in a binary classification task and $|\mathcal{Y}| > 2$ in a multi-class classification task. Once the parameters of the classification algorithm are determined, they are employed to predict the label of a new test sample.

Learning algorithms aim to perform well on new data (test set) rather than to just work well on the training set. Thus, a learning algorithm should be generalizable

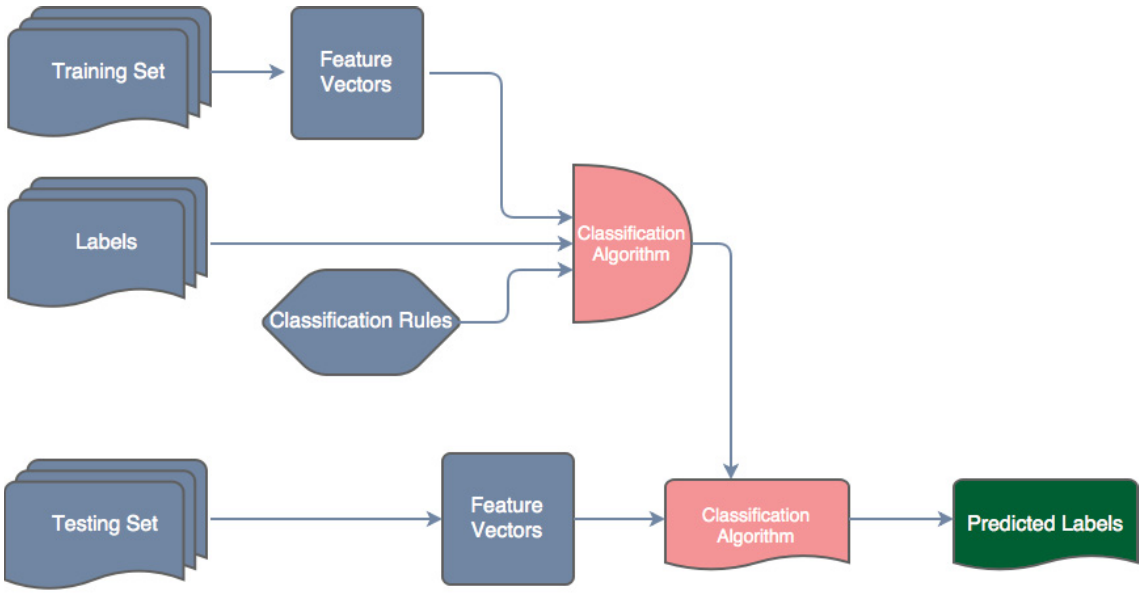


Figure 2.1: Classification work-flow

to unseen data. It is normally assumed that every instance in sample space is drawn from an unknown distribution. During data collection, we obtain each sample independently, so that our dataset is independently and identically distributed. In general, the more training samples a learning algorithm is able to process, the more information it receives about this distribution, and it is more likely to achieve better generalization on unseen data.

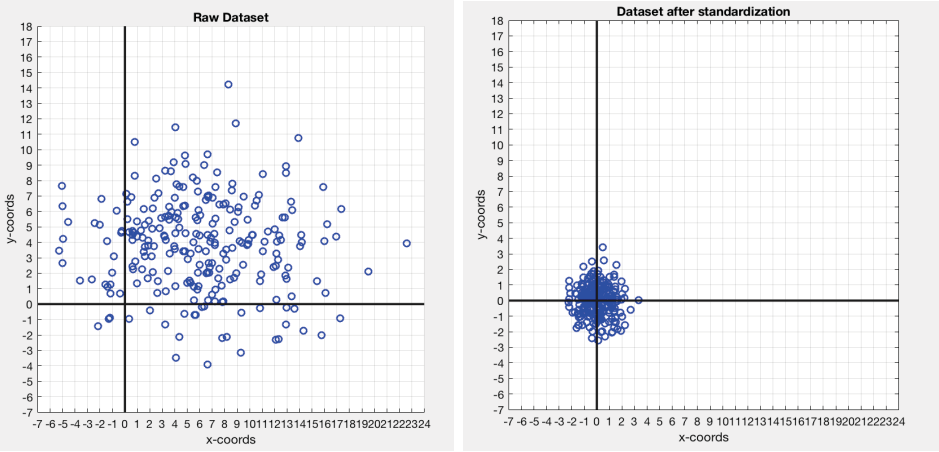
2.2 Similarity Measure

Similarity measure plays an important role in machine learning. It is usually associated with a distance function in the feature space $\mathbb{R}^D$. Distance function has various versions, according to different distance metrics. The most common distance metric is Minkowski distance, which is defined as follows:

$$dist(\mathbf{x}_i, \mathbf{x}_j) = \left(\sum_{u=1}^{d} |x_{iu} - x_{ju}|^p \right)^{\frac{1}{p}}, \tag{2.1}$$

where $p \geq 1$. For $p = 2$, Minkowski distance equals to Euclidean distance:

$$dist(\mathbf{x}_i, \mathbf{x}_j) = \|\mathbf{x}_i - \mathbf{x}_j\|_2 = \sqrt{\sum_{u=1}^{d} (x_{iu} - x_{ju})^2}. \tag{2.2}$$



(a) Raw dataset (b) Standardized dataset

Figure 2.2: Example of dataset standardization

Moreover, for $p = 1$, Minkowski distance is equal to Manhattan distance:

$$dist(\mathbf{x}_i, \mathbf{x}_j) = \|\mathbf{x}_i - \mathbf{x}_j\|_1 = \sum_{u=1}^{d} |x_{iu} - x_{ju}|. \tag{2.3}$$

After defining the distance between two samples $\mathbf{x}_i$ and $\mathbf{x}_j$, their similarity can be defined in various ways, e.g. using the inverse, i.e. $Sim(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{dist(\mathbf{x}_i, \mathbf{x}_j)}$, or the exponential, i.e. $Sim(\mathbf{x}_i, \mathbf{x}_j) = e^{-dist(\mathbf{x}_i, \mathbf{x}_j)}$, functions.

2.3 Normalization

Normalization is an essential pre-processing step in many machine learning tasks. It can be used to convert a scattered dataset into a concentrated one. There exist various normalization approaches, e.g. feature scaling and dataset standardization. The objective of feature scaling is to rescale the entire dataset to an interval [0, 1] or [-1, 1] using transformation functions.

In addition, dataset standardization aims to center the entire dataset to the origin of $\mathbb{R}^D$ and to scale each data dimension to unit deviation, as shown in Fig. (2.2). Zero-mean normalization is a common method for dataset standardization.

2.4 Linear Classifier

Linear models are notable for their simplicity and practicality for machine learning tasks. Linear models aim to generate a prediction function through a linear combination of the various attributes in training dataset. A linear model with parameters $\mathbf{w} = (w_1; w_2; \cdots; w_D)^T$ and $b \in \mathbb{R}$, when applied to a vector $\mathbf{x} = (x_1; x_2; \cdots; x_D)^T$,

is expressed as:

$$f(\mathbf{x}) = w_1x_1 + w_2x_2 + ... + w_Dx_D + b. \tag{2.4}$$

Above equation can be expressed in a vector form, as follows:

$$f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b. \tag{2.5}$$

The model's parameters $\mathbf{w}$ and b are determined through learning on the training data. Linear models can be categorized in regression and classification models, according to their learning purposes. Linear regression models aim to learn a mapping to a real value for a test sample, and the goal of a linear classifier is to assign a class label to a test sample.

2.4.1 Linear Discriminant Analysis

LDA, also referred to as Fisher's Linear Discriminant, is a classical dimensionality reduction technique. It was proposed originally by Ronald Fisher in his work " The Use of Multiple Measurements in Taxonomic Problems " for two-class tasks [23]. In Rao's work " The Utilization of multiple measurements in problems of biological classification ", two-class LDA is extended to multi-class task for the first time [24].

As a dimensionality reduction technique, the goal of LDA is to define a linear projection, mapping the input space into a discriminant subspace. The principle of Fisher's discriminant analysis is straightforward. Given a dataset formed by two classes, an optimal projection vector $\mathbf{w} \in \mathbb{R}^D$ is obtained to map the D-dimensional input into a line, where samples from the same class are clustered as much as possible, and different classes are separated from each other. As shown in Fig. (2.3), samples can not be separated well along axis $\mathbb{X}_1$. When the optimal Fisher's discriminant $\mathbf{w}$ is determined and the samples are mapped on it, the two classes are perfectly separated.

Let us assume that we have a two-class dataset $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$ and its label set is $\mathcal{Y} = \{y_i\}_{i=1}^N$, where $y_i \in \{0, 1\}$ is the label of sample $\mathbf{x}_i \in \mathbb{R}^D$. $\boldsymbol{X}_c$, $\boldsymbol{\Sigma}_c$ and $\boldsymbol{\mu}_c$ denote class data, covariance matrix and mean vector, respectively, for $c \in \{0, 1\}$. Once mapping original dataset into $\mathbf{w}$, the centers of the two classes are obtained by $\mathbf{w^T\boldsymbol{\mu}_0}$ and $\mathbf{w^T\boldsymbol{\mu}_1}$. In addition, the covariance matrix of each class after projection can be described as $\mathbf{w^T\boldsymbol{\Sigma}_0 w}$, $\mathbf{w^T\boldsymbol{\Sigma}_1 w}$ and are used to measure variability of each class.

An optimal project vector $\mathbf{w}$ should satisfy the principles of Fisher's discriminant analysis. It should minimize the class covariances (minimal $\mathbf{w^T\boldsymbol{\Sigma}_0 w} + \mathbf{w^T\boldsymbol{\Sigma}_1 w}$), and maximize the distance of the two class centers (maximal $\|\mathbf{w^T\boldsymbol{\mu}_0} - \mathbf{w^T\boldsymbol{\mu}_1}\|_2^2$). In order to achieve these two goals simultaneously, the following objective function

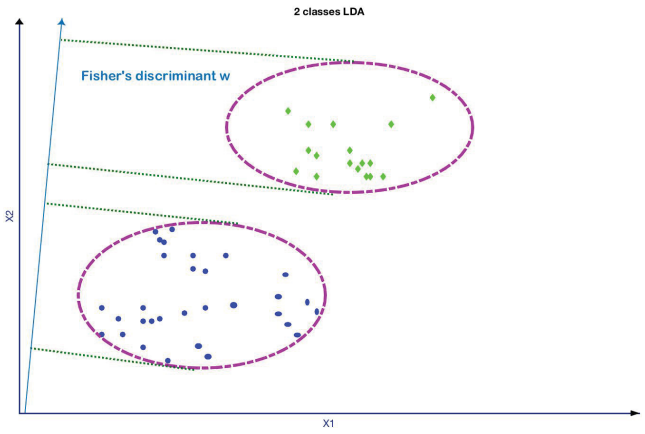


Figure 2.3: 2 classes Fisher's Discriminant Analysis

can be formulated:

$$\begin{aligned}\mathbf{J}(\mathbf{w}) &= \max_{\mathbf{w}} \frac{\|\mathbf{w}^T\boldsymbol{\mu}_0 - \mathbf{w}^T\boldsymbol{\mu}_1\|_2^2}{\mathbf{w}^T\boldsymbol{\Sigma}_0\mathbf{w} + \mathbf{w}^T\boldsymbol{\Sigma}_1\mathbf{w}}, \\ &= \max_{\mathbf{w}} \frac{\mathbf{w}^T(\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1)(\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1)^T\mathbf{w}}{\mathbf{w}^T(\boldsymbol{\Sigma}_0 + \boldsymbol{\Sigma}_1)\mathbf{w}},\end{aligned} \tag{2.6}$$

where $\boldsymbol{\Sigma}_0+\boldsymbol{\Sigma}_1$ is defined as within-class scatter matrix $\mathbf{S}_W$, and $(\boldsymbol{\mu}_0-\boldsymbol{\mu}_1)(\boldsymbol{\mu}_0-\boldsymbol{\mu}_1)^T$ is defined as between-class scatter matrix $\mathbf{S}_B$. Thus, Eq. (2.6) can be rewritten as:

$$\mathbf{J}(\mathbf{w}) = \max_{\mathbf{w}} \frac{\mathbf{w}^T\mathbf{S}_B\mathbf{w}}{\mathbf{w}^T\mathbf{S}_W\mathbf{w}}. \tag{2.7}$$

LDA aims to maximize the objective function in Eq. (2.7), which is the Generalized Rayleigh Quotient of $\mathbf{S}_B$, $\mathbf{S}_W$ [25]. Moreover, both its numerator and denominator are quadratic formulas about $\mathbf{w}$, so this maximal solution is rather related to $\mathbf{w}$'s direction than its scale. To solve this problem, Eq. (2.7) is transformed to the constrained optimization problem in Eq. (2.8), which can be solved using Lagrangian optimization as in Eq. (2.9), with respect to $\mathbf{w}$. The solution of Eq. (2.9) is equivalent to Eq. (2.10).

$$\max_{\mathbf{w}} \quad \mathbf{w}^T\mathbf{S}_B\mathbf{w} \qquad s.t. \quad \mathbf{w}^T\mathbf{S}_W\mathbf{w} = 1. \tag{2.8}$$

$$L = \mathbf{w}^T\mathbf{S}_B\mathbf{w} + \lambda(\mathbf{w}^T\mathbf{S}_W\mathbf{w} - 1). \tag{2.9}$$

$$\mathbf{S}_B\mathbf{w} = \lambda\mathbf{S}_W\mathbf{w}, \quad \lambda \geq 1. \tag{2.10}$$

Eq. (2.10) is a generalized eigenvalue problem, $\mathbf{w}$ can be generated by eigenvalue

decomposition. It is not difficult to show that the Eq. (2.10) can be transformed to Eq. (2.11), due to consistent directions of $\mathbf{S}_B\mathbf{w}$ and $\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1$:

$$\mathbf{S}_B\mathbf{w} = \lambda(\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1). \tag{2.11}$$

By replacing $\mathbf{S}_B\mathbf{w}$ in Eq. (2.10) with Eq. (2.11), $\mathbf{w}$ can be obtained by:

$$\mathbf{w} = \mathbf{S}_W^{-1}(\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1). \tag{2.12}$$

LDA can be extended to multi-class problems. In a multi-class problem, a training dataset $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$, $\mathbf{x}_i \in \mathbb{R}^D$ with C classes is labeled by $\mathcal{Y} = \{y_i\}_{i=1}^N$, where $y_i \in \{1, \cdots, C\}$. The i-th sample is mapped from the input space $\mathbb{R}^D$ to a discriminant subspace $\mathbb{R}^d$ as:

$$\mathbf{z}_i = \mathbf{W}^T\mathbf{x}_i, \tag{2.13}$$

where $\mathbf{W} \in \mathbb{R}^{D\times d}$ is the projection matrix to be learned by optimizing Fisher's discrimination criterion, and $\mathbf{z}_i \in \mathbb{R}^d$ is the projected sample.

Within-class scatter matrix $\mathbf{S}_W$ and between-class scatter matrix $\mathbf{S}_B$ in multi-class problems can be generalized from their corresponding definitions in two-class problems. Hence, within-class scatter matrix $\mathbf{S}_W$ is defined as follows:

$$\mathbf{S}_W = \sum_{c=1}^{C} \mathbf{S}_{W_c}, \tag{2.14}$$

$$\mathbf{S}_{W_c} = \sum_{\mathbf{x}_i, \alpha_i^c = 1} (\mathbf{x}_i - \boldsymbol{\mu}_c)(\mathbf{x}_i - \boldsymbol{\mu}_c)^T, \tag{2.15}$$

and between-class scatter matrix is defined as:

$$\mathbf{S}_B = \sum_{c=1}^{C} N_c(\boldsymbol{\mu}_c - \boldsymbol{\mu})(\boldsymbol{\mu}_c - \boldsymbol{\mu})^T, \tag{2.16}$$

where $\mathbf{S}_{W_c}$ is the covariance matrix of class c. α_i^c is an index denoting whether sample i belongs to class c, i.e. $\alpha_i^c = 1$ if $y_i = c$ and $\alpha_i^c = 0$ otherwise. N_c denotes the cardinality of class c, i.e. $N_c = \sum_{i=1}^N \alpha_i^c$ and $\boldsymbol{\mu}_c$ denotes the mean vector of class c, i.e. $\boldsymbol{\mu}_c = \frac{1}{N_c}\sum_{\mathbf{x}_i, \alpha_i^c = 1} \mathbf{x}_i$. $\boldsymbol{\mu}$ is the total mean vector $\boldsymbol{\mu} = \frac{1}{N}\sum_{i=1}^N \mathbf{x}_i$.

Total scatter matrix $\mathbf{S}_T$ can also be used, exploiting the between-class scatter matrix [8]. It is defined as follows:

$$\mathbf{S}_T = \sum_{i=1}^{N} (\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^T. \tag{2.17}$$

According to the definitions of $\mathbf{S}_W$, $\mathbf{S}_B$ and $\mathbf{S}_T$, $\mathbf{S}_T$ is the summation of $\mathbf{S}_W$ and $\mathbf{S}_B$.

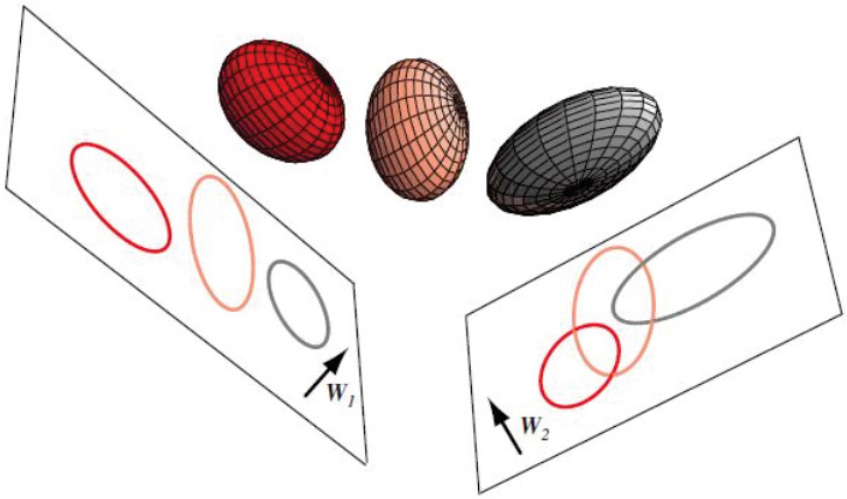


Figure 2.4: An example of Fisher's discriminant analysis with 3 classes dataset [4]

Fisher's discriminant criterion can be determined either by

$$J(W) = \max_{\mathbf{W}} \frac{tr(\mathbf{W}^T \mathbf{S}_B \mathbf{W})}{tr(\mathbf{W}^T \mathbf{S}_W \mathbf{W})}, \tag{2.18}$$

or

$$J(W) = \max_{\mathbf{W}} \frac{tr(\mathbf{W}^T \mathbf{S}_B \mathbf{W})}{tr(\mathbf{W}^T \mathbf{S}_T \mathbf{W})}, \tag{2.19}$$

where tr(.) denotes the trace of matrix. The optimal projection matrix $\mathbf{W}$ can be determined by applying eigenvalue decomposition of the matrix $\mathbf{S} = \mathbf{S}_W^{-1}\mathbf{S}_B$ (or $\mathbf{S} = \mathbf{S}_T^{-1}\mathbf{S}_B$) and keeping the eigenvectors corresponding to the largest (or smallest) eigenvalues. Since the rank of $\mathbf{S}$ is equal to $C-1$, the maximal dimensionality of the obtained subspace is equal to $C-1$. Thus, original data is mapped from a D-dimensional space into a d-dimensional subspace using the optimal $\mathbf{W}$. Fig. (2.4) demonstrates a multi-class LDA example, in which a three-class dataset is mapped from a three-dimensional input space into a two-dimensional subspace. As illustrated in this figure, $\mathbf{W}_1$ is the optimal projection matrix comparing to $\mathbf{W}_2$, due to its greatest ability of separation on the three-class dataset [4].

2.4.2 Shortcomings of Linear Discriminant Analysis

LDA has been effectively applied to solve various classification tasks. For instance, Huang et al. apply LDA to classify gene expression data to different cancer types [26], Jin et al. employ LDA approach to extract discriminant features for face recognition [27]. Although this approach gained much popularity, one can not always achieve an optimal solution in real problems. The first reason is that the optimal solution is achieved based on the assumption of a homoscedastic Gaussian model [28].

In this model, samples of each class should be sampled from a Gaussian distribution of the same characteristics, which means that the covariance matrices of all classes should be identical [17]. However, it is difficult to satisfy this requirement in most real-world problems; since e.g. imbalanced classes, as Fig. (2.5), are quite common.

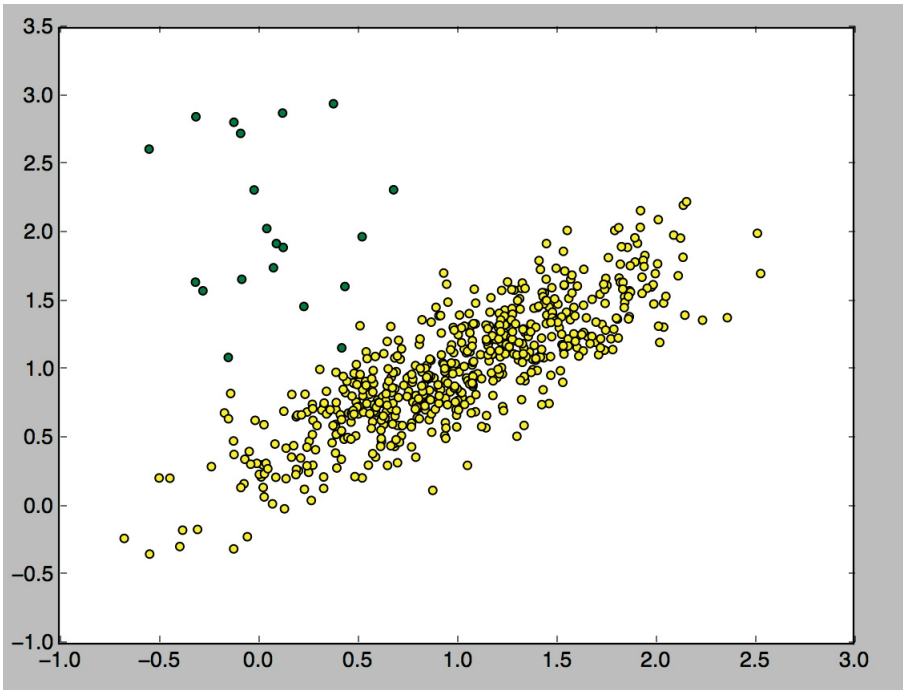


Figure 2.5: Imbalanced dataset

Moreover, traditional LDA merely takes into account the larger variation between each class mean vector and overall mean vector, according to the definition of the between-class scatter. This definition may cause a sub-optimal result due to the dominant role of outlier classes for scatter matrices.

Fig. (2.6) shows how the existence of an outlier class can influence the projection by two LDA variants. There are four classes in this example. Class 4 is the outlier class located far from the other three classes. In order to maximize variation between each class mean vector and overall mean vector using traditional LDA, we get a projection vector $\mathbf{w}_1$, due to the dominant role of class 4. When we use $\mathbf{w}_1$ for projection, mapped class 4 is located far away from other mapped datasets. Meanwhile, class 1, class 2 and class 3 overlap in the projection subspace. $\mathbf{w}_2$ is determined by an optimized LDA variant, which can alleviate the influence of class 4 by using appropriate weight functions in discriminant subspace. In this discriminant subspace, class 1, class 2 and class 3 are separated well from each other.

Last but not least, when the number of features is larger than that of samples [13], within-class scatter matrix is singular. The singularity problem will make the

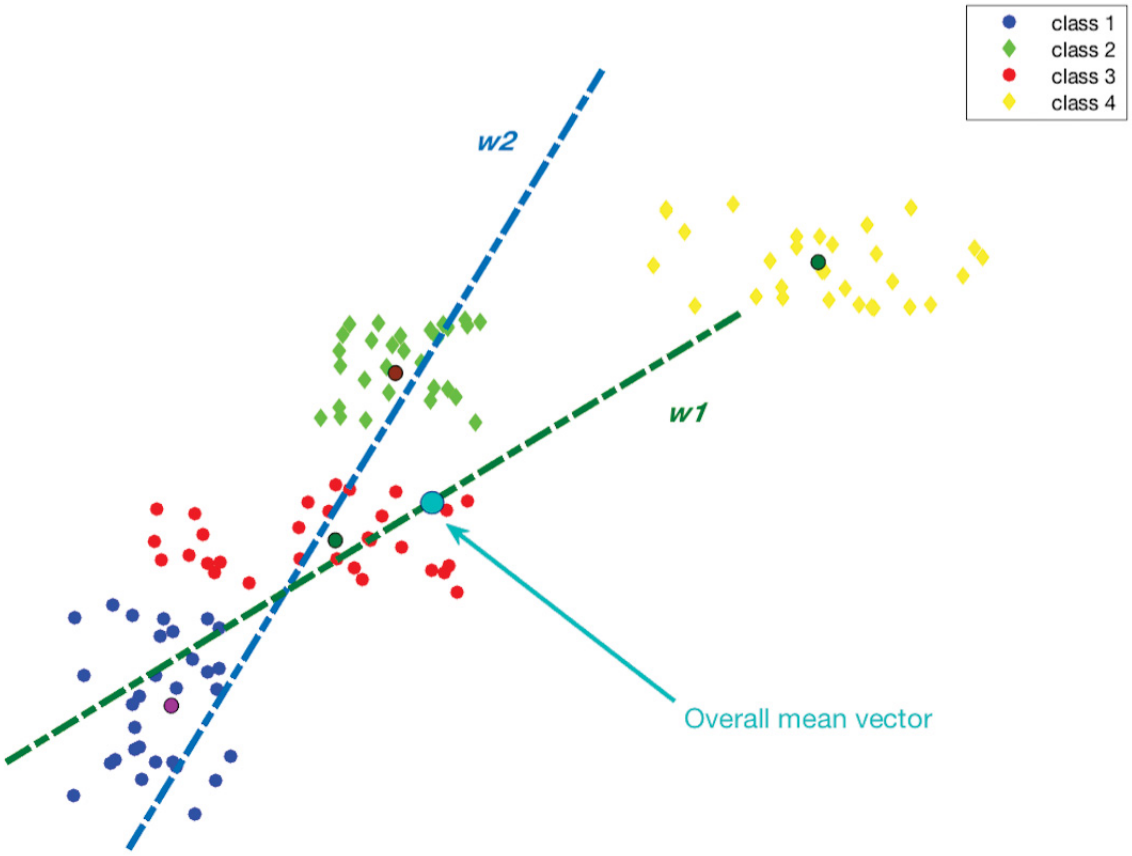


Figure 2.6: An example demonstrates how an outlier class influences the projection in Linear Discriminant Analysis

calculation of the inverse in the eigen-decomposition impossible.

2.5 Weighted Linear Discriminant Analysis

In order to resolve the outlier class and imbalanced class problems existing in traditional LDA, weighted LDA variants have been proposed. Weighted versions of LDA aim at scaling the contribution of each class based on their influences in the discriminant subspace through defining appropriate weights in between-class or/and within-class scatter matrices. Here, we demonstrate two types of weighted LDA, i.e. the relevant weighted LDA [17] and the new weighted LDA [18], which served as inspirations for the proposed weighted LDA variants.

2.5.1 Relevant Weighted LDA

The traditional definition of between-class scatter overemphasizes on outlier classes in multi-class classification tasks and leads to sub-optimal results. Hence, weighting functions investigating the role of each outlier class can be embedded into the

between-class scatter to alleviate the influence of outlier classes. Based on such consideration, Loog et al. redefine between-class scatter matrix as shown in Eq. (2.20) to conquer outlier problems [29], as follows:

$$\mathbf{S}_b = \sum_{i=1}^{C-1} \sum_{j=i+1}^{C} p_i p_j (\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)^T \tag{2.20}$$

where C is the number of classes, p_i, p_j denote the prior probabilities of class i and class j, respectively. μ_i and μ_j are the mean vectors of class i and class j. As described in [29], the new definition describes between-class scatter matrix based on the prior probability of each class and the differences between pairwise classes.

Nevertheless, as Loog et al. point out in [19], Eq. (2.20) can not describe the relationships precisely between different class pairs in multi-class problems, since the projection of one class could be interfered by the projections of other classes in its vicinity. Therefore, they extend Eq. (2.20) to Eq. (2.21) in [19] to redefine the between-class scatter matrix for enhancing the robustness of LDA variants in multi-class problems based on the Bayes error rate [19], as follows:

$$\mathbf{S}_b = \sum_{i=1}^{C-1} \sum_{j=i+1}^{C} p_i p_j \omega(\Delta_{ij}) \mathbf{S}_{ij}, \tag{2.21}$$

$$\mathbf{S}_{ij} = (\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)^T, \tag{2.22}$$

where p_i, p_j denote the priori probabilities of class i and j, respectively. Δ_{ij} is a weighting function, which expresses the similarity/dissimilarity between classes i and j.

Similarity/Dissimilarity measure is a common approach to express similarity or dissimilarity between different samples by using a distance function, as mentioned in section 2.2. In the above, Loog et al. use the Mahanalobis distance to measure the similarity of each class pair, as follows:

$$\Delta_{ij} = \sqrt{(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)^T \mathbf{S}_W^{-1} (\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)}. \tag{2.23}$$

In addition, other distance metrics can be employed to measure similarity, such as in [13], where the Euclidean distance is used for the calculation of Δ_{ij}.

In Loog's work [19], a new definition of between-class scatter matrix is used to determine Fisher's ration, as follows:

$$J(W) = \max_{\mathbf{W}} \frac{tr(\mathbf{W}^T \mathbf{S}_b \mathbf{W})}{tr(\mathbf{W}^T \mathbf{S}_W \mathbf{W})}, \tag{2.24}$$

where $\mathbf{S}_W$ is the traditional within-class scatter matrix and $\mathbf{S}_b$ is given by Eq. (2.21). The optimal projection matrix $\mathbf{W}$ is determined by applying eigenvalue decomposition to the following matrix:

$$\mathbf{S}_W^{-1} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} p_i p_j \omega(\Delta_{ij}) \mathbf{S}_{ij}. \tag{2.25}$$

Loog's work provides a great motivation to other researchers, who besides the definition of between-class scatter matrix, employ weighting functions derived from similarity measures in the definition of within-class scatter, or solve for heterogeneous Gaussian distributions. As shown in Eq.(2.14), $\mathbf{S}_W$ is the mean matrix of each $\mathbf{S}_{W_i}$ under the homoscedastic Gaussian distribution assumption. Therefore, the covariance matrix of an outlier class will dominate the calculation of $\mathbf{S}_W$, when it is larger than others. In this case, the optimal projection matrix $\mathbf{W}$ of traditional LDA pays more attention to separate outlier classes from others in the discriminant subspace, as shown in Fig. 2.6.

Tang et al. [17] propose an outlier-class-resistant weighted LDA method based on Loog's work, in order to reduce the influence of outlier classes. They express the between-class scatter using Eq. (2.21) and a new within-class scatter definition is proposed as follows:

$$\mathbf{S}_w = \sum_{c=1}^{C} p_c r_c \mathbf{S}_{w_c} \tag{2.26}$$

$$\mathbf{S}_{w_c} = \sum_{i=1}^{N_c} (\mathbf{x}_i - \boldsymbol{\mu}_c)(\mathbf{x}_i - \boldsymbol{\mu}_c)^T \tag{2.27}$$

where $r_c = \sum_{i \neq c} \frac{1}{L_{ic}}$ is a relevance-weight, reducing attention to outlier classes. N_c is the cardinality of class c, $\boldsymbol{\mu}_c$ is the mean vector of class c.

Relevance-weight r_c is calculated based on various dissimilarity/similarity measures in Tang's work, e.g. Euclidean distance, Mahalanobis distance and Bayesian classification accuracy. r_c can guarantee lowering the influence of outlier classes in $\mathbf{S}_w$. The following example demonstrates how weight r_c works.

Let us assume that there are four classes C_1, C_2, C_3 and C_4 in a dataset, where C_3 is the outlier class with relatively higher $\mathbf{S}_{w_3}$. Euclidean distance is used to evaluate the similarity between class pairs, as follows:

$$\mathbf{L}_{ic} = \sqrt{(\boldsymbol{\mu}_i - \boldsymbol{\mu}_c)^T (\boldsymbol{\mu}_i - \boldsymbol{\mu}_c)}, \tag{2.28}$$

where $\boldsymbol{\mu}_i$ and $\boldsymbol{\mu}_c$ are the mean vectors of classes i and c. Then, we can obtain the similarity measures for each pair of classes, as shown in Table (2.1), where $\mathbf{L}_{ic}$ equals to $\mathbf{L}_{ci}$.

Table 2.1: Example demonstrates how r_c works for a dataset with outlier class

C_1	C_2	C_3	C_4	r_c
0	L_{12}	L_{13}	L_{14}	$r_1 = \frac{1}{L_{12}+L_{13}+L_{14}}$
L_{12}	0	L_{23}	L_{24}	$r_2 = \frac{1}{L_{12}+L_{23}+L_{24}}$
L_{13}	L_{23}	0	L_{34}	$r_3 = \frac{1}{L_{31}+L_{32}+L_{34}}$
L_{14}	L_{24}	L_{34}	0	$r_4 = \frac{1}{L_{41}+L_{42}+L_{43}}$

When we calculate within-class scatter matrix $\mathbf{S}_w$ of this dataset, the original higher $\mathbf{S}_{w_3}$ of outlier class C_3 is replaced by a new value $r_3\mathbf{S}_{w_3}$. According to the definition of L_{ic} in Eq. (2.28), $L_{i3} \gg L_{i1}, L_{i2}, L_{i4}$, hence $r_3 \ll r_1, r_2, r_4$. Thus, a smaller r_3 will alleviate the influence of outlier class C_3 to $\mathbf{S}_w$. Therefore, an optimal $\mathbf{W}$ obtained by the weighted LDA method can minimize the influence of class 3 in the projection subspace. As concluded in Loog's work, relevant weighted LDA has been proved that it can preserve more discriminative information in the obtained subspace [19].

2.5.2 New Weighted LDA

Jarchi et al. [18] propose another version of weighted LDA to alleviate the influence of outlier classes. In Jarchi's work, weight function describes the precise distance information of every two classes. Because the optimal linear projection $\mathbf{W}$ obtained by Fisher's discriminant criterion optimization contains the separable information of pairwise classes, the determined $\mathbf{W}$ calculated from the original LDA is used as a weight function.

They define the between-class and within-class scatter matrices as follows:

$$\mathbf{S}_b = \sum_{i=1}^{C-1} \sum_{j=i+1}^{C} n_i n_j w_1(\Delta_{ij})(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)^T, \tag{2.29}$$

$$\mathbf{S}_w = \sum_{c=1}^{C} p_c w_2(\Delta_{c:})\mathbf{S}_{w_c}, \tag{2.30}$$

where n_i, n_j are the number of samples for class i and class j, respectively. $w_1(\Delta_{ij})$ and $w_2(\Delta_{c:})$ are defined as follows:

$$w_1(\Delta_{ij}) = \frac{1}{\Delta_{ij}}, \tag{2.31}$$

$$w_2(\Delta_{c:}) = \frac{1}{\sum_{j\neq c} \Delta_{cj}}, \tag{2.32}$$

and Δ_{ij} is the Fisher's discriminant criterion in the discriminant subspace de-

termined by applying traditional LDA using the between-class scatter matrix in Eq. (2.16) and the total scatter matrix $\mathbf{S}_T$ in Eq. (2.17), i.e.:

$$\mathbf{w}^* = \underset{\mathbf{w}}{argmax}\{\frac{\mathbf{w}^T\mathbf{S}_B\mathbf{w}}{\mathbf{w}^T\mathbf{S}_T\mathbf{w}}\} = \mathbf{S}_T^{-1}(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j) \tag{2.33}$$

$$\Delta_{ij} = \frac{\mathbf{w}^{*T}\mathbf{S}_B\mathbf{w}^*}{\mathbf{w}^{*T}\mathbf{S}_T\mathbf{w}*} \tag{2.34}$$

Using the above definition of Δ_{ij}, in the case where a class is well separated from all others, a smaller value of $w(\Delta_{ij})$ will be used, reducing the influence of that class on the result. Once the new $\mathbf{S}_w$ and $\mathbf{S}_t$ ($\mathbf{S}_t = \mathbf{S}_w + \mathbf{S}_b$) are obtained, the final projection matrix $\mathbf{W}$ can be determined by optimizing the following Fisher's discriminant criterion:

$$J(\mathbf{W}) = \underset{\mathbf{W}}{argmax}\frac{tr(\mathbf{W}^T\mathbf{S}_b\mathbf{W})}{tr(\mathbf{W}^T\mathbf{S}_t\mathbf{W})} \tag{2.35}$$

2.6 One-class Classification

Usually, a classification problem is formed by two classes or multiple classes. For example, in a message identification problem, training data is labeled as "yes" or "no" spam, or in the handwritten digits classification problem, the label set is $\mathcal{Y} = \{y_i\}_{i=0}^{9}$. One-class classification is different from two-class/multi-class classification tasks, because training dataset has only one target class in one-class classification. Hence, the objective of one-class classification is to determine whether a new sample belongs to the target class or not. Samples belonging to the target class are considered as target objects, whereas samples not belonging to this class are named as outlier objects [30].

Generally, one-class classification is used in cases, where training samples belong only to one class, or except one class, the other classes can not be clearly defined. For example, for a problem related to the sales history of a product, it is easier to obtain samples corresponding to actual sales of the product to a specific person, rather than to define samples of "not buy" class. This is because we do not know whether non-consumers for this product are not interested in the product or just have not purchased yet. Moreover, it is common the number of "not buy" data to be far larger than that of "buy" data, which will lead to a highly imbalanced problem. Under this situation, one-class classification is an appropriate solution, as training set only contains information of its consumers, we just need to identify whether a new sample will correspond to a potential consumer or not.

Fig. (2.7) demonstrates an example of one-class classification task, in which training dataset only contains the consumer information of infant milk powder. We can

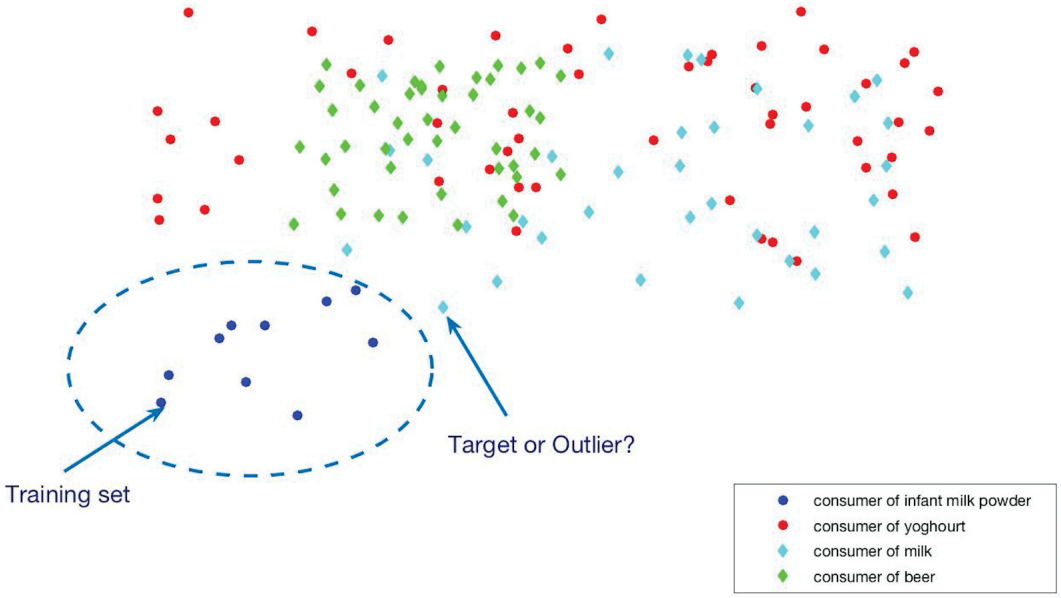


Figure 2.7: One-class classifier

classify an unknown sample as a target object or an outlier based on whether it is placed in the feature space area of the training set. One-class classification is widely used in various applications, such as concept learning [31], outlier detection [32] and novelty detection [33].

2.7 Saliency Estimation

Saliency estimation is derived from psychological science and is considered as a selective process in human visual system [34]. Saliency phenomenon arose by human visual system is usually defined as a kind visual perception, by which human visual system could process special parts in a scene in advance and distinct them (as foreground) from other parts. As shown in Fig. (2.8), the green square, the hollow circle, the irregular slash, and the solid round are more salient than other parts in each of the corresponding sub-images.

Saliency estimation is a problem which has gained attention during the last decade, since it can be applied as a pre-processing step for higher level Computer Vision tasks, like object localization and recognition [35], high-resolution [36] and compression [37], [38].

It is common knowledge that saliency estimation methods can be categorized broadly as: local methods and global methods based on salient cues [40]. Local methods inspect the discriminative information around the neighborhood of certain pixels/regions. For example, Itti et al. propose a local saliency model in their work

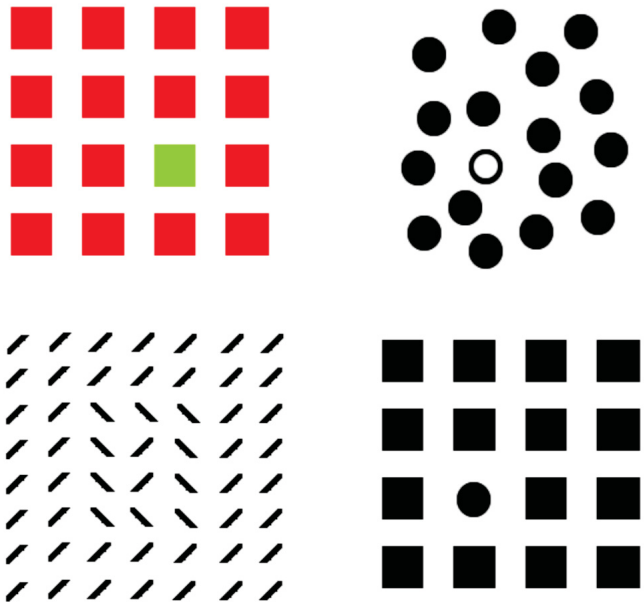


Figure 2.8: Saliency phenomena in human visual system [39]

[41] for scene analysis; Harel et al. exploit Markov chain over a local saliency model in [42]. While global methods exploit the rarity of a pixel/patch/region in whole scene [43]; as in [44] Achanta et al. investigate brightness and colour of pixels with respect to whole image. In order to take advantage of local and global methods together, some researchers utilize both of them, like in [45; 46].

Recently, Aytekin et al. formulated the salient object segmentation problem based on probabilistic interpretation. Specifically, they defined a probability mass function $\mathbf{P}(x)$ encoding the probability that an image region (in the sense of pixel, super-pixel or patch) to depict a salient region. Estimation of $P(x)$ is formulated as an optimization problem enforcing similar regions to have similar probabilities, while any prior information regarding saliency (defined based on the location of each region in the image lattice) can be exploited. This joint optimization is expressed as follows:

$$\underset{r(x)}{argmin}\left(\sum_i(\mathrm{p}(x=x_i))^2 v_i+\left(\sum_{i,j}\left(\left(\mathrm{p}(x=x_i)\right)^2-\mathrm{p}(x=x_i)\mathrm{p}(x=x_j)\right)w_{i,j}\right)\right) \qquad (2.36)$$

$$s.t. \quad \mathbf{p^T 1}=1,$$

where $v_i \geq 0$ denotes prior information for region i and w_{ij} expresses the similarity of regions i and j. The optimization problem in Eq. (2.36) can be expressed using a

matrix notation as follows:

$$\mathbf{p}^* = \underset{p}{argmin}(\mathbf{p^T H p}) \tag{2.37}$$

$$\mathbf{H} = \mathbf{D} - \mathbf{W} + \mathbf{V}, \tag{2.38}$$

$$s.t. \quad \mathbf{p^T 1} = 1,$$

where $\mathbf{p}$ is a vector having elements $p_i = P(x = x_i)$ corresponding to the probability of each region to be salient. $\mathbf{W}$ is the affinity matrix of a graph having as vertices the region representations and $\mathbf{D}$ is the corresponding diagonal matrix having elements equal to $D_{ii} = \sum_j W_{ij}$. $\mathbf{V}$ is a diagonal matrix having elements $[\mathbf{V}]_{ii} = v_i$. The optimized solution for Eq. (2.37) can be obtained by using the Lagrangian multiplier method as following:

$$\mathcal{L}(\mathbf{p}, \gamma) = (\mathbf{p^T H p}) - \gamma(\mathbf{p^T 1} - 1). \tag{2.39}$$

Then, we can calculate the partial derivative of the above equation with the respect to $\mathbf{p}$ and set it to zero. The optimization problem in Eq. (2.37) has a global optimum given by:

$$\mathbf{p}^*_{pse} = \mathbf{H}^{-1}\mathbf{1}. \tag{2.40}$$

As has been shown in [22], the above solution has close connections with one-class classification problems.

3. METHODS

In this chapter, several novel $SwLDA$ methods are proposed, combining intuitions from both weighted LDA variants and saliency estimation. Then, these methods are implemented on various datasets to exploit their saliency information in a discriminant subspace learning task. The result of this task is evaluated based on two classification algorithms, i.e. the Nearest Centroid and k-Nearest Neighbor classifiers. This chapter describes all methods or techniques involved in this task in detail. An overview of the entire process is described in section 3.1. Data pre-processing techniques are described in section 3.2. The new definitions for saliency weights, class representation, scatter matrices and Fisher's discriminant criterion are given in section 3.3. Finally, the two classifiers used in the experiments are described in section 3.4.

3.1 Overview

According to the classification algorithms used, we employ two work-flows to complete an experiment. The first one based on nearest centroid classifier is shown in Fig. (3.1). The data is normalized at the beginning, and then split into training and test data. The next step is to determine the optimal $\mathbf{W}$ by optimizing Fisher's discriminant criterion and the new class representations by proposed $SwLDA$ methods using the training data. At last, we project test data and the new class representations into a discriminant subspace, where nearest centroid classifier is used to predict labels for the test data.

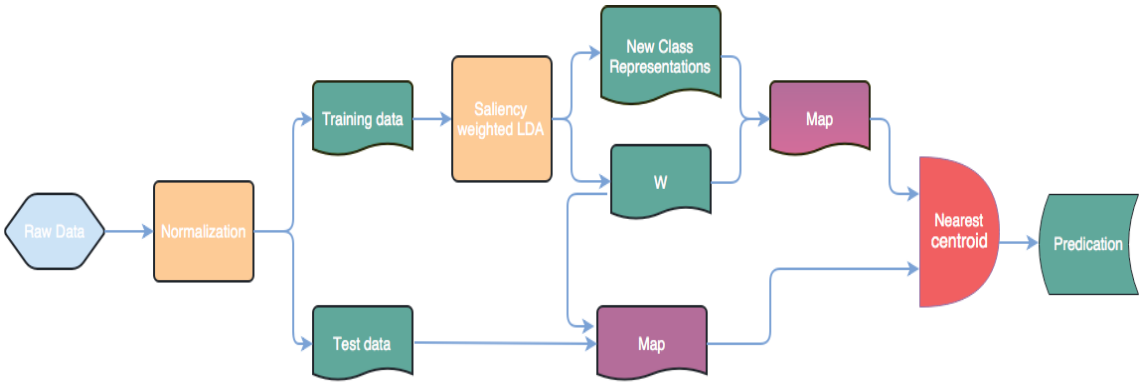


Figure 3.1: Work-flow using nearest centroid classifier

The second work-flow using the k-nearest neighbor classifier on the identical data pre-processing step as previous work-flow. Then we apply $SwLDA$ methods to obtain the optimal projection $\mathbf{W}$. The next step is to project both training and test data into the discriminant subspace. At last, k-nearest neighbor classifier is used in this subspace to predict the labels of the test data.

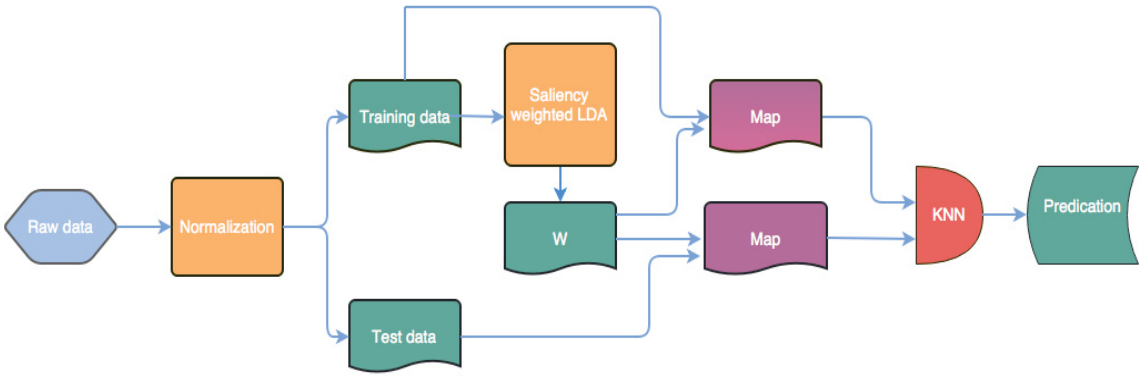


Figure 3.2: Work-flow using k nearest neighbor classifier

3.2 Pre-processing

We use the zero-mean normalization method described in section 2.3 to pre-process the data. For a training dataset $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$, $\mathbf{x}_i \in \mathbb{R}^D$, the zero-mean normalization is implemented to the i-th sample as:

$$\hat{\mathbf{x}}_i = \frac{\mathbf{x}_i - \boldsymbol{\mu}}{\boldsymbol{\sigma}}, \tag{3.1}$$

where $\boldsymbol{\mu} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i$, $\boldsymbol{\mu} \in \mathbb{R}^D$ is the mean vector of the whole dataset, $\sigma_j = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_{ij} - \mu_j)^2}$ is the standard derivation of dimension j, $j \in \{1, 2, \cdots, D\}$, $\hat{\boldsymbol{x}}_i$ is the normalized $\mathbf{x}_i$. After this step, the normalized dataset will have zero mean $\boldsymbol{\mu} = \mathbf{0}$ and standard derivation $\sigma_j = 1$ along each dimension. Test data is normalized accordingly.

In the experiments, the five-fold cross-validation is used. We implement five iterations, as shown in Fig. (3.3), for each iteration one fold is used for testing. The final result is the average prediction accuracy of these five iterations.

All training data is labeled according to their classes. The label vector is denoted as $\mathbf{y} = \{y_1, ..., y_i, ...y_N\}, y_i \in \{1, 2, ..., C\}$, where C is the number of classes in each dataset.

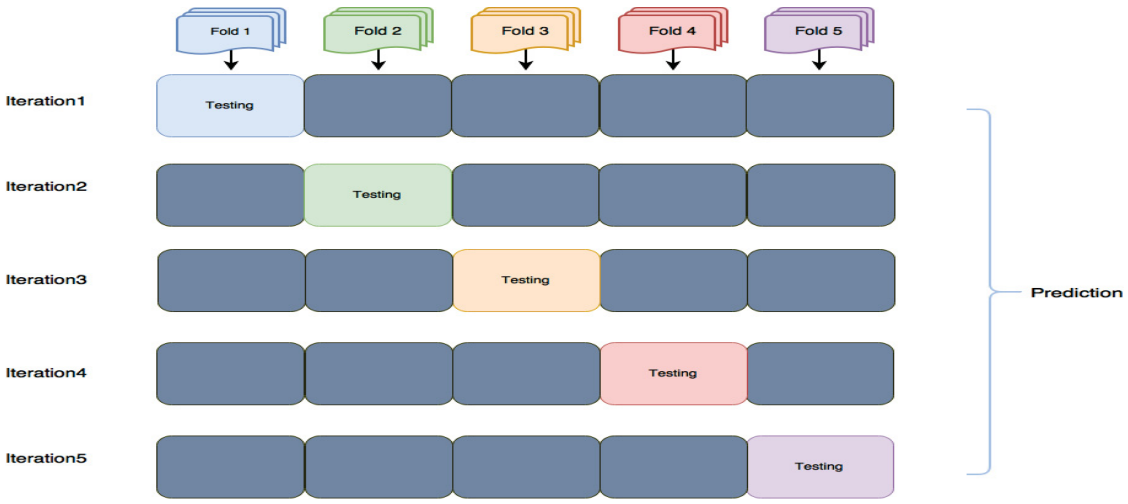


Figure 3.3: Work-flow of 5-fold cross validation

3.3 Saliency-based weighted Linear Discriminant Analysis

This section demonstrates the principle of the proposed *SwLDA* methods on the basis of a probabilistic definition of saliency. As mentioned in section 2.8, saliency estimation is usually considered as a pre-processing step to detect distinct parts (pixels/patches) in a scene. The motivation for combining weighted LDA and saliency estimation arises from the following considerations.

First of all, class Gaussian distribution dataset assumption is difficult to be satisfied. Secondly, although LDA variants inspect the contribution of each class, the importance of each sample within its class is still neglected. So we need to investigate the internal structure of each class, which can be considered as a saliency property estimation of each sample.

According to the description of saliency in section 2.8, saliency information of one sample can be used to express its importance for its corresponding class. Thus we incorporate the saliency of each sample into the calculation of scatter matrices as weights. As a result, both outlier classes and outlier samples inside a class will contribute less to the calculation of scatter matrices, which is imperative for a good result. Motivated by the connection of saliency estimation with one-class classification, we calculate the saliency of each sample within its class only.

In the following, we demonstrate the derivation process of sample weights for each class' samples in section 3.3.1, and the new representation of each class in section 3.3.2, in detail. After that, we define scatter matrices in various ways in section 3.3.3. At last, we incorporate the obtained sample weights and new class representation into scatter matrices to generate the new Fisher's discriminant criterion variant in section 3.3.4.

3.3.1 Sample Weights

Weighted LDA variants represent each class with the corresponding mean vector and define the weights based on distance measurements of pairwise classes to address the outlier class problem. Such mutation yields a certain improvement over traditional LDA. Nevertheless, it can not reveal the structure of boundary inside each class, and then neglect the influence of samples in the vicinity of boundary, which may affect the classification result greatly. This is due to the fact that all class samples equally contribute to the definition of the class representation and scatter matrix calculation.

In this thesis, we make a connection between weighted LDA and saliency estimation, so as to evaluate the contribution of each sample inside its corresponding class by exploiting *class saliency information* based on the method mentioned in section 2.8. Such class saliency information can reveal the importance of each sample in its class and describe it using a saliency score. The higher the saliency score of one sample is, the more information it contains for its class.

We use the probabilistic saliency estimation (PSE) method [22] with one-class classification model to calculate the saliency score of each sample in its class. In such an one-class classification model, the target objects are samples in one class. According to the the principle of PSE, we need to obtain affinity matrix $\mathbf{W}_c$ (as region representations), diagonal matrix $\mathbf{V}_c$ (as priori information description) and diagonal matrix $\mathbf{D}_c$ from its corresponding $\mathbf{W}_c$ for class c.

The saliency estimation process is based on an undirected graph. That is, for each class c, we form the corresponding graph $\mathcal{G}_C = \{\mathbf{X}_c, \mathbf{W}_c\}$, where $\mathbf{X}_c \in \mathbb{R}^{D\times N_c}$ is a matrix formed by the samples belonging to class c and $\mathbf{W}_c \in \mathbb{R}^{N_c\times N_c}$ is the graph weight matrix expressing the similarity between pairwise samples.

Table 3.1: Kernel matrix $\mathbf{K}_c$ for class c

Index	$i=1$	$i=2$	$\cdots$	$i=N_c$
$j=1$	$\kappa(\mathbf{x}_1,\mathbf{x}_1)=\exp\left(-\frac{\|\mathbf{x}_1-\mathbf{x}_1\|}{2\sigma^2}\right)$	$\kappa(\mathbf{x}_1,\mathbf{x}_2)=\exp\left(-\frac{\|\mathbf{x}_1-\mathbf{x}_2\|}{2\sigma^2}\right)$	$\cdots$	$\kappa(\mathbf{x}_1,\mathbf{x}_{N_c})=\exp\left(-\frac{\|\mathbf{x}_1-\mathbf{x}_{N_c}\|}{2\sigma^2}\right)$
$j=2$	$\kappa(\mathbf{x}_2,\mathbf{x}_1)=\exp\left(-\frac{\|\mathbf{x}_2-\mathbf{x}_1\|}{2\sigma^2}\right)$	$\kappa(\mathbf{x}_2,\mathbf{x}_2)=\exp\left(-\frac{\|\mathbf{x}_2-\mathbf{x}_2\|}{2\sigma^2}\right)$	$\cdots$	$\kappa(\mathbf{x}_2,\mathbf{x}_{N_c})=\exp\left(-\frac{\|\mathbf{x}_2-\mathbf{x}_{N_c}\|}{2\sigma^2}\right)$
$j=3$	$\kappa(\mathbf{x}_3,\mathbf{x}_1)=\exp\left(-\frac{\|\mathbf{x}_3-\mathbf{x}_1\|}{2\sigma^2}\right)$	$\kappa(\mathbf{x}_3,\mathbf{x}_2)=\exp\left(-\frac{\|\mathbf{x}_3-\mathbf{x}_2\|}{2\sigma^2}\right)$	$\cdots$	$\kappa(\mathbf{x}_3,\mathbf{x}_{N_c})=\exp\left(-\frac{\|\mathbf{x}_3-\mathbf{x}_{N_c}\|}{2\sigma^2}\right)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$j=N_c$	$\kappa(\mathbf{x}_{N_c},\mathbf{x}_1)=\exp\left(-\frac{\|\mathbf{x}_{N_c}-\mathbf{x}_1\|}{2\sigma^2}\right)$	$\kappa(\mathbf{x}_{N_c},\mathbf{x}_2)=\exp\left(-\frac{\|\mathbf{x}_{N_c}-\mathbf{x}_2\|}{2\sigma^2}\right)$	$\cdots$	$\kappa(\mathbf{x}_{N_c},\mathbf{x}_{N_c})=\exp\left(-\frac{\|\mathbf{x}_{N_c}-\mathbf{x}_{N_c}\|}{2\sigma^2}\right)$

According to the principle of PSE mentioned in section 2.8, saliency score is determined by Eq. (2.38) and Eq. (2.40). Firstly, we use the RBF kernel function $[\mathbf{W}_c]_{ij}=\exp\left(-\frac{\|\mathbf{x}_{ci}-\mathbf{x}_{cj}\|}{2\sigma^2}\right)$, $\sigma>0$ to calculate the values of the weight matrix $\mathbf{W}_c$, and then the diagonal matrix $\mathbf{D}_c$ is determined based on $\mathbf{W}_c$. When calculating the $\mathbf{W}_c$, we sort the similarity information $\kappa(\mathbf{x}_i,\mathbf{x}_j)$, i, $j\in(1,2,...,N_c)$ with the respect

to each sample $\mathbf{x}_i$ in a descending order. Then we employ k-NN algorithm to select the first k samples' similarity information and set others as zero, since the first k samples are the most similar to samples $\mathbf{x}_i$. By doing so, we employ the k-NN graph structure. When k is set to N_c, the similarity information of all pairwise samples are retained and $\mathcal{G}_C$ is a fully connected graph. Each element in the diagonal matrix $\mathbf{D}_c$ is given by: $[\mathbf{D}]_{ii} = \sum_j \mathbf{W}_{ij}$.

Secondly, we calculate the the matrix $\mathbf{V}_c$ expressing our a priori saliency information. For multi-class classification, we exploit three types of priori information to calculate $\mathbf{V}_c$, as follows:

1. **Equal probability**: it assumes that there is no priori information related to the saliency information of each sample. In this case, the elements of $\mathbf{V}_c$ are set equal to:
$$V_{c,ii} = \frac{1}{N_c} \tag{3.2}$$

2. **Distance-based probability**: this approach assumes that a sample is less probable to have high saliency information if it is located far from its corresponding class center. In this case, the elements of $\mathbf{V}_c$ are set equal to:
$$V_{c,ii} = \|\mathbf{x}_{c,i} - \boldsymbol{\mu}_c\|_2^2, \tag{3.3}$$
where $\mathbf{x}_{c,i}$ denotes the i-th sample of class c. Here we should note that this type of a priori information has been used in salient object segmentation methods, where it is referred to as boundary connectivity.

3. **Misclassification-based probability**: it assumes that a sample is less probable to have high saliency information, if it is closer to another class, when compared to its true class. In this case, the elements of $\mathbf{V}_c$ are set equal to:
$$V_{c,ii} = \begin{cases} 0, & if \;\; d_{c,i}^c < \underset{k \neq c}{min} \;\; d_{c,i}^k, \\ \frac{d_{c,i}^c}{\underset{k \neq c}{min} \;\; d_{c,i}^k}, & otherwise, \end{cases} \tag{3.4}$$
where $d_{c,i}^k = \|\mathbf{x}_{c,i} - \boldsymbol{\mu}_k\|_2^2$ denotes the distance information between the i-th sample in class c and the center of another class k, $k \neq c$. In this case, a sample which is close to another class is expected to have low saliency information, even if it may be close to the center of its class.

After having defined the matrices $\mathbf{W}_c$, $\mathbf{D}_c$ and $\mathbf{V}_c$, we determine $\mathbf{H}_c = \mathbf{D}_c - \mathbf{W}_c + \mathbf{V}_c$. It should be noted that both $\mathbf{W}_c$ and $\mathbf{H}_c$ are rescaled to the interval $[0, 1]$. It should be noted that when $\mathbf{H}$ is singular, a regularized version is used.

Once having $\mathbf{H}_c$, the probability of each sample $\mathbf{x}_{c,i}$ to belong to the class c is given as Eq. (2.40):

$$\mathbf{p}_c = \mathbf{H}_c^{-1}\mathbf{1}. \tag{3.5}$$

Here $\mathbf{p}_c$ is named as saliency score vector of class c, which encodes saliency information of each sample in class c.

3.3.2 Class Representation

Having obtained $\mathbf{p}_c \in \mathbb{R}^{N_c}$, $c = 1, \ldots, C$, we define a new class representation as follows:

$$\mathbf{m}_c = \mathbf{X}_c\mathbf{p}_c, \tag{3.6}$$

where $\mathbf{m}_c \in \mathbb{R}^{D\times 1}$ can also be considered as the new center of class c. Fig. (3.4) demonstrates that the saliency score vector $\mathbf{p}_c$ reallocates the contribution of each sample in class c, so as to alleviate the influences of outlier samples and generate a new class representation.

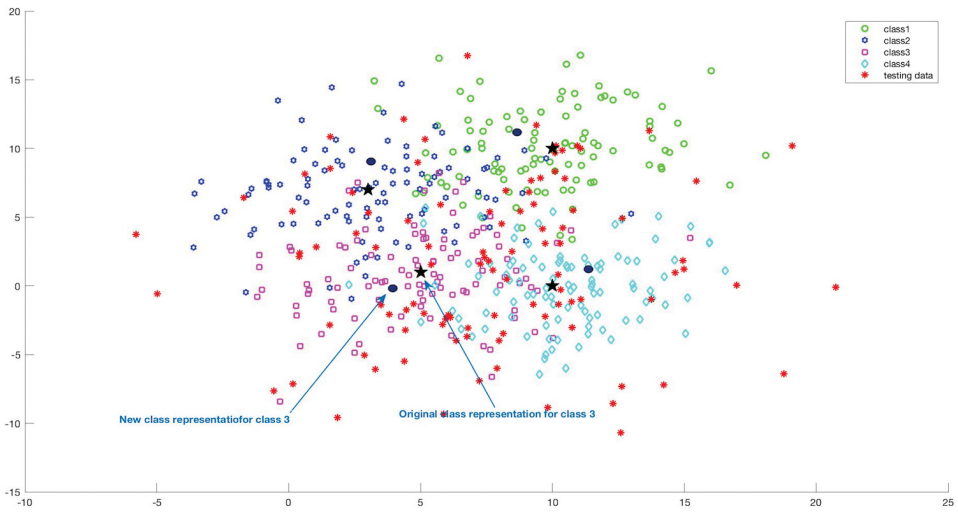


Figure 3.4: Example of new class representations with four datasets

3.3.3 Scatter Matrices Definition

By exploiting class-specific saliency scores described in the previous section, we can redefine the within-class scatter and between-class scatter matrices based on weighted LDA approaches, as mentioned in section 2.5.

Within-class scatter matrix is redefined in two different ways using the saliency

score vector $\mathbf{p}_c$. The first definition is to incorporate $\mathbf{p}_c$ in $\mathbf{S}_w$ as follows:

$$\mathbf{S}_w^{(1)} = \sum_{c=1}^{C} \mathbf{S}_{w_c}, \tag{3.7}$$

$$\mathbf{S}_{w_c} = \sum_{j=1}^{N_c} p_{c,j}(\mathbf{x}_{c,j} - \boldsymbol{\mu}_c)(\mathbf{x}_{c,j} - \boldsymbol{\mu}_c)^T, \tag{3.8}$$

where $\mathbf{x}_{c,j}$ denotes the j-th sample in class c, $p_{c,j}$ is saliency score for the j-th sample in class c, $\mathbf{S}_{w_c}$ is the covariance matrix of class c weighted by $\mathbf{p}_c$. The characteristics of this definition suggest that if a sample has more salient information, it will contribute more to its corresponding covariance matrix.

The second definition is inspired by relevance weighted LDA mentioned in section 2.5.1 and is given by:

$$\mathbf{S}_w^{(2)} = \sum_{c=1}^{C} r_c \mathbf{S}_{w_c}, \tag{3.9}$$

$$\mathbf{S}_{w_c} = \sum_{j=1}^{n_c} p_{c,j}(\mathbf{x}_j - \boldsymbol{\mu}_c)(\mathbf{x}_j - \boldsymbol{\mu}_c)^T. \tag{3.10}$$

Here $r_c = \sum_{i \neq c} \frac{1}{\mathbf{L}_{ic}}$ is the relevance-weight obtained by a similarity function $\mathbf{L}_{ic}$, where $\mathbf{L}_{ic}$ is defined based on the Euclidean distance between the mean vectors of class i and class c as:

$$\mathbf{L}_{ic} = \sqrt{(\boldsymbol{\mu}_i - \boldsymbol{\mu}_c)^T(\boldsymbol{\mu}_i - \boldsymbol{\mu}_c)}. \tag{3.11}$$

This definition can not only resolve the sub-optimal result caused by outlier classes. At the same time it incorporates the saliency information of samples into within-class scatter matrix to enhance the influence of salient samples. The principles of this approach for alleviating the influence of outlier classes are further described in section 2.5.1.

Between scatter matrix in the aforementioned LDA methods simply maximizes the variations between each class mean vector and the total mean vector or the variations between class pairs. Here, we propose four types of between-class scatter matrices, which are not only based on the aforementioned definitions of $\mathbf{S}_b$, but also capture the structure inside each class. The first definition is the same as Eq. (2.16).

$$\mathbf{S}_b^{(1)} = \sum_{c=1}^{C} N_c(\boldsymbol{\mu}_c - \boldsymbol{\mu})(\boldsymbol{\mu}_c - \boldsymbol{\mu})^T. \tag{3.12}$$

This is the standard definition of between-class scatter matrix, which only maximizes the variations between each class mean vector and the mean vector of all samples.

The second one uses saliency scores $\mathbf{p}_c$, when calculating each class' mean vector, as follows:

$$\hat{\boldsymbol{\mu}}_c = \mathbf{X}_c \mathbf{p}_c, \tag{3.13}$$

$$\mathbf{S}_b^{(2)} = \sum_{c=1}^{C} (\hat{\boldsymbol{\mu}}_c - \boldsymbol{\mu})(\hat{\boldsymbol{\mu}}_c - \boldsymbol{\mu})^T, \tag{3.14}$$

where $\mathbf{X}_c$ contains all samples in class c, $\hat{\boldsymbol{\mu}}_c$ is the new class representation or weighted center of class c, according to the saliency scores vector $\mathbf{p}_c$. This definition does not only consider the mean vector of each class, but rather exploits the contribution of each sample based on the saliency scores $\mathbf{p}_c$, and then reflects each sample's contribution in the new class representations $\hat{\boldsymbol{\mu}}_c$.

The third definition extends Eq. (3.14) to exploit the relationships between pairs of new class representation for each class, as follows:

$$\mathbf{S}_b^{(3)} = \sum_{c_1=1}^{C} \sum_{c_2=1}^{C} (\hat{\boldsymbol{\mu}}_{c_1} - \hat{\boldsymbol{\mu}}_{c_2})(\hat{\boldsymbol{\mu}}_{c_1} - \hat{\boldsymbol{\mu}}_{c_2})^T, \tag{3.15}$$

$$\hat{\boldsymbol{\mu}}_{c_1} = \mathbf{X}_{c_1} \mathbf{p}_{c_1}, \tag{3.16}$$

$$\hat{\boldsymbol{\mu}}_{c_2} = \mathbf{X}_{c_2} \mathbf{p}_{c_2}, \tag{3.17}$$

where $\mathbf{X}_{c_1}$ and $\mathbf{X}_{c_2}$ contain all samples of the corresponding classes c_1 and c_2. The new class representations or weighted centers $\hat{\boldsymbol{\mu}}_{c_1}$ and $\hat{\boldsymbol{\mu}}_{c_2}$ are calculated based on saliency scores vectors $\mathbf{p}_{c_1}$ and $\mathbf{p}_{c_2}$, respectively. In this case, original mean vector of each class is replaced by its corresponding weighted version, when calculating between-class scatter matrix. Hence this definition maximizes the variations of pairwise weighted mean vectors, so as to separate salient samples in class c_1 from salient samples in class c_2 as far as possible.

The last definition, $\mathbf{S}_b^{(4)}$, does not just encode information of each class' mean vector or weighted mean vector, but it intends to maximize discrimination between every sample in one class with other classes' weighted mean vectors, meanwhile takes into account of each sample's saliency scores, as follows:

$$\mathbf{S}_b^{(4)} = \sum_{c_1=1}^{C} \sum_{\substack{c_2=1, \\ c_2 \neq c_1}}^{C} \sum_{i=1}^{N_{c_1}} p_{c_1,i} (\mathbf{x}_{c_1,i} - \hat{\boldsymbol{\mu}}_{c_2})(\mathbf{x}_{c_1,i} - \hat{\boldsymbol{\mu}}_{c_2})^T, \tag{3.18}$$

where $\mathbf{x}_{c_1,i}$ is the i-th sample in class c_1, N_{c_1} is the cardinality of class c_1, $\hat{\boldsymbol{\mu}}_{c_2}$ is calculated by Eq. (3.17). The use of Eq. (3.18) makes variations between each sample in class c_1 and the new class representations of other classes ($c_2 \neq c_1$) as large as possible, so that samples of class c_1 with higher saliency scores concentrate around

new class representations $\hat{\boldsymbol{\mu}}_{c_1}$.

3.3.4 Saliency-based weighted Linear Discriminant Analysis

Using the above-described scatter matrices, several optimization criteria can be formed, which are listed in Table (3.2):

Table 3.2: Fisher's discrimination criteria based on different definitions

Method	Fisher's discrimination criterion $J(\mathbf{W})$
$SwLDA_{11}$	$\underset{\mathbf{W}}{argmax}\frac{tr(\mathbf{W}^T\mathbf{S}_b^{(1)}\mathbf{W})}{tr(\mathbf{W}^T\mathbf{S}_t^{(1)}\mathbf{W})}$
$SwLDA_{21}$	$\underset{\mathbf{W}}{argmax}\frac{tr(\mathbf{W}^T\mathbf{S}_b^{(2)}\mathbf{W})}{tr(\mathbf{W}^T\mathbf{S}_t^{(1)}\mathbf{W})}$
$SwLDA_{31}$	$\underset{\mathbf{W}}{argmax}\frac{tr(\mathbf{W}^T\mathbf{S}_b^{(3)}\mathbf{W})}{tr(\mathbf{W}^T\mathbf{S}_t^{(1)}\mathbf{W})}$
$SwLDA_{41}$	$\underset{\mathbf{W}}{argmax}\frac{tr(\mathbf{W}^T\mathbf{S}_b^{(4)}\mathbf{W})}{tr(\mathbf{W}^T\mathbf{S}_t^{(1)}\mathbf{W})}$
$SwLDA_{12}$	$\underset{\mathbf{W}}{argmax}\frac{tr(\mathbf{W}^T\mathbf{S}_b^{(1)}\mathbf{W})}{tr(\mathbf{W}^T\mathbf{S}_t^{(2)}\mathbf{W})}$
$SwLDA_{22}$	$\underset{\mathbf{W}}{argmax}\frac{tr(\mathbf{W}^T\mathbf{S}_b^{(2)}\mathbf{W})}{tr(\mathbf{W}^T\mathbf{S}_t^{(2)}\mathbf{W})}$
$SwLDA_{32}$	$\underset{\mathbf{W}}{argmax}\frac{tr(\mathbf{W}^T\mathbf{S}_b^{(3)}\mathbf{W})}{tr(\mathbf{W}^T\mathbf{S}_t^{(2)}\mathbf{W})}$
$SwLDA_{42}$	$\underset{\mathbf{W}}{argmax}\frac{tr(\mathbf{W}^T\mathbf{S}_b^{(4)}\mathbf{W})}{tr(\mathbf{W}^T\mathbf{S}_t^{(2)}\mathbf{W})}$

3.4 Classification

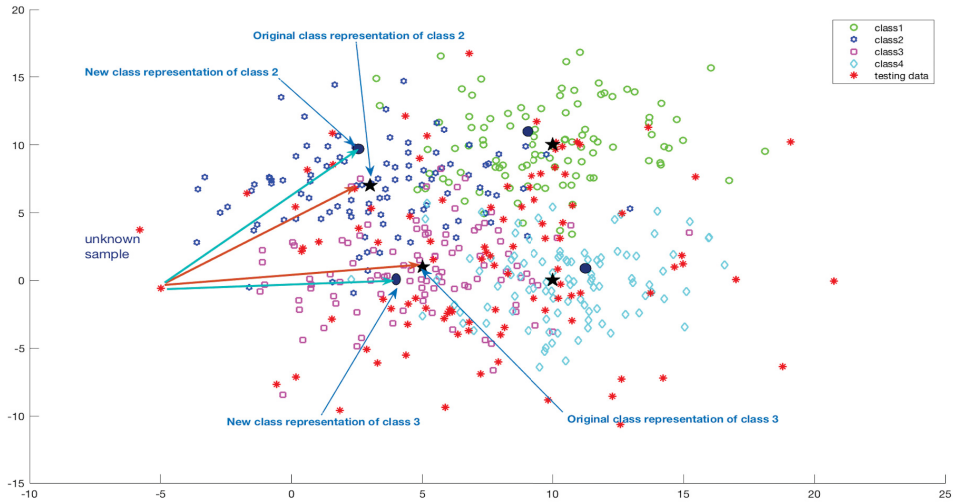
After obtaining the data representations in the discriminant space, we use k nearest neighbor or nearest centroid classifiers to predict class label of test data.

Nearest centroid classifier takes the projected test data $\mathbf{Z} = \{\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_n\}$, $\mathbf{Z} \in \mathbb{R}^{d\times n}$ and the projected centroids $\mathbf{M} = \{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, ..., \boldsymbol{\mu}_C\}$, $\mathbf{M} \in \mathbb{R}^{d\times C}$ as inputs, where n is the number of test samples and C is the number of classes (centroids).

For each sample $\mathbf{z}_i$, we employ Euclidean distance metric to measure the similarity between $\mathbf{z}_i$ and C centroids in the discriminant subspace, and then find the centroid nearest to $\mathbf{z}_i$ and assign its class $c \in (1, 2, ..., C)$ to $\mathbf{z}_i$. Centroids can be presented either by original class mean vectors, e.g. as in $SwLDA_{11}$, or new class representations (weighted class mean vectors), e.g. as in $SwLDA_{21}$. As the definition of centroid varies, the classification result may be different. Fig. (3.5) shows an unknown sample is classified to class 3 based on the new class representation, otherwise the unknown sample is labeled as class 2.

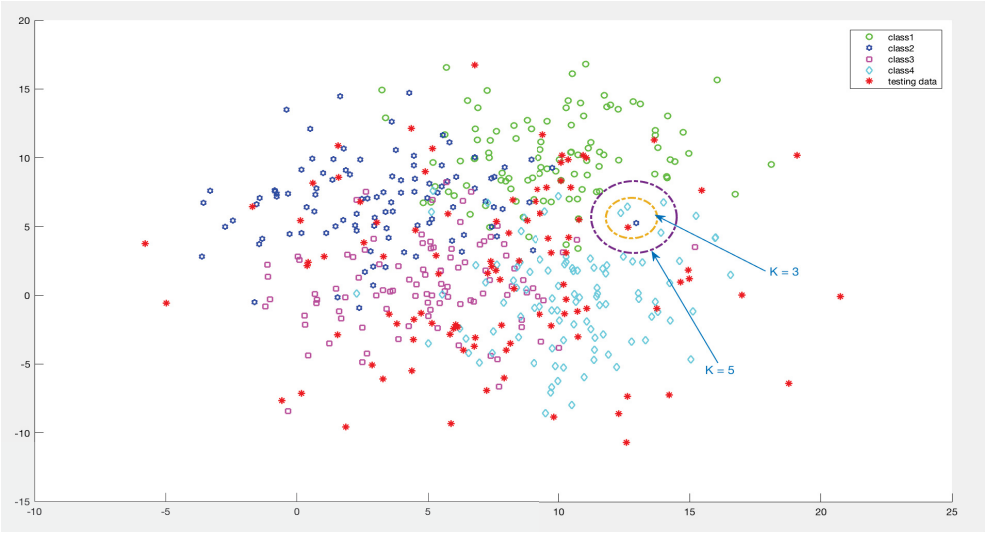
k-nearest neighbors classifier takes the projected training data $\mathbf{Z_1} = \{\mathbf{z}_1^{(1)}, \mathbf{z}_1^{(2)}, ..., \mathbf{z}_1^{(n_1)}\}$, $\mathbf{Z_1} \in \mathbb{R}^{d\times n_1}$, labels of training data $\mathbf{y} = \{y_i\}_{i=1}^{n_1}$, $y_i = 1, 2, ..., C$ and

Figure 3.5: Example of nearest centroid classification combined with $SwLDA$



the projected test data $\mathbf{Z_2} = \{\mathbf{z}_2^{(1)}, \mathbf{z}_2^{(2)}, ..., \mathbf{z}_2^{(n_2)}\}$, $\mathbf{Z_2} \in \mathbb{R}^{d \times n_2}$ as inputs, where n_1, n_2 are the number of samples in training and testing datasets, respectively. Euclidean distance metric is used to calculate the similarity between a test sample $\mathbf{z}_2^{(i)}$ and every sample in the training dataset. Then, we determine a training sample nearest to $\mathbf{z}_2^{(i)}$ sample and assign its label y_i as the label of $\mathbf{z}_2^{(i)}$, when k equals to 1. When k is larger than 1, we need to use a voting scheme between the k nearest neighbors to determine which class $y_i = 1, 2, ..., C$ has the highest frequency, and then label the test sample with it.

Figure 3.6: Example of k-nearest neighbor classification combined with $SwLDA$



In the experiments, we set three k values as 3, 5 and 7 to measure the effect of different numbers of neighbors on the accuracy. Fig. (3.6) presents an example of k-nearest neighbor classification. As shown in this figure, the unknown sample is labeled as class 4, regardless of whether k is set to 3 or 5.

4. EVALUATION

This chapter presents various datasets used in this thesis and experiments used to evaluate the performances of $SwLDA$ methods in classification tasks. We describe the results obtained from various $SwLDA$ methods and compare them to original LDA or weighted LDA variants for the same dataset.

4.1 Datasets

The proposed $SwLDA$ methods are evaluated on six publicly available facial image datasets: BU, KANADE, JAFFE, ORL, YALE and AR. Facial images are resized to a 40×30 pixels (gray-images) and vectorized to obtain facial vectors $\mathbf{x}_i \in \mathbb{R}^{1200}$. Each facial image dataset is described in detail in the followings.

BU dataset [47] is published by Binghamton University. BU dataset consists of 700 frontal face images of 100 persons, who have multiple ethnic backgrounds. Each person presents 7 facial expressions as 7 classes: anger, disgust, fear, happiness, sad, surprise and neural, as shown in Fig. (4.1).

Figure 4.1: First row: average image for each class. Second row: example images of one person for each class. Third row: example images of another person for each class.

JAFFE dataset [48] is a facial image dataset involved Japanese women, which consists of 210 images and 10 persons with 7 facial expressions as 7 classes: anger, disgust, fear, happiness, sad, surprise and neural, as shown in Fig. (4.2). Each expression is depicted by one person with 3 images.

Figure 4.2: Facial images from JAFFE dataset. First row: average image for each class. Second row: example images of one person for each class. Third row: example images of another person for each class.

COHN-KANADE dataset [49] consists of 245 frontal facial images related to 210 subjects with multiple ethnic backgrounds. This dataset is labeled with 7 classes, each class contains 35 images and depicts one kind of facial emotion, such as anger, disgust, happiness, sad, surprise and neural, as shown in Fig (4.3).

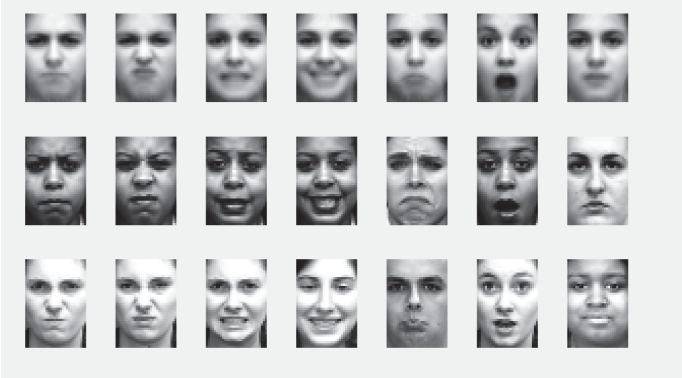
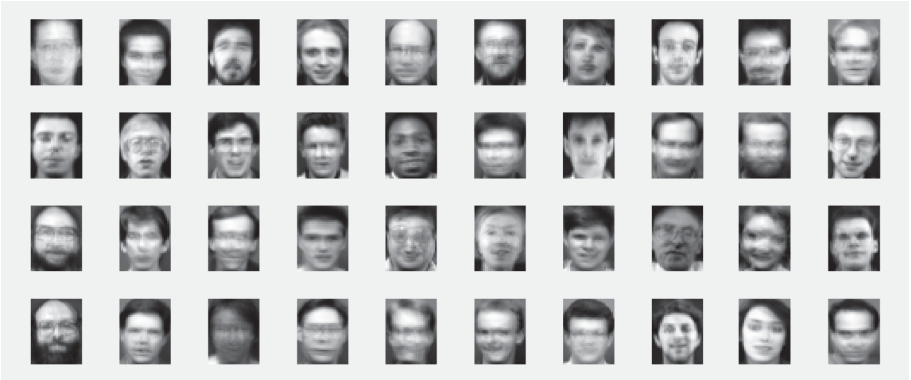


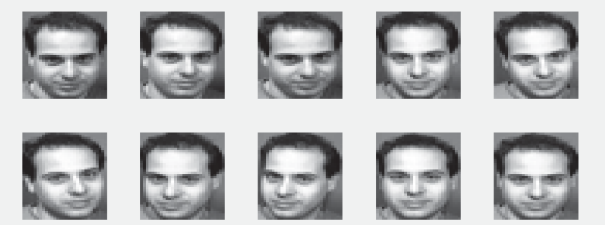
Figure 4.3: Facial images from KANADE dataset. First row: average image for each class. Second row: example images of one person for each class. Third row: example images of another person for each class.

ORL dataset [50] is provided by AT&T Laboratories Cambridge. This dataset consists of 400 facial images performed by 40 persons with variant poses, facial expressions and other facial details. Each class contains 10 images from the same person, hence there are 40 classes in ORL dataset. Fig. (4.4(a)) demonstrates the average images of each class and Fig. (4.4(b)) describes all poses performing by one person.

Extended YALE B dataset [51] consists of 2432 facial images from 38 persons, each person is labeled as one class. Each class contains 64 images which are captured



(a) Average images for each class



(b) 10 images from one class

Figure 4.4: Example images from ORL dataset

under variant illumination and different poses. Fig. (4.5(a)) demonstrates average images for each class and Fig. (4.5(b)) describes all poses performing by one person.

AR dataset [52] is provided by the Computer Vision Center (CVC) at the U.A.B. This dataset consists of 2600 facial images from 100 different persons. Each class is comprised by one person, who performs in 26 images by variant poses or different facial features. Fig. 4.6(a) demonstrates average images for 50 classes, and Fig. 4.6(b) describes all poses performing by one person or in one class.

Except for facial image datasets, the methods are evaluated on three imbalanced datasets: Balance, Contraceptive and Hayes-roth either. There are 625 samples with 3 classes in Balance dataset. In this dataset, the first class has 39 samples, the second class contains 226 samples and the third class consists of 235 samples. Contraceptive dataset contains 1473 samples with 3 classes. Its first class has 512 samples, second class has 258 samples and third class has 408 samples. Hayes-roth consists of 132 samples with 3 classes. The first class in this dataset contains 44 samples, the second class contains 42 samples and the last one has 19 samples.

(a) Average images for each class

(b) 64 images from one class

Figure 4.5: Example images from Extended Yale B dataset

(a) Average images for 50 classes

(b) 26 images from one class

Figure 4.6: Example images from AR dataset

4.2 Evaluation Procedure

Evaluation procedure in this work is quite straightforward and we just calculate accuracy from confused matrix directly. According to the combination of true result $\mathbf{y}$ and predicted result $\hat{\mathbf{y}}$, we can divide the result into four categories, as true positive (TP), false positive (FP), true negative (TN) and false negative (FN) as in Table (4.1):

Table 4.1: Confusion matrix for result

True $\mathbf{y}$	Predicted $\hat{\mathbf{y}}$	
	True	**False**
True	$TruePositive\,(TP)$	$FalseNegative\,(FN)$
False	$FalsePositive\,(FP)$	$TrueNegative\,(TN)$

- $TruePositive$: the number of true instances is classified correctly as true.
- $FalsePositive$: the number of false instances is classified wrongly as true.
- $FalseNegative$: the number of true instances is classified wrongly as false.
- $TrueNegative$: the number of false instances is classified correctly as false.

According to above items, accuracy is defined as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}. \tag{4.1}$$

The five-fold cross-validation procedure is used. For each experiment, the mean accuracy over all folds is reported.

4.3 Experimental Results

Experimental results are demonstrated separately, according to the six facial image datasets and three imbalanced datasets. For each dataset, we exploit classification performances based on two classifiers with different parameters. The result of traditional LDA is considered as baseline. Tang's work [17] is named as relevance WLDA and Jarchi's work [18] is named as new WLDA.

- **BU dataset**: Table (4.2) shows the classification accuracy of BU dataset with three types of prior information matrix $\mathbf{V}_c$ and two kinds of graph connection by nearest centroid classifier. Table (4.3), Table (4.4) and Table (4.5) describe results with three types of prior information matrix $\mathbf{V}_c$ based on k-nearest

neighbor classifier, respectively. Results of traditional LDA and other weighted LDA variants are illustrated in Table (4.6).

According to the results shown in Table (4.2), Table (4.3), Table (4.4), Table (4.5) and Table (4.6), $SwLDA_{42}$ with distance-based probability prior information combined with nearest centroid classifier in Table (4.2) achieves the best result, equals to **0.6843**. The maximal improvement is 11.14% comparing to the result of traditional LDA combined with nearest centroid classifier. That over relevance WLDA is 0.14% and over new WLDA is 2.28%. Results involved the forth definition of between-class scatter matrix, such as $SwLDA_{41}$ or $SwLDA_{42}$, are superior to the results of the other three definitions of between-class scatter matrix under the same conditions.

Table 4.2: Classification accuracy by nearest centroid classifier

Method	Equal probability		Distance-based probability		Misclassification-based probability	
K	N_c	$\min(5, 0.1 * N_c)$	N_c	$\min(5, 0.1 * N_{ci})$	N_c	$\min(5, 0.1 * N_{ci})$
$\mathbf{SwLDA}_{11}$	0.5714	0.5714	0.5714	0.5700	0.5714	0.5714
$\mathbf{SwLDA}_{21}$	0.5714	0.5714	0.5700	0.5686	0.5714	0.5700
$\mathbf{SwLDA}_{31}$	0.5871	0.5871	0.5900	0.5843	0.5886	0.5857
$\mathbf{SwLDA}_{41}$	0.6514	0.6514	0.6500	0.6543	0.6500	0.6514
$\mathbf{SwLDA}_{12}$	0.5814	0.5814	0.5843	0.5771	0.5814	0.5814
$\mathbf{SwLDA}_{22}$	0.5814	0.5814	0.5829	0.5800	0.5814	0.5814
$\mathbf{SwLDA}_{32}$	0.6243	0.6243	0.6229	0.6086	0.6243	0.6243
$\mathbf{SwLDA}_{42}$	0.6786	0.6786	0.6786	**0.6843**	0.6786	0.6786

Table 4.3: Classification accuracy by k-nearest neighbor classifier based on **Equal probability**

Method	k=3		k=5		k=7	
K	N_c	$\min(5, 0.1 * N_c)$	N_c	$\min(5, 0.1 * N_c)$	N_c	$\min(5, 0.1 * N_c)$
$\mathbf{SwLDA}_{11}$	0.5671	0.5671	0.5743	0.5743	0.5686	0.5686
$\mathbf{SwLDA}_{21}$	0.5671	0.5671	0.5743	0.5743	0.5686	0.5686
$\mathbf{SwLDA}_{31}$	0.5800	0.5800	0.5800	0.5800	0.5829	0.5829
$\mathbf{SwLDA}_{41}$	0.6486	0.6486	0.6500	0.6500	0.6571	0.6571
$\mathbf{SwLDA}_{12}$	0.5671	0.5671	0.5743	0.5743	0.5686	0.5686
$\mathbf{SwLDA}_{22}$	0.5671	0.5671	0.5743	0.5743	0.5686	0.5686
$\mathbf{SwLDA}_{32}$	0.5800	0.5800	0.5800	0.5800	0.5829	0.5829
$\mathbf{SwLDA}_{42}$	0.6486	0.6486	0.6500	0.6500	0.6571	0.6571

Table 4.4: Classification accuracy by k-nearest neighbor classifier based on **Distance-based probability**

Method	k=3		k=5		k=7	
K	N_c	$\min(5, 0.1 * N_c)$	N_c	$\min(5, 0.1 * N_c)$	N_c	$\min(5, 0.1 * N_c)$
$\mathbf{SwLDA}_{11}$	0.5671	0.5729	0.5743	0.5700	0.5686	0.5714
$\mathbf{SwLDA}_{21}$	0.5671	0.5771	0.5729	0.5729	0.5686	0.5729
$\mathbf{SwLDA}_{31}$	0.5771	0.5757	0.5800	0.5829	0.5843	0.5829
$\mathbf{SwLDA}_{41}$	0.6471	0.6443	0.6500	0.6443	0.6600	0.6586
$\mathbf{SwLDA}_{12}$	0.5671	0.5629	0.5743	0.5686	0.5686	0.5657
$\mathbf{SwLDA}_{22}$	0.5671	0.5671	0.5743	0.5743	0.5686	0.5686
$\mathbf{SwLDA}_{32}$	0.5771	0.5829	0.5800	0.5886	0.5843	0.5814
$\mathbf{SwLDA}_{42}$	0.6471	0.6500	0.6500	0.6457	0.6600	0.6514

Table 4.5: Classification accuracy by k-nearest neighbor classifier based on **Misclassification-based probability**

Method	k=3		k=5		k=7	
K	N_c	$\min(5, 0.1 * N_c)$	N_c	$\min(5, 0.1 * N_c)$	N_c	$\min(5, 0.1 * N_c)$
$\mathbf{SwLDA}_{11}$	0.5671	0.5729	0.5743	0.5743	0.5686	0.5743
$\mathbf{SwLDA}_{21}$	0.5671	0.5729	0.5743	0.5757	0.5686	0.5729
$\mathbf{SwLDA}_{31}$	0.5786	0.5814	0.5800	0.5743	0.5829	0.5843
$\mathbf{SwLDA}_{41}$	0.6500	0.6471	0.6500	0.6514	0.6571	0.6586
$\mathbf{SwLDA}_{12}$	0.5671	0.5729	0.5743	0.5743	0.5686	0.5743
$\mathbf{SwLDA}_{22}$	0.5671	0.5729	0.5743	0.5757	0.5686	0.5729
$\mathbf{SwLDA}_{32}$	0.5786	0.5814	0.5800	0.5743	0.5829	0.5843
$\mathbf{SwLDA}_{42}$	0.6500	0.6471	0.6500	0.6514	0.6571	0.6586

Table 4.6: Classification accuracy by traditional LDA and other LDA variants

Method	Nearest centroid	k-nearest neighbor		
		k=3	k=5	k=7
Original LDA	0.5729	0.5743	0.5700	0.5729
Relevance WLDA	0.5743	0.5657	0.5729	0.5729
New WLDA	0.5957	0.5714	0.5714	0.5686

- **KANADE dataset**: Table (4.7) shows classification accuracy based on nearest centroid classifier, Table (4.8), Table (4.9) and Table (4.10) present results with three types of prior information matrix $\mathbf{V}_c$, respectively. Results of traditional LDA and other weighted LDA variants are illustrated in Table (4.11).

  According to the results in the following tables, $SwLDA_{42}$ or $SwLDA_{41}$ combined with nearest centroid classifier achieves the best result **0.7224**, when the priori information is equal probability, regardless of the influence of graph connection. The maximum improvement is 3.26%, comparing to the result of traditional LDA combined with nearest centroid classifier. Results of the forth definition of between-class scatter matrix, such as $SwLDA_{41}$ or $SwLDA_{42}$, are superior to the results of the other three definitions of between-class scatter matrix under the same conditions.

Table 4.7: Classification accuracy by nearest centroid classifier

Method	Equal probability		Distance-based probability		Misclassification-based probability	
K	N_c	$\min(5, 0.1 * N_c)$	N_c	$\min(5, 0.1 * N_{ci})$	N_c	$\min(5, 0.1 * N_{ci})$
$\mathbf{SwLDA}_{11}$	0.6816	0.6816	0.6816	0.6735	0.6816	0.6898
$\mathbf{SwLDA}_{21}$	0.6816	0.6816	0.6816	0.6776	0.6816	0.6816
$\mathbf{SwLDA}_{31}$	0.6776	0.6776	0.6816	0.6776	0.6816	0.6816
$\mathbf{SwLDA}_{41}$	**0.7224**	**0.7224**	0.7020	0.6980	0.7020	0.6980
$\mathbf{SwLDA}_{12}$	0.6816	0.6816	0.6816	0.6857	0.6816	0.6857
$\mathbf{SwLDA}_{22}$	0.6816	0.6816	0.6816	0.6857	0.6817	0.6776
$\mathbf{SwLDA}_{32}$	0.6776	0.6776	0.6776	0.6816	0.6735	0.6939
$\mathbf{SwLDA}_{42}$	**0.7224**	**0.7224**	**0.7224**	**0.7224**	**0.7224**	0.7184

Table 4.8: Classification accuracy by k-nearest neighbor classifier based on **Equal probability**

Method	k=3		k=5		k=7	
K	$\min(5, 0.1 * N_c)$	N_c	$\min(5, 0.1 * N_c)$	N_c	$\min(5, 0.1 * N_c)$	
$\mathbf{SwLDA}_{11}$	0.6776	0.6776	0.6776	0.6776	0.6857	0.6857
$\mathbf{SwLDA}_{21}$	0.6776	0.6776	0.6776	0.6776	0.6857	0.6857
$\mathbf{SwLDA}_{31}$	0.6776	0.6776	0.6776	0.6776	0.6776	0.6776
$\mathbf{SwLDA}_{41}$	0.6857	0.6857	0.6898	0.6898	0.6898	0.6898
$\mathbf{SwLDA}_{12}$	0.6776	0.6776	0.6776	0.6776	0.6857	0.6857
$\mathbf{SwLDA}_{22}$	0.6776	0.6776	0.6776	0.6776	0.6857	0.6857
$\mathbf{SwLDA}_{32}$	0.6776	0.6776	0.6776	0.6776	0.6776	0.6776
$\mathbf{SwLDA}_{42}$	0.6857	0.6857	0.6898	0.6898	0.6898	0.6898

Table 4.9: Classification accuracy by k-nearest neighbor classifier based on **Distance-based probability**

Method	k=3		k=5		k=7	
K	N_c	$\min(5, 0.1 * N_c)$	N_c	$\min(5, 0.1 * N_c)$	N_c	$\min(5, 0.1 * N_c)$
$\mathbf{SwLDA}_{11}$	0.6816	0.6776	0.6776	0.6776	0.6816	0.6776
$\mathbf{SwLDA}_{21}$	0.6816	0.6735	0.6776	0.6776	0.6816	0.6776
$\mathbf{SwLDA}_{31}$	0.6776	0.6816	0.6776	0.6776	0.6816	0.6816
$\mathbf{SwLDA}_{41}$	0.6857	0.6980	0.6898	0.7020	0.6939	0.6939
$\mathbf{SwLDA}_{12}$	0.6816	0.6776	0.6776	0.6816	0.6816	0.6816
$\mathbf{SwLDA}_{22}$	0.6816	0.6816	0.6776	0.6776	0.6816	0.6816
$\mathbf{SwLDA}_{32}$	0.6776	0.6857	0.6776	0.6776	0.6816	0.6776
$\mathbf{SwLDA}_{42}$	0.6857	0.6939	0.6898	0.6980	0.6939	0.6939

Table 4.10: Classification accuracy of k-nearest neighbor classifier based on **Misclassification-based probability**

Method	k=3		k=5		k=7	
K	N_c	$\min(5, 0.1 * N_c)$	N_c	$\min(5, 0.1 * N_c)$	N_c	$\min(5, 0.1 * N_c)$
$\mathbf{SwLDA}_{11}$	0.6776	0.6816	0.6776	0.6816	0.6857	0.6816
$\mathbf{SwLDA}_{21}$	0.6776	0.6776	0.6776	0.6776	0.6857	0.6776
$\mathbf{SwLDA}_{31}$	0.6776	0.6857	0.6816	0.6857	0.6816	0.6857
$\mathbf{SwLDA}_{41}$	0.6857	0.6939	0.6898	0.6980	0.6898	0.6939
$\mathbf{SwLDA}_{12}$	0.6776	0.6816	0.6776	0.6857	0.6857	0.6857
$\mathbf{SwLDA}_{22}$	0.5306	0.5469	0.5388	0.5551	0.5469	0.5429
$\mathbf{SwLDA}_{32}$	0.6776	0.6735	0.6816	0.6735	0.6816	0.6694
$\mathbf{SwLDA}_{42}$	0.6857	0.6939	0.6898	0.6980	0.6898	0.6898

Table 4.11: Classification accuracy for KANADE dataset of traditional LDA and LDA variants

Method	Nearest centroid	k-nearest neighbor		
		k=3	k=5	k=7
Original LDA	0.6898	0.6857	0.6898	0.6857
Relevance WLDA	0.6857	0.6939	0.6939	.6898
New WLDA	0.6898	0.6857	0.6816	.6898

- **JAFFE dataset:** Table (4.12) shows the classification accuracy based on nearest centroid classifier. Table (4.13), Table (4.14) and Table (4.15) show results with three types of priori information matrix $\mathbf{V}_c$ combined with k-nearest neighbor classifier, respectively. Results of traditional LDA, relevance weighted LDA and new weighted LDA are illustrated in Table (4.16).

  The best performance is **0.5905** by $SwLDA_{41}$ based on nearest centroid classifier. The maximum improvement is 3.34%, comparing to the result of traditional LDA based on nearest centroid classifier. Moreover, the type of graph connection merely affects the accuracy with misclassification-based probability by $SwLDA_{41}$ combined with nearest centroid classifier. Results related to $\mathbf{S}_w^{(3)}$, such as $SwLDA_{31}$ or $SwLDA_{32}$, are inferior to the results related to the other three definitions of $\mathbf{S}_w$ in most cases.

Table 4.12: Classification accuracy by nearest centroid classifier

Method	Equal probability		Distance-based probability		Misclassification-based probability	
K	N_c	$\min(5, 0.1 * N_c)$	N_c	$\min(5, 0.1 * N_{ci})$	N_c	$\min(5, 0.1 * N_{ci})$
$\mathbf{SwLDA}_{11}$	0.5619	0.5619	0.5667	0.5667	0.5619	0.5762
$\mathbf{SwLDA}_{21}$	0.5619	0.5619	0.5667	0.5667	0.5619	0.5762
$\mathbf{SwLDA}_{31}$	0.5524	0.5524	0.5524	0.5571	0.5524	0.5619
$\mathbf{SwLDA}_{41}$	**0.5905**	**0.5905**	**0.5905**	**0.5905**	**0.5905**	0.5857
$\mathbf{SwLDA}_{12}$	0.5667	0.5667	0.5667	0.5714	0.5667	0.5667
$\mathbf{SwLDA}_{22}$	0.5667	0.5667	0.5667	0.5714	0.5667	0.5762
$\mathbf{SwLDA}_{32}$	0.5381	0.5381	0.5381	0.5238	0.5286	0.5333
$\mathbf{SwLDA}_{42}$	0.5476	0.5476	0.5524	0.5476	0.5476	0.5424

Table 4.13: Classification accuracy by k-nearest neighbor classifier based on **Equal probability**

Method	k=3		k=5		k=7	
K	N_c	$\min(5, 0.1 * N_c)$	N_c	$\min(5, 0.1 * N_c)$	N_c	$\min(5, 0.1 * N_c)$
$\mathbf{SwLDA}_{11}$	0.5667	0.5667	0.5667	0.5667	0.5667	0.5667
$\mathbf{SwLDA}_{21}$	0.5667	0.5667	0.5667	0.5667	0.5667	0.5667
$\mathbf{SwLDA}_{31}$	0.5571	0.5571	0.5571	0.5571	0.5571	0.5571
$\mathbf{SwLDA}_{41}$	0.5810	0.5810	0.5810	0.5810	0.5810	0.5810
$\mathbf{SwLDA}_{12}$	0.5667	0.5667	0.5667	0.5667	0.5667	0.5667
$\mathbf{SwLDA}_{22}$	0.5667	0.5667	0.5667	0.5667	0.5667	0.5667
$\mathbf{SwLDA}_{32}$	0.5571	0.5571	0.5571	0.5571	0.5571	0.5571
$\mathbf{SwLDA}_{42}$	0.5810	0.5810	0.5810	0.5810	0.5810	0.5810

Table 4.14: Classification accuracy of JAFFE dataset for k-nearest neighbor classifier based on **Distance-based probability**

Method	k=3		k=5		k=7	
K	N_c	$\min(5, 0.1 * N_c)$	N_c	$\min(5, 0.1 * N_c)$	N_c	$\min(5, 0.1 * N_c)$
$\mathbf{SwLDA}_{11}$	0.5667	0.5762	0.5667	0.5762	0.5667	0.5667
$\mathbf{SwLDA}_{21}$	0.5667	0.5619	0.5667	0.5667	0.5667	0.5667
$\mathbf{SwLDA}_{31}$	0.5524	0.5571	0.5524	0.5571	0.5524	0.5571
$\mathbf{SwLDA}_{41}$	0.5810	0.5762	0.5810	0.5810	0.5810	0.5857
$\mathbf{SwLDA}_{12}$	0.5667	0.5619	0.5667	0.5714	0.5667	0.5714
$\mathbf{SwLDA}_{22}$	0.5810	0.5810	0.5810	0.5810	0.5810	0.5810
$\mathbf{SwLDA}_{32}$	0.5524	0.5571	0.5524	0.5524	0.5524	0.5619
$\mathbf{SwLDA}_{42}$	0.5810	0.5762	0.5810	0.5762	0.5810	0.5657

Table 4.15: Classification accuracy of JAFFE dataset for k-nearest neighbor classifier based on **Misclassification-based probability**

Method	k=3		k=5		k=7	
K	N_c	$\min(5, 0.1 * N_c)$	N_c	$\min(5, 0.1 * N_c)$	N_c	$\min(5, 0.1 * N_c)$
$\mathbf{SwLDA}_{11}$	0.5667	0.5762	0.5667	0.5762	0.5667	0.5667
$\mathbf{SwLDA}_{21}$	0.5667	0.5762	0.5667	0.5762	0.5667	0.5667
$\mathbf{SwLDA}_{31}$	0.5524	0.5571	0.5571	0.5571	0.5571	0.5619
$\mathbf{SwLDA}_{41}$	0.5810	0.5857	0.5810	0.5857	0.5810	0.5857
$\mathbf{SwLDA}_{12}$	0.5667	0.5762	0.5667	0.5762	0.5667	0.5762
$\mathbf{SwLDA}_{22}$	0.5286	0.5143	0.5286	0.5095	0.5000	0.5095
$\mathbf{SwLDA}_{32}$	0.5524	0.5571	0.5571	0.5571	0.5571	0.5619
$\mathbf{SwLDA}_{42}$	0, 5810	0, 5810	0, 5810	0, 5857	0, 5810	0, 5952

Table 4.16: Classification accuracy for JAFFE dataset of traditional LDA and LDA variants

Method	Nearest centroid	k-nearest neighbor		
		k=3	k=5	k=7
Original LDA	0.5571	0.5571	0.5571	0.5571
Relevance WLDA	0.5714	0.5619	0.5619	0.5619
New WLDA	0.5381	0.5429	0.5381	0.5429

- **ORL dataset**: Table (4.17) shows the classification accuracy of ORL dataset based on nearest centroid classifier. Table (4.18), Table (4.19) and Table (4.20) present results with three types of prior information matrix $\mathbf{V}_c$ combined with k-nearest neighbor classifier, respectively. Results of traditional LDA and weighted LDA variants are illustrated in Table (4.21).

  The best performance is **0.9900** by $SwLDA_{41}$ combined with k-nearest neighbor classifier, when $k = 3$ and the prior information is either equal probability or distance-based probability, regardless of the connection of graph. When within-scatter matrix is the first definition, $SwLDA_{31}$ or $SwLDA_{41}$ works better than $SwLDA_{11}$ or $SwLDA_{21}$ under the same conditions. In addition, $SwLDA_{12}$ or $SwLDA_{22}$ works better than $SwLDA_{32}$ or $SwLDA_{42}$ with the second within-scatter matrix under the same conditions.

Table 4.17: Classification accuracy of ORL dataset for nearest centroid classifier

Method	Equal probability		Distance-based probability		Misclassification-based probability	
K	N_c	$\min(5, 0.1 * N_c)$	N_c	$\min(5, 0.1 * N_{ci})$	N_c	$\min(5, 0.1 * N_{ci})$
$\mathbf{SwLDA}_{11}$	0.9700	0.9700	0.9700	0.9700	0.9700	0.9700
$\mathbf{SwLDA}_{21}$	0.9700	0.9700	0.9700	0.9700	0.9700	0.9700
$\mathbf{SwLDA}_{31}$	0.9850	0.9850	0.9850	0.9850	0.9850	0.9850
$\mathbf{SwLDA}_{41}$	0.9825	0.9825	0.9825	0.9825	0.9850	0.9825
$\mathbf{SwLDA}_{12}$	0.9850	0.9850	0.9850	0.9850	0.9850	0.9850
$\mathbf{SwLDA}_{22}$	0.9850	0.9850	0.9850	0.9850	0.9850	0.9850
$\mathbf{SwLDA}_{32}$	0.9675	0.9675	0.9625	0.9625	0.9600	0.9600
$\mathbf{SwLDA}_{42}$	0.9425	0.9425	0.9425	0.9400	0.9475	0.9450

Table 4.18: Classification accuracy by k-nearest neighbor classifier based on **Equal probability**

Method	k=3		k=5		k=7	
K	N_c	$\min(5, 0.1 * N_c)$	N_c	$\min(5, 0.1 * N_c)$	N_c	$\min(5, 0.1 * N_c)$
$\mathbf{SwLDA}_{11}$	0.9700	0.9700	0.9700	0.9700	0.9700	0.9700
$\mathbf{SwLDA}_{21}$	0.9700	0.9700	0.9700	0.9700	0.9700	0.9700
$\mathbf{SwLDA}_{31}$	0.9850	0.9850	0.9850	0.9850	0.9850	0.9850
$\mathbf{SwLDA}_{41}$	**0.9900**	**0.9900**	0.9800	0.9800	0.9825	0.9825
$\mathbf{SwLDA}_{12}$	0.9850	0.9850	0.9850	0.9850	0.9850	0.9850
$\mathbf{SwLDA}_{22}$	0.9850	0.9850	0.9850	0.9850	0.9850	0.9850
$\mathbf{SwLDA}_{32}$	0.9675	0.9675	0.9625	0.9625	0.9375	0.9375
$\mathbf{SwLDA}_{42}$	0.9550	0.9550	0.9450	0.9450	0.9125	0.9125

Table 4.19: Classification accuracy of ORL dataset for k-nearest neighbor classifier based on **Distance-based probability**

Method	k=3		k=5		k=7	
K	N_c	$\min(5, 0.1 * N_c)$	N_c	$\min(5, 0.1 * N_c)$	N_c	$\min(5, 0.1 * N_c)$
$\mathbf{SwLDA}_{11}$	0.9700	0.9700	0.9700	0.9700	0.9700	0.9700
$\mathbf{SwLDA}_{21}$	0.9700	0.9700	0.9700	0.9700	0.9700	0.9700
$\mathbf{SwLDA}_{31}$	0.9850	0.9850	0.9850	0.9850	0.9850	0.9850
$\mathbf{SwLDA}_{41}$	**0.9900**	**0.9900**	0.9825	0.9800	0.9825	0.9825
$\mathbf{SwLDA}_{12}$	0.9850	0.9850	0.9850	0.9850	0.9850	0.9850
$\mathbf{SwLDA}_{22}$	0.9850	0.9850	0.9850	0.9850	0.9850	0.9850
$\mathbf{SwLDA}_{32}$	0.9675	0.9650	0.9575	0.9600	0.9375	0.9350
$\mathbf{SwLDA}_{42}$	0.9550	0.9550	0.9450	0.9450	0.9125	0.9150

Table 4.20: Classification accuracy of ORL dataset for k-nearest neighbor classifier based on **Misclassification-based probability**

Method	k=3		k=5		k=7	
K	N_c	$\min(5, 0.1 * N_c)$	N_c	$\min(5, 0.1 * N_c)$	N_c	$\min(5, 0.1 * N_c)$
$\mathbf{SwLDA}_{11}$	0.9700	0.9700	0.9700	0.9700	0.9700	0.9700
$\mathbf{SwLDA}_{21}$	0.9700	0.9700	0.9700	0.9700	0.9700	0.9700
$\mathbf{SwLDA}_{31}$	0.9850	0.9850	0.9850	0.9850	0.9850	0.9850
$\mathbf{SwLDA}_{41}$	0.9850	0.9875	0.9800	0.9800	0.9825	0.9825
$\mathbf{SwLDA}_{12}$	0.9850	0.9850	0.9850	0.9850	0.9850	0.9850
$\mathbf{SwLDA}_{22}$	0.9850	0.9850	0.9850	0.9850	0.9850	0.9850
$\mathbf{SwLDA}_{32}$	0.9675	0.9650	0.9575	0.9575	0.9375	0.9375
$\mathbf{SwLDA}_{42}$	0.9550	0.9550	0.9375	0.9350	0.9100	0.9150

Table 4.21: Classification accuracy by traditional LDA and other LDA variants

Method	Nearest centroid	k-nearest neighbor		
		k=3	k=5	k=7
Original LDA	0.9725	0.9725	0.9725	0.9725
Relevance WLDA	0.9800	0.9725	0.9700	0.9725
New WLDA	0.9800	0.9750	0.9750	0.9700

- **YALE dataset**: Table (4.22) shows the classification accuracy of YALE dataset based on nearest centroid classifier. Table (4.23), Table (4.24) and Table (4.25) present results with three types of prior information matrix $\mathbf{V}_c$ combined with k-nearest neighbor classifier, respectively. Results of traditional LDA and weighted LDA variants are illustrated in Table (4.26).

  The best performance is **0.9601** achieved by $SwLDA_{42}$ combined with nearest centroid classifier and distance-based probability, when graph is fully connected. The maximum improvement is 1.08% comparing to the result of traditional LDA based on nearest centroid classifier. Nearest centroid classifier achieves better performance in this dataset, comparing to k-nearest neighbor classifier under the same circumstances.

Table 4.22: Classification accuracy by nearest centroid classifier

Method	Equal probability		Distance-based probability		Misclassification-based probability	
K	N_c	$\min(5, 0.1 * N_c)$	N_c	$\min(5, 0.1 * N_{ci})$	N_c	$\min(5, 0.1 * N_{ci})$
$\mathbf{SwLDA}_{11}$	0.9597	0.9597	0.9597	0.9580	0.9597	0.9580
$\mathbf{SwLDA}_{21}$	0.9597	0.9597	0.9597	0.9572	0.9597	0.9576
$\mathbf{SwLDA}_{31}$	0.9597	0.9597	0.9597	0.9572	0.9597	0.9576
$\mathbf{SwLDA}_{41}$	0.9597	0.9597	0.9597	0.9572	0.9597	0.9576
$\mathbf{SwLDA}_{12}$	0.9593	0.9593	0.9597	0.9585	0.9593	0.9597
$\mathbf{SwLDA}_{22}$	0.9593	0.9593	0.9597	0.9584	0.9593	0.9596
$\mathbf{SwLDA}_{32}$	0.9593	0.9593	0.9597	0.9584	0.9593	0.9596
$\mathbf{SwLDA}_{42}$	0.9597	0.9597	**0.9601**	0.9589	0.9597	0.9580

Table 4.23: Classification accuracy by k-nearest neighbor classifier based on **Equal probability**

Method	k=3		k=5		k=7	
K	N_c	$\min(5, 0.1 * N_c)$	N_c	$\min(5, 0.1 * N_c)$	N_c	$\min(5, 0.1 * N_c)$
$\mathbf{SwLDA}_{11}$	0.9482	0.9482	0.9474	0.9474	0.9457	0.9457
$\mathbf{SwLDA}_{21}$	0.9482	0.9482	0.9474	0.9474	0.9457	0.9457
$\mathbf{SwLDA}_{31}$	0.9482	0.9482	0.9474	0.9474	0.9457	0.9457
$\mathbf{SwLDA}_{41}$	0.9482	0.9482	0.9474	0.9474	0.9457	0.9457
$\mathbf{SwLDA}_{12}$	0.9490	0.9490	0.9465	0.9465	0.9469	0.9469
$\mathbf{SwLDA}_{22}$	0.9490	0.9490	0.9465	0.9465	0.9469	0.9469
$\mathbf{SwLDA}_{32}$	0.9490	0.9490	0.9469	0.9469	0.9469	0.9469
$\mathbf{SwLDA}_{42}$	0.9498	0.9498	0.9478	0.9478	0.9469	0.9469

Table 4.24: Classification accuracy by k-nearest neighbor classifier based on **Distance-based probability**

Method	k=3		k=5		k=7	
K	N_c	$\min(5, 0.1 * N_c)$	N_c	$\min(5, 0.1 * N_c)$	N_c	$\min(5, 0.1 * N_c)$
$\mathbf{SwLDA}_{11}$	0.9478	0.9486	0.9474	0.9465	0.9457	0.9461
$\mathbf{SwLDA}_{21}$	0.9478	0.9482	0.9474	0.9469	0.9457	0.9457
$\mathbf{SwLDA}_{31}$	0.9478	0.9486	0.9474	0.9465	0.9457	0.9461
$\mathbf{SwLDA}_{41}$	0.9478	0.9482	0.9474	0.9469	0.9457	0.9461
$\mathbf{SwLDA}_{12}$	0.9490	0.9478	0.9469	0.9469	0.9474	0.9473
$\mathbf{SwLDA}_{22}$	0.9490	0.9478	0.9469	0.9482	0.9474	0.9474
$\mathbf{SwLDA}_{32}$	0.9490	0.9478	0.9469	0.9482	0.9474	0.9474
$\mathbf{SwLDA}_{42}$	0.9494	0.9478	0.9478	0.9482	0.9474	0.9474

Table 4.25: Classification accuracy by k-nearest neighbor classifier based on **Misclassification-based probability**

Method	k=3		k=5		k=7	
K	N_c	$\min(5, 0.1 * N_c)$	N_c	$\min(5, 0.1 * N_c)$	N_c	$\min(5, 0.1 * N_c)$
$\mathbf{SwLDA}_{11}$	0.9482	0.9478	0.9474	0.9461	0.9457	0.9457
$\mathbf{SwLDA}_{21}$	0.9482	0.9486	0.9474	0.9478	0.9457	0.9469
$\mathbf{SwLDA}_{31}$	0.9482	0.9486	0.9474	0.9478	0.9457	0.9469
$\mathbf{SwLDA}_{41}$	0.9482	0.9486	0.9474	0.9478	0.9457	0.9469
$\mathbf{SwLDA}_{12}$	0.9490	0.9486	0.9465	0.9457	0.9469	0.9469
$\mathbf{SwLDA}_{22}$	0.9490	0.9490	0.9469	0.9482	0.9469	0.9474
$\mathbf{SwLDA}_{32}$	0.9490	0.9490	0.9469	0.9482	0.9469	0.9474
$\mathbf{SwLDA}_{42}$	0.9498	0.9494	0.9486	0.9486	0.9469	0.9474

Table 4.26: Classification accuracy by traditional LDA and LDA variants

Method	Nearest centroid	k-nearest neighbor		
		k=3	k=5	k=7
Original LDA	0.9593	0.9482	0.9465	0.9461
Relevance WLDA	0.9564	0.9457	0.9437	0.9441
New WLDA	0.9597	0.9482	0.9469	0.9461

- **AR dataset**: Table (4.27) shows the classification accuracy of AR dataset based on nearest centroid classifier. Table (4.28), Table (4.29) and Table (4.30) present results with three types of prior information matrix $\mathbf{V}_c$ combined with k-nearest neighbor classifier, respectively. Results of traditional LDA and weighted LDA variants are illustrated in Table (4.31).

  The best performance is **0.9704** based on k-nearest neighbor classifier with three types of priori information under various circumstances. For instance, when the priori information is equal probability and k is set to 7, all results related to the first definition of within-class scatter matrix are the highest accuracy **0.9704**, regardless of graph connection.

Table 4.27: Classification accuracy by nearest centroid classifier

Method	Equal probability		Distance-based probability		Misclassification-based probability	
K	N_c	$\min(5, 0.1 * N_c)$	N_c	$\min(5, 0.1 * N_{ci})$	N_c	$\min(5, 0.1 * N_{ci})$
$\mathbf{SwLDA}_{11}$	0.9696	0.9696	0.9696	0.9696	0.9696	0.9696
$\mathbf{SwLDA}_{21}$	0.9696	0.9696	0.9696	0.9696	0.9696	0.9696
$\mathbf{SwLDA}_{31}$	0.9696	0.9696	0.9696	0.9696	0.9696	0.9696
$\mathbf{SwLDA}_{41}$	0.9696	0.9696	0.9696	0.9696	0.9681	0.9681
$\mathbf{SwLDA}_{12}$	0.9681	0.9681	0.9684	0.9677	0.9681	0.9681
$\mathbf{SwLDA}_{22}$	0.9681	0.9681	0.9684	0.9692	0.9681	0.9681
$\mathbf{SwLDA}_{32}$	0.9681	0.9681	0.9684	0.9692	0.9681	0.9681
$\mathbf{SwLDA}_{42}$	0.9681	0.9681	0.9684	0.9684	0.9681	0.9692

Table 4.28: Classification accuracy by k-nearest neighbor classifier based on **Equal probability**

Method	k=3		k=5		k=7	
K	N_c	$\min(5, 0.1 * N_c)$	N_c	$\min(5, 0.1 * N_c)$	N_c	$\min(5, 0.1 * N_c)$
$\mathbf{SwLDA}_{11}$	0.9696	0.9696	0.9700	0.9700	**0.9704**	**0.9704**
$\mathbf{SwLDA}_{21}$	0.9696	0.9696	0.9700	0.9700	**0.9704**	**0.9704**
$\mathbf{SwLDA}_{31}$	0.9696	0.9696	0.9700	0.9700	**0.9704**	**0.9704**
$\mathbf{SwLDA}_{41}$	0.9696	0.9696	0.9700	0.9700	**0.9704**	**0.9704**
$\mathbf{SwLDA}_{12}$	0.9669	0.9669	0.9700	0.9700	0.9700	0.9700
$\mathbf{SwLDA}_{22}$	0.9669	0.9669	0.9700	0.9700	0.9700	0.9700
$\mathbf{SwLDA}_{32}$	0.9669	0.9669	0.9700	0.9700	0.9700	0.9700
$\mathbf{SwLDA}_{42}$	0.9665	0.9665	0.9688	0.9688	0.9688	0.9688

Table 4.29: Classification accuracy by k-nearest neighbor classifier based on **Distance-based probability**

Method	k=3		k=5		k=7	
K	N_c	$\min(5, 0.1 * N_c)$	N_c	$\min(5, 0.1 * N_c)$	N_c	$\min(5, 0.1 * N_c)$
$\mathbf{SwLDA}_{11}$	**0.9704**	0.9688	0.9700	0.9692	0.9696	0.9692
$\mathbf{SwLDA}_{21}$	0.9696	0.9700	**0.9704**	0.9692	**0.9704**	0.9688
$\mathbf{SwLDA}_{31}$	0.9696	0.9700	**0.9704**	0.9692	**0.9704**	0.9688
$\mathbf{SwLDA}_{41}$	0.9696	0.9700	**0.9704**	0.9692	**0.9704**	0.9688
$\mathbf{SwLDA}_{12}$	0.9665	0.9673	0.9692	0.9700	**0.9704**	0.9696
$\mathbf{SwLDA}_{22}$	0.9669	0.9681	0.9692	0.9696	0.9692	0.9696
$\mathbf{SwLDA}_{32}$	0.9669	0.9681	0.9692	0.9692	0.9692	0.9696
$\mathbf{SwLDA}_{42}$	0.9669	0.9673	0.9692	0.9692	0.9684	0.9692

Table 4.30: Classification accuracy by k-nearest neighbor classifier based on **Misclassification-based probability**

Method	k=3		k=5		k=7	
K	N_c	$\min(5, 0.1 * N_c)$	N_c	$\min(5, 0.1 * N_c)$	N_c	$\min(5, 0.1 * N_c)$
$\mathbf{SwLDA}_{11}$	0.9696	0.9696	0.9700	0.9696	**0.9704**	0.9696
$\mathbf{SwLDA}_{21}$	0.9696	0.9696	0.9700	0.9700	**0.9704**	**0.9704**
$\mathbf{SwLDA}_{31}$	0.9696	0.9696	0.9700	0.9700	**0.9704**	**0.9704**
$\mathbf{SwLDA}_{41}$	0.9696	0.9696	0.9700	0.9700	**0.9704**	**0.9704**
$\mathbf{SwLDA}_{12}$	0.9669	0.9673	0.9700	0.9696	**0.9704**	0.9700
$\mathbf{SwLDA}_{22}$	0.9669	0.9677	0.9700	0.9696	0.9700	0.9692
$\mathbf{SwLDA}_{32}$	0.9669	0.9677	0.9700	0.9696	**0.9704**	0.9692
$\mathbf{SwLDA}_{42}$	0.9669	0.9684	0.9688	0.9692	0.9688	0.9688

Table 4.31: Classification accuracy by traditional LDA and LDA variants

Method	Nearest centroid	k-nearest neighbor		
		k=3	k=5	k=7
Original LDA	0.9688	0.9684	0.9684	0.9688
Relevance WLDA	0.9681	0.9665	0.9673	0.9684
New WLDA	0.9692	0.9692	0.9692	0.9696

- **Balance dataset**: Table (4.32) shows the classification accuracy based on nearest centroid classifier. Table (4.33), Table (4.34) and Table (4.35) present results with three types of prior information matrix $\mathbf{V}_c$ combined with k-nearest neighbor classifier, respectively. Results of traditional LDA and other weighted LDA variants are illustrated in Table (4.36).

  The best performance is **0.9120** based on k-nearest neighbor classifier using $SwLDA_{21}$ or $SwLDA_{31}$ or $SwLDA_{41}$ with distance-based probability, fully connected graph and $k = 5$. k-nearest neighbor classifier works better than nearest centroid classifier under the same circumstances. Results related to the first definition of within-scatter matrix are superior to the results related to the second one under the same conditions.

Table 4.32: Classification accuracy by nearest centroid classifier

Method	Equal probability		Distance-based probability		Misclassification-based probability	
K	N_c	$\min(5, 0.1 * N_c)$	N_c	$\min(5, 0.1 * N_{ci})$	N_c	$\min(5, 0.1 * N_{ci})$
$\mathbf{SwLDA}_{11}$	0.7248	0.7248	0.7248	0.7248	0.7248	0.7248
$\mathbf{SwLDA}_{21}$	0.7248	0.7248	0.7248	0.7248	0.7248	0.7248
$\mathbf{SwLDA}_{31}$	0.7232	0.7232	0.7232	0.7248	0.7248	0.7232
$\mathbf{SwLDA}_{41}$	0.7232	0.7232	0.7232	0.7248	0.7248	0.7232
$\mathbf{SwLDA}_{12}$	0.7200	0.7200	0.7200	0.7200	0.7200	0.7200
$\mathbf{SwLDA}_{22}$	0.7200	0.7200	0.7200	0.7200	0.7184	0.7184
$\mathbf{SwLDA}_{32}$	0.7200	0.7200	0.7200	0.7200	0.7184	0.7184
$\mathbf{SwLDA}_{42}$	0.7088	0.7088	0.7104	0.7104	0.7104	0.7104

Table 4.33: Classification accuracy by k-nearest neighbor classifier based on **Equal probability**

Method	k=3		k=5		k=7	
K	N_c	$\min(5, 0.1 * N_c)$	N_c	$\min(5, 0.1 * N_c)$	N_c	$\min(5, 0.1 * N_c)$
$\mathbf{SwLDA}_{11}$	0.9072	0.9072	0.9104	0.9104	0.9040	0.9040
$\mathbf{SwLDA}_{21}$	0.9072	0.9072	0.9104	0.9104	0.9040	0.9040
$\mathbf{SwLDA}_{31}$	0.9072	0.9072	0.9104	0.9104	0.9040	0.9040
$\mathbf{SwLDA}_{41}$	0.9072	0.9072	0.9104	0.9104	0.9040	0.9040
$\mathbf{SwLDA}_{12}$	0.8912	0.8912	0.8976	0.8976	0.9040	0.9040
$\mathbf{SwLDA}_{22}$	0.8912	0.8896	0.8960	0.8960	0.9040	0.9040
$\mathbf{SwLDA}_{32}$	0.8896	0.8912	0.8960	0.8960	0.9024	0.9040
$\mathbf{SwLDA}_{42}$	0.8816	0.8832	0.9024	0.9024	0.8928	0.8928

Table 4.34: Classification accuracy by k-nearest neighbor classifier based on **Distance-based probability**

Method	k=3		k=5		k=7	
K	N_c	$\min(5, 0.1 * N_c)$	N_c	$\min(5, 0.1 * N_c)$	N_c	$\min(5, 0.1 * N_c)$
$\mathbf{SwLDA}_{11}$	0.9072	0.9072	0.9104	0.9104	0.9040	0.9040
$\mathbf{SwLDA}_{21}$	0.9056	0.9072	**0.9120**	0.9088	0.9040	0.9024
$\mathbf{SwLDA}_{31}$	0.9056	0.9088	**0.9120**	0.9088	0.9040	0.9024
$\mathbf{SwLDA}_{41}$	0.9056	0.9088	**0.9120**	0.9088	0.9040	0.9024
$\mathbf{SwLDA}_{12}$	0.8912	0.8912	0.8976	0.8976	0.9040	0.9040
$\mathbf{SwLDA}_{22}$	0.8912	0.8912	0.8992	0.8976	0.9024	0.9040
$\mathbf{SwLDA}_{32}$	0.8912	0.8912	0.8976	0.8976	0.9040	0.9024
$\mathbf{SwLDA}_{42}$	0.8832	0.8800	0.9008	0.9008	0.8928	0.8928

Table 4.35: Classification accuracy by k-nearest neighbor classifier based on **Misclassification-based probability**

Method	k=3		k=5		k=7	
K	N_c	$\min(5, 0.1 * N_c)$	N_c	$\min(5, 0.1 * N_c)$	N_c	$\min(5, 0.1 * N_c)$
$\mathbf{SwLDA}_{11}$	0.9072	0.9072	0.9104	0.9104	0.9040	0.9040
$\mathbf{SwLDA}_{21}$	0.9072	0.9056	0.9104	**0.9120**	0.9040	0.9040
$\mathbf{SwLDA}_{31}$	0.9072	0.9056	0.9072	0.9104	0.9040	0.9040
$\mathbf{SwLDA}_{41}$	0.9072	0.9056	0.9072	0.9104	0.9040	0.9040
$\mathbf{SwLDA}_{12}$	0.8912	0.8912	0.8960	0.8976	0.9024	0.9040
$\mathbf{SwLDA}_{22}$	0.8912	0.8896	0.8960	0.8976	0.9040	0.9040
$\mathbf{SwLDA}_{32}$	0.8896	0.8912	0.8960	0.8960	0.9040	0.9024
$\mathbf{SwLDA}_{42}$	0.8832	0.8832	0.9008	0.8992	0.8928	0.8928

Table 4.36: Classification accuracy by traditional LDA and LDA variants

Method	Nearest centroid	k-nearest neighbor		
		k=3	k=5	k=7
Original LDA	0.7184	0.8912	0.8976	0.9040
Relevance WLDA	0.7184	0.8944	0.8992	0.9040
New WLDA	0.7184	0.8960	0.9008	0.9024

- **Contraceptive dataset**: Table (4.37) shows the classification accuracy based on nearest centroid classifier. Table (4.38), Table (4.39) and Table (4.40) present results with three types of prior information matrix $\mathbf{V}_c$ combined with k-nearest neighbor classifier, respectively. Results of traditional LDA and other weighted LDA variants are illustrated in Table (4.41).

  $SwLDA_{22}$ combined with nearest centroid classifier achieves the best performance **0.4983**, when the prior information is distance-based probability and graph is fully connected. Nearest centroid classifier works better than k-nearest neighbor classifier under the same conditions, generally. Furthermore, Results related to the first definition of within-scatter matrix are superior to the results related to the second one, when using k-nearest neighbor classifier, under the same conditions.

Table 4.37: Classification accuracy by nearest centroid classifier

Method	Equal probability		Distance-based probability		Misclassification-based probability	
K	N_c	$\min(5, 0.1 * N_c)$	N_c	$\min(5, 0.1 * N_{ci})$	N_c	$\min(5, 0.1 * N_{ci})$
$\mathbf{SwLDA}_{11}$	0.4956	0.4956	0.4956	0.4956	0.4956	0.4956
$\mathbf{SwLDA}_{21}$	0.4956	0.4956	0.4956	0.4942	0.4942	0.4949
$\mathbf{SwLDA}_{31}$	0.4956	0.4956	0.4956	0.4969	0.4963	0.4956
$\mathbf{SwLDA}_{41}$	0.4956	0.4956	0.4956	0.4969	0.4963	0.4956
$\mathbf{SwLDA}_{12}$	0.4969	0.4969	0.4976	0.4969	0.4976	0.4976
$\mathbf{SwLDA}_{22}$	0.4969	0.4969	**0.4983**	0.4969	0.4969	0.4969
$\mathbf{SwLDA}_{32}$	0.4976	0.4976	0.4976	0.4976	0.4983	0.4976
$\mathbf{SwLDA}_{42}$	0.4868	0.4868	0.4868	0.4881	0.4875	0.4861

Table 4.38: Classification accuracy by k-nearest neighbor classifier based on **Equal probability**

Method	k=3		k=5		k=7	
K	N_c	$\min(5, 0.1 * N_c)$	N_c	$\min(5, 0.1 * N_c)$	N_c	$\min(5, 0.1 * N_c)$
$\mathbf{SwLDA}_{11}$	0.4644	0.4637	0.4759	0.4780	0.4759	0.4759
$\mathbf{SwLDA}_{21}$	0.4637	0.4631	0.4766	0.4759	0.4759	0.4759
$\mathbf{SwLDA}_{31}$	0.4685	0.4671	0.4780	0.4780	0.4780	0.4780
$\mathbf{SwLDA}_{41}$	0.4678	0.4671	0.4800	0.4786	0.4780	0.4780
$\mathbf{SwLDA}_{12}$	0.4298	0.4298	0.4814	0.4807	0.4820	0.4820
$\mathbf{SwLDA}_{22}$	0.4292	0.4298	0.4820	0.4814	0.4820	0.4820
$\mathbf{SwLDA}_{32}$	0.4278	0.4278	0.4793	0.4814	0.4827	0.4827
$\mathbf{SwLDA}_{42}$	0.4556	0.4542	0.4732	0.4746	0.4780	0.4780

Table 4.39: Classification accuracy by k-nearest neighbor classifier based on **Distance-based probability**

Method	k=3		k=5		k=7	
K	N_c	$\min(5, 0.1 * N_c)$	N_c	$\min(5, 0.1 * N_c)$	N_c	$\min(5, 0.1 * N_c)$
$\mathbf{SwLDA}_{11}$	0.4651	0.4617	0.4786	0.4759	0.4786	0.4827
$\mathbf{SwLDA}_{21}$	0.4664	0.4631	0.4807	0.4780	0.4847	0.4847
$\mathbf{SwLDA}_{31}$	0.4664	0.4637	0.4807	0.4814	0.4847	0.4814
$\mathbf{SwLDA}_{41}$	0.4658	0.4637	0.4820	0.4800	0.4847	0.4807
$\mathbf{SwLDA}_{12}$	0.4298	0.4312	0.4820	0.4820	0.4800	0.4820
$\mathbf{SwLDA}_{22}$	0.4353	0.4393	0.4793	0.4685	0.4827	0.4793
$\mathbf{SwLDA}_{32}$	0.4312	0.4366	0.4820	0.4692	0.4841	0.4827
$\mathbf{SwLDA}_{42}$	0.4522	0.4515	0.4719	0.4746	0.4725	0.4800

Table 4.40: Classification accuracy by k-nearest neighbor classifier based on **Misclassification-based probability**

Method	k=3		k=5		k=7	
K	N_c	$\min(5, 0.1 * N_c)$	N_c	$\min(5, 0.1 * N_c)$	N_c	$\min(5, 0.1 * N_c)$
$\mathbf{SwLDA}_{11}$	0.4644	0.4644	0.4780	0.4786	0.4759	0.4766
$\mathbf{SwLDA}_{21}$	0.4631	0.4631	0.4766	0.4766	0.4786	0.4773
$\mathbf{SwLDA}_{31}$	0.4651	0.4671	0.4800	0.4773	0.4807	0.4786
$\mathbf{SwLDA}_{41}$	0.4658	0.4671	0.4807	0.4780	0.4807	0.4786
$\mathbf{SwLDA}_{12}$	0.4292	0.4305	0.4814	0.4834	0.4820	0.4820
$\mathbf{SwLDA}_{22}$	0.4339	0.4298	0.4814	0.4834	0.4807	0.4814
$\mathbf{SwLDA}_{32}$	0.4346	0.4285	0.4786	0.4820	0.4820	0.4807
$\mathbf{SwLDA}_{42}$	0.4563	0.4569	0.4732	0.4732	0.4746	0.4739

Table 4.41: Classification accuracy by traditional LDA and LDA variants

Method	Nearest centroid	k-nearest neighbor		
		k=3	k=5	k=7
Original LDA	0.4963	0.4454	0.4739	0.4861
Relevance WLDA	0.4956	0.4529	0.4671	0.4847
New WLDA	0.4969	0.4380	0.4719	0.4814

- **Hayes-roth dataset**: Table (4.42) shows the classification accuracy based on nearest centroid classifier. Table (4.43), Table (4.44), and Table (4.45) present results with three types of prior information matrix $\mathbf{V}_c$ combined with k-nearest neighbor classifier, respectively. Results of traditional LDA and other weighted LDA variants are illustrated in Table (4.46).

  $SwLDA_{21}$ or $SwLDA_{31}$ or $SwLDA_{41}$ combined with k-nearest neighbor classifier achieves the best result **0.6963**, when priori information is misclassification-based probability, k is set to 3 and graph is partly connected. k-nearest neighbor classifier works better than nearest centroid classifier generally. Results related to the first definition of within-scatter matrix are superior to the results related to the second one in most cases, when using k-nearest neighbor classifier.

Table 4.42: Classification accuracy by nearest centroid classifier

Method	Equal probability		Distance-based probability		Misclassification-based probability	
K	N_c	$\min(5, 0.1 * N_c)$	N_c	$\min(5, 0.1 * N_{ci})$	N_c	$\min(5, 0.1 * N_{ci})$
$\mathbf{SwLDA}_{11}$	0.5556	0.5556	0.5556	0.5556	0.5556	0.5481
$\mathbf{SwLDA}_{21}$	0.5556	0.5556	0.5556	0.5704	0.5556	0.5407
$\mathbf{SwLDA}_{31}$	0.5556	0.5556	0.5556	0.5704	0.5556	0.5481
$\mathbf{SwLDA}_{41}$	0.5556	0.5556	0.5556	0.5704	0.5556	0.5481
$\mathbf{SwLDA}_{12}$	0.5481	0.5481	0.5481	0.5481	0.5481	0.5481
$\mathbf{SwLDA}_{22}$	0.5481	0.5481	0.5481	0.5481	0.5481	0.5481
$\mathbf{SwLDA}_{32}$	0.5481	0.5481	0.5481	0.5481	0.5481	0.5481
$\mathbf{SwLDA}_{42}$	0.5556	0.5556	0.5556	0.5556	0.5556	0.5556

Table 4.43: Classification accuracy by k-nearest neighbor classifier based on **Equal probability**

Method	k=3		k=5		k=7	
K	N_c	$\min(5, 0.1 * N_c)$	N_c	$\min(5, 0.1 * N_c)$	N_c	$\min(5, 0.1 * N_c)$
$\mathbf{SwLDA}_{11}$	0.6815	0.6815	0.6889	0.6889	0.6741	0.6741
$\mathbf{SwLDA}_{21}$	0.6815	0.6815	0.6889	0.6889	0.6741	0.6741
$\mathbf{SwLDA}_{31}$	0.6815	0.6815	0.6889	0.6889	0.6741	0.6741
$\mathbf{SwLDA}_{41}$	0.6815	0.6815	0.6889	0.6889	0.6741	0.6741
$\mathbf{SwLDA}_{12}$	0.6741	0.6741	0.6889	0.6889	0.6741	0.6741
$\mathbf{SwLDA}_{22}$	0.6741	0.6741	0.6889	0.6889	0.6741	0.6741
$\mathbf{SwLDA}_{32}$	0.6741	0.6741	0.6889	0.6889	0.6741	0.6741
$\mathbf{SwLDA}_{42}$	0.6667	0.6667	0.6593	0.6593	0.6667	0.6667

Table 4.44: Classification accuracy by k-nearest neighbor classifier based on **Distance-based probability**

Method	k=3		k=5		k=7	
K	N_c	$\min(5, 0.1 * N_c)$	N_c	$\min(5, 0.1 * N_c)$	N_c	$\min(5, 0.1 * N_c)$
$\mathbf{SwLDA}_{11}$	0.6815	0.6889	0.6889	0.6889	0.6741	0.6741
$\mathbf{SwLDA}_{21}$	0.6889	0.6815	0.6889	0.6741	0.6741	0.6741
$\mathbf{SwLDA}_{31}$	0.6889	0.6815	0.6889	0.6667	0.6741	0.6741
$\mathbf{SwLDA}_{41}$	0.6889	0.6815	0.6889	0.6667	0.6741	0.6741
$\mathbf{SwLDA}_{12}$	0.6741	0.6741	0.6889	0.6889	0.6741	0.6741
$\mathbf{SwLDA}_{22}$	0.6741	0.6741	0.6889	0.6889	0.6741	0.6741
$\mathbf{SwLDA}_{32}$	0.6741	0.6741	0.6889	0.6889	0.6741	0.6741
$\mathbf{SwLDA}_{42}$	0.6741	0.6741	0.6519	0.6519	0.6667	0.6667

Table 4.45: Classification accuracy by k-nearest neighbor classifier based on **Misclassification-based probability**

Method	k=3		k=5		k=7	
K	N_c	$\min(5, 0.1 * N_c)$	N_c	$\min(5, 0.1 * N_c)$	N_c	$\min(5, 0.1 * N_c)$
$\mathbf{SwLDA}_{11}$	0.6815	0.6889	0.6889	0.6889	0.6741	0.6741
$\mathbf{SwLDA}_{21}$	0.6815	**0.6963**	0.6889	0.6741	0.6741	0.6593
$\mathbf{SwLDA}_{31}$	0.6815	**0.6963**	0.6889	0.6815	0.6741	0.6741
$\mathbf{SwLDA}_{41}$	0.6815	**0.6963**	0.6889	0.6815	0.6741	0.6741
$\mathbf{SwLDA}_{12}$	0.6741	0.6741	0.6889	0.6889	0.6667	0.6667
$\mathbf{SwLDA}_{22}$	0.6741	0.6741	0.6889	0.6889	0.6741	0.6741
$\mathbf{SwLDA}_{32}$	0.6741	0.6741	0.6889	0.6889	0.6667	0.6667
$\mathbf{SwLDA}_{42}$	0.6593	0.6593	0.6519	0.6519	0.6667	0.6667

Table 4.46: Classification accuracy by traditional LDA and LDA variants

Method	Nearest centroid	k-nearest neighbor		
		k=3	k=5	k=7
Original LDA	0.5481	0.6741	0.6889	0.6741
Relevance WLDA	0.5556	0.6741	0.6815	0.6667
New WLDA	0.5481	0.6741	0.6889	0.6741

4.4 Discussion

In this thesis, we evaluated the performance of the proposed $SwLDA$ methods on six facial datasets and three imbalanced datasets, and then compared the results with

those of traditional LDA and two weighted LDA approaches mentioned in section 2.6. According to the experimental results, $SwLDA$ methods present the following characteristics.

Firstly, $SwLDA$ methods can achieve better performance with nearest centroid classifier than k-nearest neighbor classifier on BU, KANEDE, YALE, AR and Contraceptive dataset in most cases, with the same type of prior information and graph connection.

Secondly, the classification accuracy is not always higher, when using a fully connected graph than k-NN graph, as shown in Table (4.19) and Table (4.20). Moreover, graph connection does not affect the classification accuracy, when prior information is equal probability on facial image datasets. Not all $SwLDA$ methods with fully connected graph can achieve a better performance, sometimes $SwLDA$ methods with k-NN graphs can achieve a better performance, as in Table (4.9) and Table (4.10) by $SwLDA_{31}$, $SwLDA_{41}$, $SwLDA_{32}$, and $SwLDA_{42}$.

Thirdly, when using the between-class scatter matrix $\mathbf{S}_b^{(4)}$, $SwLDA$ could work better than when using other definitions of between-class scatter matrix in some cases on facial image datasets. This is because the forth definition of between-class scatter reveals the structure of each class. However, this kind of definition may result in over-fitting, leading to a worse result, as in ORL dataset.

Last but not least, k-nearest neighbor classifier can achieve a higher accuracy either, as in AR or ORL datasets. Meanwhile, the worst result appears on ORL dataset, when $SwLDA_{42}$ is combined with k-nearest neighbor classifier. The results with k-nearest neighbor classifier have high improvements comparing to with nearest centroid classifier on certain imbalanced datasets under the same conditions, e.g. $SwLDA_{21}$ with distance-based probability and fully connected graph on Balance dataset, the biggest improvement is 18.72%. Except on Contraceptive dataset, the second definition of within-class scatter matrix is inferior to the first one on Balance and Hayes-roth datasets under the same conditions.

5. CONCLUSIONS

In this thesis the combination of weighted LDA and saliency estimation is proposed to derive a new $SwLDA$ variant and solve the sub-optimal problems existing in LDA variant. Weighted LDA approaches can improve on the sub-optimal problem caused by heterogeneous Gaussian distribution dataset to a certain extent. However, weighted LDA methods can not reflect the importance of samples inside each class and neglect the influence of outlier samples. Hence, sub-optimal results still exist.

In order to further improve the performance of LDA variants, a class-specific saliency estimation process is followed and novel $SwLDA$ approaches are proposed to determine the contribution of each sample in the optimization problem solved for discriminant subspace learning. This saliency estimation process has a connection with one-class classification, when estimating saliency score of each class.

The new approaches were tested on six public facial image datasets and three imbalanced datasets for evaluation and comparison with related LDA methods. The new definitions can reveal connections between each sample in every class to a certain extent, and further improve the classification results on these facial image and class-imbalanced classification problems. The improvement is particularly large on BU dataset, which is 11.14% comparing to the result of traditional LDA. Moreover, experimental results sufficiently demonstrate that the highest classification accuracy is always with one of the proposed variants over these six facial image and the three imbalanced datasets.

Concerning $SwLDA$ methods, further studies will concentrate on solving particularly imbalanced datasets, and more effectively exploiting prior information of imbalanced dataset for saliency score and class representation. Moreover, regularization can be considered to reduce over-fitting, when defining scatter matrices.

Nevertheless, kernel strategy can be considered to combine with $SwLDA$ for nonlinear discriminant analysis as in [53; 54; 55]. Therefore, scatter matrices and Fisher's discriminant criterion can be defined in a non-linear space.

REFERENCES

[1] E. Greenstein and D. Penner, "Japanese-to-English Machine Translation Using Recurrent Neural Networks," *Stanford Deep Learning for NLP Course*, pp. 1–7, 2015.

[2] E. Garcia Amaro, M. Nuno-Maganda, and M. Morales-Sandoval, "Evaluation of Machine Learning Techniques for Face Detection and Recognition," *CONIELECOMP 2012, 22nd International Conference on Electrical Communications and Computers*, pp. 213–218, 2012.

[3] A. R. Webb and K. D. Copsey, *Statistical Pattern Recognition*. Wiley, 2011.

[4] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. Wiley, 2001.

[5] H. S. Dadi and G. K. Mohan Pillutla, "Improved Face Recognition Rate Using HOG Features and SVM Classifier," *IOSR Journal of Electronics and Communication Engineering*, vol. 11, no. 04, pp. 34–44, 2016.

[6] A. Liaw and M. Wiener, "Classification and Regression by randomForest," *R News*, vol. 2, no. 3, pp. 18–22, 2002.

[7] Y.-l. Cai, D. Ji, and D.-f. Cai, "A KNN Research Paper Classification Method Based on Shared Nearest Neighbor," *Proceedings of the 8th NTCIR Workshop Meeting on Evaluation of Information Access Technologies: Information Retrieval, Question Answering and Cross-Lingual Information Access*, pp. 336–340, 2010.

[8] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer-Verlag New York, 2013.

[9] H. Yu and J. Yang, "A Direct LDA Algorithm for High-Dimensional Data – with Application to Face Recognition," *Pattern Recognition*, vol. 34, pp. 2067–2070, 2001.

[10] A. Iosifidis, A. Tefas, N. Nikolaidis, and I. Pitas, "Multi-view human movement recognition based on fuzzy distances and linear discriminant analysis," *Computer Vision and Image Understanding*, vol. 116, no. 3, pp. 347–360, 2012.

[11] A. Iosifidis, A. Tefas, and I. Pitas, "Regularized extreme learning machine for multi-view semi-supervised action recognition," *Neurocomputing*, vol. 145, pp. 250–262, 2014.

[12] ——, "Activity-based person identification using fuzzy representation and discriminant learning," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 2, pp. 530–542, 2012.

[13] B. Yu, L. Jin, and P. Chen, "A New LDA-based Method for Face Recognition," *Proc. 16th international conference in Pattern recognition*, vol. 1, pp. 168–171, 2002.

[14] E. K. Tang, P. N. Suganthan, and X. Yao, "Generalized LDA Using Relevance Weighting and Evolution Strategy," *Congress on Evolutionary Computation*, vol. 2, no. 1, pp. 2230 – 2234, 2004.

[15] A. Iosifidis, A. Tefas, and I. Pitas, "On the Optimal Class Representation in Linear Discriminant Analysis," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 9, pp. 1491–1497, 2013.

[16] ——, "Kernel Reference Discriminant Analysis," *Pattern Recognition Letters*, vol. 49, pp. 85–91, 2014.

[17] E. K. Tang, P. N. Suganthan, X. Yao, and A. K. Qin, "Linear Dimensionality Reduction Using Relevance Weighted LDA," *Pattern Recognition*, vol. 38, no. 4, pp. 485–493, 2005.

[18] D. Jarchi and R. Boostani, "A New Weighted LDA Method in Comparison to Some Versions of LDA," *Proceedings of Word Academy of Science, Engineering and Technology*, vol. 18, no. 12, pp. 233–238, 2008.

[19] M. Loog, R. Duin, and R. Haeb-Umbach, "Multiclass Linear Dimension Reduction by Weighted Pairwise Fisher Criteria," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 7, pp. 762–766, 2001.

[20] Z. Li, D. Lin, and X. Tang, "Nonparametric discriminant analysis for face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 4, pp. 755–761, 2009.

[21] C. Aytekin, S. Kiranyaz, M. Gabbouj, and A. Iosifidis, "Recent Advances in Salient Object Detection," *Futura-BigData*, vol. 35, no. 2, pp. 80–92, 2016.

[22] C. Aytekin, A. Iosifidis, and M. Gabbouj, "Probabilistic Saliency Estimation," *arXiv:1609.03868*, pp. 1–28, 2016.

[23] R. A. Fisher, "The Use of Multiple Measurements in Taxonomic Problems," *Annals of Eugenics*, vol. 7, no. 2, pp. 179–188, 1936.

[24] C. Rao, "The Utilization of Multiple Measurements in Problems of Biological Classification," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 10, no. 2, pp. 159–203, 1948.

[25] Y. Jia, F. Nie, and C. Zhang, "Trace ratio problem revisited," *IEEE Transactions on Neural Networks*, vol. 20, no. 4, pp. 729–735, 2009.

[26] D. Huang, Y. Quan, M. He, and B. Zhou, "Comparison of linear discriminant analysis methods for the classification of cancer based on gene expression data." *Journal of experimental & clinical cancer research*, vol. 28:149, 2009.

[27] Z. Jin, J. Y. Yang, Z. S. Hu, and Z. Lou, "Face recognition based on the uncorrelated discriminant transformation," *Pattern Recognition*, vol. 34, no. 7, pp. 1405–1416, 2001.

[28] S. Petridis and S. J. Perantonis, "On the Relation between Discriminant Analysis and Mutual Information for Supervised Linear Feature Extraction," *Pattern Recognition*, vol. 37, no. 5, pp. 857–874, 2004.

[29] M. Loog, *Approximate Pairwise Accuracy Criteria for Multiclass Linear Dimension Reduction: Generalisations of the Fisher Criterion.* Delft Univ. Press, 1999.

[30] T. U. Delft and R. Magnificus, "One-class classification," Ph.D. dissertation, de natuurkunde, geboren te Ede, 2001.

[31] N. Japkowicz, "Concept-Learning in the Absence of Counter-Examples : an Autoassociation-Based Approach To Classification," Ph.D. dissertation, The State University of New Jersey, 1999.

[32] G. Ritter and M. Gallegos, "Outliers in statistical pattern recognition and an application to automatic chromosome classification," *Pattern Recognition Letters*, vol. 18, no. 6, pp. 525–539, 1997.

[33] C. Bishop, "Novelty detection and neural network validation," *IEE Proceedings - Vision, Image, and Signal Processing*, vol. 141, no. 4, p. 217, 1994.

[34] Q. Yan, L. Xu, J. Shi, and J. Jia, "Hierarchical Saliency Detection," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1155–1162, 2013.

[35] S. Hong, T. You, S. Kwak, and B. Han, "Online Tracking by Learning Discriminative Saliency Map with Convolutional Neural Network," *Proceedings of the 32nd International Conference on Machine Learning, PMLR*, vol. 37, pp. 597–606, 2015.

[36] Y. Wang, L. Pan, D. Wang, and Y. Kang, "Detection of Harbours From High Resolution Remote Sensing Imagery Via Saliency Analysis and Feature Learning," *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, pp. 573–578, 2016.

[37] X. Y. Stella and D. A. Lisin, "Image Compression based on Visual Saliency at Individual Scales," *Advances in Visual Computing*, vol. 5875, pp. 157–166, 2009.

[38] S. Barua, K. Mitra, and A. Veeraraghavan, "Saliency guided wavelet compression for low-bitrate image and video coding," *IEEE Global Conference on Signal and Information Processing, GlobalSIP*, pp. 1185–1189, 2015.

[39] T. Kadir and J. M. Brady, "Scale, Saliency and Image Description," *International Journal of Computer Vision*, vol. 45, no. 2, pp. 83–105, 2001.

[40] M. Cheng, G. Zhang, N. J. Mitra, X. Huang, and S. Hu, "Global Contrast based Salient Region Detection," *2011 IEEE Conference on Computer Vision and Pattern Recognition*, vol. 37, no. 3, pp. 409–416, 2011.

[41] L. Itti, C. Koch, and E. Niebur, "A Model of Saliency-Based Visual Attention for Rapid Scene Analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 11, pp. 1254–1259, 1998.

[42] J. Harel, C. Koch, and P. Perona, "Graph-Based Visual Saliency," *NIPS 2006*, pp. 545–552, 2006.

[43] X. Li, H. Lu, L. Zhang, X. Ruan, and M. H. Yang, "Saliency Detection via Dense and Sparse Reconstruction," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2976–2983, 2013.

[44] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk, "Frequency-tuned Salient Region Detection," *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1597–1604, 2009.

[45] R. Zhao, H. Li, and X. Wang, "Saliency Detection by Multi-Context Deep Learning," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1265–1274, 2015.

[46] A. Borji and L. Itti, "Exploiting Local and Global Patch Rarities for Saliency Detection," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 478–485, 2012.

[47] L. Yin, X. Wei, Y. Sun, J. Wang, and M. J. Rosato, "A 3D Facial Expression Database For Facial Behavior Research," *7th International Conference on Automatic Face and Gesture Recognition*, vol. 10, no. 12, pp. 211–216, 2006.

[48] M. Lyons, S. Akamatsu, M. Kamachi, and J. Gyoba, "Coding Facial Expressions with Gabor Wavelets," *Proceedings - 3rd IEEE International Conference on Automatic Face and Gesture Recognition, FG 1998*, pp. 200–205, 1998.

[49] T. Kanade and J. Cohn, "Comprehensive Database for Facial Expression Analysis," *Proceedings of the 4th IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 46–53, 2000.

[50] F. Samaria and A. Harter, "Parameterisation of a Stochastic Model for Human Face Identification," *Proceedings of 1994 IEEE Workshop on Applications of Computer Vision*, pp. 138–142, 1994.

[51] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman, "From Few to Many: Illumination Cone Models for Face Recognition Under Variable Lighting and Pose," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 643–660, 2001.

[52] A. Martinez and R. Benavente, "The AR Face Database," Tech. Rep., 1998.

[53] C. Bouveyron, M. Fauvel, and S. Girard, "Kernel discriminant analysis and clustering with parsimonious Gaussian process models," *Statistics and Computing*, vol. 25, no. 6, pp. 1143–1162, 2015.

[54] T. Xiong, J. Ye, Q. Li, R. Janardan, and V. Cherkassky, "Efficient Kernel Discriminant Analysis via QR Decomposition," *Advances in Neural Information Processing Systems 17*, pp. 1529–1536, 2005.

[55] X. Z. Liu, P. C. Yuen, G. C. Feng, and W. S. Chen, "Learning Kernel in Kernel-based LDA for Face Recognition Under Illumination Variations," *IEEE Signal Processing Letters*, vol. 16, no. 12, pp. 1019–1022, 2009.