Semi-Supervised Generalized Discriminant Analysis

Yu Zhang, *Student Member, IEEE* and Dit-Yan Yeung, *Member, IEEE*

***Abstract*—*Generalized discriminant analysis* (GDA) is a commonly used method for dimensionality reduction. In its general form, it seeks a nonlinear projection that simultaneously maximizes the between-class dissimilarity and minimizes the within-class dissimilarity to increase class separability. In real-world applications where labeled data are scarce, GDA may not work very well. However, unlabeled data are often available in large quantities at very low cost. In this paper, we propose a novel *semi-supervised generalized discriminant analysis* algorithm which is abbreviated as SSGDA. We utilize unlabeled data to maximize an optimality criterion of GDA and formulate the problem as an optimization problem that is solved using the *constrained concave-convex procedure* (CCCP). The optimization procedure leads to estimation of the class labels for the unlabeled data. We propose a novel confidence measure and a method for selecting those unlabeled data points whose labels are estimated with high confidence. The selected unlabeled data can then be used to augment the original labeled data set for performing GDA. We also propose a variant of SSGDA, called M-SSGDA, which adopts the *manifold assumption* to utilize the unlabeled data. Extensive experiments on many benchmark data sets demonstrate the effectiveness of our proposed methods.**

***Index Terms*—Generalized Discriminant Analysis, Semi-Supervised Learning, Dimensionality Reduction, Constrained Concave-Convex Procedure**

I. Introduction

Linear discriminant analysis (LDA) [14], [26] 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 some 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 [13] due to severe under-sampling 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 [22], RLDA [15] PCA+LDA [3], LDA/QR [36], NullLDA [13], and DualLDA [30]. PseudoLDA overcomes the singularity problem by substituting the inverse of the within-class scatter matrix with its pseudo-inverse. RLDA adds a positive constant value to each eigenvalue of the within-class scatter matrix to overcome this problem. PCA+LDA first applies PCA [19] 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 and then maximizes the between-class scatter in this space. It is similar to the discriminative common vectors (DCV) method [10]. 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. [16] proposes a unified framework for RLDA, NullLDA and other variants of LDA. There is also another approach to address the SSS problem, with 2DLDA [35] 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 *generalized discriminant analysis* (GDA) (or referred to as kernel discriminant analysis in some papers), that can project the data points nonlinearly, e.g., [2]. Similar to the linear case, there are many variants of GDA corresponding to those of LDA, such as, KPCA+LDA [32], KDA/QR [31], kernel-DCV [9], Kernel Uncorrelated Discriminant Analysis [17], and 2D-GDA [21]. Unlike in LDA, using GDA requires one to select the kernel and set the kernel parameters. Inspired by previous research in multi-kernel learning [23], [20] and [34] address this problem by learning the optimal kernel matrix as a convex combination of some predefined kernel matrices using semi-definite programming (SDP).

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 [37], to reformulate multi-class LDA as a multivariate linear regression problem [33], [43].

In many real-world applications, it is impractical to expect the availability of large quantities of labeled data because labeling data is a costly process. 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* [11], 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-

Y. Zhang and D.-Y. Yeung are with the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Hong Kong, China. E-mail: {zhangyu,dyyeung}@cse.ust.hk.

supervised learning methods can be found in [45]. Some early semi-supervised learning methods include Co-Training [6] and transductive SVM (TSVM) [5], [18]. Recently, graph-based semi-supervised learning methods [4], [44], [46] have aroused the interest 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.

There are some works on semi-supervised extension of LDA, such as [8], [40], [41] and [42], to utilize unlabeled data to alleviate the SSS problem. [8] and [40] assume that the low-dimensional representations of similar data points are also similar and formulate this notion by means of a regularization term in the objective function utilizing some similarity measure. Inspired by TSVM, [41] utilizes unlabeled data to maximize the optimality criterion of LDA. However, these works mainly focus on the linear case, i.e., LDA, but not GDA which is the more general case.

The objective of this paper is to alleviate the SSS problem of GDA by exploiting unlabeled data. We propose a novel *semi-supervised generalized discriminant analysis* algorithm, abbreviated as SSGDA. Specifically, in SSGDA, we utilize unlabeled data to maximize an optimality criterion of GDA and formulate the problem as a constrained optimization problem that can be solved using the *constrained concave-convex procedure* (CCCP) [38], [29]. 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 GDA again. Besides SSGDA, we also propose a variant of SSGDA, called M-SSGDA, which adopts the *manifold assumption* [4] to utilize the unlabeled data. Note that M-SSGDA 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 GDA algorithm in Section II. We then present our SSGDA and M-SSGDA algorithms in Section III. Section IV reports experimental results based on some commonly used data sets. Performance comparison with some representative methods is reported there to demonstrate the effectiveness of our methods. Finally, some concluding remarks are offered in the last section.

II. Generalized Discriminant Analysis

We are given a training set of n data points, $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, where $\mathbf{x}_i \in \mathbb{R}^N$, $i = 1, \ldots, n$. 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 total scatter matrix $\mathbf{S}_t$ and the between-class scatter matrix $\mathbf{S}_b$ are defined as

$$\mathbf{S}_t = \sum_{i=1}^{n} (\phi(\mathbf{x}_i) - \bar{\mathbf{m}})(\phi(\mathbf{x}_i) - \bar{\mathbf{m}})^T \tag{1}$$

$$\mathbf{S}_b = \sum_{k=1}^{C} n_k (\bar{\mathbf{m}}_k - \bar{\mathbf{m}})(\bar{\mathbf{m}}_k - \bar{\mathbf{m}})^T, \tag{2}$$

where $\phi(\cdot)$ denotes the feature mapping corresponding to a kernel function $k(\cdot, \cdot)$, $\bar{\mathbf{m}} = (\sum_{i=1}^{n} \phi(\mathbf{x}_i))/n$ is the sample mean of the whole data set $\mathcal{D}$ and $\bar{\mathbf{m}}_k = (\sum_{\mathbf{x}_i \in \Pi_k} \phi(\mathbf{x}_i))/n_k$ is the class mean of Π_k.

Let $\mathbf{X} = (\phi(\mathbf{x}_1), \ldots, \phi(\mathbf{x}_n))$, $\mathbf{M} = (\bar{\mathbf{m}}_1, \ldots, \bar{\mathbf{m}}_C)$, $\boldsymbol{\pi} = (n_1, \ldots, n_C)^T$, $\mathbf{H}_n = \mathbf{I}_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T$ be the $n \times n$ centering matrix where $\mathbf{I}_n$ is an $n \times n$ identity matrix and $\mathbf{1}_n$ is an $n \times 1$ vector of all ones, $\mathbf{D} = \mathrm{diag}(n_1, \ldots, n_C)$ be a diagonal matrix whose (i, i)th element is n_i, $\mathbf{E}$ be an $n \times C$ class indicator matrix whose (i, j)th element is equal to 1 if $\mathbf{x}_i$ is from the jth class and 0 otherwise. It is easy to see that $\mathbf{1}_n^T\mathbf{E} = \mathbf{1}_C^T\mathbf{D} = \boldsymbol{\pi}^T$ and $\mathbf{M} = \mathbf{X}\mathbf{E}\mathbf{D}^{-1}$ where $\mathbf{A}^{-1}$ denotes the inverse of matrix $\mathbf{A}$ if $\mathbf{A}$ is nonsingular and the pseudo-inverse if $\mathbf{A}$ is singular.

From the definitions of $\mathbf{S}_t$ and $\mathbf{S}_b$ in Eqs. (1) and (2), we can rewrite them in matrix form as

$$\mathbf{S}_t = \mathbf{X}\mathbf{H}_n\mathbf{H}_n\mathbf{X}^T = \mathbf{X}\mathbf{H}_n\mathbf{X}^T$$

and

$$\begin{aligned}
\mathbf{S}_b &= (\mathbf{M} - \frac{1}{n}\mathbf{X}\mathbf{1}_n\mathbf{1}_C^T)\mathbf{D}(\mathbf{M} - \frac{1}{n}\mathbf{X}\mathbf{1}_n\mathbf{1}_C^T)^T \\
&= (\mathbf{X}\mathbf{E}\mathbf{D}^{-1} - \frac{1}{n}\mathbf{X}\mathbf{1}_n\mathbf{1}_C^T)\mathbf{D}(\mathbf{X}\mathbf{E}\mathbf{D}^{-1} - \frac{1}{n}\mathbf{X}\mathbf{1}_n\mathbf{1}_C^T)^T \\
&= \mathbf{X}(\mathbf{E}\mathbf{D}^{-1} - \frac{1}{n}\mathbf{1}_n\mathbf{1}_C^T)\mathbf{D}(\mathbf{D}^{-1}\mathbf{E}^T - \frac{1}{n}\mathbf{1}_C\mathbf{1}_n^T)\mathbf{X}^T \\
&= \mathbf{X}(\mathbf{E} - \frac{1}{n}\mathbf{1}_n\mathbf{1}_C^T\mathbf{D})\mathbf{D}^{-1}(\mathbf{E}^T - \frac{1}{n}\mathbf{D}\mathbf{1}_C\mathbf{1}_n^T)\mathbf{X}^T \\
&= \mathbf{X}(\mathbf{E} - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T\mathbf{E})\mathbf{D}^{-1}(\mathbf{E}^T - \frac{1}{n}\mathbf{E}^T\mathbf{1}_n\mathbf{1}_n^T)\mathbf{X}^T \\
&= \mathbf{X}\mathbf{H}_n\mathbf{E}\mathbf{D}^{-1}\mathbf{E}^T\mathbf{H}_n\mathbf{X}^T.
\end{aligned}$$

The second last equation holds because $\mathbf{1}_n^T\mathbf{E} = \mathbf{1}_C^T\mathbf{D}$.

GDA seeks to find a projection matrix $\mathbf{W}^*$ that maximizes the trace function of $\mathbf{S}_b$ and $\mathbf{S}_t$:

$$\mathbf{W}^* = \arg\max_{\mathbf{W}} \mathrm{trace}((\mathbf{W}^T\mathbf{S}_t\mathbf{W})^{-1}\mathbf{W}^T\mathbf{S}_b\mathbf{W}), \tag{3}$$

where $\mathrm{trace}(\cdot)$ denotes the trace of a square matrix. Since we do not know the explicit form of $\phi(\cdot)$ for most kernel functions, we cannot solve Eq. (3) directly. From the representer theorem in [25], we have $\mathbf{W} = \mathbf{X}\mathbf{P}$. So problem (3) becomes

$$\mathbf{P}^* = \arg\max_{\mathbf{P}} \mathrm{trace}((\mathbf{P}^T\mathbf{K}\mathbf{H}_n\mathbf{K}\mathbf{P})^{-1}\mathbf{P}^T\mathbf{K}\mathbf{H}_n\mathbf{E}\mathbf{D}^{-1}\mathbf{E}^T\mathbf{H}_n\mathbf{K}\mathbf{P}). \tag{4}$$

According to [2], the optimal solution $\mathbf{P}^*$ for the problem (4) can be computed from the eigenvectors of $(\mathbf{K}\mathbf{H}_n\mathbf{K})^{-1}\mathbf{K}\mathbf{H}_n\mathbf{E}\mathbf{D}^{-1}\mathbf{E}^T\mathbf{H}_n\mathbf{K}$. Because the rank of $\mathbf{K}\mathbf{H}_n\mathbf{E}\mathbf{D}^{-1}\mathbf{E}^T\mathbf{H}_n\mathbf{K}$ is at most $C - 1$, $\mathbf{P}^*$ contains $C - 1$ columns in most situations.

III. Semi-Supervised Generalized Discriminant Analysis

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

A. Optimal Solution for GDA

The following theorem on the optimal solution to the problem (4) is relevant here.

Theorem 1: Given $\mathbf{A} \in \mathbb{R}^{n\times n}$ and $\mathbf{B} \in \mathbb{R}^{n\times n}$ are positive semi-definite matrices and the rank of $\mathbf{B}$ is t, then for $\mathbf{P} \in \mathbb{R}^{n\times t}$,

$$\max_{\mathbf{P}} \mathrm{trace}((\mathbf{P}^T\mathbf{A}\mathbf{P})^{-1}\mathbf{P}^T\mathbf{B}\mathbf{P}) = \mathrm{trace}(\mathbf{A}^{-1}\mathbf{B}).$$

The proof of this theorem can be found in [27].

From Theorem 1, we can easily get the following corollary since $\mathbf{K}\mathbf{H}_n\mathbf{K}$ and $\mathbf{K}\mathbf{H}_n\mathbf{E}\mathbf{D}^{-1}\mathbf{E}^T\mathbf{H}_n\mathbf{K}$ are positive semi-definite matrices and the rank of $\mathbf{K}\mathbf{H}_n\mathbf{E}\mathbf{D}^{-1}\mathbf{E}^T\mathbf{H}_n\mathbf{K}$ is $C-1$.

Corollary 1: For $\mathbf{P} \in \mathbb{R}^{n\times(C-1)}$,

$$\begin{aligned} & \max_{\mathbf{P}} \mathrm{trace}((\mathbf{P}^T\mathbf{K}\mathbf{H}_n\mathbf{K}\mathbf{P})^{-1}\mathbf{P}^T\mathbf{K}\mathbf{H}_n\mathbf{E}\mathbf{D}^{-1}\mathbf{E}^T\mathbf{H}_n\mathbf{K}\mathbf{P}) \\ = \; & \mathrm{trace}((\mathbf{K}\mathbf{H}_n\mathbf{K})^{-1}\mathbf{K}\mathbf{H}_n\mathbf{E}\mathbf{D}^{-1}\mathbf{E}^T\mathbf{H}_n\mathbf{K}). \end{aligned}$$

B. SSGDA: 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 and these n data points consist of the training set $\mathcal{D}$ where each class Π_i is just a subset of $\mathcal{D}$. Usually $l \ll m$. When l is very small compared with the input dimensionality, GDA generally does not perform very well. To remedy this problem, we want to incorporate unlabeled data to improve its performance.

Inspired by TSVM [5], [18], which utilizes unlabeled data to maximize the margin, we use unlabeled data here to maximize the optimality criterion of LDA. According to Corollary 1, we utilize unlabeled data to maximize $\mathrm{trace}((\mathbf{K}\mathbf{H}_n\mathbf{K})^{-1}\mathbf{K}\mathbf{H}_n\mathbf{E}\mathbf{D}^{-1}\mathbf{E}^T\mathbf{H}_n\mathbf{K})$ via estimating the class labels of the unlabeled data points.

We first rewrite the objective function as $\mathrm{trace}(\mathbf{E}\mathbf{D}^{-1}\mathbf{E}^T\mathbf{S})$ where $\mathbf{S} = \mathbf{H}_n\mathbf{K}(\mathbf{K}\mathbf{H}_n\mathbf{K})^{-1}\mathbf{K}\mathbf{H}_n$. Since $\mathbf{E}\mathbf{D}^{-1}\mathbf{E}^T = \sum_{i=1}^C \frac{\mathbf{e}_i\mathbf{e}_i^T}{n_i}$ where $\mathbf{e}_i$ is the ith column of $\mathbf{E}$, the objective function can be formulated as

$$\mathrm{trace}(\mathbf{E}\mathbf{D}^{-1}\mathbf{E}^T\mathbf{S}) = \mathrm{trace}(\sum_{i=1}^C \frac{\mathbf{e}_i\mathbf{e}_i^T}{n_i}\mathbf{S}) = \sum_{i=1}^C \frac{\mathbf{e}_i^T\mathbf{S}\mathbf{e}_i}{n_i}.$$

Since those entries in $\mathbf{E}$ for the unlabeled data points are unknown, we maximize the objective function with respect to $\mathbf{E}$. Recalled that n_i is the number of data points in the ith class and so $n_i = \mathbf{e}_i^T\mathbf{1}_n$. By defining some new variables for the sake of notational simplicity, we formulate the optimization problem as:

$$\begin{aligned} \max_{\mathbf{E},\{t_k\}} \quad & \sum_{k=1}^C \frac{\mathbf{e}_k^T\mathbf{S}\mathbf{e}_k}{t_k} \\ s.t. \quad & t_k = \mathbf{e}_k^T\mathbf{1}_n, \; k = 1, \ldots, C \\ & e_{ij} = \begin{cases} 1 & \text{if } \mathbf{x}_i \in \Pi_j \\ 0 & \text{otherwise} \end{cases} \; i = 1, \ldots, l \\ & e_{ij} \in \{0, 1\}, \; i = l+1, \ldots, n, \; j = 1, \ldots, C \\ & \sum_{j=1}^C e_{ij} = 1, \; i = l+1, \ldots, n, \end{aligned} \tag{5}$$

where e_{ij} is the jth element of $\mathbf{e}_i$ and also the (i, j)th element of $\mathbf{E}$.

Unfortunately this is an integer programming problem which is known to be NP-hard and often cannot be efficiently solved. We seek to make this integer programming problem tractable by relaxing the constraint $e_{ij} \in \{0, 1\}$ in (5) to $e_{ij} \geq 0$, giving rise to a modified formulation of the optimization problem:

$$\begin{aligned} \max_{\mathbf{E},\{t_k\}} \quad & \sum_{k=1}^C \frac{\mathbf{e}_k^T\mathbf{S}\mathbf{e}_k}{t_k} \\ s.t. \quad & t_k = \mathbf{e}_k^T\mathbf{1}_n, \; k = 1, \ldots, C \\ & e_{ij} = \begin{cases} 1 & \text{if } \mathbf{x}_i \in \Pi_j \\ 0 & \text{otherwise} \end{cases} \; i = 1, \ldots, l \\ & e_{ij} \geq 0, \; i = l+1, \ldots, n, \; j = 1, \ldots, C \\ & \sum_{j=1}^C e_{ij} = 1, \; i = l+1, \ldots, n. \end{aligned} \tag{6}$$

With such relaxation, the matrix entries of $\mathbf{E}$ for the unlabeled data points may be interpreted as posterior class probabilities. However, even though the constraints in the optimization problem (6) are linear, the problem seeks to maximize a convex function which, unfortunately, does not correspond to a convex optimization problem [7]. If we re-express the optimization problem in (6) as minimizing a concave function, we can adopt the *constrained concave-convex procedure* (CCCP) [38], [29] 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:

$$\begin{aligned} \max_{\mathbf{E},\{t_k\}} \quad & \sum_{k=1}^{C} \left(\frac{2(\mathbf{e}_k^{(p)})^T\mathbf{S}}{t_k^{(p)}}\mathbf{e}_k - \frac{(\mathbf{e}_k^{(p)})^T\mathbf{S}\mathbf{e}_k^{(p)}}{(t_k^{(p)})^2} t_k \right) \\ s.t. \quad & t_k = \mathbf{e}_k^T \mathbf{1}_n,\ k=1,\ldots,C \\ & e_{ij} = \left\{ \begin{array}{ll} 1 & \text{if } \mathbf{x}_i \in \Pi_j \\ 0 & \text{otherwise} \end{array} \right. \quad i=1,\ldots,l \\ & e_{ij} \geq 0,\ i=l{+}1,\ldots,n,\ j=1,\ldots,C \\ & \sum_{j=1}^{C} e_{ij} = 1,\ i=l{+}1,\ldots,n, \end{aligned} \tag{7}$$

where $e_k^{(p)}$ and $t_k^{(p)}$ $(k = 1,\ldots,C)$ were obtained in the pth iteration. The objective function in (7) is just the first-order Taylor series approximation of that in (6) by ignoring some constant terms.

Since the optimization problem (7) 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 e_{ij} computed in each iteration must be in $\{0, 1\}$, which automatically satisfies the constraints in (5).

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 e_{ij} are held fixed based on their class labels. For the unlabeled data points, we initialize the corresponding entries in e_{ij} with equal prior probabilities for all classes:

$$\begin{aligned} e_{ij}^{(0)} &= \left\{ \begin{array}{ll} 1 & \text{if } \mathbf{x}_i \in \Pi_j \\ 0 & \text{otherwise} \end{array} \right. \quad i=1,\ldots,l,\ j=1,\ldots,C \\ e_{ij}^{(0)} &= \frac{1}{C},\ i=l{+}1,\ldots,n,\ j=1,\ldots,C. \end{aligned} \tag{8}$$

The initial values for $\{t_k^{(0)}\}$ can be computed based on the equality constraints in (7) which establish the relationships between $\mathbf{E}$ and t_k.

C. M-SSGDA: Incorporating the Manifold Assumption

The manifold assumption [4] 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 SSGDA to M-SSGDA.

Given the data set $\mathcal{D} = \{\mathbf{x}_1,\ldots,\mathbf{x}_n\}$, 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 $\mathbf{x}_i$ and $\mathbf{x}_j$:

$$w_{ij} = \left\{ \begin{array}{ll} \exp\left(-\frac{\|\mathbf{x}_i-\mathbf{x}_j\|_2^2}{\sigma_i\sigma_j}\right) & \text{if } \mathbf{x}_i \in N_K(\mathbf{x}_j) \text{ or } \mathbf{x}_j \in N_K(\mathbf{x}_i) \\ 0 & \text{otherwise} \end{array} \right.$$

where $\|\cdot\|_2$ denotes the 2-norm of a vector, $N_K(\mathbf{x}_i)$ denotes the neighborhood set of K-nearest neighbors of $\mathbf{x}_i$, σ_i the distance between $\mathbf{x}_i$ and its Kth nearest neighbor, and σ_j the distance between $\mathbf{x}_j$ and its Kth nearest neighbor. This way of constructing the nearest neighbor graph is called *local scaling* [39], which is different from that in SDA [8]. 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 $\mathbf{E}$ are more likely to be the same. We thus modify the optimization problem (6) by adding one more term to the objective function:

$$\begin{aligned} \max_{\mathbf{E},\{t_k\}} \quad & \sum_{k=1}^{C} \frac{\mathbf{e}_k^T\mathbf{S}\mathbf{e}_k}{t_k} - \lambda \sum_{i=1}^{n}\sum_{j=i+1}^{n} w_{ij}\|\mathbf{e}^i - \mathbf{e}^j\|_1 \\ s.t. \quad & t_k = \mathbf{e}_k^T \mathbf{1}_n,\ k=1,\ldots,C \\ & e_{ij} = \left\{ \begin{array}{ll} 1 & \text{if } \mathbf{x}_i \in \Pi_j \\ 0 & \text{otherwise} \end{array} \right. \quad i=1,\ldots,l \\ & e_{ij} \geq 0,\ i=l{+}1,\ldots,n,\ j=1,\ldots,C \\ & \sum_{j=1}^{C} e_{ij} = 1,\ i=l{+}1,\ldots,n, \end{aligned} \tag{9}$$

where $\lambda > 0$ is a regularization parameter, $\mathbf{e}^i$ denotes the ith row of $\mathbf{E}$, and $\|\cdot\|_1$ is the 1-norm of a vector.

Since the objective function of the optimization problem (9) is the difference of two convex functions, we can also adopt CCCP to solve it. Similar to SSGDA, in each iteration of CCCP, we also need to solve an LP problem:

$$\begin{aligned} \max_{\mathbf{E},\{t_k\}} \quad & \sum_{k=1}^{C} \left(\frac{2(\mathbf{e}_k^{(p)})^T\mathbf{S}}{t_k^{(p)}}\mathbf{e}_k - \frac{(\mathbf{e}_k^{(p)})^T\mathbf{S}\mathbf{e}_k^{(p)}}{(t_k^{(p)})^2} t_k \right) \\ & -\lambda \sum_{i=1}^{n}\sum_{j=i+1}^{n} w_{ij}\|\mathbf{e}^i - \mathbf{e}^j\|_1 \\ s.t. \quad & t_k = \mathbf{e}_k^T \mathbf{1}_n,\ k=1,\ldots,C \\ & e_{ij} = \left\{ \begin{array}{ll} 1 & \text{if } \mathbf{x}_i \in \Pi_j \\ 0 & \text{otherwise} \end{array} \right. \quad i=1,\ldots,l \\ & e_{ij} \geq 0,\ i=l{+}1,\ldots,n,\ j=1,\ldots,C \\ & \sum_{j=1}^{C} e_{ij} = 1,\ i=l{+}1,\ldots,n. \end{aligned} \tag{10}$$

One reason for choosing the 1-norm in the problem (9) is to keep the problem (10) as an LP problem which has an efficient and effective solution.

D. Augmenting the Labeled Data Set with Unlabeled Data

For both SSGDA and M-SSGDA, CCCP estimates the class labels of all the unlabeled data points by solving the corresponding optimization problems with respect to $\mathbf{E}$. One might then use all these unlabeled data points with estimated class labels to expand the labeled data set and then apply GDA 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 GDA. 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 e_{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 SSGDA or M-SSGDA to estimate the class labels for the unlabeled data. Then we use all the unlabeled data points with their estimated labels as well as the original labeled data set to perform GDA. 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 the estimated class label of this unlabeled point to have high confidence and hence select it to augment the labeled data set. Finally we performance GDA on the augmented labeled data set to get the final transformation.

The SSGDA (or M-SSGDA) algorithm is summarized in Table I.

E. Discussions

In order to gain some insight into our method, we investigate the dual form of the optimization problem (7). We denote $\mathbf{r}_k^{(p)} = \frac{(\mathbf{e}_k^{(p)})^T \mathbf{S}\mathbf{e}_k^{(p)}}{(t_k^{(p)})^2}\mathbf{1}_n - \frac{2\mathbf{S}\mathbf{e}_k^{(p)}}{t_k^{(p)}}$. We plug the first equality constraint of the optimization problem (7) into its objective function and get the following Lagrangian:

$$\begin{aligned} L(\mathbf{E}, \boldsymbol{\alpha}, \boldsymbol{\beta}) &= \sum_{k=1}^{C} (\mathbf{r}_k^{(p)})^T \mathbf{e}_k - \sum_{k=1}^{C}\sum_{i=1}^{l} \alpha_{ki}(e_{ik} - \delta_k^{c(i)}) \\ &\quad - \sum_{k=1}^{C}\sum_{i=l+1}^{n} \alpha_{ki} e_{ik} - \sum_{i=l+1}^{n} \beta_i (\sum_{k=1}^{C} e_{ik} - 1), \end{aligned}$$

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 (7) is

$$\begin{aligned} \max_{\alpha,\beta} \quad & \sum_{k=1}^{C}\sum_{i=1}^{l} \alpha_{ki}\delta_k^{c(i)} + \sum_{i=l+1}^{n} \beta_i \\ s.t. \quad & \alpha_{ki} = r_{ki}^{(p)},\ i = 1, \ldots, l,\ k = 1, \ldots, C \\ & \alpha_{ki} + \beta_i = r_{ki}^{(p)},\ i = l{+}1, \ldots, n,\ k = 1, \ldots, C \\ & \alpha_{ki} \geq 0,\ i = l{+}1, \ldots, n,\ k = 1, \ldots, C, \end{aligned} \tag{11}$$

where $r_{ki}^{(p)}$ is the ith element of vector $\mathbf{r}_k^{(p)}$.

The *Karush-Kuhn-Tucker* (KKT) condition [7] for the optimization problem (11) is

$$\alpha_{ki} e_{ik} = 0,\ i = l{+}1, \ldots, n,\ k = 1, \ldots, C. \tag{12}$$

From the first constraint of the optimization problem (11), 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 (11) by eliminating the first summation term in the objective function and the first constraint as

$$\begin{aligned} \max_{\alpha,\beta} \quad & \sum_{i=l+1}^{n} \beta_i \\ s.t. \quad & \alpha_{ki} + \beta_i = r_{ki}^{(p)},\ i = l{+}1, \ldots, n,\ k = 1, \ldots, C \\ & \alpha_{ki} \geq 0,\ i = l{+}1, \ldots, n,\ k = 1, \ldots, C, \end{aligned} \tag{13}$$

which can be further simplified as

$$\begin{aligned} \max_{\beta} \quad & \sum_{i=l+1}^{n} \beta_i \\ s.t. \quad & \beta_i \leq r_{ki}^{(p)},\ i = l{+}1, \ldots, n,\ k = 1, \ldots, C. \end{aligned} \tag{14}$$

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

For each unlabeled data point, if we assume $e_{ik^\star} > 0$, then from the KKT condition (12) we can get $\alpha_{k^\star i} = 0$ and also $\beta_i = r_{k^\star i}^{(p)}$ according to the first constraint of the optimization problem (13). So

$$r_{k^\star i}^{(p)} = \min_k \{r_{ki}^{(p)}\}$$

and

$$k^\star = \arg\min_k \{r_{ki}^{(p)}\}.$$

So $r_{ki}^{(p)}$ 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 $e_{ik^\star} = 1$ and $e_{ik'} = 0$ for $k' \neq k^\star$; otherwise, we can first find the set of unlabeled data points for which there exists a unique minimum and e_{ik} can be easily determined, and then we can solve a smaller LP problem (7) by plugging in the known elements e_{ij}. From our experiments, the latter situation seldom occurs and this can speed up the optimization problem (7), which even does not need to solve an LP problem.

For problem (10), when the number of unlabeled data points is large, the computational cost to find the optimal value is still very large. Here we use an alternating method to solve problem (10). That is, at one time we optimize problem (10) with respect to $\mathbf{e}^i$ with $\{\mathbf{e}^j (j \neq i)\}$ fixed. Let $\mathbf{r}_k = \frac{(\mathbf{e}_k^{(p)})^T \mathbf{S}\mathbf{e}_k^{(p)}}{(t_k^{(p)})^2}\mathbf{1}_n - \frac{2\mathbf{S}\mathbf{e}_k^{(p)}}{t_k^{(p)}}$. We first rewrite problem (10) as

$$\begin{aligned} \min_{\mathbf{E}} \quad & \sum_{k=1}^{C} \mathbf{r}_k^T \mathbf{e}_k + \lambda \sum_{i=1}^{n}\sum_{j=i+1}^{n} w_{ij} \|\mathbf{e}^i - \mathbf{e}^j\|_1 \\ s.t. \quad & e_{ij} = \left\{ \begin{array}{ll} 1 & \text{if } \mathbf{x}_i \in \Pi_j \\ 0 & \text{otherwise} \end{array} \right. \quad i = 1, \ldots, l \\ & e_{ij} \geq 0,\ i = l{+}1, \ldots, n,\ j = 1, \ldots, C \\ & \sum_{j=1}^{C} e_{ij} = 1,\ i = l{+}1, \ldots, n, \end{aligned} \tag{15}$$

TABLE I
ALGORITHM FOR SSGDA OR M-SSGDA

```
Input: labeled data $x_i$ $(i = 1, \ldots, l)$, unlabeled data $x_i$ $(i = l{+}1, \ldots, n)$, $K$, $\theta$, $\varepsilon$
Initialize $\mathbf{E}^{(0)}$ using Eq. (8);
Initialize $t_k^{(0)}$ based on $\mathbf{E}^{(0)}$ for $k = 1, \ldots, C$;
Construct the $K$-nearest neighbor graph;
$p = 0$;
Repeat
    $p = p + 1$;
    Solve the optimization problem (7) or (10);
    Update $\mathbf{E}^{(p)}$ and $t_k^{(p)}$ using the result of the optimization problem for $k = 1, \ldots, C$;
Until $\|\mathbf{E}^{(p)} - \mathbf{E}^{(p-1)}\|_F \leq \varepsilon$
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 GDA on the augmented labeled data set to get the transformation $\mathbf{P}$.
Output: the transformation $\mathbf{P}$
```

which can be reformulated as

$$\begin{aligned}
\min_{\mathbf{E}} \quad & \sum_{k=1}^{n} \mathbf{e}^k (\mathbf{r}^k)^T + \lambda \sum_{i=1}^{n} \sum_{j=i+1}^{n} w_{ij} \|\mathbf{e}^i - \mathbf{e}^j\|_1 \\
s.t. \quad & e_{ij} = \begin{cases} 1 & \text{if } \mathbf{x}_i \in \Pi_j \\ 0 & \text{otherwise} \end{cases} \quad i = 1, \ldots, l \\
& e_{ij} \geq 0, \ i = l{+}1, \ldots, n, \ j = 1, \ldots, C \\
& \sum_{j=1}^{C} e_{ij} = 1, \ i = l{+}1, \ldots, n, \qquad (16)
\end{aligned}$$

where $\mathbf{R} = (\mathbf{r}_1, \ldots, \mathbf{r}_C)$ and $\mathbf{r}^k$ is the kth row of $\mathbf{R}$. So when using an alternating method, the optimization problem with respect to $\mathbf{e}^i$ $(i > l)$ can be formulated as

$$\begin{aligned}
\min_{\mathbf{e}^i} \quad & \mathbf{e}^i (\mathbf{r}^i)^T + \lambda \sum_{j \neq i \& w_{ij} > 0} w_{ij} \|\mathbf{e}^i - \mathbf{e}^j\|_1 \\
s.t. \quad & e_{ij} \geq 0, \ j = 1, \ldots, C \\
& \sum_{j=1}^{C} e_{ij} = 1. \qquad (17)
\end{aligned}$$

Compared with the original problem (10), there is no need to specify the constraints for labeled data in problem (17) which reduces the complexity of the optimization problem. The problem (17) has C variables and $C + 1$ constraints and so it can be solved efficiently. Since problem (10) is a convex problem and each step in the alternating method decreases the objective function value, the learning procedure can converge to the global optimum.

The computational cost of SSGDA and M-SSGDA includes performing GDA twice and solving the optimization problem using CCCP. The complexity of GDA 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 SSGDA and M-SSGDA are efficient under most situations.

IV. EXPERIMENTS

In this section, we first study SSGDA and M-SSGDA empirically.

We compare their performance with several other dimensionality reduction methods. After dimensionality reduction has been performed, we apply a simple nearest-neighbor classifier to perform classification in the embedding space. We also compare SSGDA and M-SSGDA with two state-of-the-art inductive semi-supervised learning methods, LapSVM and LapRLS [4]. 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 13 benchmark data sets, including 8 UCI data sets [1], a brain-computer interface data set BCI[3] and four image data sets: COIL[3], PIE [28], ORL [3] and AR [24]. See Table II for more details.

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 II 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-SSGDA, 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.

We use 5-fold cross validation to determine the values of the hyperparameters. The candidate set for the λ used in M-SSGDA is $\{10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^2, 10^3\}$ and the width parameter of the RBF kernel is chosen from $\sigma_0 \times \{10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^2, 10^3\}$ where σ_0 is the average distance between two points in the data set.

A. Experimental Results for Linear Kernel

In this section, we compare our methods with some linear methods by using the linear kernel. We first compare our methods with dimensionality reduction methods including PCA, LDA [3], SDA and LapLDA [12]. Note that PCA is unsupervised, LDA is supervised, and SDA and LapLDA are semi-supervised in nature. Moreover, the nearest-neighbor (NN) classifier is used as a baseline. The experimental results are listed in Table III. 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 bold.

[1] `http://www.stanford.edu/~boyd/cvx/`
[2] `http://manifold.cs.uchicago.edu/`
[3] `http://www.kyb.tuebingen.mpg.de/ssl-book/`

TABLE II
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
ORL	644	200	20	2	6
AR	792	520	20	5	15

From the results, we can see that the performance of SSGDA or M-SSGDA 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 any manifold structure, the performance of SSGDA is better than M-SSGDA. For MFEAT-PIXEL, PIE and others which may contain manifold structure, the performance of M-SSGDA is better than SSGDA. Thus for data sets such as images which may have manifold structure, we recommend using M-SSGDA. Otherwise SSGDA is preferred. Compared with SDA, SSGDA and M-SSGDA are more stable. Specifically, the performance of SSGDA or M-SSGDA is comparable to or better than that of LDA in most situations. For SDA, however, performance degradation can sometimes be very severe, especially for MFEAT-PIXEL and PIE.

We also investigate the selection method described in Section III-D. We record the mean error rate of label estimation for the unlabeled data over 20 trials before and after applying the selection method. The results in Table IV show that the estimation error rate after applying the selection method is almost always smaller, sometimes very significantly. This confirms that our selection method for unlabeled data is very effective.

TABLE IV
ERROR RATE OF LABEL ESTIMATION FOR THE UNLABELED DATA BEFORE AND AFTER APPLYING THE SELECTION METHOD

	SSGDA (%)		M-SSGDA (%)	
Data set	Before	After	Before	After
diabetes	35.97	33.33	45.90	48.80
heart-statlog	27.73	27.38	44.75	33.30
ionosphere	30.95	12.49	25.90	17.93
hayes-roth	53.25	47.27	58.00	57.36
iris	24.58	6.61	8.58	4.94
mfeat-pixel	67.51	0.00	5.79	1.09
pendigits	24.69	13.92	11.08	5.98
vehicle	43.70	30.12	55.20	47.74
BCI	49.25	34.58	51.00	50.85
COIL	66.43	3.93	57.36	39.97
PIE	69.52	15.00	47.36	29.59
ORL	95.58	0.50	26.67	13.00
AR	94.43	35.00	68.63	24.17

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 [6], methods in [44], [46], LapSVM and LapRLS [4]. Co-Training requires two independent and sufficient views for the data, but data used in our experiment cannot satisfy this requirement. The methods in [44], [46] 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 cannot satisfy our experimental settings and hence are not included 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 [4] 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 V. From the experimental results, we can see that the performance of SSGDA and M-SSGDA is comparable to or even better than that of LapSVM and LapRLS. Moreover, one advantage of SSGDA and M-SSGDA 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.

In Table VI, we record the number of iterations in the CCCP optimization method for SSGDA and M-SSGDA as well as their computation time. From the results, it is clear that both methods exhibit fast convergence requiring no more than 20 iterations and relatively short computation time.

B. Experimental Results for RBF Kernel

We next compare our methods with some nonlinear methods using the RBF kernel, including KPCA, GDA, LapRLS and LapSVM. The experimental results are shown in Table VII. For each data set, the lowest classification error is shown in bold. From the results, we can see that the performance of SSGDA and M-SSGDA using the RBF kernel is comparable to or even better than that of KPCA, GDA, LapSVM and LapRLS. This demonstrates the effectiveness of our methods.

Moreover, from Table VIII, we can see that our methods are efficient in terms of both the number of iterations and the

TABLE III

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	NN	PCA	LDA	SDA	LapLDA	SSGDA	M-SSGDA
diabetes	0.4630(0.0436)	0.4335(0.0775)	0.4438(0.0878)	0.4022(0.0638)	0.4537(0.0514)	**0.3898(0.0674)**	0.4360(0.0605)
	0.4657(0.0675)	0.4253(0.1154)	0.4311(0.0997)	0.3763(0.0864)	0.4466(0.0838)	**0.3276(0.0643)**	0.4125(0.1074)
heart-statlog	0.4160(0.0803)	0.4288(0.0689)	0.3978(0.0582)	0.3680(0.0564)	0.3695(0.0568)	**0.3293(0.0976)**	0.3818(0.0662)
	0.4283(0.1181)	0.3975(0.0669)	0.3767(0.1055)	0.3783(0.1076)	0.3958(0.0856)	**0.3133(0.1174)**	0.3258(0.1493)
ionosphere	0.3250(0.0982)	0.2895(0.1032)	0.2850(0.0876)	**0.2695(0.1056)**	0.2825(0.0552)	0.2860(0.1015)	0.2830(0.1029)
	0.3261(0.0566)	**0.2189(0.0632)**	0.2365(0.0972)	0.2241(0.0863)	0.2203(0.0627)	0.2351(0.1032)	0.2399(0.1278)
hayes-roth	0.5100(0.0969)	0.5175(0.0571)	0.4942(0.0531)	0.5058(0.0661)	0.5183(0.0630)	0.4867(0.0569)	**0.4758(0.0586)**
	0.5211(0.0598)	0.5115(0.0605)	0.5165(0.0690)	0.5077(0.0752)	0.5984(0.0642)	0.5121(0.0770)	**0.5060(0.0627)**
iris	0.0917(0.0393)	0.0917(0.0417)	0.0933(0.0613)	0.0825(0.0506)	0.0792(0.0609)	0.0708(0.0445)	**0.0667(0.0493)**
	0.1052(0.0409)	0.0907(0.0333)	0.0833(0.0586)	0.0809(0.0395)	0.0728(0.0485)	0.0611(0.0370)	**0.0611(0.0454)**
mfeat-pixel	0.1770(0.0168)	0.1450(0.0232)	0.1501(0.0290)	0.2783(0.0435)	0.1571(0.0268)	0.1501(0.0289)	**0.1367(0.0210)**
	0.1692(0.0182)	0.1429(0.0228)	0.1486(0.0264)	0.3428(0.0298)	0.1673(0.0181)	0.1485(0.0264)	**0.1329(0.0213)**
pendigits	0.1947(0.0151)	0.1724(0.0305)	0.2238(0.0364)	0.2547(0.0447)	0.1758(0.0185)	0.1785(0.0266)	**0.1617(0.0242)**
	0.1971(0.0178)	0.1761(0.0276)	0.2192(0.0332)	0.2544(0.0382)	0.1725(0.0185)	0.1779(0.0190)	**0.1650(0.0225)**
vehicle	0.5485(0.0521)	0.5739(0.0375)	0.5741(0.0365)	0.5400(0.0402)	0.4816(0.0538)	**0.4396(0.0734)**	0.4838(0.0901)
	0.5580(0.0570)	0.5808(0.0453)	0.5879(0.0429)	0.5462(0.0312)	0.4918(0.0575)	**0.4329(0.0672)**	0.4739(0.0791)
BCI	0.5260(0.0378)	0.4835(0.0460)	0.4830(0.0557)	0.4960(0.0476)	0.4990(0.0468)	**0.4750(0.0432)**	0.4975(0.0484)
	0.5169(0.0266)	0.5000(0.0324)	0.4803(0.0249)	0.4812(0.0326)	0.4948(0.0488)	**0.4732(0.0331)**	0.4741(0.0346)
COIL	0.5202(0.0535)	**0.4443(0.0418)**	0.5247(0.0371)	0.5419(0.0607)	0.4561(0.0525)	0.5236(0.0374)	0.5193(0.0401)
	0.5399(0.0523)	**0.4391(0.0364)**	0.5194(0.0421)	0.5461(0.0482)	0.4553(0.0508)	0.5178(0.0434)	0.5096(0.0398)
PIE	0.5608(0.0223)	0.6156(0.0275)	0.5055(0.1624)	0.7629(0.0377)	0.4010(0.0394)	0.4674(0.1757)	**0.2381(0.0552)**
	0.5654(0.0310)	0.6207(0.0251)	0.5126(0.1512)	0.8277(0.0208)	0.3853(0.0382)	0.4777(0.1696)	**0.2424(0.0592)**
ORL	0.1117(0.0341)	0.1217(0.0292)	0.1050(0.0300)	0.1142(0.0556)	0.1108(0.0427)	**0.0967(0.0391)**	0.1025(0.0515)
	0.1300(0.0483)	0.1400(0.0530)	0.1375(0.0489)	**0.0950(0.0524)**	0.0975(0.0416)	0.1325(0.0613)	0.1300(0.0468)
AR	0.6563(0.0403)	0.6610(0.0369)	0.1570(0.0385)	0.6603(0.0334)	0.6447(0.0365)	**0.1480(0.0280)**	0.1547(0.0273)
	0.6392(0.0611)	0.6525(0.0547)	0.1458(0.0240)	0.5250(0.0450)	0.5267(0.0360)	**0.1325(0.0234)**	0.1342(0.0217)

TABLE V

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	SSGDA	M-SSGDA
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)**
ORL	0.2442(0.0466)	0.1158(0.0494)	**0.0967(0.0391)**	0.1025(0.0515)
	0.2600(0.0518)	**0.1025(0.0416)**	0.1325(0.0613)	0.1300(0.0468)
AR	0.4647(0.0270)	0.4163(0.0252)	**0.1480(0.0280)**	0.1547(0.0273)
	0.3317(0.0473)	0.2850(0.0374)	**0.1325(0.0234)**	0.1342(0.0217)

total computation time.

V. CONCLUSION

In this paper, we have presented a new approach for semi-supervised generalized 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 GDA under situations where labeled data are limited. In our future work, we will apply the ideas here to some other dimensionality reduction methods.

ACKNOWLEDGMENTS

This research has been supported by General Research Fund 621407 from the Research Grants Council of the Hong Kong

TABLE VI
NUMBER OF ITERATIONS FOR THE CONVERGENCE OF THE CCCP OPTIMIZATION METHOD AND THE TOTAL COMPUTATION TIME (IN SECONDS) OF OUR TWO METHODS, SSGDA AND M-SSGDA, USING THE LINEAR KERNEL.

	SSGDA		M-SSGDA	
Data set	Number of Iterations	Computation Time (s)	Number of Iterations	Computation Time (s)
diabetes	10	1.69	9	8.54
heart-statlog	12	2.54	10	8.10
ionosphere	14	1.41	15	5.26
hayes-roth	13	0.87	16	5.65
iris	10	0.62	14	4.71
mfeat-pixel	8	3.58	10	76.08
pendigits	11	24.69	15	13.92
vehicle	13	4.11	15	56.02
BCI	14	0.33	16	7.19
COIL	15	3.87	14	91.84
PIE	16	12.52	11	165.61
ORL	16	3.71	12	111.82
AR	16	2.11	13	263.45

TABLE VII
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	NN	KPCA	GDA	LapSVM	LapRLS	SSGDA	M-SSGDA
diabetes	0.4625(0.0334)	0.4195(0.0589)	**0.3915(0.0645)**	0.4235(0.0477)	0.3990(0.0653)	0.4510(0.0662)	0.4100(0.0489)
	0.4809(0.0637)	0.3828(0.0696)	0.3857(0.0647)	0.3685(0.0696)	0.3685(0.0543)	0.3932(0.0751)	**0.3579(0.0589)**
heart-statlog	0.5080(0.0640)	0.4320(0.0547)	0.4255(0.0839)	**0.4185(0.0645)**	0.4225(0.0688)	0.4685(0.0730)	0.4285(0.0555)
	0.5750(0.0873)	0.4717(0.0828)	0.4583(0.0988)	0.5000(0.1227)	0.4933(0.0988)	0.5217(0.1267)	**0.4317(0.0869)**
ionosphere	0.3650(0.0854)	0.3540(0.0878)	**0.2530(0.0831)**	0.3390(0.1440)	0.3070(0.1192)	0.3270(0.0460)	0.3010(0.0794)
	0.2929(0.0305)	0.2660(0.0495)	0.2091(0.0559)	0.2357(0.0582)	0.2212(0.0523)	0.2519(0.0585)	**0.1789(0.0687)**
hayes-roth	0.5683(0.0703)	0.5417(0.0672)	0.4879(0.0549)	0.5283(0.1034)	0.5367(0.0749)	**0.4750(0.0589)**	0.5533(0.1206)
	0.5543(0.0477)	0.4769(0.0788)	0.4389(0.0961)	0.4692(0.0798)	0.4890(0.0824)	**0.4286(0.0930)**	0.4846(0.0790)
iris	0.1550(0.0681)	0.1033(0.0361)	0.1000(0.0425)	0.1567(0.0462)	0.1600(0.0459)	**0.0967(0.0431)**	0.1500(0.0791)
	0.1464(0.0768)	0.0889(0.0534)	0.0914(0.0542)	0.1228(0.0376)	0.1230(0.0477)	**0.0840(0.0489)**	0.1556(0.0478)
mfeat-pixel	0.1690(0.0221)	0.1560(0.0171)	0.1288(0.0140)	0.0920(0.0146)	0.0948(0.0173)	0.0846(0.0159)	**0.0812(0.0177)**
	0.1646(0.0199)	0.1535(0.0101)	0.1194(0.0102)	0.0978(0.0089)	0.1011(0.0132)	0.0774(0.0132)	**0.0630(0.0161)**
pendigits	0.1905(0.0191)	0.1664(0.0387)	0.1664(0.0367)	0.1648(0.0324)	0.1644(0.0325)	**0.1549(0.0355)**	0.1592(0.0459)
	0.1890(0.0199)	0.1657(0.0307)	0.1641(0.0303)	0.1652(0.0212)	0.1669(0.0199)	**0.1538(0.0294)**	0.1597(0.0400)
vehicle	0.5695(0.0292)	0.5875(0.0418)	0.5895(0.0341)	0.5895(0.0341)	0.5965(0.0280)	**0.5275(0.0316)**	0.5405(0.0472)
	0.5800(0.0296)	0.5690(0.0536)	0.5728(0.0435)	0.5793(0.0327)	0.5925(0.0340)	0.5512(0.0376)	**0.5460(0.0368)**
BCI	0.5280(0.0326)	0.5320(0.0789)	0.4760(0.0594)	0.4900(0.0354)	0.4900(0.0510)	**0.4660(0.0680)**	0.4710(0.0667)
	0.5179(0.0356)	0.4772(0.0542)	0.4762(0.0282)	0.4924(0.0344)	0.4745(0.0352)	**0.4621(0.0360)**	0.4717(0.0434)
COIL	0.4877(0.0461)	0.4803(0.0337)	0.4813(0.0345)	0.4573(0.0277)	0.4653(0.0358)	**0.4407(0.0423)**	0.4533(0.0147)
	0.4889(0.0416)	0.4846(0.0478)	0.4809(0.0356)	0.4480(0.0422)	0.4579(0.0541)	0.4518(0.0264)	**0.4380(0.0245)**
PIE	0.5757(0.0229)	0.6017(0.0327)	0.4743(0.0239)	0.3460(0.0253)	0.3793(0.0257)	0.3730(0.0352)	**0.3223(0.0550)**
	0.5719(0.0218)	0.5980(0.0223)	0.4822(0.0278)	0.3380(0.0250)	0.3716(0.0248)	0.3711(0.0281)	**0.3002(0.0470)**
ORL	0.1117(0.0341)	0.1333(0.0471)	0.1092(0.0401)	0.0783(0.0205)	0.0825(0.0287)	**0.0703(0.0132)**	0.0725(0.0279)
	0.1300(0.0483)	0.1425(0.0514)	0.1200(0.0497)	0.0975(0.0399)	0.0675(0.0501)	**0.0615(0.0208)**	0.0691(0.0278)
AR	0.6563(0.0403)	0.6710(0.0377)	0.4593(0.0493)	0.5023(0.0241)	0.5853(0.0329)	0.4153(0.0292)	**0.4143(0.0267)**
	0.6392(0.0611)	0.6525(0.0539)	0.4483(0.0408)	0.4533(0.0653)	0.5275(0.0630)	**0.4350(0.0527)**	0.4692(0.0333)

TABLE VIII
NUMBER OF ITERATIONS FOR THE CONVERGENCE OF THE CCCP OPTIMIZATION METHOD AND THE TOTAL COMPUTATION TIME (IN SECONDS) OF OUR TWO METHODS, SSGDA AND M-SSGDA, UNDER THE RBF KERNEL.

	SSGDA		M-SSGDA	
Data set	Number of Iterations	Computation Time (s)	Number of Iterations	Computation Time (s)
diabetes	12	0.89	11	19.29
heart-statlog	14	0.69	10	23.82
ionosphere	15	0.22	9	21.33
hayes-roth	16	0.07	14	44.94
iris	9	0.06	12	41.33
mfeat-pixel	10	12.36	14	96.95
pendigits	8	11.08	15	125.38
vehicle	9	3.87	15	155.39
BCI	13	0.25	12	70.21
COIL	14	12.26	13	157.10
PIE	15	22.35	14	199.01
ORL	14	0.82	15	162.28
AR	16	4.49	16	388.55

Special Administrative Region, China.

References

[1] A. Asuncion and D.J. Newman. UCI machine learning repository, 2007.

[2] G. Baudat and Fatiha Anouar. Generalized discriminant analysis using a kernel approach. *Neural Computation*, 12(10):2385–2404, 2000.

[3] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):711–720, 1997.

[4] M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: A geometric framework for learning from examples. *Journal of Machine Learning Research*, 7:2399–2434, 2006.

[5] K. Bennett and A. Demiriz. Semi-supervised support vector machines. In *Advances in Neural Information Processing Systems 11*, pages 368–374, Vancouver, British Columbia, Canada, 1998.

[6] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the Workshop on Computational Learning Theory*, pages 92–100, Madison, Wisconsin, USA, 1998.

[7] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, 2004.

[8] D. Cai, X. He, and J. Han. Semi-supervised discriminant analysis. In *Proceedings of the IEEE International Conference on Computer Vision*, Rio de Janeiro, Brazil, 2007.

[9] H. Cevikalp, M. Neamtu, and M. Wilkes. Discriminative common vector method with kernels. *IEEE Transactions on Neural Networks*, 17(6):1550–1565, 2006.

[10] H. Cevikalp, M. Neamtu, M. Wilkes, and A. Barkana. Discriminative common vectors for face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(1):4–13, 2005.

[11] O. Chapelle, A. Zien, and B. Schölkopf, editors. *Semi-Supervised Learning*. MIT Press, Boston, 2006.

[12] J. Chen, J. Ye, and Q. Li. Integrating global and local structures: A least squares framework for dimensionality reduction. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2007.

[13] L. Chen, H. Liao, M. Ko, J. Lin, and G. Yu. A new LDA-based face recognition system which can solve the small sample size problem. *Pattern Recognition*, 33(10):1713–1726, 2000.

[14] R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7:179–188, 1936.

[15] Y. Guo, T. Hastie, and R. Tibshirani. Regularized linear discriminant analysis and its application in microarrays. *Biostatistics*, 8(1):86–100, 2007.

[16] S. Ji and J. Ye. Generalized linear discriminant analysis: A unified framework and efficient model selection. *IEEE Transactions on Neural Networks*, 19(10):1768–1782, 2008.

[17] S. Ji and J. Ye. Kernel uncorrelated and regularized discriminant analysis: A theoretical and computational study. *IEEE Transactions on Knowledge and Data Engineering*, 20(10):1131–1321, 2008.

[18] T. Joachims. Transductive inference for text classification using support vector machines. In *Proceedings of the Sixteenth International Conference on Machine Learning*, pages 200–209, San Francisco, CA, USA, 1999.

[19] I. T. Jolliffe. *Principal Component Analysis*. Springer-Verlag, New York, 1986.

[20] S.-J. Kim, A. Magnani, and S. Boyd. Optimal kernel selection in kernel fisher discriminant analysis. In *Proceedings of the Twenty-Third International Conference on Machine Learning*, pages 465–472, Pittsburgh, Pennsylvania, USA, 2006.

[21] H. Kong, E. K. Teoh, J. G. Wang, and C. Kambhamettu. Generalized 2d fisher discriminant analysis. In *Proceedings of the British Machine Vision Conference*, Oxford Brookes University, Oxford, 2005.

[22] W. J. Krzanowski, P. Jonathan, W. V. McCarthy, and M. R. Thomas. Discriminant analysis with singular covariance matrices: methods and applications to spectroscopic data. *Applied Statistics*, 44(1):101–115, 1995.

[23] G. R. G. Lanckriet, N. Cristianini, L. El Ghaoui, P. L. Bartlett, and M. I. Jordan. Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Research*, 5:27–72, 2004.

[24] A. M. Martínez and R. Benavente. The AR-face database. Technical Report 24, CVC, 1998.

[25] S. Mika, G. Rätsch, J. Weston, B. Schölkopf, A. Smola, and K.-R. Müller. Constructing descriptive and discriminative nonlinear features: Rayleigh coefficients in kernel feature spaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5):623–628, 2003.

[26] C. R. Rao. The utilization of multiple measurements in problems of biological classification. *Journal of the Royal Statictical Society*, 10:159–203, 1948.

[27] J. R. Schott. *Matrix Analysis for Statistics*. Wiley, New York, 2nd edition, 2005.

[28] T. Sim, S. Baker, and M. Bsat. The CMU pose, illumination and expression database. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(12):1615–1618, 2003.

[29] A. J. Smola, S. V. N. Vishwanathan, and T. Hofmann. Kernel methods for missing variables. In *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*, Barbados, 2005.

[30] X. Wang and X. Tang. Dual-space linear discriminant analysis for face recognition. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 564–569, Washington, DC, 2004.

[31] T. Xiong, J. Ye, Q. Li, R. Janardan, and V. Cherkassky. Efficient kernel discriminant analysis via QR decomposition. In *Advances in Neural Information Processing Systems 17*, Vancouver, British Columbia, Canada, 2004.

[32] J. Yang, A. F. Frangi, J.-Y. Yang, D. Zhang, and Z. Jin. KPCA plus LDA: A complete kernel fisher discriminant framework for feature extraction and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(2):230–244, 2005.

[33] J.-P. Ye. Least squares linear discriminant analysis. In *Proceedings of the Twenty-Fourth International Conference on Machine Learning*, pages 1087–1093, Corvalis, Oregon, USA, 2007.

[34] J.-P. Ye, J. Chen, and S. Ji. Discriminant kernel and regularization parameter learning via semidefinite programming. In *Proceedings of the Twenty-Fourth International Conference on Machine Learning*, pages 1095–1102, Corvalis, Oregon, USA, 2007.

[35] J.-P. Ye, R. Janardan, and Q. Li. Two-dimensional linear discriminant analysis. In *Advances in Neural Information Processing Systems 17*, pages 1529–1536, Vancouver, British Columbia, Canada, 2004.

[36] J.-P. Ye and Q. Li. A two-stage linear discriminant analysis via QR-Decomposition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(6):929–941, 2005.

[37] J.-P. Ye and T. Xiong. Computational and theoretical analysis of null space and orthogonal linear discriminant analysis. *Journal of Machine Learning Research*, 7:1183–1204, 2006.

[38] A. Yuille and A. Rangarajan. The concave-convex procedure. *Neural Computation*, 15(4):915–936, 2003.

[39] L. Zelnik-Manor and P. Perona. Self-tuning spectral clustering. In *Advances in Neural Information Processing Systems 17*, pages 1601–1608, Vancouver, British Columbia, Canada, 2004.

[40] Y. Zhang and D.-Y. Yeung. Semi-supervised discriminant analysis using robust path-based similarity. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Anchorage, Alaska, 2008.

[41] Y. Zhang and D.-Y. Yeung. Semi-supervised discriminant analysis via CCCP. In *Proceedings of the 19th European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, pages 644–659, Antwerp, Belgium, 2008.

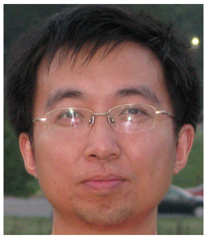
[42] Y. Zhang and D.-Y. Yeung. Heteroscedastic probabilistic linear discriminant analysis with semi-supervised extension. In *Proceedings of the 20th European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, pages 602–616, Bled, Slovenia, 2009.

[43] Z. Zhang, G. Dai, and M. I. Jordan. A flexible and efficient algorithm for regularized fisher discriminant analysis. In *Proceedings of the 20th European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, pages 632–647, Bled, Slovenia, 2009.

[44] D. Zhou, O. Bousquet, T. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. In *Advances in Neural Information Processing Systems 16*, Vancouver, British Columbia, Canada, 2003.

[45] X. Zhu. Semi-supervised learning literature survey. Technical Report 1530, Department of Computer Sciences, University of Wisconsin at Madison, Madison, WI, 2006.

[46] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using Gaussian fields and harmonic functions. In *Proceedings of the Twentieth International Conference on Machine Learning*, pages 912–919, Washington, DC, 2003.



Yu Zhang is a PhD candidate at the Department of Computer Science and Engineering of Hong Kong University of Science and Technology. He received the BSc and MEng degrees in the Department of Computer Science and Technology of Nanjing University, China, in 2004 and 2007 respectively. His research interests mainly include machine learning and data mining, especially in multi-task learning, transfer learning, dimensionality reduction, metric learning and semi-supervised learning. He was the winner of the best paper award in the 26th Conference on Uncertainty in Artificial Intelligence (UAI) 2010.

Dit-Yan Yeung is a Professor in the Department of Computer Science and Engineering of the Hong Kong University of Science and Technology (HKUST). He started his academic career as an Assistant Professor at the Illinois Institute of Technology (IIT) before he joined HKUST. His research interests are in computational and statistical approaches to problems in machine learning, neural networks and pattern recognition. He served on the editorial board of Journal of Artificial Intelligence Research (JAIR) and is currently on the editorial boards of ACM Transactions on Intelligent Systems and Technology (TIST) and Pattern Recognition (PR).