Semi-supervised Discriminant Analysis Via CCCP*

Yu Zhang and Dit-Yan Yeung

Hong Kong University of Science and Technology
{zhangyu,dyyeung}@cse.ust.hk

Abstract. Linear discriminant analysis (LDA) is commonly used for dimensionality reduction. In real-world applications where labeled data are scarce, LDA does not work very well. However, unlabeled data are often available in large quantities. We propose a novel semi-supervised discriminant analysis algorithm called SSDA$_{CCCP}$. We utilize unlabeled data to maximize an optimality criterion of LDA and use the constrained concave-convex procedure to solve the optimization problem. The optimization procedure leads to estimation of the class labels for the unlabeled data. We propose a novel confidence measure for selecting those unlabeled data points with high confidence. The selected unlabeled data can then be used to augment the original labeled data set for performing LDA. We also propose a variant of SSDA$_{CCCP}$, called M-SSDA$_{CCCP}$, which adopts the manifold assumption to utilize the unlabeled data. Extensive experiments on many benchmark data sets demonstrate the effectiveness of our proposed methods.

1 Introduction

Linear discriminant analysis (LDA) [1, 2] is a commonly used method for dimensionality reduction. It seeks a linear projection that simultaneously maximizes the between-class dissimilarity and minimizes the within-class dissimilarity to increase class separability, typically for classification applications. Despite its simplicity, the effectiveness and computational efficiency of LDA make it a popular choice for many applications. Nevertheless, LDA does have its limitations. One of these arises in situations when the sample size is much smaller than the dimensionality of the feature space, leading to the so-called *small sample size* (SSS) problem [3] due to severe undersampling of the underlying data distribution. As a result, the within-class scatter matrix that characterizes the within-class variability is not of full rank and hence it is not invertible. A number of methods have been proposed to overcome this problem, e.g., PseudoLDA [4], PCA+LDA [5], LDA/QR [6], NullLDA [3], and DualLDA [7]. PseudoLDA overcomes the singularity problem by substituting the inverse of the within-class scatter matrix with its pseudo-inverse. PCA+LDA first applies PCA [8] to project the data into a lower-dimensional space so that the within-class scatter matrix computed there is nonsingular, and then applies LDA in the lower-dimensional space. LDA/QR is also a two-stage method which can be divided into two steps: first project the data to the range space of the between-class scatter matrix and then apply LDA in this space. NullLDA first projects the data to the null space of the within-class scatter matrix

* This research has been supported by General Research Fund 621407 from the Research Grants Council of the Hong Kong Special Administrative Region, China.

W. Daelemans et al. (Eds.): ECML PKDD 2008, Part II, LNAI 5212, pp. 644–659, 2008.

and then maximizes the between-class scatter in this space. It is similar to the Discriminative Common Vectors method [9]. DualLDA, which combines the ideas from PCA+LDA and NullLDA, maximizes the between-class scatter matrix in the range space and the null space of the within-class scatter matrix separately and then integrates the two parts together to get the final transformation. There is also another approach to address the SSS problem, with 2DLDA [10] being the representative of this approach. The major difference between 2DLDA and the algorithms above lies in their data representation. Specifically, 2DLDA operates on data represented as (2D) matrices, instead of (1D) vectors, so that the dimensionality of the data representation can be kept small as a way to alleviate the SSS problem. Another limitation of LDA is that it only gives a linear projection of the data points. Fortunately, the kernel approach can be applied easily via the so-called kernel trick to extend LDA to its kernel version, called *kernel discriminant analysis* (KDA), that can project the data points nonlinearly, e.g., [11]. Besides addressing these two limitations of LDA, some interesting recent works also address other issues, e.g., to study the relationships between two variants of LDA [12], to reformulate multi-class LDA as a multivariate linear regression problem [13], and to learn the optimal kernel matrix for KDA using semi-definite programming (SDP) [14, 15].

In many real-world applications, it is impractical to expect the availability of large quantities of labeled data because labeling data requires laborious human effort. On the other hand, unlabeled data are available in large quantities at very low cost. Over the past decade or so, one form of *semi-supervised learning*, which attempts to utilize unlabeled data to aid classification or regression tasks under situations with limited labeled data, has emerged as a hot and promising research topic within the machine learning community. A good survey of semi-supervised learning methods can be found in [16]. Some early semi-supervised learnng methods include Co-Training [17] and transductive SVM (TSVM) [18, 19]. Recently, graph-based semi-supervised learning methods [20, 21, 22] have attracted the interests of many researchers. Unlike earlier methods, these methods model the geometric relationships between all data points in the form of a graph and then propagate the label information from the labeled data points through the graph to the unlabeled data points.

The objective of this paper is to alleviate the SSS problem of LDA by exploiting unlabeled data. We propose a novel *semi-supervised discriminant analysis* algorithm called SSDA$_{CCCP}$. Although there already exists another semi-supervised LDA algorithm, called SDA [23], which exploits the local neighborhood information of data points in performing dimensionality reduction, our SSDA$_{CCCP}$ algorithm works in a very different way. Specifically, we utilize unlabeled data to maximize an optimality criterion of LDA and formulate the problem as a constrained optimization problem that can be solved using the *constrained concave-convex procedure* (CCCP) [24, 25]. This procedure essentially estimates the class labels of the unlabeled data points. For those unlabeled data points whose labels are estimated with sufficiently high confidence based on some novel confidence measure proposed by us, we select them to expand the original labeled data set and then perform LDA again. Besides SSDA$_{CCCP}$, we also propose a variant of SSDA$_{CCCP}$, called M-SSDA$_{CCCP}$, which adopts the *manifold assumption* [20] to utilize the unlabeled data. Note that M-SSDA$_{CCCP}$ shares the spirit of both TSVM and graph-based semi-supervised learning methods.

The remainder of this paper is organized as follows. We first briefly review the traditional LDA algorithm in Section 2. We then present our SSDA$_{CCCP}$ and M-SSDA$_{CCCP}$ algorithms in Section 3. Section 4 reports experimental results based on some commonly used data sets. Performance comparison with some representative methods are reported there to demonstrate the effectiveness of our methods. Finally, some concluding remarks are offered in the last section.

2 Background

We are given a training set of n data points, $\mathcal{D} = \{x_1, \ldots, x_n\}$, where $x_i \in \mathbb{R}^N$, $i = 1, \ldots, n$. Let $\mathcal{D}$ be partitioned into $C \geq 2$ disjoint classes Π_i, $i = 1, \ldots, C$, where class Π_i contains n_i examples. The between-class scatter matrix S_b and the within-class scatter matrix S_w are defined as

$$S_b = \sum_{k=1}^{C} n_k(\bar{m}_k - \bar{m})(\bar{m}_k - \bar{m})^T$$

$$S_w = \sum_{k=1}^{C} \sum_{x_i \in \Pi_k} (x_i - \bar{m}_k)(x_i - \bar{m}_k)^T,$$

where $\bar{m} = (\sum_{i=1}^{n} x_i)/n$ is the sample mean of the whole data set $\mathcal{D}$ and $\bar{m}_k = (\sum_{x_i \in \Pi_k} x_i)/n_k$ is the class mean of Π_k. LDA seeks to find a projection matrix W^* that maximizes the trace function of S_b and S_w:

$$W^* = \arg\max_{W} \operatorname{trace}((W^T S_w W)^{-1} W^T S_b W), \tag{1}$$

which has an analytically tractable solution. According to [26], the optimal solution W^* for the problem (1) can be computed from the eigenvectors of $S_w^{-1} S_b$, where S_w^{-1} denotes the matrix inverse of S_w. Since W^* computed this way is computationally simple yet effective for many applications, the optimality criterion in (1) is often used for many applications. Because the rank of S_b is at most $C - 1$, W contains $C - 1$ columns in most situations.

3 Semi-supervised Discriminant Analysis Via CCCP

In this section, we first present a theoretical result on the optimal solution for LDA. We then show how to utilize unlabeled data to solve the optimization problem, leading to the SSDA$_{CCCP}$ algorithm. Next, we incorporate the manifold assumption into SSDA$_{CCCP}$ to give M-SSDA$_{CCCP}$. Finally we give some discussions about our methods.

3.1 Optimal Solution for LDA

In our work, we use the following optimality criterion:

$$W^* = \arg\max_W \operatorname{trace}((W^T S_t W)^{-1} W^T S_b W), \tag{2}$$

where S_t is the total scatter matrix with $S_t = S_b + S_w$. It is easy to prove that the optimal solution to the problem (2) is equivalent to that to the problem (1).

We assume that S_t is of full rank, or else we can apply principal component analysis (PCA) [8] first to eliminate the null space of S_t without affecting the performance of LDA since the null space makes no contribution to the discrimination ability of LDA [27].

The following theorem on the optimal solution to the problem (2) is relevant here.

Theorem 1. *For* $W \in \mathbb{R}^{N \times (C-1)}$,

$$\max_W \operatorname{trace}((W^T S_t W)^{-1} W^T S_b W) = \operatorname{trace}(S_t^{-1} S_b).$$

The proof of this theorem can be found in [26].

3.2 SSDA$_{CCCP}$: Exploiting Unlabeled Data to Maximize the Optimality Criterion

Suppose we have l labeled data points $x_1, \ldots, x_l \in \mathbb{R}^N$ with class labels from C classes Π_i, $i = 1, \ldots, C$, and m unlabeled data points $x_{l+1}, \ldots, x_{l+m} \in \mathbb{R}^N$ with unknown class labels. So we have totally $n = l + m$ examples available for training. Usually $l \ll m$. When l is too small compared with the input dimensionality, LDA generally does not perform very well. To remedy this problem, we want to incorporate unlabeled data to improve its performance.

Inspired by TSVM [18, 19], which utilizes unlabeled data to maximize the margin, we use unlabeled data here to maximize the optimality criterion of LDA. Since the optimal criterion value is $\operatorname{trace}(S_t^{-1} S_b)$ (from **Theorem 1**), we utilize unlabeled data to maximize $\operatorname{trace}(S_t^{-1} S_b)$ via estimating the class labels of the unlabeled data points.

We first calculate S_t as $S_t = \sum_{i=1}^n (x_i - \bar{m})(x_i - \bar{m})^T$, where $\bar{m} = (\sum_{i=1}^n x_i)/n$ is the sample mean of all the data points. We define the class indicator matrix $A \in \mathbb{R}^{n \times C}$, where the (i, j)th element A_{ij} is given by

$$A_{ij} = \begin{cases} 1 & \text{if } x_i \in \Pi_j \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

If $D = (x_1, \ldots, x_l, x_{l+1}, \ldots, x_n)$ is the data matrix and A_k is a vector for the kth column of A, then the class mean can be expressed as $\bar{m}_k = DA_k/n_k$, where $n_k = A_k^T 1_n$ is the number of data points that belong to the kth class and 1_n is an

n-dimensional column vector of ones. Similarly, we can also express the sample mean as $\bar{m} = D1_n/n$. Then S_b can be calculated as

$$\begin{aligned} S_b &= \sum_{k=1}^{C} n_k(\bar{m}_k - \bar{m})(\bar{m}_k - \bar{m})^T \\ &= \sum_{k=1}^{C} n_k D \left(\frac{A_k}{n_k} - \frac{1_n}{n}\right)\left(\frac{A_k^T}{n_k} - \frac{1_n^T}{n}\right) D^T \\ &= D\left[\sum_{k=1}^{C} n_k \left(\frac{A_k}{n_k} - \frac{1_n}{n}\right)\left(\frac{A_k^T}{n_k} - \frac{1_n^T}{n}\right)\right] D^T. \end{aligned}$$

So $\mathrm{trace}(S_t^{-1}S_b)$ can be calculated as

$$\begin{aligned} \mathrm{trace}(S_t^{-1}S_b) &= \mathrm{trace}\left(S_t^{-1} D\left[\sum_{k=1}^{C} n_k \left(\frac{A_k}{n_k} - \frac{1_n}{n}\right)\left(\frac{A_k^T}{n_k} - \frac{1_n^T}{n}\right)\right] D^T\right) \\ &= \mathrm{trace}\left(\left[\sum_{k=1}^{C} n_k \left(\frac{A_k}{n_k} - \frac{1_n}{n}\right)\left(\frac{A_k^T}{n_k} - \frac{1_n^T}{n}\right)\right] D^T S_t^{-1} D\right) \\ &= \mathrm{trace}\left(\sum_{k=1}^{C} n_k \left(\frac{A_k^T}{n_k} - \frac{1_n^T}{n}\right) S \left(\frac{A_k}{n_k} - \frac{1_n}{n}\right)\right) \\ &= \sum_{k=1}^{C} \frac{1}{n_k}\left(A_k^T - \frac{n_k}{n}1_n^T\right) S \left(A_k - \frac{n_k}{n}1_n\right), \end{aligned}$$

where $S = D^T S_t^{-1} D$ is a positive semi-definite matrix.

Since those entries in A for the unlabeled data points are unknown, we maximize $\mathrm{trace}(S_t^{-1}S_b)$ with respect to A. By defining some new variables for the sake of notational simplicity, we formulate the optimization problem as:

$$\begin{aligned} \max_{A,B_k,t_k}\ & \sum_{k=1}^{C} \frac{B_k^T S B_k}{t_k} \\ s.t.\ & t_k = A_k^T 1_n,\ k = 1,\ldots,C \\ & B_k = A_k - \frac{t_k}{n}1_n,\ k = 1,\ldots,C \\ & A_{ij} = \begin{cases} 1 \text{ if } x_i \in \Pi_j \\ 0 \text{ otherwise} \end{cases} i = 1,\ldots,l \\ & A_{ij} \in \{0,1\},\ i = l{+}1,\ldots,n,\ j = 1,\ldots,C \\ & \sum_{j=1}^{C} A_{ij} = 1,\ i = l{+}1,\ldots,n. \end{aligned} \tag{4}$$

Unfortunately this is an integer programming problem which is known to be NP-hard and often has no efficient solution. We seek to make this integer programming problem tractable by relaxing the constraint $A_{ij} \in \{0,1\}$ in (4) to $A_{ij} \geq 0$, giving rise to a modified formulation of the optimization problem:

$$\begin{aligned}
\max_{A,B_k,t_k} \quad & \sum_{k=1}^{C} \frac{B_k^T S B_k}{t_k} \\
s.t. \quad & t_k = A_k^T 1_n, \; k = 1, \ldots, C \\
& B_k = A_k - \frac{t_k}{n} 1_n, \; k = 1, \ldots, C \\
& A_{ij} = \begin{cases} 1 \text{ if } x_i \in \mathit{\Pi}_j \\ 0 \text{ otherwise} \end{cases} \; i = 1, \ldots, l \\
& A_{ij} \geq 0, \; i = l{+}1, \ldots, n, \; j = 1, \ldots, C \\
& \sum_{j=1}^{C} A_{ij} = 1, \; i = l{+}1, \ldots, n.
\end{aligned} \tag{5}$$

With such relaxation, the matrix entries of A for the unlabeled data points may be interpreted as posterior class probabilities. However, even though the constraints in the optimization problem (5) are linear, the problem seeks to maximize a convex function which, unfortunately, does not correspond to a convex optimization problem [28]. If we re-express the optimization problem in (5) as minimizing a concave function, we can adopt the *constrained concave-convex procedure* (CCCP) [24, 25] to solve this non-convex optimization problem. For our case, the convex part of the objective function degenerates to the special case of a constant function which always returns zero.

CCCP is an iterative algorithm. In each iteration, the concave part of the objective function for the optimization problem is replaced by its first-order Taylor series approximation at the point which corresponds to the result obtained in the previous iteration. Specifically, in the $(p{+}1)$th iteration, we solve the following optimization problem:

$$\begin{aligned}
\max_{A,B_k,t_k} \quad & \sum_{k=1}^{C} \left(\frac{2(B_k^{(p)})^T S}{t_k^{(p)}} B_k - \frac{(B_k^{(p)})^T S B_k^{(p)}}{(t_k^{(p)})^2} t_k \right) \\
s.t. \quad & t_k = A_k^T 1_n, \; k = 1, \ldots, C \\
& B_k = A_k - \frac{t_k}{n} 1_n, \; k = 1, \ldots, C \\
& A_{ij} = \begin{cases} 1 \text{ if } x_i \in \mathit{\Pi}_j \\ 0 \text{ otherwise} \end{cases} \; i = 1, \ldots, l \\
& A_{ij} \geq 0, \; i = l{+}1, \ldots, n, \; j = 1, \ldots, C \\
& \sum_{j=1}^{C} A_{ij} = 1, \; i = l{+}1, \ldots, n,
\end{aligned} \tag{6}$$

where $B_k^{(p)}, t_k^{(p)}$, $k = 1, \ldots, C$ were obtained in the pth iteration. The objective function in (6) is just the first-order Taylor series approximation of that in (5) by ignoring some constant terms.

Since the optimization problem (6) is a linear programming (LP) problem, it can be solved efficiently and hence can handle large-scale applications. Because the optimal solution of an LP problem falls on the boundary of its feasible set (or called constraint set), the matrix entries of the optimal A_{ij} computed in each iteration must be in $\{0, 1\}$, which automatically satisfies the constraints in (4).

As the optimization problem is non-convex, the final solution that CCCP obtains generally depends on its initial value. For the labeled data points, the corresponding entries in A_{ij} are held fixed based on their class labels. For the unlabeled data points, we initialize the corresponding entries in A_{ij} with equal prior probabilities for all classes:

$$\begin{aligned} A_{ij}^{(0)} &= \begin{cases} 1 & \text{if } x_i \in \Pi_j \\ 0 & \text{otherwise} \end{cases} \quad i = 1, \ldots, l,\ j = 1, \ldots, C \\ A_{ij}^{(0)} &= \frac{1}{C},\ \ i = l{+}1, \ldots, n,\ j = 1, \ldots, C. \end{aligned} \tag{7}$$

The initial values for $B_k^{(0)}$ and $t_k^{(0)}$ can be computed based on the equality constraints in (6) which establish the relationships between A, B_k and t_k.

3.3 M-SSDA$_{CCCP}$: Incorporating the Manifold Assumption

The manifold assumption [20] is adopted by many graph-based semi-supervised learning methods. Under this assumption, nearby points are more likely to have the same class label for classification problems and similar low-dimensional representations for dimensionality reduction problems. We adopt this assumption to extend SSDA$_{CCCP}$ to M-SSDA$_{CCCP}$.

Given the data set $\mathcal{D} = \{x_1, \ldots, x_n\}$, we first construct a K-nearest neighbor graph $G = (V, E)$, with the vertex set $V = \{1, \ldots, n\}$ corresponding to the labeled and unlabeled data points and the edge set $E \subseteq V \times V$ representing the relationships between data points. Each edge is assigned a weight w_{ij} which reflects the similarity between points x_i and x_j:

$$w_{ij} = \begin{cases} \exp\left(-\frac{\|x_i - x_j\|^2}{\sigma_i \sigma_j}\right) & \text{if } x_i \in N_K(x_j) \text{ or } x_j \in N_K(x_i) \\ 0 & \text{otherwise} \end{cases}$$

where $N_K(x_i)$ denotes the neighborhood set of K-nearest neighbors of x_i, σ_i the distance between x_i and its Kth nearest neighbor, and σ_j the distance between x_j and its Kth nearest neighbor. This way of constructing the nearest neighbor graph is called *local scaling* [29], which is different from that in SDA [23]. In SDA, a constant value of 1 is set for all neighbors. This is unsatisfactory especially when some neighbors are relatively far away.

By incorporating the manifold assumption into our problem, we expect nearby points to be more likely to have the same class label and hence the two corresponding rows in A are more likely to be the same. We thus modify the optimization problem (5) by adding one more term to the objective function:

$$
\begin{aligned}
\max_{A,B_k,t_k}\ & \sum_{k=1}^{C}\frac{B_k^T S B_k}{t_k}-\lambda\sum_{i=1}^{n}\sum_{j=i+1}^{n}w_{ij}\|A(i)-A(j)\|_1\\
s.t.\ & t_k=A_k^T 1_n,\ k=1,\ldots,C\\
& B_k=A_k-\frac{t_k}{n}1_n,\ k=1,\ldots,C\\
& A_{ij}=\begin{cases}1 \text{ if } x_i\in\Pi_j\\ 0 \text{ otherwise}\end{cases}\ i=1,\ldots,l\\
& A_{ij}\geq 0,\ i=l{+}1,\ldots,n,\ j=1,\ldots,C\\
& \sum_{j=1}^{C}A_{ij}=1,\ i=l{+}1,\ldots,n,
\end{aligned}
\tag{8}
$$

where $\lambda > 0$ is a regularization parameter, $A(i)$ denotes the ith row of A, and $\|x\|_1$ is the L_1-norm of vector x.

Since the objective function of the optimization problem (8) is the difference of two convex functions, we can also adopt CCCP to solve it. Similar to SSDA$_{CCCP}$, in each iteration of CCCP, we also need to solve an LP problem:

$$
\begin{aligned}
\max_{A,B_k,t_k}\ & \sum_{k=1}^{C}\left(\frac{2(B_k^{(p)})^T S}{t_k^{(p)}}B_k-\frac{(B_k^{(p)})^T S B_k^{(p)}}{(t_k^{(p)})^2}t_k\right)-\lambda\sum_{i=1}^{n}\sum_{j=i+1}^{n}w_{ij}\|A(i)-A(j)\|_1\\
s.t.\ & t_k=A_k^T 1_n,\ k=1,\ldots,C\\
& B_k=A_k-\frac{t_k}{n}1_n,\ k=1,\ldots,C\\
& A_{ij}=\begin{cases}1 \text{ if } x_i\in\Pi_j\\ 0 \text{ otherwise}\end{cases}\ i=1,\ldots,l\\
& A_{ij}\geq 0,\ i=l{+}1,\ldots,n,\ j=1,\ldots,C\\
& \sum_{j=1}^{C}A_{ij}=1,\ i=l{+}1,\ldots,n.
\end{aligned}
\tag{9}
$$

One reason for choosing the L_1-norm in the problem (8) is to keep the problem (9) as an LP problem which has an efficient and effective solution.

3.4 Augmenting the Labeled Data Set with Unlabeled Data

For both SSDA$_{CCCP}$ and M-SSDA$_{CCCP}$, CCCP estimates the class labels of all the unlabeled data points by solving the corresponding optimization problems with respect to A. One might then use all these unlabeled data points with estimated class labels to expand the labeled data set and then apply LDA again. However, it should be noted that not all the class labels can be estimated accurately. Thus, including those points with noisy class labels may impair the performance of LDA. Here we propose an effective method for selecting only those unlabeled data points whose labels are estimated with sufficiently high confidence.

Since all matrix entries in A_{ij} obtained by CCCP are either 0 or 1, they cannot serve as posterior class probabilities for defining a measure to characterize the label estimation confidence. Here we propose an alternative scheme. We first use all the unlabeled data points with their estimated labels and the original labeled data set to perform LDA. Then, in the embedding space, we consider the neighborhood of each unlabeled data point by taking into account unlabeled data points only. If an unlabeled point has a sufficiently large proportion (determined by some threshold θ, usually chosen to be larger than 0.5) of neighboring unlabeled points with the same estimated class label as its own, we consider this unlabeled point to have an estimated class label with high confidence and hence select it to augment the labeled data set for performing LDA again.

3.5 Discussions

In order to gain some insight into our method, we investigate the dual form of the optimization problem (6). We denote $R_k^{(p)} = \frac{2(B_k^{(p)})^T S}{t_k^{(p)}}$ and $q_k^{(p)} = \frac{(B_k^{(p)})^T S B_k^{(p)}}{(t_k^{(p)})^2}$, for $k = 1, \ldots, C$. We plug the first two equality constraints of the optimization problem (6) into its objective funciton and get the following Lagrangian:

$$L(A, \alpha, \beta) = \sum_{k=1}^{C} \left[\left(q_k^{(p)} + \frac{R_k^{(p)} 1_n}{n} \right) 1_n^T - R_k^{(p)} \right] A_k - \sum_{k=1}^{C} \sum_{i=1}^{l} \alpha_{ki} (A_{ik} - \delta_k^{c(i)})$$
$$- \sum_{k=1}^{C} \sum_{i=l+1}^{n} \alpha_{ki} A_{ik} - \sum_{i=l+1}^{n} \beta_i (\sum_{k=1}^{C} A_{ik} - 1),$$

where $c(i)$ is the class label of labeled data point i and $\delta_k^{c(i)}$ is the delta function whose value is 1 if $c(i) = k$ and 0 otherwise.

So the dual form of the optimization problem (6) is

$$\begin{aligned} \max_{\alpha,\beta} \ & \sum_{k=1}^{C} \sum_{i=1}^{l} \alpha_{ki} \delta_k^{c(i)} + \sum_{i=l+1}^{n} \beta_i \\ s.t. \ & \alpha_{ki} = q_k^{(p)} - R_{ki}^{(p)} + \frac{R_k^{(p)} 1_n}{n}, \ i = 1, \ldots, l, \ k = 1, \ldots, C \\ & \alpha_{ki} + \beta_i = q_k^{(p)} - R_{ki}^{(p)} + \frac{R_k^{(p)} 1_n}{n}, \ i = l+1, \ldots, n, \ k = 1, \ldots, C \\ & \alpha_{ki} \geq 0, \ i = l+1, \ldots, n, \ k = 1, \ldots, C \end{aligned} \tag{10}$$

where $R_{ki}^{(p)}$ is the ith element of vector $R_k^{(p)}$.

The *Karush-Kuhn-Tucker* (KKT) condition [28] for the optimization problem (10) is

$$\alpha_{ki} A_{ik} = 0, \ i = l+1, \ldots, n, \ k = 1, \ldots, C. \tag{11}$$

From the first constraint of the optimization problem (10), we can see that each α_{ki} has a constant value for $i = 1, \ldots, l, \ k = 1, \ldots, C$. So we can simplify the optimization problem (10) by eliminating the first summation term in the objective function and the first constraint as

$$\begin{aligned}
&\max_{\alpha,\beta} \sum_{i=l+1}^{n} \beta_i \\
&s.t.\ \ \alpha_{ki} + \beta_i = q_k^{(p)} - R_{ki}^{(p)} + \frac{R_k^{(p)} 1_n}{n},\ i = l{+}1, \ldots, n,\ k = 1, \ldots, C \\
&\qquad \alpha_{ki} \geq 0,\ i = l{+}1, \ldots, n,\ k = 1, \ldots, C,
\end{aligned} \tag{12}$$

which can be further simplified as

$$\begin{aligned}
&\max_{\beta} \sum_{i=l+1}^{n} \beta_i \\
&s.t.\ \ \beta_i \leq q_k^{(p)} - R_{ki}^{(p)} + \frac{R_k^{(p)} 1_n}{n},\ i = l{+}1, \ldots, n,\ k = 1, \ldots, C.
\end{aligned} \tag{13}$$

So the optimal solution of β_i can be obtained as $\beta_i = \min_k \{q_k^{(p)} - R_{ki}^{(p)} + \frac{R_k^{(p)} 1_n}{n}\}$ for $i = l + 1, \ldots, n$.

For each unlabeled data point, if we assume $A_{ik^\star} > 0$, then from the KKT condition (11) we can get $\alpha_{k^\star i} = 0$ and also $\beta_i = q_{k^\star}^{(p)} - R_{k^\star i}^{(p)} + \frac{R_{k^\star}^{(p)} 1_n}{n}$ according to the first constraint of the optimization problem (12). So

$$q_{k^\star}^{(p)} - R_{k^\star i}^{(p)} + \frac{R_{k^\star}^{(p)} 1_n}{n} = \min_k \left\{ q_k^{(p)} - R_{ki}^{(p)} + \frac{R_k^{(p)} 1_n}{n} \right\}$$

and

$$k^\star = \arg\min_k \left\{ q_k^{(p)} - R_{ki}^{(p)} + \frac{R_k^{(p)} 1_n}{n} \right\}.$$

So $q_k^{(p)} - R_{ki}^{(p)} + \frac{R_k^{(p)} 1_n}{n}$ can be seen as the negative confidence that the ith data point belongs to the kth class and hence we can classify each data point to the class corresponding to the minimal negative confidence. If there is a unique minimum, then we can get $A_{ik^\star} = 1$ and $A_{ik'} = 0$ for $k' \neq k^\star$; otherwise, we can first find the set of unlabeled data points for which there exist unique minimum and A_{ik} can be easily determined, and then we can solve a smaller LP problem (6) by plugging in the known elements A_{ij}. From our experiments, the latter situation seldom occurs and this can speed up the optimization problem (6), which even does not need to solve a LP problem.

[30] proposed a novel clustering method called DisKmeans which also maximize the optimality criterion of LDA to do clustering. However, its purpose is very different. In our work, M-SSDA$_{CCCP}$ and SSDA$_{CCCP}$ utilize unlabeled data to alleviate the SSS problem of LDA and we formulate the learning problem under the semi-supervised setting. On the other hand, DisKmeans aims at clustering high-dimensional data which is an unsupervised learning problem.

The computation cost of SSDA$_{CCCP}$ and M-SSDA$_{CCCP}$ includes performing LDA twice and solving the optimization problem using CCCP. The complexity of LDA

Table 1. Algorithm for SSDA$_{CCCP}$ or M-SSDA$_{CCCP}$

```
Input: labeled data $x_i$ $(i = 1, \ldots, l)$, unlabeled data $x_i$ $(i = l{+}1, \ldots, n)$, $K$, $\theta$, $\varepsilon$
Initialize $A^{(0)}$ using Eq. (7);
Initialize $B_k^{(0)}$ and $t_k^{(0)}$ based on $A^{(0)}$ for $k = 1, \ldots, C$;
Construct the $K$-nearest neighbor graph;
$p = 0$;
Repeat
    $p = p + 1$;
    Solve the optimization problem (6) or (9);
    Update $A^{(p)}$, $B_k^{(p)}$ and $t_k^{(p)}$ using the result of the optimization problem for $k = 1, \ldots, C$;
Until$\|A^{(p)} - A^{(p-1)}\|_F \leq \varepsilon$
Select the unlabeled data points with high confidence based on the threshold $\theta$;
Add the selected unlabeled data points with their estimated labels into the labeled data set
and perform LDA on the augmented labeled data set to get the transformation $W$.
Output: the transformation $W$
```

is $O(N^3)$. The LP problem inside each iteration of CCCP can be solved efficiently. From our experimental results, CCCP converges very fast in less than 10 iterations. So SSDA$_{CCCP}$ and M-SSDA$_{CCCP}$ are efficient under most situations.

Finally, we summary this section by presenting the SSDA$_{CCCP}$ (or M-SSDA$_{CCCP}$) algorithm in Table 1.

4 Experiments

In this section, we first study SSDA$_{CCCP}$ and M-SSDA$_{CCCP}$ empirically and compare their performance with several other dimensionality reduction methods, including PCA, LDA [5] and SDA. Note that PCA is unsupervised, LDA is supervised, and SDA is semi-supervised in nature. After dimensionality reduction has been performed, we apply a simple nearest-neighbor classifier to perform classification in the embedding space. We also compare SSDA$_{CCCP}$ and M-SSDA$_{CCCP}$ with two state-of-the-art inductive semi-supervised learning methods, LapSVM and LapRLS [20].

4.1 Experimental Setup

We use MATLAB to implement all the algorithms and the CVX toolbox[1] for solving the optimization problems. We use the source code offered by Belkin et al. for LapSVM and LapRLS.[2] We evaluate these algorithms on 11 benchmark data sets, including 8 UCI data sets [31], a brain-computer interface dataset BCI[3] and two image data sets: COIL[3] and PIE [32]. See Table 2 for more details.

[1] http://www.stanford.edu/~boyd/cvx/

[2] http://manifold.cs.uchicago.edu/manifold_regularization/manifold.html

[3] http://www.kyb.tuebingen.mpg.de/ssl-book/

Table 2. Summary of data sets used and data partitioning for each data set

Data set	#Dim (N)	#Inst (n)	#Class (C)	#Labeled (q)	#Unlabeled (r)
diabetes	8	768	2	5	100
heart-statlog	13	270	2	5	100
ionosphere	34	351	2	5	50
hayes-roth	4	160	3	3	20
iris	4	150	3	3	20
mfeat-pixel	240	2000	10	5	50
pendigits	16	10992	10	5	95
vehicle	18	864	4	5	100
BCI	117	400	2	5	50
COIL	241	1500	6	5	100
PIE	1024	1470	30	2	20

For each data set, we randomly select q data points from each class as labeled data and r points from each class as unlabeled data. The remaining data form the test set. Table 2 shows the data partitioning for each data set. For each partitioning, we perform 20 random splits and report the mean and standard derivation over the 20 trials. For M-SSDA$_{CCCP}$, we choose the number of nearest neighbors K for constructing the K-nearest neighbor graph to be the same as that for SDA, LapSVM, and LapRLS.

4.2 Experimental Results

We first compare our methods with dimensionality reduction methods and the experimental results are listed in Table 3. There are two rows for each data set: the upper one being the classification error on the unlabeled training data and the lower one being that on the test data. For each data set, the lowest classification error is shown in boldface. From the results, we can see that the performance of SSDA$_{CCCP}$ or M-SSDA$_{CCCP}$ is better than other methods in most situations. For DIABETES, HEART-STATLOG, PENDIGITS, VEHICLE and PIE, the improvement is very significant. Moreover, for the data sets such as DIABETES and HEART-STATLOG which may not contain manifold structure, the performance of SSDA$_{CCCP}$ is better than M-SSDA$_{CCCP}$. For MFEAT-PIXEL, PIE and others which may contain manifold structure, the performance of M-SSDA$_{CCCP}$ is better than SSDA$_{CCCP}$. Thus for data sets such as images which may have manifold structure, we recommend to use M-SSDA$_{CCCP}$. Otherwise SSDA$_{CCCP}$ is preferred. Compared with SDA, SSDA$_{CCCP}$ and M-SSDA$_{CCCP}$ are more stable. Specifically, the performance of SSDA$_{CCCP}$ or M-SSDA$_{CCCP}$ is comparable to or better than that of LDA in most situations. For SDA, however, the performance degradation can sometimes be very severe, especially for MFEAT-PIXEL and PIE.

We also investigate the selection method described in Section 3.4. We record the mean accuracy of label estimation for the unlabeled data over 20 trials before and after applying the selection method. The results in Table 4 show that the estimation accuracy after applying the selection method is almost always higher, sometimes very significantly. This confirms that our selection method for unlabeled data is very effective.

Table 3. Average classification errors for each method on each data set. Each number inside brackets shows the corresponding standard derivation. The upper row for each data set is the classification error on the unlabeled training data and the lower row is that on the test data.

Data set	PCA	LDA	SDA	SSDA$_{CCCP}$	M-SSDA$_{CCCP}$
diabetes	0.4335(0.0775)	0.4438(0.0878)	0.4022(0.0638)	**0.3898(0.0674)**	0.4360(0.0605)
	0.4253(0.1154)	0.4311(0.0997)	0.3763(0.0864)	**0.3276(0.0643)**	0.4125(0.1074)
heart-statlog	0.4288(0.0689)	0.3978(0.0582)	0.3680(0.0564)	**0.3293(0.0976)**	0.3818(0.0662)
	0.3975(0.0669)	0.3767(0.1055)	0.3783(0.1076)	**0.3133(0.1174)**	0.3258(0.1493)
ionosphere	0.2895(0.1032)	0.2850(0.0876)	**0.2695(0.1056)**	0.2860(0.1015)	0.2830(0.1029)
	0.2189(0.0632)	0.2365(0.0972)	0.2241(0.0863)	0.2351(0.1032)	0.2399(0.1278)
hayes-roth	0.5175(0.0571)	0.4942(0.0531)	0.5058(0.0661)	0.4867(0.0569)	**0.4758(0.0586)**
	0.5115(0.0605)	0.5165(0.0690)	0.5077(0.0752)	0.5121(0.0770)	**0.5060(0.0627)**
iris	0.0917(0.0417)	0.0933(0.0613)	0.0825(0.0506)	0.0708(0.0445)	**0.0667(0.0493)**
	0.0907(0.0333)	0.0833(0.0586)	0.0809(0.0395)	0.0611(0.0370)	**0.0611(0.0454)**
mfeat-pixel	0.1450(0.0232)	0.1501(0.0290)	0.2783(0.0435)	0.1501(0.0289)	**0.1367(0.0210)**
	0.1429(0.0228)	0.1486(0.0264)	0.3428(0.0298)	0.1485(0.0264)	**0.1329(0.0213)**
pendigits	0.1724(0.0305)	0.2238(0.0364)	0.2547(0.0447)	0.1785(0.0266)	**0.1617(0.0242)**
	0.1761(0.0276)	0.2192(0.0332)	0.2544(0.0382)	0.1779(0.0190)	**0.1650(0.0225)**
vehicle	0.5739(0.0375)	0.5741(0.0365)	0.5400(0.0402)	**0.4396(0.0734)**	0.4838(0.0901)
	0.5808(0.0453)	0.5879(0.0429)	0.5462(0.0312)	**0.4329(0.0672)**	0.4739(0.0791)
BCI	0.4835(0.0460)	0.4830(0.0557)	0.4960(0.0476)	**0.4750(0.0432)**	0.4975(0.0484)
	0.5000(0.0324)	0.4803(0.0249)	0.4812(0.0326)	**0.4732(0.0331)**	0.4741(0.0346)
COIL	**0.4443(0.0418)**	0.5247(0.0371)	0.5419(0.0607)	0.5236(0.0374)	0.5193(0.0401)
	0.4391(0.0364)	0.5194(0.0421)	0.5461(0.04821)	0.5178(0.0434)	0.5096(0.0398)
PIE	0.6156(0.0275)	0.5055(0.1624)	0.7629(0.0377)	0.4674(0.1757)	**0.2381(0.0552)**
	0.6207(0.0251)	0.5126(0.1512)	0.8277(0.0208)	0.4777(0.1696)	**0.2424(0.0592)**

Table 4. Accuracy of label estimation for the unlabeled data before and after applying the selection method

	SSDA$_{CCCP}$ (%)		M-SSDA$_{CCCP}$ (%)	
Data set	Before	After	Before	After
diabetes	64.03	66.67	54.10	51.20
heart-statlog	72.27	72.62	55.25	66.70
ionosphere	69.05	87.51	74.10	82.07
hayes-roth	46.75	52.73	42.00	42.64
iris	75.42	93.39	91.42	95.06
mfeat-pixel	32.49	100.0	94.21	98.91
pendigits	75.31	86.08	88.92	94.02
vehicle	56.30	69.88	44.80	52.26
BCI	50.75	65.42	49.00	49.15
COIL	33.57	96.07	42.64	60.03
PIE	30.48	85.00	52.64	70.41

Table 5. Average classification errors for each method on each data set. Each number inside brackets shows the corresponding standard derivation. The upper row for each data set is the classification error on the unlabeled training data and the lower row is that on the test data.

Data set	LapSVM	LapRLS	SSDA$_{CCCP}$	M-SSDA$_{CCCP}$
diabetes	0.4763(0.0586)	0.4523(0.0650)	**0.3620(0.0680)**	0.4015(0.0893)
	0.5643(0.0684)	0.5009(0.0775)	**0.3488(0.0514)**	0.4234(0.1107)
heart-statlog	0.3478(0.1059)	0.3348(0.1070)	**0.3108(0.0901)**	0.3758(0.0914)
	0.3517(0.1458)	0.3375(0.1366)	**0.3091(0.0989)**	0.3442(0.1226)
ionosphere	0.3525(0.0539)	0.3260(0.0527)	0.3340(0.0902)	**0.3185(0.0719)**
	0.2245(0.0697)	0.2266(0.0732)	0.2705(0.0969)	0.2905(0.0933)
hayes-roth	0.6633(0.0149)	0.6608(0.0261)	**0.4833(0.0824)**	0.5225(0.0466)
	0.5550(0.0737)	0.5500(0.0516)	**0.4901(0.0705)**	0.5104(0.0711)
iris	0.3175(0.1390)	0.2708(0.1474)	0.0650(0.0516)	**0.0525(0.0437)**
	0.3049(0.1426)	0.2741(0.1473)	0.0772(0.0508)	**0.0593(0.0379)**
mfeat-pixel	0.1488(0.0236)	**0.1359(0.0257)**	0.1578(0.0268)	0.1420(0.0249)
	0.2252(0.0187)	0.2075(0.0181)	0.1555(0.0263)	**0.1427(0.0183)**
pendigits	0.2571(0.0379)	0.2368(0.0312)	0.1856(0.0226)	**0.1697(0.0245)**
	0.2539(0.0334)	0.2377(0.0283)	0.1866(0.0244)	**0.1735(0.0217)**
vehicle	0.4713(0.0449)	0.4921(0.0460)	**0.4219(0.0623)**	0.4645(0.0770)
	0.4758(0.0477)	0.5007(0.0452)	**0.4181(0.0600)**	0.4641(0.0777)
BCI	0.4805(0.0551)	0.4695(0.0612)	**0.4515(0.0543)**	0.4665(0.0479)
	0.4631(0.0456)	**0.4562(0.0390)**	0.4752(0.0362)	0.4864(0.0372)
COIL	0.5414(0.0496)	0.5855(0.0617)	**0.5028(0.0576)**	0.5030(0.0488)
	0.5421(0.0497)	0.5864(0.0598)	**0.5057(0.0533)**	0.5062(0.0423)
PIE	0.2561(0.0311)	0.3405(0.0227)	0.4096(0.1600)	**0.2497(0.0313)**
	0.2671(0.0235)	0.3523(0.0151)	0.4160(0.1575)	**0.2556(0.0235)**

Next we compare our methods with some representative semi-supervised learning methods. The experimental settings are the same as those in the first experiment. There are many popular semi-supervised learning methods, such as Co-Training [17], TSVM [18, 19], methods in [21, 22], LapSVM and LapRLS [20]. Co-Training requires two independent and sufficient views for the data, but data used in our experiment can not satisfy this requirement. TSVM has high computation cost and hence cannot be used for large-scale problems. Thus it is not included in our experiment. The methods in [21,22] can only work under the transductive setting, in which the test data, in addition to the training data, must be available during model training and the learned model cannot be applied to unseen test data easily. So these methods can not satisfy our experimental settings and are excluded in our experiments. LapSVM and LapRLS, which also adopt the manifold assumption, have efficient solutions and can work under the inductive setting. So we have included them in our experiment for performance comparison. The standard LapSVM and LapRLS algorithms are for two-class problems. For multi-class problems, we adopt the *one vs. rest* strategy as in [20] for LapSVM and LapRLS. Since the methods used here are all linear methods, we use a linear kernel for LapSVM and LapRLS. The experimental results are shown in Table 5. From the experimental results, we can see that the performance of SSDA$_{CCCP}$ and M-SSDA$_{CCCP}$ is comparable to or even better than that of LapSVM and LapRLS. Moreover, One advantage of SSDA$_{CCCP}$

and M-SSDA$_{CCCP}$ is that their formulation and optimization procedure are the same for two-class and multi-class problems. However, this is not the case for LapSVM and LapRLS which require training the models multiple times for multi-class problems.

5 Conclusion

In this paper, we have presented a new approach for semi-supervised discriminant analysis. By making use of both labeled and unlabeled data in learning a transformation for dimensionality reduction, this approach overcomes a serious limitation of LDA under situations where labeled data are limited. In our future work, we will investigate kernel extensions to our proposed methods in dealing with nonlinearity. Moreover, we will also apply the ideas here to some other dimensionality reduction methods.

References

1. Fisher, R.A.: The use of multiple measurements in taxonomic problems. Annals of Eugenics 7, 179–188 (1936)
2. Rao, C.R.: The utilization of multiple measurements in problems of biological classification. Journal of the Royal Statictical Society 10, 159–203 (1948)
3. Chen, L., Liao, H., Ko, M., Lin, J., Yu, G.: A new LDA-based face recognition system which can solve the small sample size problem. Pattern Recognition 33(10), 1713–1726 (2000)
4. Krzanowski, W.J., Jonathan, P., McCarthy, W.V., Thomas, M.R.: Discriminant analysis with singular covariance matrices: methods and applications to spectroscopic data. Applied Statistics 44(1), 101–115 (1995)
5. Belhumeur, P.N., Hespanha, J.P., Kriegman, D.J.: Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. IEEE Transactions on Pattern Analysis and Machine Intelligence 19(7), 711–720 (1997)
6. Ye, J.P., Li, Q.: A two-stage linear discriminant analysis via QR-Decomposition. IEEE Transactions on Pattern Analysis and Machine Intelligence 27(6), 929–941 (2005)
7. Wang, X., Tang, X.: Dual-space linear discriminant analysis for face recognition. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Washington, DC, pp. 564–569 (2004)
8. Jolliffe, I.T.: Principal Component Analysis. Springer, New York (1986)
9. Cevikalp, H., Neamtu, M., Wilkes, M., Barkana, A.: Discriminative common vectors for face recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 27(1), 4–13 (2005)
10. Ye, J.P., Janardan, R., Li, Q.: Two-dimensional linear discriminant analysis. In: Advances in Neural Information Processing Systems 17, Vancouver, British Columbia, Canada, pp. 1529–1536 (2004)
11. Baudat, G., Anouar, F.: Generalized discriminant analysis using a kernel approach. Neural Computation 12(10), 2385–2404 (2000)
12. Ye, J.P., Xiong, T.: Computational and theoretical analysis of null space and orthogonal linear discriminant analysis. Journal of Machine Learning Research 7, 1183–1204 (2006)
13. Ye, J.P.: Least squares linear discriminant analysis. In: Proceedings of the Twenty-Fourth International Conference on Machine Learning, Corvalis, Oregon, USA, pp. 1087–1093 (2007)
14. Kim, S.J., Magnani, A., Boyd, S.: Optimal kernel selection in kernel fisher discriminant analysis. In: Proceedings of the Twenty-Third International Conference on Machine Learning, Pittsburgh, Pennsylvania, USA, pp. 465–472 (2006)

15. Ye, J.P., Chen, J., Ji, S.: Discriminant kernel and regularization parameter learning via semidefinite programming. In: Proceedings of the Twenty-Fourth International Conference on Machine Learning, Corvalis, Oregon, USA, pp. 1095–1102 (2007)
16. Zhu, X.: Semi-supervised learning literature survey. Technical Report 1530, Department of Computer Sciences, University of Wisconsin at Madison, Madison, WI (2006)
17. Blum, A., Mitchell, T.: Combining labeled and unlabeled data with co-training. In: Proceedings of the Workshop on Computational Learning Theory, Madison, Wisconsin, USA, pp. 92–100 (1998)
18. Bennett, K., Demiriz, A.: Semi-supervised support vector machines. In: Advances in Neural Information Processing Systems 11, Vancouver, British Columbia, Canada, pp. 368–374 (1998)
19. Joachims, T.: Transductive inference for text classification using support vector machines. In: Proceedings of the Sixteenth International Conference on Machine Learning, San Francisco, CA, USA, pp. 200–209 (1999)
20. Belkin, M., Niyogi, P., Sindhwani, V.: Manifold regularization: A geometric framework for learning from examples. Journal of Machine Learning Research 7, 2399–2434 (2006)
21. Zhou, D., Bousquet, O., Lal, T., Weston, J., Schölkopf, B.: Learning with local and global consistency. In: Advances in Neural Information Processing Systems 16, Vancouver, British Columbia, Canada (2003)
22. Zhu, X., Ghahramani, Z., Lafferty, J.: Semi-supervised learning using Gaussian fields and harmonic functions. In: Proceedings of the Twentieth International Conference on Machine Learning, Washington, DC, pp. 912–919 (2003)
23. Cai, D., He, X., Han, J.: Semi-supervised discriminant analysis. In: Proceedings of the IEEE International Conference on Computer Vision, Rio de Janeiro, Brazil (2007)
24. Yuille, A., Rangarajan, A.: The concave-convex procedure. Neural Computation 15(4), 915–936 (2003)
25. Smola, A.J., Vishwanathan, S.V.N., Hofmann, T.: Kernel methods for missing variables. In: Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics, Barbados (2005)
26. Fukunnaga, K.: Introduction to Statistical Pattern Recognition. Academic Press, New York (1991)
27. Yang, J., Yang, J.Y.: Why can LDA be performed in PCA transformed space? Pattern Recognition 36(2), 563–566 (2003)
28. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press, New York (2004)
29. Zelnik-Manor, L., Perona, P.: Self-tuning spectral clustering. In: Advances in Neural Information Processing Systems 17, Vancouver, British Columbia, Canada, pp. 1601–1608 (2004)
30. Ye, J.P., Zhao, Z., Wu, M.: Discriminative k-means for clustering. In: Advances in Neural Information Processing Systems 20 (2007)
31. Asuncion, A., Newman, D.: UCI machine learning repository (2007)
32. Sim, T., Baker, S., Bsat, M.: The CMU pose, illumination and expression database. IEEE Transactions on Pattern Analysis and Machine Intelligence 25(12), 1615–1618 (2003)