Separability-Oriented Subclass Discriminant Analysis

Huan Wan, Hui Wang, Gongde Guo, Xin Wei

Abstract—Linear discriminant analysis (LDA) is a classical method for discriminative dimensionality reduction. The original LDA may degrade in its performance for non-Gaussian data, and may be unable to extract sufficient features to satisfactorily explain the data when the number of classes is small. Two prominent extensions to address these problems are *subclass discriminant analysis* (SDA) and *mixture subclass discriminant analysis* (MSDA). They divide every class into subclasses and re-define the within-class and between-class scatter matrices on the basis of subclass. In this paper we study the issue of how to obtain subclasses more effectively in order to achieve higher class separation. We observe that there is significant overlap between models of the subclasses, which we hypothesise is undesirable. In order to reduce their overlap we propose an extension of LDA, *separability oriented subclass discriminant analysis* (SSDA), which employs hierarchical clustering to divide a class into subclasses using a separability oriented criterion, before applying LDA optimisation using re-defined scatter matrices. Extensive experiments have shown that SSDA has better performance than LDA, SDA and MSDA in most cases. Additional experiments have further shown that SSDA can project data into LDA space that has higher class separation than LDA, SDA and MSDA in most cases.

Index Terms—Dimensionality reduction, feature extraction, linear discriminant analysis, subclass discriminant analysis, classification

1 Introduction

Dimensionality reduction is a key process in machine learning and statistics to reduce the number of variables under consideration, by way of feature selection or feature extraction. Data analysis such as regression or classification can usually be done in the reduced space more accurately than in the original space when the same analysis model is used. This is usually the case for high dimensional data analysis tasks such as image, video, and spectral data analysis.

Dimensionality reduction transforms data from a high-dimensional space into a lower-dimensional space. The transformation may be *linear* as in principal component analysis (PCA), or *nonlinear* as in kernel PCA and manifold learning, or *supervised* (or *discriminant*) as in linear discriminant analysis (LDA). LDA is a classical approach to discriminant dimensionality reduction, dating back to Fisher [1]. It has been widely used in many fields of pattern recognition such as face recognition and verification [2], [3], image retrieval [4], and document recognition [5].

There are three limitations with the original LDA. The mathematics of LDA is derived on the assumption that the instances of all classes are generated from Gaussian distributions of same covariance but different means. If the distributions are significantly non-Gaussian, the LDA projections may not preserve complex structure in the data needed for classification. The assumption has significantly restricted the application of LDA, as in real life many data distributions are not Gaussian [6].

The mathematics of LDA also implies that LDA produces at most $C-1$ feature projections, where C is the number of classes, because the number of projections is constrained by the rank of the between-class scatter matrix which is at most $C-1$. As a result the original data space can be transformed to a new space of at most $C-1$ dimensions. When there are many classes this is not a problem; however, when there are few classes this could be a serious problem. For example, in a binary classification problem there are only two classes. When LDA is applied there will be only one feature projection to represent the data. One feature may be insufficient to describe the class boundary in many cases especially when the class boundary is complex. This is the so-called *over-reducing problem* [7]. The third limitation is that LDA will fail when the discriminatory information is not in the mean but rather in the variance of the data .

To overcome these limitations, several variants of the original LDA have been proposed in recent years including *non-parametric discriminant analysis* (NDA) [8], [9], *subclass discriminant analysis* (SDA) [10] and *mixture subclass discriminant analysis* (MSDA) [11], [12].

Fukunaga [8] argued that LDA is parametric discriminant analysis since it uses the parametric form of the scatter matrix based on the Gaussian distribution assumption [9]. As a result LDA suffers performance degradation in cases of non-Gaussian distribution. Fukunaga then re-defined the between-class scatter matrix[1] to overcome this non-Gaussian problem and called the resulting LDA as *non-*

- *H. Wan, G. Guo and X. Wei are with Key Lab of Network Security and Cryptology, School of Mathematics and Computer Science, Fujian Normal University, P.R. China*
- *H. Wang is with School of Computing and Mathematics, Ulster University, UK; also with Fujian Normal University as a visiting professor.*

[1] Fukunaga defined the new between-class scatter matrix as: $S_b^N = \sum_{j=1}^{N_1} w(1,j)(x_{1j} - \mu_2(x_{1j}))(x_{1j} - \mu_2(x_{1j}))^T + \sum_{j=1}^{N_2} w(2,j)(x_{2j} - \mu_1(x_{2j}))(x_{2j} - \mu_1(x_{2j}))^T$, where x_{ij} denotes the jth vector of class $i(i = 1,2)$ and $\mu_i(x_{ij})$ is the local KNN mean, defined by $\mu_i(x_{ij}) = \frac{1}{k}\sum_{p=1}^{k} NN_p(x_{ij}, l)$ where $NN_p(x_{ij}, l)$ is the pth nearest neighbor from class l to the vector $x_{ij}(i \neq l)$ and $w(i,l)$ is the value of the weighting function in class i.

parametric discriminant analysis (NDA). In NDA's between-class scatter matrix, weighting is introduced to consider the boundary information in the training set. In this way, for any distributions, we can separate classes by maximizing the distance between data instances in one class (especially those instances that stand at the boundary) and the mean of p nearest neighbours from another class. In addition, NDA uses data instances to construct the between-class scatter matrix instead of merely the class centres, therefore it also solves the over-reducing problem (i.e., $C-1$ limitation). However, NDA can only deal with the two-class case, so Li et al. [9] extended NDA to address the multiclass case and proposed a series of variants of NDA, such as *nonparametric subspace analysis* (NSA) and *nonparametric feature subspace* (NFA), and applied these variants of NDA for face recognition.

Zhu and Martinez [10] argued that classes are usually multimodal so distribution within classes should be considered when constructing between-class scatter matrix. They proposed a variant of LDA, called *subclass discriminative analysis* (SDA), to address this issue. Based on a nearest neighbor based clustering algorithm and stability criterion they proposed, SDA divides each class into same number of subclasses and computes centres of those subclasses or *subcentres* of the class. Then SDA uses the differences between the subcentres of one class and the subcentres of the other classes to construct a new between-class scatter matrix and maximizes the new between-class scatter matrix through the LDA optimisation machinery. Since SDA uses subcentres rather than centres of every class to compute between-class scatter matrix, it does not have the $C-1$ limitation.

Mixture subclass discriminant analysis (MSDA) [11] uses the same criterion as SDA to obtain optimal number of subclasses for each class but it only divides those classes that do not have Gaussian distribution based on nongaussianity criterion proposed by [11]. MSDA uses the same between-class matrix as SDA so it does not have the $C-1$ limitation either.

Although both SDA and MSDA have overcome some of the LDA limitations, they have not answered one important question. Most if not all LDA variants, including SDA and MSDA, perform dimensionality reduction by maximising the same (Fisher) objective which is the between-class scatter matrix normalised by the within-class scatter matrix, with different definitions for the between-class and within-class scatter matrices. What is the true objective of discriminative dimensionality reduction? Fig. 6 shows the subclass distribution of the Iris data obtained by SDA, MSDA and SSDA (to be presented in this paper), where each circle corresponds to one subclass. It is clear that SDA and MSDA have different number of circles thus resulting in very different LDA dimensions[2], suggesting that SDA and MSDA have different goals in dimensionality reduction. At the same time they have one thing in common – the circles have significant overlap. A research question naturally arises: could reducing the overlap of these circles be a good objective of dimensionality reduction?

More recently a new LDA variant, orLDA, is proposed to address the over-reducing problem associated with LDA [7]. Unlike LDA, which measures the between-class separation by subtracting the mean of every class by the mean of the whole data instances, orLDA does so by subtracting every instance in one class by the mean of another class. In this way, the original data space can be reduced to one with at most $min(d, n-2)$ dimensions[3] where d is the number of dimensions in the original data space and n is the number of instances.

This has indeed overcome the over-reducing problem. However, when the number of instances and the number of dimensions are both large, the reduced (LDA) space may still have many dimensions, thus dimensionalty reduction being possibly insufficient. This may then affect the overall performance. Furthermore, in its current form, orLDA can only be used for binary classification problems due to its way of computing the between-class scatter matrix.

Inspired by SDA and MSDA, also motivated by the desire to extend orLDA to work on multiclass problems, we present in this paper *separability-oriented subclass discriminant analysis* (SSDA), a variant of LDA as a result of our effort to answer the above overlapping question. SSDA is aimed to reduce the overlap between models of the subclasses within each class during the LDA dimensionality reduction process. This is achieved by finding the *optimal*[4] subclasses for each class and maximising Fisher's separation objective function using re-defined within-class and between-class scatter matrices – instead of using every instance of one class to subtract the mean of another class (as in orLDA), we use the means of the subclasses in one class to subtract the mean of all data instances. In this way, SSDA is expected to find subclasses for each class that are 'innate' to the data, and are not much overlapping in their models. Furthermore the number of dimensions in the reduced LDA space is not restricted by the number of classes or the number of data

[2] The number of LDA dimensions is closely related to the number of subclass (i.e. circles). The number of circles denotes the number of subclasses. When the between-class scatter matrix and within-class scatter matrix are constructed by the centres of subclasses rather the centres of classes, the ranks of between-class scatter matrix and within-class scatter matrix are strongly correlated with the number of subclasses (circles). The number of reduced (LDA) dimensions is given by the rank of $S_w^{-1}S_b$, where $rank(S_w^{-1}S_b) \leqq min(rank(S_w^{-1}), rank(S_b))$. Therefore, the number of dimensions in the reduced (LDA) space correlates strongly with the number of circles. If we want to quantify the relation precisely, we must know the exact definition of S_b and S_w. Therefore there is no general formula for their relation.

[3] According to linear algebra and the formula that we compute the between-class scatter matrix S_b and within-class scatter matrix S_w, we can obtain $rank(S_b) = min(d, N_1 - 1 + N_2 - 1)$ and $rank(S_w) = min(d, N_1 - 1 + N_2 - 1)$, where d is the number of dimensions, N_i ($i = 1, 2$) is the number of ith class. Therefore $rank(S_b) = min(d, n-2)$, $rank(S_w) = min(d, n-2)$, where n is the number of instances. Because the number of dimensions in the reduced space of orLDA is equal to the $rank(S_w^{-1}S_b)$ and $rank(S_w^{-1}S_b) \leqq min(rank(S_w^{-1}), rank(S_b))$, the original data space can be reduced to one with at most $min(d, n-2)$.

[4] Throughout this paper, when we say 'optimal' we mean a choice that gives the best prediction performance.

instances. Extensive experiments show SSDA indeed has superior performance.

The rest of the paper is organised as follows. Section 2 presents related work including the classical linear discriminant analysis, subclass discriminant analysis, mixture subclass discriminant analysis, and LDA for over-reducing problem. Section 3 presents our separability-oriented subclass discriminant analysis. Experimental results are presented in Section 4. Section 5 presents further evaluation results about separability. Finally, Section 6 concludes the paper.

2 Related Work

To provide the context and to introduce the necessary technical notations we present an overview of related work, including the original LDA and its recent extensions SDA, MSDA and orLDA.

2.1 Linear Discriminant Analysis

Linear discriminant analysis (LDA) is a classical method for discriminant analysis that is used in statistics, pattern recognition and machine learning to find a linear combination of features that separates two or more classes of objects. The resulting combination may be used as a linear classifier, or more commonly, for dimensionality reduction before later classification [13]. It has been successfully used in many data centric applications.

LDA uses a between-class scatter matrix S_b to measure class separability, and uses within-class scatter matrix S_w to measure class compactness. The goal of LDA is to find a projective matrix W that projects data from one *data space* to a new one, *LDA space* that is spanned by *LDA features* (or *LDA dimensions*), such that a measure of the between-class scatter matrix S_b in the new space is maximsed and simultaneously the same measure of the within-class scatter matrix S_w in the new space is minimised. S_b and S_w are defined respectively as:

$$S_b = \frac{1}{N}\sum_{i=1}^{C} N_i(\mu_i - \mu)(\mu_i - \mu)^T \quad (1)$$

$$S_w = \frac{1}{N}\sum_{i=1}^{C}\sum_{j=1}^{N_i}(x_{ij} - \mu_i)(x_{ij} - \mu_i)^T \quad (2)$$

where N is the number of instances, N_i is the number of instances in class i, C is the number of classes, μ_i is the mean of class i, μ is global mean of all instances, and x_{ij} denotes the jth instance in class i.

LDA is an optimisation process. Its optimisation objective, the *Fisher objective*, is defined as follows

$$J^{LDA}(W) = \frac{tr(W^T S_b W)}{tr(W^T S_w W)} \quad (3)$$

where W is a projective matrix that projects data from the data space to the LDA space. The *LDA optimisation* is to find one projective matrix, W^*, that maximises the Fisher objective; that is

$$W^* = \arg\max_W J^{LDA}(W) \quad (4)$$

It turns out the sought-after projective matrix W^* is composed of the eigenvectors corresponding to the largest eigenvalues of $S_w^{-1}S_b$, based on the assumptions of normally distributed classes and equal class covariances[5].

Note that S_b is the sum of C matrices of rank ≤ 1 and the mean vectors are constrained by $\frac{1}{C}\sum_{j=1}^{C}\mu_i = \mu$, so the rank of S_b is at most $C-1$. Therefore the variability between features will be contained in the subspace spanned by the eigenvectors corresponding to the $C-1$ largest eigenvalues. In other words the dimensionality of the (new reduced) LDA space is at most $C-1$. If $C=2$, the LDA space will have only one dimension thus the *over-reducing problem* – insufficient number of features are extracted for describing the class boundaries, so there is loss of between-class information.

The terms LDA and Fisher's linear discriminant (FLD) are often used interchangeably, although Fisher's original article [1] actually describes a slightly different discriminant, which does not make some of the assumptions of LDA such as normally distributed classes or equal class covariances.

Various variants have been proposed to extend LDA, which fall under two main categories – incremental learning LDA and batch learning LDA. Incremental learning LDA methods focus on processing data streams, which update *LDA features* based on new 'burst' of data. Many of them update LDA features through updating between-class and within-class scatter matrices [14], [15], [16], [17].

Batch learning LDA methods require that all instances are available and construct LDA features with all data instances. Most batch learning LDA methods also address the *small sample size* (SSS) or *singularity* problem, which is a well known problem with the original LDA – when the number of features is much larger than the number of instances we cannot use $S_w^{-1}S_b$ to obtain the projective matrix W. Solutions include using LDA after PCA [18], using random matrix multiplication with scatter matrices to extract the most discriminant information [19], and using regularization [20], [21], [22], [23]. The regularization based methods are an important approach to solving the SSS problem. Its key idea is to find a regularization parameter α and add α to the diagonal elements of the within-class scatter matrix, which will guarantee that the new within-class scatter matrix is positive definite and non-singular. Other batch learning LDA methods include tensor-based LDA [24], heteroscedastic LDA [25], and sparse discriminant analysis [26].

2.2 Subclass Discriminant Analysis

Subclass discriminant analysis (SDA) [10] is a variant of LDA that aims to separate classes at a subclass level rather

[5] To show how to obtain W^*, we let $f = W^T S_b W$ and $g = W^T S_w W - \alpha = 0$, where $\alpha > 0$ is any constant. The LDA optimisation is thus equivalent to finding a projective matrix W to maximize f under the g constraint. For this we define $L = f - \lambda g$, where $\lambda \neq 0$ is Lagrange's multiplier. By setting the derivative of L with respect to W to zero, we get

$$\frac{\partial L}{\partial W} = 2S_bW - 2\lambda S_wW = 0 \Longrightarrow S_bW = \lambda S_wW \quad (5)$$

If S_w is nonsingular, we obtain $S_w^{-1}S_bW = \lambda W$. Therefore, the columns of projective matrix W^* are the eigenvectors corresponding to the largest eigenvalues of $S_w^{-1}S_b$.

than at a class level, based on the observation that the data distribution in a class may be multimodal (i.e., forming clusters). This is achieved by dividing each class into several subclasses and then maximising the redefined Fisher objective function, where the original between-class scatter matrix S_b is replaced by a new ***between-subclass scatter matrix*** S_{bsb}:

$$S_b^{SDA} = S_{bsb} = \sum_{i=1}^{C-1}\sum_{j=1}^{K_i}\sum_{l=i+1}^{C}\sum_{n=1}^{K_l} p_{ij}p_{ln}(\mu_{ij}-\mu_{ln})(\mu_{ij}-\mu_{ln})^T \tag{6}$$

where C denotes the number of classes, K_i denotes the number of subclasses in class i, p_{ij} and p_{ln} denote priors, and μ_{ij} denotes the mean of the jth subclass in class i. The within-class scatter matrix is the instance covariance matrix

$$S_w^{SDA} = \Sigma X = \frac{1}{N}\sum_{j=1}^{N}(x_j - \mu)(x_j - \mu)^T \tag{7}$$

where N, x_j, and μ are the number of instances, the jth instance and the mean of all instances respectively. The redefined objective function is the following:

$$J^{SDA}(W) = \frac{tr(W^T S_b^{SDA} W)}{tr(W^T S_w^{SDA} W)} = \frac{tr(W^T S_{bsb} W)}{tr(W^T \Sigma X W)}. \tag{8}$$

SDA uses a nearest neighbor based clustering algorithm to divide each class into several subclasses, and automatically determines the optimal number of subclasses by the *leave-one-out-test* (LOOT) criterion proposed in [10], or by a faster *stability criterion* [27] [6].

2.3 Mixture Subclass Discriminant Analysis

Mixture subclass discriminant analysis (MSDA) [11] is an extension of SDA. It adopts SDA's between-subclass scatter matrix S_{bsb}, but replaces SDA's within-class scatter matrix by a new ***within-subclass scatter matrix***, which is defined as

$$S_w^{MSDA} = \widehat{\Sigma X} = S_{bsb} + S_{ws}, \tag{9}$$

where S_{bsb} is defined as in SDA, and S_{ws} is defined as follows:

$$S_{ws} = \sum_{i=1}^{C}\sum_{j=1}^{K_i} p_{ij}(x_{ij} - \mu_{ij})(x_{ij} - \mu_{ij})^T \tag{10}$$

where C, K_i, p_{ij}, x_{ij} and μ_{ij} are the number of classes, the number of subclasses in class i, prior, the instances in subclass j of class i, and the mean of subclass j in class i. The Fisher objective function of MSDA is the:

$$J^{MSDA}(W) = \frac{tr(W^T S_b^{SDA} W)}{tr(W^T S_w^{MSDA} W)} = \frac{tr(W^T S_{bsb} W)}{tr(W^T \widehat{\Sigma X} W)} \tag{11}$$

MSDA applies nongaussianity criterion, based on skewness and kurtosis, to select a class or subclass that is not Gaussian distribution. It then applies the LOOT criterion (or stability criterion if speed is required) to re-partiton the selected classes or subclasses to getting optimal number of subclasses.

2.4 LDA for Over-Reducing Problem

LDA for over-reducing problem (orLDA) [7] is a variant of LDA that is aimed to address the *over-reducing* problem for binary classification by using a new between-class scatter matrix. Instead of using the mean of every class to subtract the mean of the whole data, orLDA uses every instance in one class to subtract the mean of the other class. In this way orLDA can get more LDA features than the original LDA. The new between-class scatter matrix is defined as follows:

$$\begin{aligned} S_b{}^{orLDA} = \; & \tfrac{1}{N}(N_1 \textstyle\sum_{j=1}^{N_1}(x_{1j} - \mu_2)(x_{1j} - \mu_2)^T + \\ & N_2 \textstyle\sum_{j=1}^{N_2}(x_{2j} - \mu_1)(x_{2j} - \mu_1)^T) \end{aligned}$$

where N is the number of instances, N_i is the number of instances in class i ($i = 1, 2$) such that $\sum_{i=1}^{2} N_i = N$, μ_i is the mean of the instances in class i, and x_{ij} is the jth instance in class i.

3 Separability-Oriented Subclass Discriminant Analysis

In order to have an insight into LDA in general and to answer the research question on Page 2 specifically, we propose *separability-oriented subclass discriminant analysis* (SSDA) – an extension of subclass discriminant analysis. The objective is to find those LDA features that together (1) maximise the between-subclass separation within a class (2) minimise the within-class scatterness and (3) maximise the overall between-class scatterness. The between-subclass separation objective is achieved through clustering guided by a separability criterion. The within-class scatterness objective and the between-class scatterness objective are achieved through re-defined within-class scatter matrix and between-class scatter matrix.

3.1 Between-subclass Separation by Hierarchical Clustering

Like SDA and MSDA, we divide each class into subclasses through clustering before the LDA optimisation procedure is applied. We select agglomerative hierarchical clustering[7] for this, because hierarchical clustering is stable in that for a given K and a given data set, the same clustering is obtained no matter when the algorithm is run. This is not the case with K-means clustering as it sets the initial clustering randomly.

We are interested in the optimal number, K^*, of clusters (i.e., subclasses) for each class, which gives the best prediction performance using SSDA with other parameters being fixed. This is challenging if not impossible. One solution to the problem of finding the optimal K^* is a wrapper approach[8], but it is clearly very costly. We therefore need

[6] The LOOT criterion is computationally expensive so a computationally efficient criterion, the *stability criterion*, was introduced in order to find the optimal number of subclasses faster.

[7] Hierarchical clustering [28] is a method of cluster analysis which seeks to build a hierarchy of clusters. There are two approaches to hierarchical clustering: agglomerative (bottom up) and divisive (top down). In agglomerative approach each data instance starts in its own cluster, and pairs of clusters are merged as one moves up the hierarchy. In divisive approach all data instances start in one cluster, and splits are performed recursively as one moves down the hierarchy. In both approaches the clustering process continues (in either direction) until the given number (K) of clusters are obtained.

[8] A wrapper approach should work this way: for each K we run SSDA once to get one validation result, and we repeat this process for all K and choose the optimal K based on the validation results.

to find an efficient heuristic method. One approach is to design a criterion measuring the quality of a clustering for a given K; repeat the clustering process for all K values; and then select the K value that results in a clustering with the best value for the criterion. SDA and MSDA both take this approach and use the leave-one-out criterion or stability criterion.

We consider a different criterion, called *separability criterion*, which is aimed to find the clustering that maximises the average distance between the mean of a class and the means of subclasses in this class. This criterion is defined as follows:

$$K_i^* = \arg\max_K (AED_{K,i}) \tag{12}$$

where K_i^* is optimal number of subclasses in class i and $AED_{K,i}$ is the *average Euclidean distance* (AED) between the mean of the class and the means of its K subclasses, which is given below:

$$AED_{K,i} = \frac{1}{K}\sum_{j=1}^{K}\|\mu_{ij} - \mu_i\|_2 \tag{13}$$

where μ_{ij} is the mean of the jth subclass in class i and μ_i is the mean of class i. It is clear the bigger the AED, the more separated the subclasses are from each other.

The algorithm for finding the optimal number of subclasses (i.e., clusters) K^* for each class, in terms of the separability criterion in Eq.(12) is described in Algorithm 1. Given a max value K_{max}, we run the hierarchical clustering algorithm K_{max} times with $K = 1$ to K_{max}, calculating our separability criterion for every K then taking the K corresponding to the best separability criterion value as the optimal K^*.

Algorithm 1 Finding the optimal number K^* of subclasses automatically. In this algorithm, *nClass* is the number of classes, *classMean* is the mean of a class, *subclassMean* is the mean of a subclass, K_{max} denotes the maximum number of subclasses to consider for a class, K^* is the optimal number of subclasses for a class, K is the number of subclasses, TED denotes the total of all Euclidean distances in a class and ED_i is the Euclidean distance between *classMean* and ith *subclassMean*.

```
Input: A set of training instances and $K_{max}$
Output: $K^*$
  for C = 1 : nClass do
      Calculate classMean;
    for K = 1: $K_{max}$ do
        Apply hierarchical clustering algorithm and obtain subclasses;
        Calculate subclassMean;
        $TED = 0$;
      for i = 1:K do
          Calculate $ED_i$;
          Calculate $TED = TED + ED_i$;
      end for
        Calculate and record $AED_K = \frac{TED}{K}$;
    end for
    $K^* = \arg\max_K (AED_K)$;
  end for
```

After the optimal number of subclasses is found for every class, the subclasses are subsequently obtained for every class. We use these subclasses to calculate our new between-class scatter matrix and within-class scatter matrix.

3.2 The Re-defined Within-class Scatterness

The original LDA uses individual instances and the mean of the class in its definition of the within-class scatter matrix (See Fig. 2(a) for an illustration), and MSDA uses instances and means of subclasses. SDA defines the within-class scatter as instance covariance matrix.

In SSDA, we use individual instances, means of subclasses and the mean of the class to define the new within-class scatter matrix (See Fig. 2(b) for an illustration). It measures within-class scatterness at two levels: *local scatterness* at subclass level and *global scatterness* at class level. Local scatterness measures the degree to which data instances in a subclass of a class are scattered around the subclass mean, and global scatterness measures the degree to which subclass means in a class are scattered around the class mean. It should be noted that in most research on LDA, including the original LDA, only global scatterness is considered. Fig. 1 illustrates local scatterness and global scatterness for a class.

In classification tasks it is possible that some class may have different modalities (or clusters). One example is face recognition where a person's face images may be front view or side view, resulting in different modalities when all images are represented in the same data space. The new within-class scatterness represents multimodality information, therefore optimising this within-class scatterness will separate different modalities more clearly and hopefully the resulting data reduction will have better performance.

Suppose there are N instances $x_i \in R^n$ for $i = 1, 2, \ldots, N$ from C classes, N_i is the number of instances in class i (i= 1, 2,. . . ,C) such that $\sum_{i=1}^{C} N_i = N$, K_i is the number of subclasses in class i, N_{ij} is the number of instances in subclass j of class i, μ_{ij} is the mean of subclass j of class i, μ_i is the mean of class i and x_{ijm} is the mth instance in subclass j of class i. The new within-class scatter matrix for SSDA is defined as follows:

$$S_w^{SSDA} = S_{sw} \quad = \quad S_{sw_1} + S_{sw_2} \tag{14}$$

where

$$S_{sw_1} \quad = \quad \sum_{i=1}^{C}\sum_{j=1}^{K_i}\sum_{m=1}^{N_{ij}}(x_{ijm} - \mu_{ij})(x_{ijm} - \mu_{ij})^T \tag{15}$$

$$S_{sw_2} \quad = \quad \sum_{i=1}^{C}\frac{N_i}{N}\sum_{j=1}^{K_i}(\mu_{ij} - \mu_i)(\mu_{ij} - \mu_i)^T \tag{16}$$

where S_{sw_1} measures local scatterness and S_{sw_2} measures global scatterness.

3.3 The Re-defined Between-class Scatterness

In the original LDA the between-class scatter matrix is defined in terms of the means of classes and the mean of all instances (See Fig. 3(a)). In SDA and MSDA, the between-class scatter matrices are all defined only by the means of subclasses (see Eq.6).

In SSDA we define a new between-class scatter matrix using the means of subclasses and the mean of all instances; that is, we measure between-class scatterness by the difference between subclass means and a single point of reference, i.e., the mean of all data instances (See Fig. 3(b)).

Using the same notation as in Section 3.2, our new between-class scatter matrix for SSDA is defined as follows:

$$S_b^{SSDA} = S_{sb} = \sum_{i=1}^{C} \frac{N_i}{N} \sum_{j=1}^{K_i} (\mu_{ij} - \mu)(\mu_{ij} - \mu)^T \tag{17}$$

It is clear that this breaks the $C-1$ limitation.

S_b^{SSDA} is similar to the between-class scatter measure $S_b^{ZM2004} = \sum_{i=1}^{C} \sum_{j=1}^{K_i} p_{ij}(\mu_{ij} - \mu)(\mu_{ij} - \mu)^T$ [29]. There is however a key difference between them. The subclass prior in S_b^{SSDA} is N_i/N, while the subclass prior in S_b^{ZM2004} is N_{ij}/N. As a result of this, our approach places emphasis on subclass separability and increases the contribution of subclasses in separating different classes.

3.4 The LDA Optimisation for SSDA

The LDA optimisation for SSDA is done through the Fisher objective (Eq.3), where we replace S_b, or S_w or both by our new versions, thus resulting in three different versions of SSDA.

3.4.1 SSDA-1

In this version, we use the new between-class scatter matrix and the original within-class scatter matrix so the Fisher objective function is

$$J^{SSDA-1}(W) = \frac{tr(W^T S_b^{SSDA} W)}{tr(W^T S_w W)} = \frac{tr(W^T S_{sb} W)}{tr(W^T S_w W)}.$$

SSDA-1 maximises class separation through maximising subclass separation thus preserving the within-class structure whilst separating different classes. Therefore we expect this to lead to data reductions that have better performance. Additionally our S_{sb} is defined in terms of subclass means rather than class means, therefore SSDA-1 is not $C-1$ limited. Extensive experiments confirm our hypothesis.

It should be noted that the solution to the LDA optimisation using the above Fisher objective is the optimal projective matrix whose columns are the eigenvectors corresponding to the largest eigenvalues of $S_w^{-1} S_{sb}$ rather than $S_w^{-1} S_b$.

3.4.2 SSDA-2

In this version, we use the new within-class scatter matrix S_{sw} and the original between-class scatter matrix S_b so the Fisher objective function is

$$J^{SSDA-2}(W) = \frac{tr(W^T S_b W)}{tr(W^T S_w^{SSDA} W)} = \frac{tr(W^T S_b W)}{tr(W^T S_{sw} W)}.$$

The new within-class scatter matrix encodes within-class scatterness at two levels – the subclass level and the class level. As discussed earlier, this within-class scatter matrix measures within-class multimodality thus maximisng the above Fisher objective can be expected to result in data reductions leading to good classification performance. Again the optimal projective matrix consists of eigenvectors corresponding to the largest eigenvalues of $S_{sw}^{-1} S_b$.

3.4.3 SSDA-3

In this version we use both new matrices thus the Fisher objective function is

$$J^{SSDA-3}(W) = \frac{tr(W^T S_b^{SSDA} W)}{tr(W^T S_w^{SSDA} W)} = \frac{tr(W^T S_{sb} W)}{tr(W^T S_{sw} W)}. \tag{18}$$

We have argued the advantages of both new matrices, therefore there is reason to expect SSDA-3 to perform well. The optimal projective matrix consists of eigenvectors corresponding to the largest eigenvalues of $S_{sw}^{-1} S_{sb}$.

3.5 Discussion

The SSDA algorithm has two sub-procedures: (1) between-subclass separation by hierarchical agglomerative clustering, which is equipped with our separability criterion; (2) LDA optimisation using within-class and between-class scattering matrices. Both procedures are known to converge.

Algorithm 1 has a time complexity of $\sum_{i=1}^{C}(K_{max}^2 + K_{max} * O(N_i^2 * \log(N_i)))$, where C is the number of classes, K_{max} is the maxmum number of subclasses to consider, N_i is the number of data instances in class i and $O(N_i^2 * \log(N_i))$ is the time complexity of the hierarchical agglomerative clustering algorithm [30].

4 Experimental Evaluation

We have designed a series of experiments to evaluate SSDA. Firstly, we want to compare our new SSDA with its closest counterparts, LDA, SDA and MSDA, and two nonlinear discriminative analysis (nonlinear-DA) methods, GDA and KMSDA, in terms of classification performance and run time. Secondly, we want to compare the three versions of SSDA in order to gain a better understanding of SSDA. Thirdly, we compare SSDA with two SDA versions with different clustering approaches in order to gain an insight into SSDA and to see the reasons behind their behaviours in the experiments.

4.1 The Experiments

Linear Discriminant Analysis (and its variants) has been applied to a wide range of data intensive domains, especially in computer vision, as a data reduction technique. In our experiments, we consider a range of classification tasks from general data mining, imbalanced data mining, to face recognition and face verification.

We select ten data sets from UCI Data Repository [31] for general data mining; eight imbalanced data sets from the KEEL Data Repository [32] for imbalanced data mining; and four face databases in the public domain for face recognition/verification – ORL face database [33], AR face database [34], YouTube face database [35], and Labeled Faces in the Wild (LFW) face database [36].

We use k-Nearest Neighbor (kNN, k=1) as the classifier and ten-fold cross-validation as evaluation framework. As evaluation metrics we consider *estimated mean accuracy* (EMA) and *standard error of the mean* (SEM): $EMA = \frac{\sum_{i=1}^{10} p_i}{10}$, where p_i denotes the percentage of correct classification in the ith fold validation; $SEM = \frac{\delta}{\sqrt{10}}$, where $\delta = \sqrt{\frac{\sum_{i=1}^{10}(p_i - EMA)^2}{9}}$.

General information about the ten data sets from UCI Data Repository [31] is shown in TABLE 1. The same information about the eight imbalanced data sets from KEEL Data Repository [32] is shown in TABLE 2. All data sets are numerical and have no missing information as we need to compute the mean and distance.

Name of data set (Acronym)	#Instance	#Class	#Attribute
WDBC	569	2	30
Iris	150	3	4
Vehicle	846	4	18
Red Wine Quality (RWQ)	1599	6	11
Breast Tissue (BT)	106	6	9
Seeds	210	3	7
Banknote Authentication (BA)	1372	2	4
Leaf	340	30	14
Urban Land Cover (ULC)	675	9	147
Forest Type Mapping (FTM)	523	4	27

TABLE 1 General information about ten UCI data sets used in experiments

Name of data set	#Attribute	#Instance	#Class	# IR
Glass1	9	214	2	1.82
New-thyroid1	5	215	2	5.14
Dermatology	34	366	6	5.55
Hayes-roth	4	132	3	1.7
Led7digit-0-2-4-5-6-7-8-9_vs_1	7	443	2	10.97
Pima	8	768	2	1.87
Vowel0	13	988	2	9.98
Wisconsin	9	683	2	1.86

TABLE 2 General information about the eight imbalanced data sets used in the experiments. IR is short for *Imbalanced Ratio*.

Face Image Data

Face recognition is a multi-class classification problem. The goal of face recognition is to determine if an image is from someone in the database when we have a collection of images for each person in the database. In our experiments we use three face image databases which are commonly used in face recognition research literature: the YouTube faces database [35], the ORL face database [33] and the AR face database [34].

The YouTube faces database: It contains 3425 videos of 1595 different people collected from the YouTube website. The average length of each video clip is 181.3 frames, there are large variations in expression, pose and illumination in each video. In our experiments, we use the aligned image database which contains aligned face frames taken from videos, which are represented using YouTube's Center-Symmetric LBP (CSLBP) descriptor [37].

The ORL face database: It consists of a total of 400 images of 40 distinct persons. Each parson has ten different images and the size of each image is 92 by 112 pixels, which will generate a feature space of 10,304 dimensions. All the images were taken against a dark homogeneous background with the subjects in an upright, frontal position (see Fig. 4(a) for some examples).

The AR face database: It contains frontal-view face images of 126 different persons (70 males and 56 females). Each person was photographed under different lighting conditions and distinct facial expressions, and some images have partial occlusions (sunglasses or scarf). A total of 13 images were taken in each session for a total of two sessions, which were separated by an interval of two weeks. Therefore, there are 26 frontal face images per person. In our experiments, we use a subset of AR-face data set. We use face images of 100 persons and 7 nonoccluded face images of each person from the first session. Besides, we crop the face part of the image and then resize all images to a standard image size of 80 by 100 pixels (see Fig. 4(b) for some examples). This yields a 8,000-dimensional feature space.

Labeled Faces in the Wild (LFW) Face Database: We use this database for face verification[9]. The database consists of 13,233 images of 5,749 people, which are organised into 2 views: view one is a development set of 3,200 pairs, which is used for building models and selecting features; view two is a ten-fold cross-validation set of 6,000 pairs for evaluation. The size of each image is 250 by 250 pixels. All the images in LFW were collected from the Internet with large intra-personal variations (see Fig. 4(c) for some examples). There are three versions of the LFW: original, funneled and aligned. In our experiments, we use the aligned version [38]. Besides, we use a subset of view two of LFW. We randomly choose 200 matched face pairs and 200 mismatched face pairs from view two and crop each image to an image of 80 by 150 pixels as in [3]. We thus have 24,000 features for each image of LFW that we use.

Dimensionality Reduction for Face Recognition/Verification

The images are initially represented using pixel-based representation thus having large numbers of features. Therefore face recognition and verification both have the *small sample size* problem. To deal with this problem, we use the two-stage PCA + LDA [18]. We use PCA to reduce data dimensionality retaining principal components which can explain 95% of variance, before LDA, SDA, MSDA ,GDA, KMSDA and SSDA are used.

4.2 The Results: All Methods

UCI Data: Our experimental results on UCI data are presented in TABLE 3 and TABLE 4. It can be seen that SSDA (SSDA-1, SSDA-2, SSDA-3) have better performance than LDA, SDA, MSDA,GDA and KMSDA on a majority of the data sets. In particular SSDA-1 has better performance than LDA on all ten data sets. This set of results suggests that SSDA captures more discriminant information than LDA, SDA, MSDA,GDA and KMSDA .

Imbalanced Data: Our experimental results on imbalanced data are shown in TABLE 5 and TABLE 6. It can be seen that SSDA has better performance than other methods on most of the imbalanced data sets; in particular, SSDA3 has the best performance on all imbalanced data sets.

Face Image Data: Our experimental results are presented in TABLE 7 and TABLE 8. It can be seen that the SSDA variants have better performance than the other methods on most of the face databases; in particular, SSDA1 has the best performance on 3 out of the 4 face databases.

[9] Face verification is a binary classification problem, and its goal is to decide if two given face images are from the same person (i.e. match or not).

Datasets Methods	BA	BT	FTM	Iris	Leaf	RWQ	Seeds	ULC	Vehicle	WDBC
LDA	0.9956 ± 0.0016	0.7164 ± 0.0506	0.8468 ± 0.0142	0.9667 ± 0.0149	0.6912 ± 0.0417	0.6529 ± 0.0085	0.9524 ± 0.0142	0.7749 ± 0.0145	0.7398 ± 0.0168	0.9579 ± 0.0079
SSDA-1	**1.0000** ± **0.0000**	0.7173 ± 0.0409	0.8621 ± 0.0132	0.9733 ± 0.0109	0.7735 ± 0.0232	0.6716 ± 0.0097	**0.9762** ± **0.0106**	**0.7807** ± **0.0091**	0.8050 ± 0.0148	0.9614 ± 0.0078
SSDA-2	0.9971 ± 0.0012	**0.7182** ± **0.0335**	0.8755 ± 0.0127	0.9733 ± 0.0147	0.7382 ± 0.0242	0.6591 ± 0.0102	0.9429 ± 0.0138	0.7764 ± 0.0115	0.7529 ± 0.0134	0.9579 ± 0.0083
SSDA-3	**1.0000** ± **0.0000**	0.7164 ± 0.0401	0.8777 ± 0.0095	0.9733 ± 0.0109	0.7500 ± 0.0245	0.6667 ± 0.0110	0.9476 ± 0.0111	0.7734 ± 0.0098	0.8049 ± 0.0148	**0.9649** ± **0.0091**
SDA	0.9956 ± 0.0016	0.5473 ± 0.0478	**0.8794** ± **0.0105**	0.9667 ± 0.0149	0.5618 ± 0.0272	0.6360 ± 0.0101	0.9524 ± 0.0142	0.4889 ± 0.0176	0.7245 ± 0.0176	0.9297 ± 0.0070
MSDA	0.9913 ± 0.0026	0.6318 ± 0.0568	0.8546 ± 0.0116	0.9533 ± 0.0142	**0.7794** ± **0.0249**	**0.6754** ± **0.0116**	0.9667 ± 0.0073	0.6325 ± 0.0175	0.7812 ± 0.0125	0.9510 ± 0.0060
KMSDA(gaussian)	0.9990 ± 0.0008	0.2082 ± 0.0134	0.3725 ± 0.0188	0.9373 ± 0.0171	0.7585 ± 0.0211	0.6069 ± 0.0118	0.9129 ± 0.0219	0.1483 ± 0.0133	0.2293 ± 0.0140	0.6100 ± 0.0258
KMSDA(linear)	0.9993 ± 0.0007	0.6412 ± 0.0396	0.8677 ± 0.0173	**0.9773** ± **0.0140**	0.7535 ± 0.0214	0.6732 ± 0.0137	0.9333 ± 0.0150	0.7703 ± 0.0205	**0.8136** ± **0.0143**	0.9313 ± 0.0103
KMSDA(poly)	0.9727 ± 0.0048	0.5282 ± 0.0527	0.7823 ± 0.0231	0.8740 ± 0.0217	0.7244 ± 0.0198	0.6401 ± 0.0121	0.8490 ± 0.0195	0.5644 ± 0.0210	0.7909 ± 0.0144	0.8442 ± 0.0167
GDA(gaussian)	0.7369 ± 0.0162	0.1800 ± 0.0170	0.2863 ± 0.0380	0.9200 ± 0.0194	0.7588 ± 0.0214	0.5559 ± 0.0130	0.9238 ± 0.0238	0.1706 ± 0.0225	0.2540 ± 0.0151	0.4993 ± 0.0458
GDA(linear)	0.8863 ± 0.0361	0.6400 ± 0.0382	0.7151 ± 0.0179	0.9600 ± 0.0147	0.7706 ± 0.0227	0.5828 ± 0.0143	0.9429 ± 0.0119	0.3009 ± 0.0212	0.4727 ± 0.0168	0.8752 ± 0.0150
GDA(poly)	0.9177 ± 0.0198	0.6400 ± 0.0382	0.7152 ± 0.0160	0.9467 ± 0.0166	0.7706 ± 0.0227	0.5891 ± 0.0131	0.9524 ± 0.0123	0.3009 ± 0.0212	0.4727 ± 0.0168	0.8752 ± 0.0150

TABLE 3 EMA±SEM of all methods on ten UCI data sets

Datasets Methods	BA	BT	FTM	Iris	Leaf	RWQ	Seeds	ULC	Vehicle	WDBC
LDA	0.4608	0.4684	0.4116	0.3469	0.4088	0.4470	0.3726	1.0508	0.3943	0.3931
SSDA-1	13.4603	8.4878	9.0574	4.9063	5.8354	3.7611	5.3393	28.1698	11.5249	7.9278
SSDA-2	13.1489	8.6453	8.9739	5.0511	6.0323	3.7358	5.4260	22.5437	11.4149	7.6538
SSDA-3	13.2278	8.6484	8.9999	4.9841	7.1088	3.9986	5.5386	24.6621	11.5299	7.8622
SDA	14.0491	5.3398	10.9097	3.3197	7.1660	9.6376	9.3230	73.7869	21.2983	20.1334
MSDA	84.7338	44.3218	1197.4534	85.1082	1158.5275	441.1192	176.1178	41389.6441	847.1926	493.4477
KMSDA(gaussian)	2690.3158	30.1266	302.3954	66.6830	4416.8484	3037.8693	112.1421	7065.1503	415.5467	300.1595
KMSDA(linear)	2522.4015	28.3255	380.1810	66.5756	4921.0894	3210.1374	102.2283	6565.1182	733.3666	345.4465
KMSDA(poly)	2812.9827	25.7762	414.3065	66.7427	4474.3195	3068.0893	96.6440	8139.9187	586.9152	339.9491
GDA(gaussian)	73.4912	0.2170	4.6560	0.3143	1.6497	114.3346	1.0386	9.8661	16.1372	7.6408
GDA(linear)	34.0265	0.1945	1.3793	0.1394	1.5265	53.4882	0.3356	8.0627	4.9252	6.2217
GDA(poly)	34.4164	0.1598	1.3974	0.1416	1.4434	52.3882	0.2470	7.3010	4.5603	5.5423

TABLE 4 Run time, in second, of all methods on ten UCI data sets

4.3 The Results: SSDA-1, SSDA-2 and SSDA-3

Although all versions of SSDA have better performance than LDA, SDA, MSDA, KMSDA and GDA in most of the cases, SSDA-1 appears to be the best of the three versions in many cases. To validate this further, a comparative study is conducted on the three versions of SSDA. We varied K_{max} from 1 to 15 and, for each K_{max} value, we run the three SSDA algorithms on 10 UCI data sets and 3 face databases, and we counted how many times each algorithm obtained the best performance. These counts are charted in Fig. 5. It should be noted that the sum of the counts for each K_{max} may be higher than 13 due to ties. It is clear that SSDA-1 is a clear winner especially for bigger K_{max}.

This evaluation suggests that combining S_{sb} and S_w can get more discriminant information than other ways of combination.

4.4 The Results: SDA Versions Using Different Clustering Approaches

We modified the SDA implementation by replacing SDA's NN-clustering by the *separability criterion based hierarchical clustering as used for SSDA – named* SSDA clustering. We conducted experiments with the two versions of SDA (SDA with NN-clustering and SDA with SSDA clustering) and SSDA on 10 UCI data sets. Experimental results are shown in TABLE 9. It can be seen that SDA with SSDA clustering is

Datasets Methods	Glass1	New-thyroid1	Dermatology	Hayes-roth	Led7digit	Pima	Vowel0	Wisconsin
LDA	0.6119 ± 0.0226	0.9483 ± 0.0150	0.9645 ± 0.0134	0.7725 ± 0.0499	0.9390 ± 0.0067	0.6782 ± 0.0216	0.9393 ± 0.0067	0.9635 ± 0.0090
SSDA-1	0.8316 ± 0.0175	0.9859 ± 0.0072	0.9700 ± 0.0075	0.8022 ± 0.0435	0.9390 ± 0.0067	0.6874 ± 0.0196	**1.0000 ± 0.0000**	0.9649 ± 0.0082
SSDA-2	0.6175 ± 0.0236	0.9814 ± 0.0124	0.9670 ± 0.0089	0.7582 ± 0.0478	**0.9391 ± 0.0067**	0.6808 ± 0.0207	0.9433 ± 0.0064	0.9590 ± 0.0093
SSDA-3	**0.8374 ± 0.0244**	**0.9952 ± 0.0048**	**0.9700 ± 0.0027**	**0.8176 ± 0.0449**	**0.9391 ± 0.0067**	**0.6938 ± 0.0172**	**1.0000 ± 0.0000**	**0.9692 ± 0.0067**
SDA	0.5799 ± 0.0412	0.9810 ± 0.0105	0.9589 ± 0.0085	0.7725 ± 0.0499	0.9345 ± 0.0070	0.6886 ± 0.0182	0.9393 ± 0.0067	0.9590 ± 0.0065
MSDA	0.7623 ± 0.0273	0.9905 ± 0.0063	0.9370 ± 0.0158	0.6429 ± 0.0519	0.9345 ± 0.0062	0.5896 ± 0.0244	**1.0000 ± 0.0000**	0.8902 ± 0.0107
KMSDA(gaussian)	0.7402 ± 0.0281	0.8802 ± 0.0216	0.3558 ± 0.0426	0.7764 ± 0.0455	0.9265 ± 0.0106	0.6475 ± 0.0236	0.9993 ± 0.0006	0.9053 ± 0.0266
KMSDA(linear)	0.7680 ± 0.0321	0.9765 ± 0.0112	0.9661 ± 0.0094	0.7553 ± 0.0399	0.9388 ± 0.0067	0.6789 ± 0.0179	0.9986 ± 0.0012	0.9517 ± 0.0071
KMSDA(poly)	0.7408 ± 0.0282	0.9381 ± 0.0144	0.9609 ± 0.0078	0.8117 ± 0.0363	0.9368 ± 0.0084	0.6442 ± 0.0168	**1.0000 ± 0.0000**	0.9442 ± 0.0078
GDA(gaussian)	0.6965 ± 0.0152	0.6881 ± 0.0327	0.1532 ± 0.0321	0.7357 ± 0.0589	0.9345 ± 0.0070	0.5131 ± 0.0546	0.9160 ± 0.0105	0.8042 ± 0.0424
GDA(linear)	0.7433 ± 0.0376	0.8883 ± 0.0244	0.5795 ± 0.0375	0.7495 ± 0.0651	0.9391 ± 0.0067	0.6444 ± 0.0211	0.9717 ± 0.0039	0.9590 ± 0.0094
GDA(poly)	0.7528 ± 0.0374	0.8879 ± 0.0245	0.5902 ± 0.0433	0.7198 ± 0.0615	0.9390 ± 0.0058	0.6560 ± 0.0222	0.9787 ± 0.0053	0.9561 ± 0.0092

TABLE 5 EMA±SEM of all methods on eight imbalanced data sets

Datasets Methods	Glass1	New-thyroid1	Dermatology	Hayes-roth	Led7digit	Pima	Vowel0	Wisconsin
LDA	0.6590	0.0643	0.1265	0.0597	0.1355	0.1429	0.2050	0.1257
SSDA-1	7.4430	6.2264	21.9584	8.9783	8.8396	12.7986	28.3226	12.5941
SSDA-2	7.3851	6.4622	20.3601	8.9707	8.3202	10.6121	26.0256	10.5545
SSDA-3	7.7689	6.3447	20.6194	9.0347	8.8047	11.9960	27.0266	12.0818
SDA	14.8135	10.2511	48.5555	10.7161	14.8995	23.9342	32.0563	23.8213
MSDA	14.2680	8.9955	162.0084	34.5359	14.8178	5.2682	15.4729	7.6626
KMSDA(gaussian)	91.7083	78.0098	785.2526	52.7884	428.1921	896.2775	2245.4591	853.9795
KMSDA(linear)	75.1814	65.2564	964.5750	54.4705	241.8900	762.5810	1401.7950	437.2513
KMSDA(poly)	70.6793	62.1042	834.6825	51.0337	350.5261	983.0632	1799.6884	779.3523
GDA(gaussian)	1.1456	1.5269	2.6936	0.2528	1.4012	25.1368	28.4806	9.0260
GDA(linear)	0.6904	0.3037	1.4987	0.1497	0.7809	8.0520	15.3227	4.7597
GDA(poly)	0.6404	0.3240	1.5158	0.1678	0.7831	8.4522	15.9725	5.1790

TABLE 6 Run time, in second, of all methods on eight imbalanced data sets

Datasets Methods	YouTube	LFW	AR	ORL
LDA	0.9790±0.0043	0.5450±0.0298	0.9157±0.0105	0.9700±0.0122
SSDA-1	**0.9830±0.0030**	**0.7100±0.0155**	**0.9471±0.0068**	0.9725±0.0058
SSDA-2	0.9800±0.0045	0.5900±0.0369	0.9200±0.0080	0.9725±0.0115
SSDA-3	0.9800±0.0045	0.7000±0.0144	0.9114±0.0090	0.9725±0.0115
SDA	0.9790±0.0043	0.7075±0.0247	0.9157±0.0105	0.9700±0.0122
MSDA	0.9830±0.0037	0.6600±0.0208	0.8720±0.0129	0.9775±0.0096
KMSDA(gaussian)	0.9830±0.0047	0.5000±0.0000	0.0000 ±0.0000	0.0250±0.0000
KMSDA(linear)	0.9790±0.0038	0.6750±0.0227	0.9200±0.0105	0.9625±0.0125
KMSDA(poly)	0.9800±0.0049	0.6900±0.0292	0.9443 ±0.0065	**0.9825±0.0065**
GDA(gaussian)	0.9730±0.0060	0.5000±0.0000	0.0014±0.0014	0.0250±0.0000
GDA(linear)	0.9790±0.0043	0.5575±0.0140	0.9157±0.0105	0.9700±0.0122
GDA(poly)	0.9800±0.0039	0.5625±0.0136	0.9157±0.0105	0.9700±0.0122

TABLE 7 EMA±SEM of all methods on four face databases

Datasets Methods	AR	LFW	ORL	YouTube
LDA	3.8463	4.1128	2.6092	7.7797
SSDA-1	54.9050	367.5684	32.4669	523.2413
SSDA-2	57.0464	301.0679	30.4252	609.9837
SSDA-3	59.2389	318.6475	36.1923	628.6595
SDA	363.7470	1179.4425	654.5643	8761.0927
MSDA	1042403.8487	6639.7915	382253.9291	609021.1228
KMSDA(gaussian)	416208.2398	2225.8405	162803.3881	—
KMSDA(linear)	256660.7750	2547.5120	144394.6722	—
KMSDA(poly)	479095.5967	2276.5896	131822.3761	—
GDA(gaussian)	22.8550	8.5004	6.6577	61.8623
GDA(linear)	14.1721	8.9017	5.6373	29.1588
GDA(poly)	13.6173	8.2842	5.7491	29.5443

TABLE 8 Run time, in second, of all methods on four face databases. The missing entries indicate the respective algorithms ran for too long.

not consistently different to SDA with NN clustering. Furthermore SSDA is better than both versions of SDA on most of the data sets. This suggests that the good performance of SSDA should be due to the new separability-oriented scatter-ness proposed in this paper.

4.5 The Results: Runtime Performance

Runtime results of these algorithms are presented in TABLE 4, TABLE 6, and TABLE 8. It is clear that all versions of SSDA are slower than LDA but faster than SDA and MSDA in most of the cases.

5 Separability

In this section we evaluate the separability of LDA, SDA, MSDA and SSDA in terms of two measures: (1) visually, the overlapping of the subclasses obtained and (2) the extent to which the classifying of data in the LDA space are compact (the variance between members of a class is small) and are also well separated (the means of different classes are sufficiently far apart). We select SSDA-1 in this evaluation as it is the best of the three versions of SSDA.

For measure (1) we create charts, Fig. 6, showing subclass structures sought after by SDA, MSDA and SSDA-1 in the original data space. We can see subclass structure of SSDA-1 has much less overlapping than SDA and MSDA.

For measure (2) we use the well known Dunn index[10] [40]. TABLE 10 shows the Dunn index value for the classifying of data in the original data space and the LDA spaces. It is clear that SSDA-1 has highest Dunn index value in 8 out of 13 data sets used in this experiment.

To show their difference in the LDA space further we take a closer look at a data set that has two classes – Banknote Authentication, which has 1372 instances and 4 features. We divide it equally into two parts – one for training and another for testing with 686 instances each. We apply LDA, SDA, MSDA and SSDA-1 to the training data to map them to the LDA space. The distributions of the training data in these LDA spaces are shown in Fig. 7. Note that LDA and SDA reduced the training data to 1 LDA dimension thus the data distribution is a straight line, MSDA to 2 LDA dimensions, and SSDA-1 to 4 LDA dimensions where we select the first two components to show the data distribution. It is clear that using the first two LDA components, the training data can not be completely separated by their class memberships. There is no more LDA component to use in all cases but SSDA-1. When we consider the third LDA component in SSDA-1, the training data can be completely separated as shown in Fig. 7(d).

We also apply nearest neighbour classifier to the test data set, using the projected training data set in the LDA space as the model. The classification accuracies are: 99.13%, 99.13%, 99.27%, and 100.00% for LDA, SDA, MSDA and SSDA-1 respectively.

6 Conclusion

In this paper we propose a new LDA variant, *separability oriented subclass discriminant analysis* (SSDA), which uses a *separability criterion* we devise to divide every class into subclasses. This subclass structure has much less overlapping than those obtained by SDA and MSDA, and is used as the basis of constructing an LDA projection (from the original data space to the LDA space) using *new scatter matrices* we defined here. The projected data in the LDA space have consistently higher Dunn index values than LDA/SDA/MSDA, meaning they have higher within-class compactness and higher between-class separation. Extensive experimentation has shown that in most cases SSDA outperforms the original LDA as well as SDA and MSDA, the two state of the art subclass-based LDA variants. SSDA is also faster than SDA and MSDA in most cases.

The difference between SSDA and LDA is the fact that SSDA has exploited the subclass structure within classes. The difference between SSDA and SDA/MSDA is the fact that, although they all exploit the subclass structure, they use different criteria to divide a class into subclasses and they use different scatter matrices. SSDA aims to separate different subclasses within every class as well different classes whereas SDA and MSDA do not have the same aim.

It is well known that reducing variance (as a means of reducing overefitting) under given learning bias is one way of reducing generalisation errors. The usual approach is to control the size of hypothesis space by e.g. keeping the dimensionality of the hypothesis space small or keeping the norm of the hypothesis space small. SSDA advocates separating different subclasses within every class as well as

[10] The Dunn index is a metric for evaluating clustering algorithms. It measures the "clusterness" of a clustering (i.e. partition) of a data set – the extent to which the "clusters are compact, with a small variance between members of the cluster, and well separated, where the means of different clusters are sufficiently far apart, as compared to the within cluster variance" [39]. For a given data set, a higher Dunn index indicates better clustering.

Methods Datasets	SDA with NN clustering	SDA with SSDA clustering	SSDA-1	SSDA-2	SSDA-3
BA	0.9956 ± 0.0016	0.9964 ± 0.0012	**1.0000** ± **0.0000**	0.9971 ± 0.0012	**1.0000** ± **0.0000**
BT	0.5473 ± 0.0478	0.6691 ± 0.0436	0.7173 ± 0.0409	**0.7182** ± **0.0335**	0.7164 ± 00.0401
FTM	**0.8794** ± **0.0105**	0.6895 ± 0.0721	0.8621 ± 0.0132	0.8755 ± 0.0127	0.8777 ± 0.0095
Iris	0.9667 ± 0.0149	**0.9733** ± **0.0109**	**0.9733** ± **0.0109**	0.9733 ± 0.0147	**0.9733** ± **0.0109**
Leaf	0.5618 ± 0.0272	0.5824 ± 0.0297	**0.7735** ± **0.0232**	0.7382 ± 0.0242	0.7500 ± 0.0245
RWQ	0.6360 ± 0.0101	0.6492 ± 0.0129	**0.6716** ± **0.0097**	0.6591 ± 0.0102	0.6667 ± 0.0110
Seeds	0.9524 ± 0.0142	0.9524 ± 0.0142	**0.9762** ± **0.0106**	0.9429 ± 0.0138	0.9476 ± 0.0111
ULC	0.4889 ± 0.0176	0.4950 ± 0.0227	**0.7807** ± **0.0091**	0.7764 ± 0.0115	0.7734 ± 0.0098
Vehicle	0.7245 ± 0.0176	0.6608 ± 0.0319	**0.8050** ± **0.0148**	0.7529 ± 0.0134	0.8049 ± 0.0148
WDBC	0.9297 ± 0.0070	0.9121 ± 0.0109	0.9614 ± 0.0078	0.9579 ± 0.0083	**0.9649** ± **0.0091**

TABLE 9 EMA±SEM of SDA with NN-clustering, SDA with SSDA clustering, and SSDA on 10 UCI database.

Datasets	Original	LDA	SSDA-1	SDA	MSDA
BA	0.0395	0.0000	**0.0914**	0.0000	0.0043
BT	0.0001	**0.0285**	0.0062	0.0001	0.0254
FTM	0.0280	0.0093	0.0445	**0.0468**	0.0001
Iris	0.0585	0.0146	0.0817	**0.0819**	0.0176
Leaf	0.0050	0.0140	**0.0490**	0.0074	0.0106
RWQ	0.0007	0.0051	**0.0096**	0.0040	0.0049
Seeds	0.0456	0.0068	**0.0767**	0.0466	0.0014
ULC	0.0068	0.0382	**0.0570**	0.0245	0.0113
Vehicle	0.0095	0.0058	**0.0325**	0.0235	0.0185
WDBC	0.0025	0.0001	**0.0080**	0.0027	0.0005
AR	0.1741	0.2497	**0.2689**	0.2497	0.2562
LFW	**0.2288**	0.0000	0.0808	0.0000	0.0632
ORL	0.4299	**0.8119**	0.4791	0.4743	0.3140

TABLE 10 Dunn index for a classifying of data in the original data space, and an LDA space by LDA/SSDA-1/SDA/MSDA on all ten UCI data sets and the AR/LFW/ORL face data sets.

separating different classes. In other words, SSDA advocates within-class multimodality to classification in contrast to within-class unimodality. In the case of multimodality, each modality has small hypothesis space whereas in the case of unimodality, the single modality has large hypothesis space. As a result the multimodality approach is expected to have lower variance than the unimodality approach. In practice the unimodality approach may still result in more than one modality in the model, depending on the data and the learning algorithm, and the multimodality approach may end up with only one modality. A thorough investigation of multimodality/unimodality will involve a lot of testing and validating using both artificial and real data, which is clearly beyond the scope of this paper and is a direction for future work.

Other future work on SSDA will focus on two directions: further increase of classification accuracy and further reduction in computation time. In the first direction we will explore other (possibly nonlinear) ways of processing data to increase Dunn index of data. In the second direction we will redesign the whole SSDA pipeline to speed up the clustering process of finding the optimal K^*. Due to the way hierarchical clustering works, we can run the hierarchical clustering algorithm **once** (instead of K_{max} times in the current SSDA pipeline) from $K = n$ (when every data instance is a cluster) to $K = 1$ (when all data instances are merged into a single cluster), calculating our separability criterion for every K thus being able to find the optimal K^* that has the best separability criterion value at a lower computational cost. .

References

[1] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of eugenics*, vol. 7, no. 2, pp. 179–188, 1936.

[2] L. Zhang, M. Yang, Z. Feng, and D. Zhang, "On the dimensionality reduction for sparse representation based face recognition." in *ICPR*, 2010, pp. 1237–1240.

[3] M. Kan, D. Xu, S. Shan, W. Li, and X. Chen, "Learning prototype hyperplanes for face verification in the wild," *Image Processing, IEEE Transactions on*, vol. 22, no. 8, pp. 3310–3316, 2013.

[4] X. He, D. Cai, and J. Han, "Learning a maximum margin subspace for image retrieval," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 20, no. 2, pp. 189–201, 2008.

[5] C. L. He, L. Lam, and C. Y. Suen, "Rejection measurement based on linear discriminant analysis for document recognition," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 14, no. 3, pp. 263–272, 2011.

[6] Z. Fan, Y. Xu, and D. Zhang, "Local linear discriminant analysis framework using sample neighbors." *IEEE Transactions on Neural Networks*, vol. 22, no. 7, pp. 1119–1132, 2011.

[7] H. Wan, G. Guo, H. Wang, and X. Wei, *A New Linear Discriminant Analysis Method to Address the Over-Reducing Problem*. Springer International Publishing, 2015.

[8] K. Fukunaga, *Introduction to statistical pattern recognition*. Academic press, 2013.

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

[10] M. Zhu and A. M. Martinez, “Subclass discriminant analysis,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 28, no. 8, pp. 1274–1286, 2006.

[11] N. Gkalelis, V. Mezaris, and I. Kompatsiaris, “Mixture subclass discriminant analysis,” *Signal Processing Letters, IEEE*, vol. 18, no. 5, pp. 319–322, 2011.

[12] N. Gkalelis, V. Mezaris, I. Kompatsiaris, and T. Stathaki, “Mixture subclass discriminant analysis link to restricted gaussian model and other generalizations,” *Neural Networks and Learning Systems, IEEE Transactions on*, vol. 24, no. 1, pp. 8–21, 2013.

[13] Wikipedia, “Linear discriminant analysis,” https://en.wikipedia.org/wiki/Linear_discriminant_analysis, accessed: 2015-09-5.

[14] S. Pang, S. Ozawa, and N. Kasabov, “Incremental linear discriminant analysis for classification of data streams,” *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 35, no. 5, pp. 905–914, 2005.

[15] T.-K. Kim, B. Stenger, J. Kittler, and R. Cipolla, “Incremental linear discriminant analysis using sufficient spanning sets and its applications,” *International Journal of Computer Vision*, vol. 91, no. 2, pp. 216–232, 2011.

[16] D. Chu, L.-Z. Liao, M.-P. Ng, and X. Wang, “Incremental linear discriminant analysis: A fast algorithm and comparisons,” *Neural Networks and Learning Systems, IEEE Transactions on*, vol. 26, no. 11, pp. 2716–2735, Nov 2015.

[17] Y. A. Ghassabeh, F. Rudzicz, and H. A. Moghaddam, “Fast incremental lda feature extraction,” *Pattern Recognition*, vol. 48, pp. 1999–2012, 2015.

[18] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, “Eigenfaces vs. fisherfaces: Recognition using class specific linear projection,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 19, no. 7, pp. 711–720, 1997.

[19] A. Sharma and K. K. Paliwal, “A new perspective to null linear discriminant analysis method and its fast implementation using random matrix multiplication with scatter matrices.” *Pattern Recognition*, vol. 45, no. 6, pp. 2205–2213, 2012.

[20] Y. Guo, “Regularized linear discriminant analysis and its application in microarrays,” *Biostatistics*, vol. 8, no. 1, pp. 86–100, 2007.

[21] A. Sharma, K. K. Paliwal, S. Imoto, and S. Miyano, “A feature selection method using improved regularized linear discriminant analysis,” *Machine vision and applications*, vol. 25, no. 3, pp. 775–786, 2014.

[22] A. Sharma and K. K. Paliwal, “A deterministic approach to regularized linear discriminant analysis,” *Neurocomputing*, vol. 151, pp. 207–214, 2015.

[23] X. Shu and H. Lu, “Linear discriminant analysis with spectral regularization,” *Applied intelligence*, vol. 40, no. 4, pp. 724–731, 2014.

[24] M. Li and B. Yuan, “2d-lda: A statistical linear discriminant analysis for image matrix,” *Pattern Recognition Letters*, vol. 26, no. 5, pp. 527–532, 2005.

[25] R. P. Duin and M. Loog, “Linear dimensionality reduction via a heteroscedastic extension of lda: the chernoff criterion,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 26, no. 6, pp. 732–739, 2004.

[26] L. Clemmensen, “Sparse discriminant analysis,” *Technometrics*, vol. 53, no. 4, pp. 406–413, 2011.

[27] A. M. Martinez and M. Zhu, “Where are linear feature extraction methods applicable?” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, no. 12, pp. 1934–1944, 2005.

[28] S. C. Johnson, “Hierarchical clustering schemes,” *Psychometrika*, vol. 32, no. 3, pp. 241–254, 1967.

[29] M. Zhu and A. M. Martnez, “Optimal subclass discovery for discriminant analysis,” in *Computer Vision and Pattern Recognition Workshop, 2004. CVPRW ’04. Conference on*, 2004, pp. 97–97.

[30] L. Rokach and O. Maimon, “Clustering methods,” in *Data mining and knowledge discovery handbook*. Springer, 2005, pp. 321–352.

[31] C. Blake and C. J. Merz, “{UCI} repository of machine learning databases,” 1998.

[32] J. Alcalá, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, and F. Herrera, “Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework,” *Journal of Multiple-Valued Logic and Soft Computing*, vol. 17, no. 2-3, pp. 255–287, 2010.

[33] F. S. Samaria and A. C. Harter, “Parameterisation of a stochastic model for human face identification,” in *Applications of Computer Vision, 1994., Proceedings of the Second IEEE Workshop on*. IEEE, 1994, pp. 138–142.

[34] A. M. Martínez and A. C. Kak, “Pca versus lda,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 23, no. 2, pp. 228–233, 2001.

[35] L. Wolf, T. Hassner, and I. Maoz, “Face recognition in unconstrained videos with matched background similarity,” vol. 42, no. 7, pp. 529–534, 2011.

[36] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, “Labeled faces in the wild: A database for studying face recognition in unconstrained environments,” Technical Report 07-49, University of Massachusetts, Amherst, Tech. Rep., 2007.

[37] M. Heikkil, M. Pietikinen, and C. Schmid, “Description of interest regions with center-symmetric local binary patterns,” in *Computer Vision, Graphics and Image Processing, Indian Conference, Icvgip 2006, Madurai, India, December 13-16, 2006, Proceedings*, 2006, pp. 58–69.

[38] L. Wolf, T. Hassner, and Y. Taigman, “Similarity scores based on background samples,” in *Computer Vision–ACCV 2009*. Springer, 2010, pp. 88–97.

[39] Wikipedia, “Dunn index,” https://en.wikipedia.org/wiki/Dunn_index, accessed: 2015-12-21.

[40] J. C. Dunn, “A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters,” *Journal of Cybernetics*, vol. 3, no. 3, pp. 32–57, 1973.

Huan Wan received the bachelor of engineering degree in computer science and technology from Shangrao Normal University, Jiangxi, China, in 2013. She is currently pursuing the M.S. degree in computer application and technology in the School of Mathematics and Computer Science, Fujian Normal University, China. Her current research interests are feature extraction and face verification.

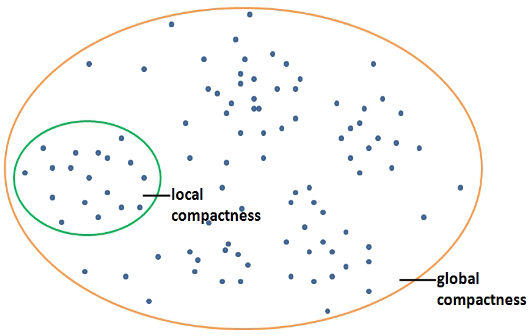
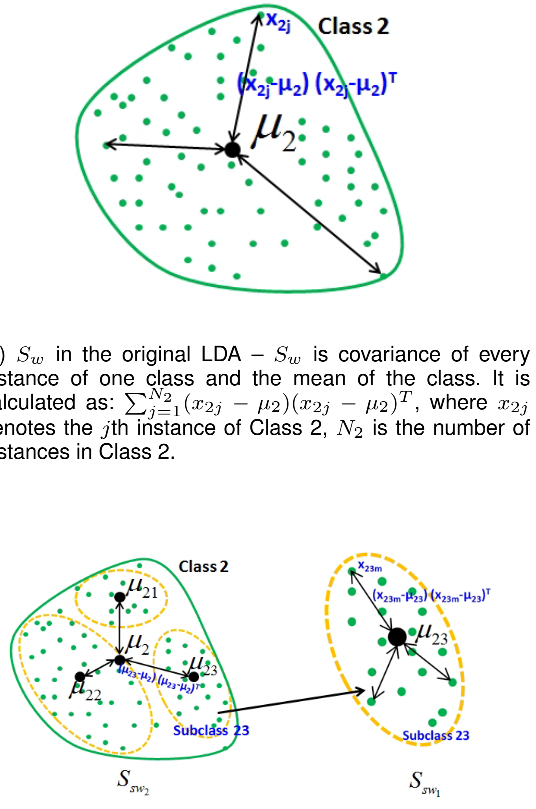


Fig. 1 Illustration of local scatterness and global scatterness of a class of data.



(a) S_w in the original LDA – S_w is covariance of every instance of one class and the mean of the class. It is calculated as: $\sum_{j=1}^{N_2}(x_{2j}-\mu_2)(x_{2j}-\mu_2)^T$, where x_{2j} denotes the jth instance of Class 2, N_2 is the number of instances in Class 2.

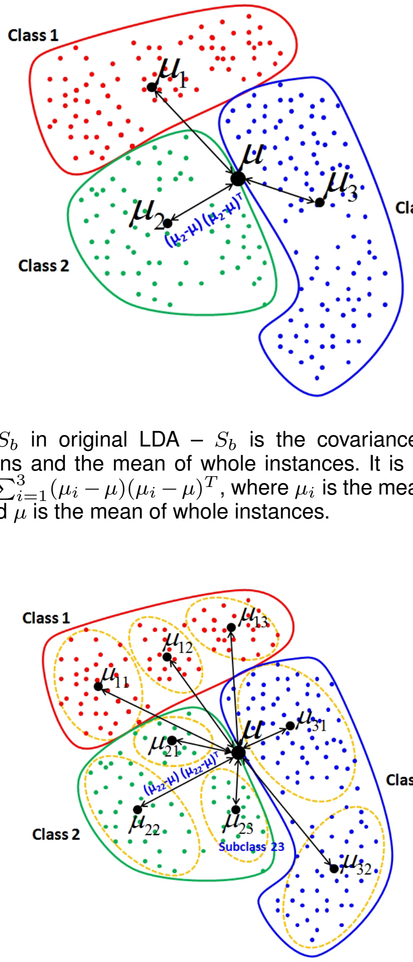
(b) S_{sw} in our SSDA – $S_{sw} = S_{sw_1} + S_{sw_2}$. S_{sw_1} is covariance of subclass instances and corresponding subclass mean, which is calculated as: $\sum_{m=1}^{N_{23}}(x_{23m}-\mu_{23})(x_{23m}-\mu_{23})^T$, where Subclass 23 denotes the 3th subclass in Class 2, x_{23m} denotes the mth instance in Subclass 23 and μ_{23} is the mean of Subclass 23. S_{sw_2} is covariance of subclass means and class mean. It is calculated as: $\sum_{j=1}^{3}(\mu_{2j}-\mu_2)(\mu_{2j}-\mu_2)^T$, where μ_{2j} is the jth subclass mean in Class 2 and μ_2 is the mean of Class 2.

Fig. 2 Illustration of S_w and S_{sw}.

Hui Wang is Professor of Computer Science at Ulster University, and is Visiting Professor at Fujian Normal University. His research interests are machine learning, logics and reasoning, combinatorial data analytics, and their applications in image, video, spectra and text analysis. He has over 200 publications in these areas.

He played a pivotal role in the development of a new algebraic framework for machine learning, Lattice Machine. He proposed the original concept of contextual probability, which can be used for uncertainty reasoning/quantification, probability estimation and machine learning. He also proposed a generic similarity measure, neighbourhood counting, and its specialisations on multivariate data, sequences, tree and graph structures. The contextual probability and neighbourhood counting similarity bear strong similarity to kernel methods, and were independently developed.

He is an associate editor of IEEE Transactions on Cybernetics, and an associate editor of International Journal of Machine Learning and Cybernetics. He is the Chair of IEEE SMCS Northern Ireland Chapter, and a member of IEEE SMCS Board of Governors (2010-2013). He is principal investigator of a number of regional, national and international projects in the areas of image/video analytics (Horizon 2020 funded DESIREE, FP7 funded SAVASA, Royal Society funded VIAD), text analytics (INI funded DEEPFLOW, Royal Society funded BEACON), and intelligent content management (FP5 funded ICONS); and is co-investigator of several other EU funded projects.

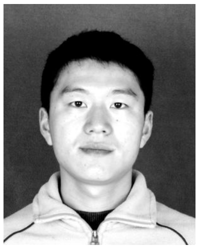


(a) S_b in original LDA – S_b is the covariance of class means and the mean of whole instances. It is calculated as: $\sum_{i=1}^{3}(\mu_i-\mu)(\mu_i-\mu)^T$, where μ_i is the mean of class i and μ is the mean of whole instances.

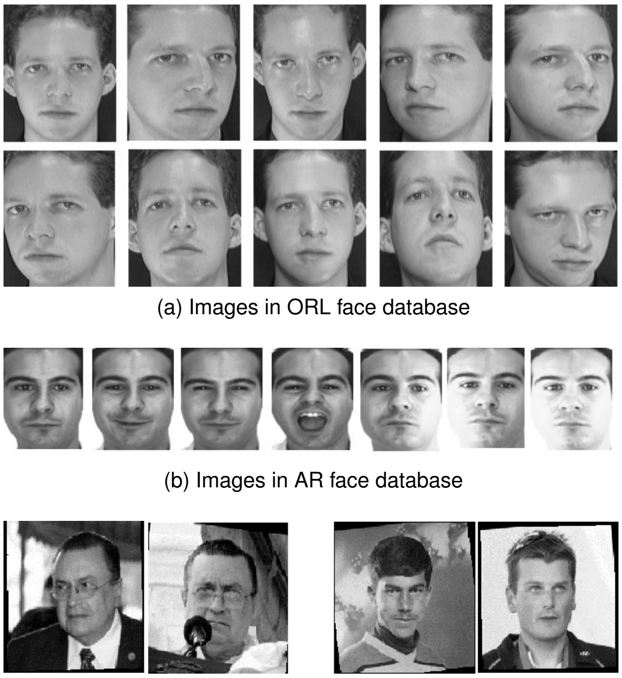
(b) S_{sb} in our SSDA – S_{sb} is covariance of subclass means and the mean of whole instances, which is calculated as: $\sum_{i=1}^{3}\sum_{j=1}^{k_i}(\mu_{ij}-\mu)(\mu_{ij}-\mu)^T$, k_i is the number of sublcasses in class i and μ_{ij} is the mean of subclass j in class i.

Fig. 3 Illustration of S_b and S_{sb}.

Gongde Guo received the bachelor of engineering degree in computer software from Zhejiang University, China in 1985 and PhD degree in computer science from University of Ulster in 2004. He is currently working at School of Mathematics and Computer Science, Fujian Normal Univeristy as a professor. His research interests include data mining and machine learning.



Xin Wei received the bachelor of engineering degree in computer science and technology from Shangrao Normal University, Jiangxi, China, in 2013; and the master degree in computer application and technology from the School of Mathematics and Computer Science, Fujian Normal University, China. He is currently pursuing his PhD at Ulster University, UK. His current research interests are face recognition and image representation.



(a) Images in ORL face database

(b) Images in AR face database

(c) LFW face database: matched face pairs in the left; mismatched face paris in the right

Fig. 4 Sample images from the face databases

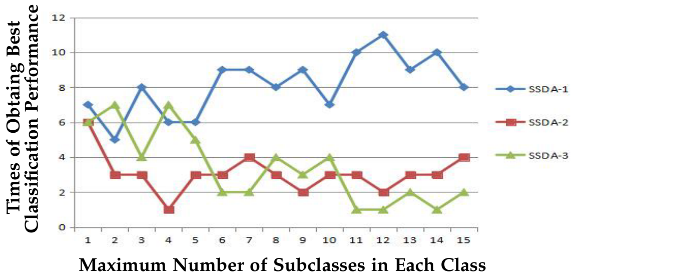
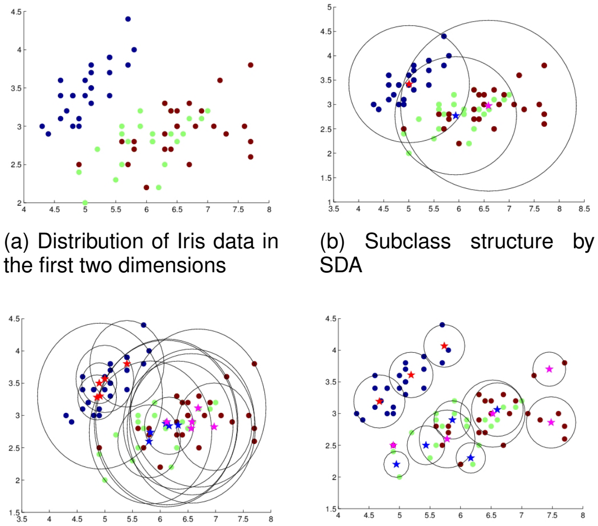


Fig. 5 Number of wins by SSDA-1, SSDA-2 and SSDA-3 for different K_{max} on all 10 UCI data sets and 3 face databases – AR, ORL and LFW.



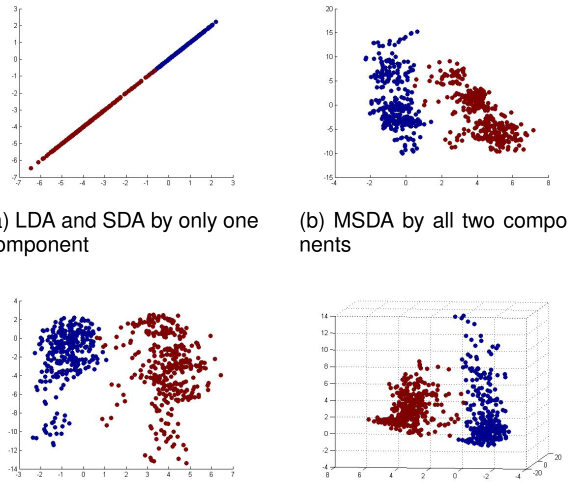
(a) Distribution of Iris data in the first two dimensions

(b) Subclass structure by SDA

(c) Subclass structure by MSDA

(d) Subclass structure by SSDA-1

Fig. 6 Iris data: subclass structures sought after by SDA, MSDA and SSDA-1. Every circle represents one cluster, which is centred at the mean of the cluster and has a radius as the distance between the centre and furtherest point in the cluster.



(a) LDA and SDA by only one component

(b) MSDA by all two components

(c) SSDA-1 by first two components

(d) SSDA-1 by first three components

Fig. 7 Visualisation of data in the first few dimensions of the LDA space