Pattern Recognition 45 (2012) 1482–1499

Contents lists available at SciVerse ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Trace ratio criterion based generalized discriminative learning for semi-supervised dimensionality reduction

Mingbo Zhao, Zhao Zhang, Tommy W.S. Chow*

Electronic Engineering Department, City University of Hong Kong, Tat Chee Avenue, Kowloon, Hong Kong Special Administration Region

ARTICLE INFO

Article history:
Received 24 January 2011
Received in revised form
13 October 2011
Accepted 16 October 2011
Available online 12 November 2011

Keywords:
Dimensionality reduction
Semi-supervised learning
Trace ratio criterion
Discriminative learning

ABSTRACT

Dealing with high-dimensional data has always been a major problem in many pattern recognition and machine learning applications. Trace ratio criterion is a criterion that can be applicable to many dimensionality reduction methods as it directly reflects Euclidean distance between data points of within or between classes. In this paper, we analyze the trace ratio problem and propose a new efficient algorithm to find the optimal solution. Based on the proposed algorithm, we are able to derive an orthogonal constrained semi-supervised learning framework. The new algorithm incorporates unlabeled data into training procedure so that it is able to preserve the discriminative structure as well as geometrical structure embedded in the original dataset. Under such a framework, many existing semi-supervised dimensionality reduction methods such as *SDA*, *Lap-LDA*, *SSDR*, *SSMMC*, can be improved using our proposed framework, which can also be used to formulate a corresponding kernel framework for handling nonlinear problems. Theoretical analysis indicates that there are certain relationships between linear and nonlinear methods. Finally, extensive simulations on synthetic dataset and real world dataset are presented to show the effectiveness of our algorithms. The results demonstrate that our proposed algorithm can achieve great superiority to other state-of-art algorithms.

1. Introduction

Dealing with high-dimensional data has always been a major problem for pattern recognition and machine learning. Typical applications involving high-dimensional data include face recognition, document categorization and image retrieval. Finding a low-dimensional representation of high-dimensional space, namely dimensionality reduction is thus of great practical importance. The goal of dimensionality reduction is to reduce the complexity of the original space and embed high-dimensional space into a low-dimensional space while keeping most of the desired intrinsic information [1,2]. The desired information can be discriminative [11,12,15–17], geometrical [1,2,13,14,46] or both discriminative and geometrical [19–23]. Among all the dimensionality reduction methods, Linear Discriminant Analysis (*LDA*) [11,12] is the most popular method and has been widely used in many classification applications. The goal of *LDA* is to find the optimal low-dimensional presentation to the original dataset by maximizing between-class scatter matrix S_b, while minimizing within-class scatter matrix S_w. The original formulation of *LDA*, known as Fisher *LDA* [11], can only deal with binary-class classification. When solving multi-class classification problem, the basic *LDA* has to be extended using two main criterions including ratio trace criterion $\max_W Tr[(W^T S_b W)^{-1}(W^T S_w W)]$ and trace ratio criterion $\max_{W^T W=I}((Tr(W^T S_b W)/(Tr(W^T S_w W)))$.

In ratio trace or determinant ratio *LDA*, it is assumed that the within-class scatter matrix is nonsingular. Finding the optimal projection can be solved by generalized eigen-value decomposition (*GEVD*) [35]. However, trace ratio *LDA* may confront ill-posed problem when the number of data points is smaller than that of the features [34,44,45]. Several variants of ratio trace *LDA* are proposed to solve this problem such as null-space *LDA* [25], uncorrelated *LDA* [26], *LDA/GSVD* [27], Discriminative Common Vectors [28]. Another widely used criterion of *LDA* is the trace ratio criterion. Different from the former one, the trace ratio criterion can directly reflect Euclidean distances between data points of inter and intra classes. In addition, the optimal projection obtained by trace ratio *LDA* is orthogonal, while the one obtained by ratio trace *LDA* is non-orthogonal. Recently, there has been increasing interest in the issue of finding orthogonal projection for dimensionality reduction methods [29–31]. As described in [4], when evaluating the similarities between data points based on Euclidean distance, the non-orthogonal projection may put different weights on different projection directions thus changing the similarities, while for orthogonal projection, such similarities can be preserved. Thus trace ratio *LDA* tends to perform empirically better than ratio trace *LDA* in many

* Corresponding author. Tel.: +852 27887756; fax: +852 27887791.
E-mail addresses: mzhao4@student.cityu.edu.hk (M. Zhao), zhaozhang5@student.cityu.edu.hk (Z. Zhang), eetchow@cityu.edu.hk (T.W.S. Chow).

0031-3203/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2011.10.008

classification problems. In this paper, we will focus on trace ratio *LDA*. For convenience, in this paper we denote it as *TR-LDA*.

Solving trace ratio problem of *LDA* directly has always been a problem, because there is no close-form solution [7]. Several attempts have been proposed to find the optimal solution [3–8]. Guo et al. [3] has pointed out that the original *TR* problem can be converted to an equivalent trace difference problem, which can be solved by a heuristic bisection method. Recently, Wang et al. [4] has proposed another efficient algorithm, called *ITR* algorithm to find the optimal solution based on an iterative procedure, which is faster than the former one. In this paper, we further analyze *ITR* algorithm, and discuss the drawbacks of its training strategy. We then propose a new efficient algorithm, called *ITR-Score* algorithm, to improve the original *ITR* algorithm. The proposed algorithm can be viewed as a greedy strategy to find the optimum of *TR* problem. Hence it is more efficient than the previous ones.

In general, the *TR-LDA* is supervised, which means it requires labeled information. Although *TR-LDA* works pretty well [3,4], it needs considerable number of labeled data in order to be able to deliver satisfactory results. But in many practical cases, obtaining sufficient number of labeled data for training can be problematic because labeling large number of data is time-consuming and costly. On the other hand, unlabeled data may be abundant and can easily be obtained in the real world. Thus, using semi-supervised learning methods [19–24,47], which incorporate both labeled and unlabeled data into learning procedure, has become an effective option instead of only relying on supervised learning. In this paper, we will propose an orthogonal constrained framework for semi-supervised learning. Under such a framework, the *TR-LDA* can be extended to its corresponding semi-supervised version called trace ratio based semi-supervised discriminant analysis (*TR-SDA*). Furthermore, through analyzing the relationship between supervised and semi-supervised *TR* problems, we show that the proposed *ITR-Score* algorithm can be extended to solve semi-supervised *TR* problem.

The main contributions of this paper are summarized as follows:

(1) As an extended algorithm of *TR-LDA*, the proposed *TR-SDA* can find an optimal low-dimensional projection by preserving the discriminative information embedded in the labeled set as well as the geometric information embedded in both labeled and unlabeled set. Also similar to *TR-LDA*, the optimal projection obtained by *TR-SDA* is orthogonal that can preserve the similarity between data points without any change if it is based on Euclidean distance.

(2) We propose a new method called *ITR-Score* algorithm to solve supervised and semi-supervised *TR* problem. By improving the original *ITR* algorithm both from the initialization and training strategy, the proposed method can converge faster. This indicates that *ITR-Score* algorithm is more efficient than the *ITR* algorithm.

(3) We propose an orthogonal constrained framework for semi-supervised learning. Under such a framework, the *TR-SDA* algorithm can be related to several existing semi-supervised algorithms such as *SDA* [19], *Lap-LDA* [21], *SSMMC* [22], *SSDR* [20]. In short, our algorithm can be viewed as an improved or extended method to these algorithms.

(4) The proposed *TR-SDA* can easily be extended to a nonlinear version using kernel trick [32,33]. In this paper, we restrict the nonlinear projection to be in an orthogonal basis of high-dimensional Hilbert space. We then perform linear dimensionality reduction based on such basis. Finally, we connect *TR-LDA*, *TR-SDA* and their corresponding kernel versions in a unified form.

The rest of this paper is organized as follows: In Section 2, we briefly describe the basic idea of *LDA* and *TR-LDA*. We then review the previews work for solving *TR* problem and propose our improved method. In Section 3, we propose an orthogonal constrained framework for semi-supervised learning. We extend *TR-LDA* to its corresponding semi-supervised version *TR-SDA*. In Section 4, we extend our algorithm for solving nonlinear problem using kernel trick. The simulation results are presented in Section 5 and the conclusions are drawn in Section 6.

2. Review of Linear Discriminant Analysis

2.1. Trace ratio problem

In this section, we first review the basic idea of Linear Discriminant Analysis. The goal of *LDA* is to find a linear transformation matrix $W \in R^{D\times d}$, for which the between-class scatter matrix is maximized, while the within-class scatter matrix is minimized. Let $X=\{x_1,x_2,\ldots,x_l\} \in \mathbb{R}^{D\times l}$ be the training set, each x_i belongs to a class c_i={1,2,…,c}. Let l_i be the number of data points in ith class, l be the number of data points in all classes, we define the between-class scatter matrix S_b, within-class scatter matrix S_w and total-class scatter matrix S_t as

$$S_b=\sum_{i=1}^{c} l_i(\mu_i-\mu)(\mu_i-\mu)^T$$
$$S_w=\sum_{i=1}^{c}\sum_{x_i\in c_i}(x_i-\mu_{c_i})(x_i-\mu_{c_i})^T$$
$$S_t=\sum_{i=1}^{l}(x_i-\mu)(x_i-\mu)^T \tag{1}$$

where $\mu_i=1/l_i\sum_{xi\in ci}x_i$ is the mean of data points in the ith class, $\mu_i=1/l\sum_{i=1}^{l}x_i$ is the mean of data points in all classes. The original formulation of *LDA*, called Fisher *LDA* [11] can only deal with binary classification. Two optimization criterions can be used to extend Fisher *LDA* to solving multi-class classification problem. For the first criterion, the optimization of *LDA* can be given by

$$W^*=\arg\max_W Tr((W^TS_wW)^{-1}(W^TS_bW)). \tag{2}$$

For the convenience to distinguish the trace ratio problem introduced in later section, we call the above optimization ratio trace problem. If we assume S_w is nonsingular, the optimization problem in Eq. (2) can be solved by generalized eigen-value decomposition (*GEVD*) as [34]

$$S_bw_k=\tau_kS_ww_k \tag{3}$$

where $w_k \in R^d$ is the eigenvector corresponding to the kth largest eigenvalue τ_k. We then form W^* by the top w_k. Finally, the original high-dimensional set X can be projected into a low-dimensional set $Y\in R^{d\times l}$ by $Y=W^{*T}X$.

Another reasonable optimization criterion of *LDA* is to maximize $Tr(W^TS_bW)$, while minimizing $Tr(W^TS_bW)$. The optimization problem can be given by

$$W^*=\arg\max_{W^TW=I}(Tr(W^TS_bW)/Tr(W^TS_wW)). \tag{4}$$

We call the above problem trace ratio problem. Compared with ratio trace problem in Eq. (2), solving *TR* problem can deliver an empirically better discriminative projection when the classification problem is based on Euclidean distance, as both $Tr(W^TS_bW)$ and $Tr(W^TS_bW)$ directly reflect the Euclidean distances between data points of inter and intra classes. In addition, the optimal projection obtained by *TR* problem is orthogonal, so the similarity between data points can be preserved without any change [4].

But solving the *TR* problem has never been a straightforward issue, because it does not have a closed-form solution [7]. Thus, instead of dealing with *TR* problem directly, many works tend to solve an equivalent trace difference problem [3–7]. Let λ^* be the optimal trace ratio value satisfying

$$\lambda^* = \max_{W^TW=I}(Tr(W^TS_bW)/Tr(W^TS_wW)). \tag{5}$$

According to Guo et al. [3], it follows:

$$\max_{W^TW=I}Tr[W^T(S_b-\lambda^*S_w)W]=0. \tag{6}$$

Thus, inspired by Eq. (6), we cite the theorem in [3] without proof as

Theorem 1. *TR problem can be solved equivalent to find the zero point of the trace difference function defined as*

$$g(\lambda)=\max_{W^TW=I}Tr[W^T(S_b-\lambda S_w)W] \tag{7}$$

i.e., to solve a trace difference equation $g(\lambda^*)=0$, which is called trace difference problem. The optimal projection matrix W^* can then be calculated by

$$W^*=\mathrm{argmax}_{W^TW=I}Tr[W^T(S_b-\lambda^*S_w)W]. \tag{8}$$

2.2. Efficient algorithm for solving TR problem

2.2.1. Previous work

In this section, we first review some previous work for solving *TR* problem. Recently, there are several methods proposed to deal with the problem [3–8]. Guo et al. [3] has solved the trace difference problem using the notion of Foley–Sammon transform. The basic steps of the algorithm are

(1) Initialize λ_1 and λ_2 satisfying $g(\lambda_1)>0>g(\lambda_2)$.
(2) Compute $\lambda=(\lambda_1+\lambda_2)/2$ and $g(\lambda)$.
(3) If $f(\lambda)>0$, let$\lambda_1=\lambda$, else let $\lambda_2=\lambda$.
(4) Iterate the steps (2) and (3) until convergence, the optimal matrix W^* is formed by the d eigenvectors of $S_b-\lambda^*S_w$ corresponding to the d largest eigenvalues.

It is easy to recognize that Guo et al. is equivalent to the heuristic bisection method to find the zero point of trace difference function [7]. Xiang et al. [6] has improved this method by determining a lower bound and an upper bound for λ^*. The binary search can be efficiently achieved when λ_1 and λ_2 are well initialized.

Wang et al. [4] has proposed a more efficient method, called *ITR* algorithm, to solve the *TR* problem. The basic steps of the algorithm are

(1) Initialize the projection matrix W_0 as an arbitrary column-orthogonal matrix.
(2) Compute the trace ratio value $\lambda=Tr(W^TS_bW)/Tr(W^TS_wW)$.
(3) Update $W=\mathrm{argmax}_{W^TW=I}Tr[W^T(S_b-\lambda S_w)W]$.
(4) Iterate the steps (2) and (3) until convergence. The optimal matrix W^* is formed by the d eigenvectors of $S_b-\lambda^*S_w$ corresponding to the d largest eigenvalues.

The *ITR* algorithm is proved empirically efficient than the former algorithm [4]. A more theoretical discussion can be seen in the work of Jia et al. [5]. According to Jia et al., the *ITR* algorithm is equivalent to the naive Newton–Raphson method for solving *TR* problem. Due to the nature of the Newton–Raphson method, it is generally faster than the simple bisection method in [3].

2.2.2. A more efficient algorithm

Though *ITR* algorithm works well for solving *TR* problem, it has its drawbacks. First, the initialized orthogonal matrix W_0 is arbitrary and hard to choose. In some cases when W_0 is well chosen, the algorithm is able to converge relatively faster, while in most cases, inappropriate W_0 dramatically increases the number of iterations. On the other hand, initializing λ_0 seems much easier for any λ_0 satisfying $g(\lambda_0)\geq 0$. Thus, similar to the work in [3], we can initialize the trace ratio value λ_0 instead of the projection matrix W_0. Furthermore, since a proper initialized value can speed up the training procedure, we need to determine a lower bound of λ^* as λ_0. We have the following Theorem 2:

Theorem 2. *The lower bound of λ^* can be given by $Tr(S_b)/Tr(S_w)$.*

The proof of Theorem 2 is given in Appendix A. Therefore, in practice we can set $\lambda_0=(Tr(S_b))/(Tr(S_w))$ for initialization. Second, the method of *ITR* algorithm has chosen d eigenvectors corresponding to the d largest eigenvalues of $S_b-\lambda^*S_w$ to form W^* maximizing the trace difference value $Tr[W^T(S_b-\lambda^*S_w)W]$. But these top eigenvectors cannot maximize the trace ratio value $(Tr(W^TS_bW))/(Tr(W^TS_wW))$. Thus, we need to find better choices of d eigenvectors satisfying $\max[(Tr(W^TS_bW))/(Tr(W^TS_wW))]$. Motivated by all these issues, we propose an improved *ITR* algorithm, called *ITR-Score* algorithm in this paper, to solve the problem. For all the eigenvectors of $S_b-\lambda^*S_w$, our algorithm computes a score $s_i=(w_i^TS_bw_i)/(w_i^TS_ww_i)$ for each of eigenvector w_i. We then choose d eigenvectors having the largest scores to form the optimal matrix W^*. The basic steps of the proposed algorithm are in Table 1.

One may easily note that the main difference between *ITR* algorithm and the proposed *ITR-Score* algorithm is in Step 3–4 which are about choosing d eigenvectors of $S_b-\lambda_tS_w$ to update W_t. The method of *ITR* algorithm has chosen d eigenvectors corresponding to the largest eigenvalues of $S_b-\lambda_tS_w$ to form W_t. But the renewed W_t are not necessarily formed by these top eigenvectors as they may not be able to maximize $(Tr(W_t^TS_bW_t))/(Tr(W_t^TS_wW_t))$. On the other hand, the proposed *ITR-Score* algorithm has chosen d eigenvectors having the largest scores of $s_i=(w_i^TS_bw_i)/(w_i^TS_ww_i)$ to form W_t. This procedures can be viewed as a greedy algorithm that optimizes $\max_i\sum_{k=1}^d(w_{i_k}^TS_bw_{i_k})/(w_{i_k}^TS_ww_{i_k})$, which is an approximation to

$$\max_i\frac{\sum_{k=1}^d w_{i_k}^TS_bw_{i_k}}{\sum_{k=1}^d w_{i_k}^TS_ww_{i_k}}=\max\frac{Tr(W^TS_bW)}{Tr(W^TS_wW)}, \tag{10}$$

where $i=\{i_1,i_2,\ldots,i_d\}$ is a certain permutation chosen from $\{1,2,\ldots,D\}$. Hence for any initial $\lambda_t<\lambda^*$, the updated λ_{t+1} proposed in our algorithm is usually greater than that in the *ITR* algorithm, which indicates that our *ITR-Score* algorithm is more efficient than the *ITR* algorithm. But in practice, it may also confront with the situation that the updated λ_{t+1} of our *ITR-Score* algorithm is smaller than that of *ITR* algorithm, especially when the iterative process is close to the convergence. Hence in this case, a feasible solution is to update λ_{t+1} by choosing the larger one. This can guarantee that the updated trace ratio value is always no smaller than that of *ITR* algorithm.

2.2.3. Convergence analysis

We next analyze the convergence of our *ITR-Score* algorithm. It has been rigorously proved that for any initial $\lambda_t<\lambda^*$, the updated

Table 1
ITR-Score algorithm for solving trace ratio problem.

(1) Initialize $\lambda_0=Tr(S_b)/Tr(S_w)$.
(2) Compute the eigen-decomposition of $S_b-\lambda_tS_w$ as $(S_b-\lambda_tS_w)w_i=\tau_iw_i$, where $w_i(i=1,2,\ldots,D)$ is the eigenvector of $S_b-\lambda_tS_w$.
(3) Compute the score $s_i=(w_i^TS_bw_i)/(w_i^TS_ww_i)$ for each of eigenvector w_i.
(4) Choose the top d eigenvectors w_i having the d largest scores s_i to form W_t.
(5) Update $\lambda_{t+1}=(Tr(W_t^TS_bW_t))/(Tr(W_t^TS_wW_t+\alpha I_d))$.
(6) Iterate the steps (2–5) until $|\lambda_{t+1}-\lambda_t|<\varepsilon$. Output W^*.

λ_{t+1}^{ITR} of *ITR* algorithm is larger than λ_t [4,24]. We next prove that for *ITR-Score* algorithm, the updated λ_{t+1} satisfies (1) $\lambda_{t+1} \geq \lambda_{t+1}^{ITR} > \lambda_t$ and (2) $\lambda_{t+1} \leq \lambda^*$, which indicates that the *ITR-Score* algorithm can converge to the global optimum.

The first inequality is straightforward as analyzed in the last paragraph of Section 2.2.2. Hence we only prove the second inequality.

Since $\lambda_{t+1} = (Tr(W_t^T S_b W_t))/(Tr(W_t^T S_w W_t))$, we have $Tr(W_t^T S_b W_t) - \lambda_{t+1} Tr(W_t^T S_w W_t) = Tr(W_t^T(S_b - \lambda_{t+1} S_w)W_t) = 0$. According to the definition of trace difference function, i.e. $g(\lambda) = \max_{W^TW=I} Tr[W^T(S_b - \lambda S_w)W]$, it follows

$$\begin{aligned} g(\lambda_{t+1}) &= \max_{W^TW=I} Tr(W^T(S_b - \lambda_{t+1}S_w)W) \\ &\geq Tr(W_t^T(S_b - \lambda_{t+1}S_w)W_t) = 0. \end{aligned}$$

This indicates that $g(\lambda_{t+1}) \geq 0$. In addition, according to Theorem 1, it follows $g(\lambda^*) = Tr(W^{*T}(S_b - \lambda^* S_w)W^*) = 0$, where W^* is its optimal projection matrix. We thus have

$$\begin{aligned} g(\lambda^*) = 0 = Tr(W^{*T}(S_b - \lambda^* S_w)W^*) &\geq Tr(W_t^T(S_b - \lambda^* S_w)W_t) \\ = g(\lambda_{t+1}) + (\lambda_{t+1} - \lambda^*)Tr(W_t^T S_w W_t). \end{aligned}$$

To satisfy this inequality, i.e. $g(\lambda_{t+1}) + (\lambda_{t+1} - \lambda^*)Tr(W_t^T S_w W_t) \leq 0$, we can only have $\lambda_{t+1} - \lambda^* \leq 0$ as $g(\lambda'_{t+1}) \geq 0$ and $Tr(W_t^T S_w W_t) \geq 0$ (S_w is semi-positive definite). Therefore, we prove $\lambda_{t+1} \leq \lambda^*$.

2.2.4. Singularity case

In the above algorithm, it assumes S_w is nonsingular. Otherwise, if W is in the null space of S_w, i.e. $W^T S_w W = 0$, the trace ratio value $Tr(W^T S_b W)/Tr(W^T S_w W)$ can go infinite. This is the so called singularity problem and can often occur when the null space of S_w has dimensionality d' larger than d (the reduced dimensionality) [5,6]. This is because for $d' > d$, the d column vectors of W can all lie in the null space of S_w hence causing $Tr(W^T S_w W) = 0$. To solve this problem, Xiang et al. [6] and Jia et al. [5] choose to find the optimal solution by maximizing $Tr(W^T S_b W)$ in the null space of S_w as

$$W^* = \text{argmax}_{W^T S_w W = 0} Tr(W^T S_b W) \quad \text{or}$$
$$W^* = \text{argmax}_{W^T S_w W = 0} Tr(W^T S_t W).$$

One may easily find that the above algorithm is similar to that of null space *LDA* (*NLDA*) [25] or Discriminative Common Vectors (*DCV*) [28]. Hence the work in Xiang et al. [6] and Jia et al. [5] solve *TR* problem in two cases: for $d \leq d'$, using the algorithm of *NLDA* (or *DCV*), for $d > d'$, using the algorithm in the previous work [3,4].

In this paper, we solve singularity problem by adding a regularization term to the objective function of *TR* problem

$$W^* = \text{argmax}_{W^TW=I}(Tr(W^T S_b W)/(Tr(W^T S_w W + \alpha I_d)))$$

where $\alpha I_d \in \mathbb{R}^{d \times d}$ is a multiply of identity matrix. From the above equation, we can see that no matter whether W is in the null space of S_w, it always holds $Tr(W^T S_w W + \alpha I_d) > 0$. Hence the singularity problem can be solved. In addition, if W^* converges to the null space of S_w, i.e. $W^T S_w W = 0$ and $Tr(W^T S_w W + \alpha I_d) = \alpha d$, then it follows $W^* = \text{argmax}_{W^T S_w W = 0} Tr(W^T S_b W)$. This indicates that for singularity case, the optimal solution of *TR-LDA* is equivalent to that of *NLDA* (or *DCV*). But our algorithm is more efficient as it does not need to consider the cases of $d \leq d'$ and $d > d'$.

3. Trace ratio based semi-supervised dimensionality reduction

3.1. Orthogonal constrained semi-supervised learning framework

The algorithms to solve TR problem are all supervised. In order to use unlabeled data points to achieve satisfactory results, there are many works incorporating both labeled and unlabeled set into learning procedure [19–23]. In this paper, we first introduce a semi-supervised learning framework. Denote $X = \{X^l, X^u\}$ representing the whole dataset, $X^l = \{x_i\}_{i=1}^l$ is the labeled set corresponding with the labeled matrix $Y = \{y_i\}_{i=1}^l$ and $X^u = \{x_i\}_{i=l}^{l+u}$ is the unlabeled set, the framework can be given as

$$f^* = \arg\min_f [V^l(X^l, f) + \gamma V^u(X^u, f)], \tag{11}$$

where V^l and V^u are certain cost functions corresponding to the labeled and unlabeled set, $f = [f(x_1), \cdots f(x_{l+u})]^T$ is an output space associated with a basis, which can be represented either by Euclidean space $f = W^T X$ or Reproducing Kernel Hilbert Space $f(x) = \sum_{i=1}^{l+u} \alpha_i K(x_i, x)$ [32,33], γ is a parameter control the tradeoff between two cost functions.

The goal of minimizing the labeled cost function V^l is to find an optimal output space f^* to preserve the discriminative structure embedded in a low-dimensional set. Specifically, after performing dimensionality reduction under such cost function, it is our objective that the distance between data points in the same class is close, while those in different classes are far apart. Furthermore, when dealing with dimensionality reduction problems, we often use a matrix with pairwise form to describe the distance relationship between data points regarding whether they are close or far apart [14]. Therefore, based on the notation of matrix with pairwise form, we give the labeled cost function V^l as

$$V^l\left\{X^l, f\right\} = \frac{1}{2}\min_f \left(\sum_{i,j=1}^{l} c_{ij}^d \|f(x_i) - f(x_j)\|^2 + \lambda \sum_{i,j=1}^{l} c_{ij}^s \|f(x_i) - f(x_j)\|^2 \right) \tag{12}$$

where $C^d = \{c_{ij}^d\}$ and $C^s = \{c_{ij}^s\}$ are the cost matrixes penalizing the pairwise distances for any two data points of inter and intra class, respectively. Let $D^d = \{d_{ii}^d\}_{i=1}^l$ and $D^s = \{d_{ii}^s\}_{i=1}^l$ be the diagonal matrixes satisfying $d_{ii}^d = \sum_{j=1}^l c_{ij}^d$ and $d_{ii}^s = \sum_{j=1}^l c_{ij}^s$, the cost function in Eq. (12) can then be rewritten as

$$V^l\{X^l, f\} = \min_f Tr(f^T L^d f + f^T L^s f), \tag{13}$$

where $L^d = D^d - C^d$ and $L^s = D^s - C^s$. In addition, if we regard C^d or C^s as a weight matrix of a graph, L^d or L^s can be viewed as a graph Laplacian matrix in spectral graph theory [36].

On the other hand, the unlabeled cost function V^u is optimized to best preserve the geometric structure of dataset. Assuming this geometric structure is smoothly embedded in a low-dimensional manifold, it can be approximated using the graph Laplacian associated with both labeled and unlabeled set [14]. Let $G = (V, E)$ denotes the above graph, where V is vertex set of the graph representing both labeled and unlabeled samples, and E is edge set containing the neighborhood information between two nearby data points. Let C^u be the corresponding weight matrix of E, a nature method for defining the weight matrix is using Gaussian function

$$c_{ij}^u = \begin{cases} \exp(-(\|x_i - x_j\|^2)/\sigma^2) & \text{if } x_i \in N_k(x_j) \text{ or } x_j \in N_k(x_j) \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

where $N_k(x_i)$ denotes the k nearest neighborhood set of x_i. The unlabeled cost function V^u can then be written as

$$V^u(X^u, f) = \frac{1}{2}\min_f \sum_{i,j=1}^{l+u} c_{ij}^u \|f(x_i) - f(x_j)\|^2 = \min_f Tr(f^T L^u f). \tag{15}$$

If we only consider the linear embedded space $f = W^T X$ (an efficient nonlinear extension based on reproducing kernel Hilbert space is discussed in Section 4) and impose the projection matrix W with an orthogonal constraint, the total cost function can be written as

$$\min_{W^TW=I} Tr[W^T(XL^dX^T+\lambda XL^sX^T+\gamma XL^uX^T)W]. \tag{16}$$

3.2. Semi-supervised trace ratio problem

The above framework in Eq. (16) has motivated us to extend the *TR* problem to its corresponding semi-supervised version. To establish the relationship between framework and *TR* problem, we first rewrite between-class scatter matrix and within-class scatter matrix using pairwise form. Let C^d and C^s be defined

$$c_{ij}^d=\begin{cases}\frac{1}{l_k}-\frac{1}{l} & x_i \text{ and } x_j \text{ belongs to the same class}\\ -\frac{1}{l} & \text{otherwise}\end{cases}$$

$$c_{ij}^s=\begin{cases}\frac{1}{l_k} & x_i \text{ and } x_j \text{ belongs to the same class}\\ 0 & \text{otherwise}\end{cases}. \tag{17}$$

According to He et al. [13], we have

$$\frac{1}{2}\sum_{i,j=1}^{l} c_{ij}^d\left\|W^Tx_i-W^Tx_j\right\|^2=Tr(W^TXL^dX^TW)=-Tr(W^TS_bW)$$

$$\frac{1}{2}\sum_{i,j=1}^{l} c_{ij}^s\left\|W^Tx_i-W^Tx_j\right\|^2=Tr(W^TXL^sX^TW)=Tr(W^TS_wW). \tag{18}$$

By putting Eq. (18) into Eq. (13) and let $f=W^TX$, one may easily find the labeled cost function $V^l=\min_W[-Tr(W^TS_bW)+\lambda Tr(W^TS_wW)]=\max_W Tr(W^TS_bW-\lambda Tr(W^TS_wW)]$, which is exactly the trace difference function of *TR* problem. For convenience, we denote L_b, L_w, L_m as the graph Laplacian matrix of between-class scatter matrix, within-class scatter matrix and manifold matrix, respectively. Let $\gamma=\lambda\lambda_m$ and $L_b=L^d$, $L_w=L^s$ and $L_m=L^u$, where L^d, L^s and L^u are defined in Eq. (17) and (14), the semi-supervised trace ratio problem can be written as

$$W^*=\arg_{W^TW=I}\min W^T(-XL_bX^T+\lambda^*(XL_wX^T+\lambda_mXL_mX^T))W. \tag{19}$$

The corresponding semi-supervised version of trace ratio function and trace difference function can be given by

$$W^*=\operatorname{argmax}_{W^TW=I}\frac{Tr(W^TXL_bX^TW)}{Tr(W^T(XL_wX^T+\lambda_mXL_mX^T)W)} \tag{20}$$

$$g(\lambda)=\max_{W^TW=I}Tr[W^T(XL_bX-\lambda(XL_wX+\lambda_mXL_mX^T)W]. \tag{21}$$

One may easily find for semi-supervised *TR* problem, an extra manifold matrix $\lambda_mX^TL_mX$ is added to the original object function in the *TR* problem. The manifold matrix can be positive semi-definite due to the graph Laplacian property. Thus, we can simply use the proposed *ITR-Score* algorithm to solving semi-surprised *TR* problem by replacing XL_wX with $XL_wX+\lambda_mXL_mX^T$. The basic steps of *ITR-Score* algorithm for solving semi-supervised *TR* problem are in Table 2.

Table 2
ITR-Score algorithm for solving semi-supervised trace ratio problem.

```
(1) Initialize $\lambda_0=(Tr(XL_bX^T))/(Tr(XL_wX^T+\lambda_mXL_mX^T))$.
(2) Compute the eigen-decomposition of $XL_bX^T-\lambda_t(XL_wX^T+\lambda_mXL_mX^T)$ as
    $(XL_bX^T-\lambda_t(XL_wX^T+\lambda_mXL_mX^T)w_i=\tau_iw_i$, where $w_i(i=1,2,\ldots,D)$ is the
    eigenvector of $XL_bX^T-\lambda_t(XL_wX^T+\lambda_mXL_mX^T)$.
(3) Compute the score $s_i=(w_i^TS_bw_i)/(w_i^TS_ww_i)$ corresponding to each
    eigenvector $w_i$.
(4) Choose the top $d$ eigenvectors $w_i$ having the $d$ largest scores $s_i$ to form $W_t$.
(5) Update $\lambda_{t+1}=(Tr(W_t^TXL_bX^TW_t))/(Tr(W_t^T(XL_wX^T+\lambda_mXL_mX^T)W_t+\alpha I_d))$.
(6) Iterate the steps (2–5) until $|\lambda_{t+1}-\lambda_t|<\varepsilon$. Output $W^*$.
```

4. Kernelization

The proposed *TR-SDA* is a linear algorithm. In this section, we will extend it to solve the nonlinear problem using kernel trick [32,33]. For convenience, we denote the kernel version of *TR-SDA* as *TR-KSDA*.

The basic idea of the kernel trick is to map the original data space to a high-dimensional Hilbert space given by $\phi:X\to F$, then perform linear dimensionality reduction on the new space. Let $\phi(X)=\{\phi(x_1),\phi(x_2),\ldots,\phi(x_{l+u})\}$ be such high-dimensional space, we assume the map can be implicitly implemented in a kernel function $K(x_i,x_j)=\phi(x_i)^T\phi(x_i)$. The goal of *TR-KSDA* is to find an optimal projection $W^{\phi*}\in\mathbb{R}^{d\times(l+u)}$ satisfying

$$W^{\phi*}=\operatorname{argmax}_{W^{\phi T}W^{\phi}=I}\frac{Tr(W^{\phi T}\phi(X)L_b\phi(X)^TW^{\phi})}{Tr(W^{\phi T}\phi(X)(L_b+L_m)\phi(X)^TW^{\phi})}. \tag{22}$$

Note $\phi(X)$ is not available as it is only implicit. Thus we cannot directly solve the problem in Eq. (22). In order to compute the optimal projection $W^{\phi*}$, we can add some restricts to W^{ϕ}, making the solution to the problem in Eq. (22) available. Supposing the QR decomposition of $\phi(X)$ is $\phi(X)=QR$, where R is an upper triangular matrix, and Q is an orthogonal matrix satisfying $Q^TQ=I$. Then, we have

$$\phi(X)^T\phi(X)=R^TR=K \tag{23}$$

which indicates that R^TR can be viewed as Cholesky decomposition factorization of K. Furthermore, since Q is now an orthogonal basis of $\phi(X)$, assuming W^{ϕ} is mapped into the span of Q, we then have $W^{\phi}=QV^{\phi}$, where $V^{\phi}\in\mathbb{R}^{d\times(l+n)}$ is an orthogonal matrix with the columns satisfying $V^{\phi T}V^{\phi}=I$. Thus, the original object function in Eq. (22) can be rewritten as

$$V^{\phi*}=\underset{V^{\phi T}V^{\phi}=I}{\arg\ \max}\frac{Tr(V^{\phi T}RL_bR^TV^{\phi})}{Tr(V^{\phi T}R(L_b+L_m)R^TV^{\phi})}. \tag{24}$$

The output data points in the reduced space can be given by

$$Y^{\phi}=(W^{\phi*})^T\phi(X)=(V^{\phi*})^TQ^TQR=(V^{\phi*})^TR. \tag{25}$$

The basic steps of using *ITR-Score* algorithm to solve *TR-KSDA* are shown in Table 3.

Recalling the objective function of *TR-SDA* in Eq. (20), it is noticed that it has the same form as *TR-KSDA* in Eq. (24). If we rewrite the objective function of *TR-LDA*, *TR-SDA* and their corresponding kernel version in the form of $V^*=\operatorname{argmax}_{V^TV=I}((Tr(V^TRM_1R^TV))/(Tr(V^TRM_2R^TV)))$, these different algorithms can be connected according to different choices of matrix M_i and R in Table 4.

Table 3
ITR-Score algorithm for solving *TR-KSDA*.

```
(1) Perform Cholesky decomposition to the kernel matrix $K=R^TR$.
(2) Initialize $\lambda_0=(Tr(RL_bR^T))/(Tr(RL_wR^T+\lambda_mRL_mR^T))$
(3) Compute the eigen-decomposition of $RL_bR^T-\lambda_t(RL_wR^T+\lambda_mRL_mR^T)$ as
    $(RL_bR^T-\lambda_t(RL_wR^T+\lambda_mRL_mR^T)v_i^{\phi}=\tau_iv_i^{\phi}$, where $v_i^{\phi}(i=1,2,\ldots,D)$ is the
    eigenvector of $RL_bR^T-\lambda_t(RL_wR^T+\lambda_mRL_mR^T)$.
(4) Compute the score $s_i=(v_i^{\phi T}S_bv_i^{\phi})/(v_i^{\phi T}S_wv_i^{\phi})$ corresponding to each
    eigenvector $v_i^{\phi}$.
(5) Choose the top $d$ eigenvectors $v_i^{\phi}$ having the $d$ largest scores $s_i$ to form $V_t^{\phi}$.
(6) Update $\lambda_{t+1}=(Tr(V_t^{\phi T}RL_bR^TV_t^{\phi}))/(Tr(V_t^{\phi T}(RL_wR^T+\lambda_mRL_mR^T)V_t^{\phi}+\alpha I_d))$.
(7) Iterate the steps (2–5) until $|\lambda_{t+1}-\lambda_t|<\varepsilon$. Output $W^{\phi*}=(V^{\phi*})^TR$.
```

Table 4
Connection between *TR-LDA*, *TR-SDA*, *TR-KLDA* and *TR-KSDA*.

Method	Matrix M_1	Matrix M_2	Matrix R	Output Y
TR-LDA	L_b	L_w	X	$Y=V^{*T}X$
TR-SDA	L_b	L_w+L_m	X	$Y=V^{*T}X$
TR-KLDA	L_b	L_w	$R^TR=K$(Chol.)	$Y=V^{*T}R$
TR-KSDA	L_b	L_w+L_m	$R^TR=K$(Chol.)	$Y=V^{*T}R$

5. Related work

In the paper, we propose a semi-supervised version of *TR-LDA*. It is to the best of our knowledge that there are several recently proposed semi-supervised dimensionality reduction methods using the same objectives of this paper [19–22]. By analyzing the strategy and mechanism of these algorithms, we show that our proposed algorithm is, in fact, an improved or extended method among these algorithms.

5.1. Relation to SDA [19], Lap-LDA [21] and SS-CCA [23]

The method of *SDA* algorithm [19] can be viewed as adding a manifold regularization term to the original objective function of Regularized *LDA*. The objective function of *SDA* is

$$J(W)=\max_W \frac{\left|W^TXL_bX^TW\right|}{\left|W^T(XL_wX+\lambda_1XL_mX^T+\lambda_2I)W\right|} \tag{26}$$

where I is Tikhonov regularization term choosing an identity matrix for imposing the smoothes of possible solution, λ_i are the parameters balanced the tradeoff of two regularization terms. *Lap-LDA* [21] is another semi-supervised dimensionality reduction method with the objective function given as

$$\min_W \|Y-X^TW\|_F^2+\lambda_1 Tr(W^TXL_mX^TW)+\lambda_2 Tr(W^TW). \tag{27}$$

Note that *Lap-LDA* is proposed under a least square framework. Considering the relationship between *LDA* and multivariate linear regression with certain class indicator matrix [10], *Lap-LDA* can be viewed equivalent to *SDA*. The solution of *SDA* and *Lap-LDA* can then be obtained by solving generalized eigen-value decomposition problem (*GEVD*) as

$$XL_bX^Tw_k=\tau_k(XL_wX+\lambda_1XL_mX^T+\lambda_2I)w_k$$

where $w_k \in \mathbb{R}^d$ is the eigenvector corresponding to the kth largest eigenvalue τ_k, note $\lambda_2=0$ for *Lap-LDA* as it has no Tikhonov regularization term.

We next build the relationship between *SDA* (or *Lap-LDA*) and *TR-SDA*. Actually, as described in [24], given an uncorrelated constraint, the objective function of *LDA* can be rewritten in a form of trace ratio criterion $\max_{W^TS_tW=I}((Tr(W^TS_bW))/(Tr(W^TS_wW)))$. Here, we extend this form to a semi-supervised vision as

$$\max_{W^T(S_t+\lambda_mX^TL_mX)W=I}(Tr(W^TS_bW/(Tr(W^T(S_w+\lambda_mX^TL_mX)W))). \tag{26}$$

Since $S_t=S_w+S_b$, the objective function in Eq. (26) can be rewritten as $\max_{W^T(S_t+\lambda_mX^TL_mX)W=I}Tr(W^TS_bW)$ and the optimal solution can be obtained by solving *GEVD* of $S_bW=\tau_k(S_t+\lambda_mXL_mX^T)W$, which is equivalent to that of *SDA*. Hence from Eqs. (26) and (20), we can observe that the main difference between *SDA* (or *Lap-LDA*) and *TR-SDA* is whether the objective function is based on uncorrelated constraint or orthogonal constraint. When the classification problem is based on Euclidean Distance, our proposed *TR-SDA* algorithm is superior to *SDA* as elaborated in the analysis of Section 2. Another semi-supervised method having the similar thought is *SS-CCA* [23]. Given a certain class indicator matrix [18], *SS-CCA* can be equivalent to *SDA*. Thus our proposed *TR-SDA* is also superior to *SS-CCA*.

5.2. Relation to SSMMC [22] and MMC [15]

Other semi-supervised and supervised learning algorithms sharing the same concept include *SSMMC* [22] and *MMC* [15]. The basic objective function of *SSMMC* is

$$J(W)=\max_{W^TW=I}Tr[W^T(XL_bX^T-\lambda_1XL_wX-\lambda_2XL_mX^T)W]. \tag{27}$$

We can observe that there is a great similarity between Eq. (27) and our proposed semi-supervised trace difference function in Eq. (21). But according to the work in [22], λ_i is empirically selected and adjusted by a 5-fold cross validation. This can be arbitrary time-consuming. On the other hand, by adjusting λ_m slightly in Eq. (20) so that $Tr(XL_wX^T)$ and $Tr(XL_mX^T)$ are in the same level, the optimal trace ratio value λ^* and the projection matrix W^* can be found using an iterative procedure. Thus our algorithm can be seen as a parameter-adaptive algorithm finding the optimal solution; this is a great improvement to *SSMMC*. In addition, by fixing $\lambda_1=1$ and $\lambda_2=0$, the objective function in Eq. (27) is equivalent to that of *MMC* [15]

$$J(W)=\max_{W^TW=I}Tr[W^T(XL_bX^T-XL_wX)W]. \tag{28}$$

We can observe that *MMC* can be seen as an approximated solution to the *TR* problem with $\lambda^*=1$. Since *MMC* cannot preserve the geometric structure (it has no manifold terms), our algorithm is superior to it.

5.3. Relation to SSDR [20]

Zhang et al. have proposed another semi-supervised dimensionality reduction, called *SSDR* [20] using a must-link and cannot-link constraints. Though it is different from our algorithm as it is based on pairwise constraints instead of the labeled information directly, it can be covered into our framework. Let C and M be the sets of cannot-links must-links, respectively, the objective function of *SSDR* can be formulated as

$$J(W)=\max_{W^TW=I}\left[\frac{1}{2n^2}\sum_{i,j}\|W^Tx_i-W^Tx_j\|^2+\frac{\alpha}{2n_C}\sum_{(x_i,x_j)\in C}\|W^Tx_i-W^Tx_j\|^2 -\frac{\beta}{2n_M}\sum_{(x_i,x_j)\in M}\|W^Tx_i-W^Tx_j\|^2\right] \tag{28}$$

where n_C and n_M are the number of cannot-links and must-links, α and β are two parameter balanced the tradeoff of two terms. Recalling the framework in Eq. (21), let

$$c_{ij}^d=\begin{cases}-(\alpha/n_C) & \text{if } x_i \text{ and } x_j \text{ belong to different classes}\\ 0 & \text{otherwise}\end{cases}$$

$$c_{ij}^s=\begin{cases}\alpha/n_M & \text{if } x_i \text{ and } x_j \text{ belong to the same class}\\ 0 & \text{otherwise}\end{cases}$$

$$c_{ij}^u=-(1/n^2) \quad \text{for all data set}.$$

The algorithm can be equivalent to *SSDR*.

6. Simulations and results

We will evaluate our algorithms with several synthetic datasets and real world datasets. For synthetic datasets, we use *2d* Gaussian dataset to show the discriminative boundary learned by our algorithm and a *3d* Gaussian dataset to visualize the output set in a *2d* reduced space. We also use a two-moon dataset to deal with nonlinear classification problem and show the discriminative boundary learned by our algorithm. In addition, we demonstrate visualization and classification problem on four real world datasets including the *UMIST* dataset [37], *ORL* dataset [38], *USPS* dataset [39] and *MNIST* dataset [40]. We also compare the performance of our proposed algorithms with other state-of-art algorithms.

6.1. Toy examples for synthetic dataset

Three toy examples based on *2d* Gaussian dataset, *3d* Gaussian dataset and two moon dataset are studied. In the first toy example, we generated a *2d* dataset with two classes, each of which follows a Gaussian distribution. In this dataset, we randomly selected two data points per class as labeled set and the remaining as unlabeled set. Fig. 1 shows the boundary obtained by *LDA*, *SDA*, *TR-LDA* and *TR-SDA*. The results show that both *SDA* and *TR-SDA* are superior to *LDA* and *TR-LDA*, as the boundaries learned by *SDA* and *TR-SDA* can directly divide the data points into two classes, while for *LDA* and *TR-SDA*, some of data points may be divided into false class. The improved performance of our proposed algorithm show they are able to improve *LDA* and *TR-LDA* by incorporating both labeled and unlabeled set to preserve the geometrical structure of dataset. In addition, *TR-SDA* performs better than *SDA*. This enhanced performance is believed to be due to the fact that optimal projection obtained by *TR-SDA* is orthogonal, which results in preserving the similarities (Euclidean distance) between data points. In contrast, *SDA* may change such similarities as the optimal projection of *SDA* is not required to be orthogonal, When evaluating the similarities (Euclidean distance) between data points, it may put different weights on different projection directs.

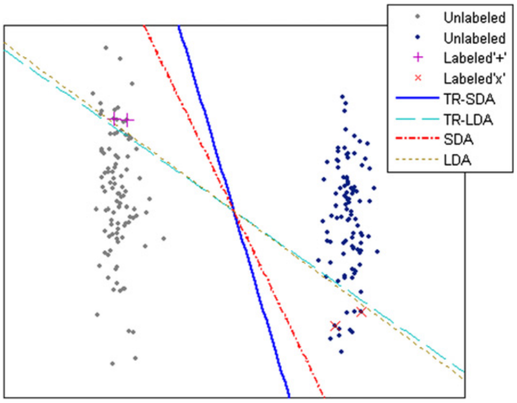

Fig. 1. Boundary obtained by *LDA*, *SDA*, *TR-LDA* and *TR-SDA*: *2d* Gaussian dataset.

In the second toy example, we generated a *3d* dataset with two classes, each of which are Gaussian distributed. In each class, two data points were selected as labeled set and the remaining as unlabeled set. We used *SDA* and *TR-SDA* to perform dimensionality reduction. Fig. 2 shows the output set in the reduced *2d* space of *SDA* and *TR-SDA*. It shows that our proposed *TR-SDA* is superior to *SDA*. In Fig. 2c, it is clear that in the reduced space the data points in a class are pulled together, while the data points in different classes appear to be far apart. In contrast, Fig. 2b shows that *SDA* is unable to deliver a clear class boundary.

In the third toy example, we generated another dataset with two classes, each of which follows a half-moon distribution. In each class, two data points were selected as labels set and the remaining as unlabeled set. Since the distribution of the two moon dataset is non-Gaussian, we only performed *KSDA* and *TR-KSDA* for dimensionality reduction. Fig. 3 shows the gray images of reduced space learned by *KSDA* and *TR-KSDA*. The value of each pixel in the images represents

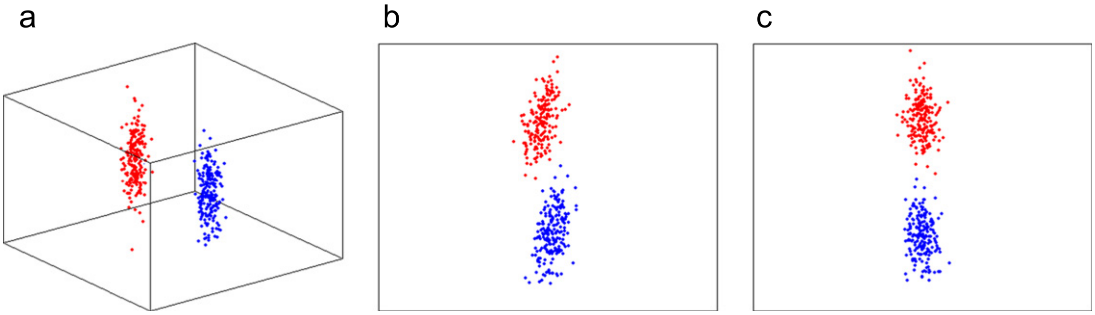

Fig. 2. Output set in the reduced *2d* space of *SDA* and *TR-SDA*: *3d* Gaussian dataset (a) original set, (b) output set of *SDA* and (c) output set of *TR-SDA*.

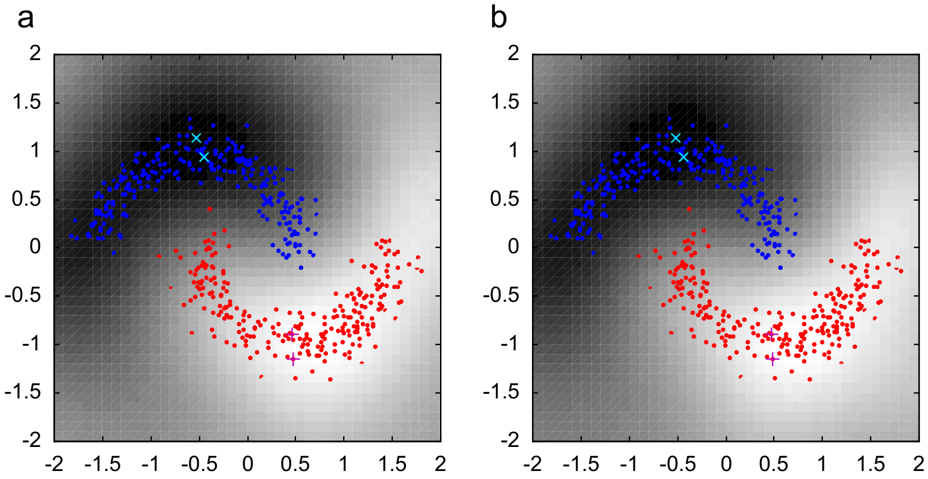

Fig. 3. Gray image of reduced space learned by *KSDA* and *TR-KSDA*: two moon dataset (a) *KSDA* and (b) *TR-KSDA*.

the distance difference from a pixel to its nearest labeled data points after dimensionality reduction by *KSDA* and *TR-KSDA*. In this example, we set the dimensionality of projection as 1. From Fig. 3a and b, we can see both *KSDA* and *TR-KSDA* can obtain a desired classification boundary indicating that the two algorithms can deal with nonlinear problems. Our proposed *TR-SDA* can deliver slightly better performance than *SDA*. The sketch of two half-moon learned by *TR-KSDA* is smoother and clearer than that learned by *KSDA*.

6.2. Sub-manifold visualization

We demonstrate the sub-manifold visualization of our proposed *TR-SDA* algorithm and compare it with *PCA*, *LPP* and *SDA*. In this study, two real-world dataset including the *UMIST* face dataset [37] and *USPS* handwritten digit dataset [39] are used.

In the *UMIST* face dataset, we selected five individuals to illustrate the sub-manifolds of the dataset. For each individual, we randomly selected four data points as labeled set and the remaining as unlabeled set. Fig. 4 shows the *2d* sub-manifolds learned by *PCA*, *LPP*, *SDA* and *TR-SDA*. From the results in Fig. 4a and b, we can see that in unsupervised method such as *PCA* and *LPP*, the sub-manifold structure, i.e. the ordering of poses from profile to frontal views, can be well preserved. *LPP* delivers slightly better performance than *PCA*, as the manifold lines of face poses are more smoothly preserved. This improved performance is mainly due to the characteristics of *LPP* that local information embedded in dataset is preserved. But we can also see from Fig. 4a and b that the boundaries of sub-manifolds in different classes are heavily overlapped and confused, which means both *PCA* and *LPP* cannot preserve the discriminative structure. On the other hand, from Fig. 4c and d, we can see by providing discriminative information based on the labeled set, *SDA* and *TR-SDA* are able to preserve the discriminative structure as well as geometric structure. In Fig. 4d, it demonstrates that our proposed *TR-SDA* algorithm is able to outperform *SDA* in a way that the sub-manifold of each individual is closely conglomerated, while those belonging to different individuals are clearly separated. Fig. 5 shows the sub-manifold of a typical class learned by *TR-SDA*, which represents the face subset of one individual. From Fig. 5 we can see that the poses of faces are turned from frontal views to profile views along the red lines indicating *TR-SDA* can smoothly preserve the sub-manifold of face subset. In contrast, *SDA* cannot preserve the sub-manifold of each class satisfactory as there are two classes seriously overlapped in Fig. 4c.

In the *USPS* handwritten digit dataset, we selected four digits 0–3 to illustrate the sub-manifolds of the dataset. For each digit, we randomly selected fifty data points as labeled set and the remaining as unlabeled set. Fig. 6 shows that the *2d* sub-manifolds learned by *PCA*, *LPP*, *SDA* and *TR-SDA*. In Fig. 6 we can see that the sub-manifolds learned by *LPP*, *SDA* and *TR-SDA* in Figs. 6b–d are much better than those learned by *PCA* in Fig. 6a. The local structures are smoothly preserved in *LPP*, *SDA* and *TR-SDA*, while in *PCA*, these localities are mostly overlapped. This indicates that the *USPS* dataset has lots of local information, hence the local method (*LPP*) or other methods, which involve local strategy (*SDA*, *TR-SDA*), can preserve such useful local information. In contrast, the global method (*PCA*) clearly cannot preserve the locality. In addition, the results in Fig. 6c and d show that the boundary between different classes learned by *SDA* and *TR-SDA* are more distinctive and less confused compared with *PCA* and *LPP* as shown in Fig. 6a and b. This indicates that semi-supervised methods such as *SDA* and *TR-SDA* can provide more discriminative information than unsupervised method such as *PCA* and *LPP*. Our proposed *TR-SDA* is superior to *SDA*, because the sub-manifolds of different classes are more separated and less overlapped, while the sub-manifold in each class are smoothly preserved. Fig. 7 further details the sub-manifold of digit 0 learned by *TR-SDA*. From Fig. 7, we can see that different hand-writing styles of digit 0 are smoothly varied along a sub-manifold line (red line in Fig. 7) indicating that *TR-SDA* can preserve the manifold structure.

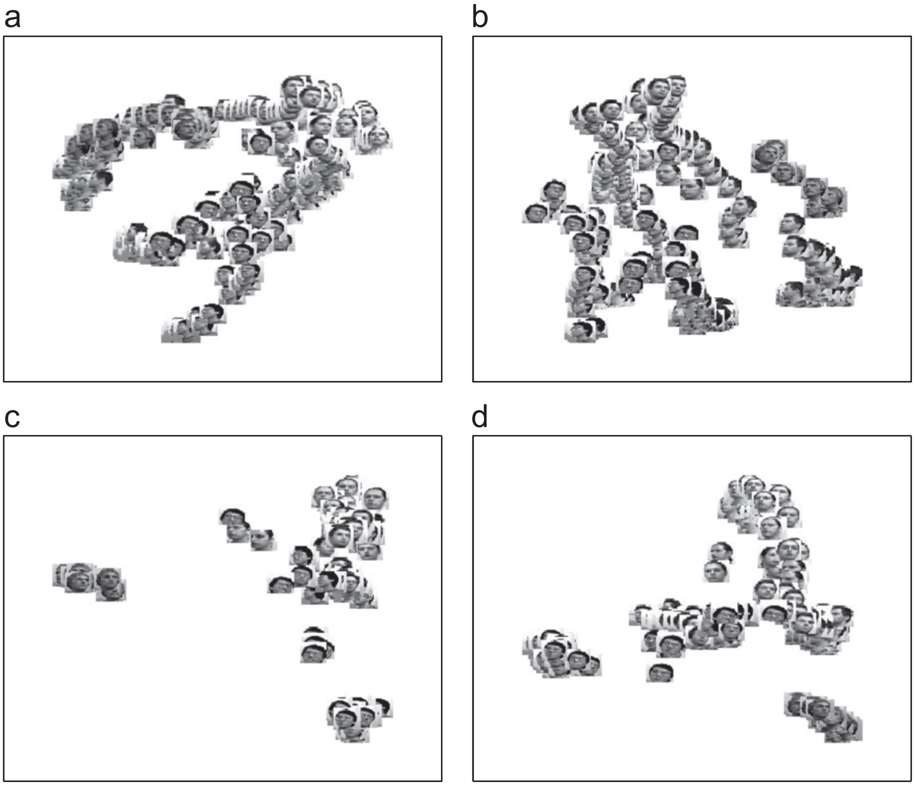

Fig. 4. 2*d* sub-manifolds learned by *PCA*, *LPP*, *SDA* and *TR-SDA*: four individuals of *UMIST* dataset (a) *PCA*, (b) *LPP*, (c) *SDA* and (d) *TR-SDA*.

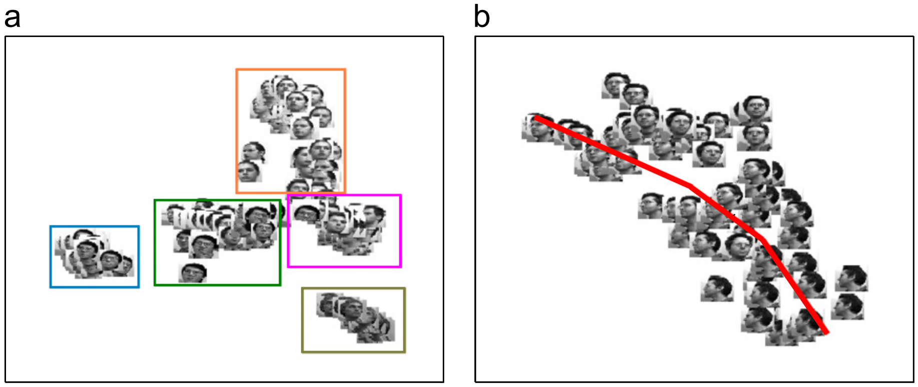

Fig. 5. 2*d* sub-manifold learned by *TR-SDA* and zoom in one individual: (a) *TR-SDA* of four individuals and (b) Zooming in one individual.

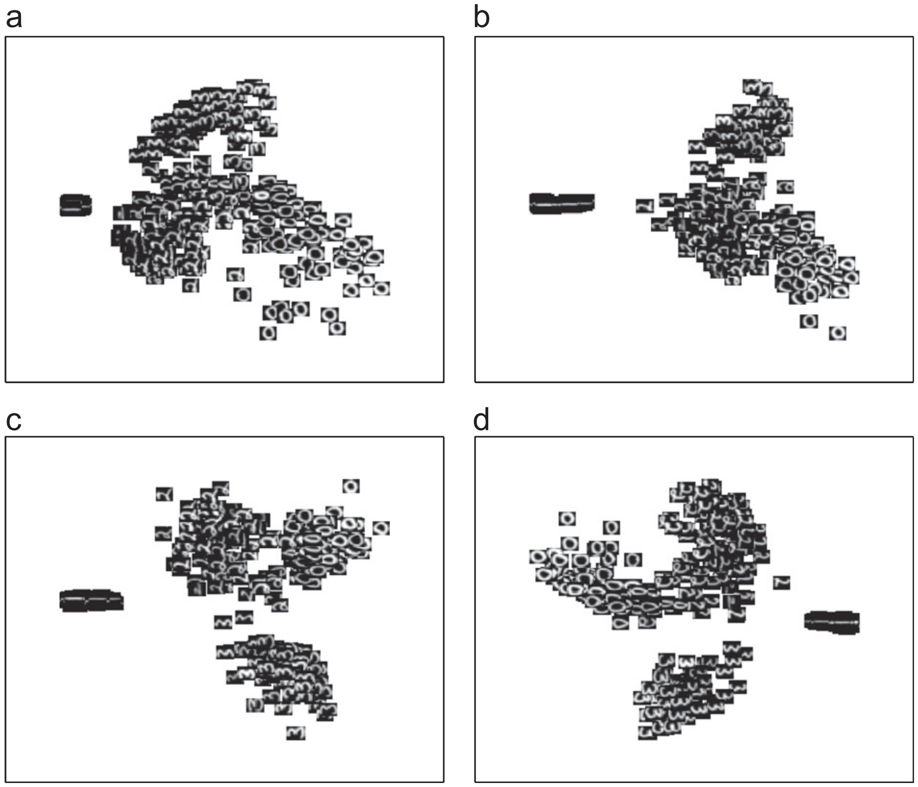

Fig. 6. 2*d* sub-manifold learned by *PCA*, *LPP*, *SDA* and *TR-SDA*: handwritten digits *0–4* of *USPS* dataset (a) *PCA*, (b) *LPP*, (c) *SDA* and (d) *TR-SDA*.

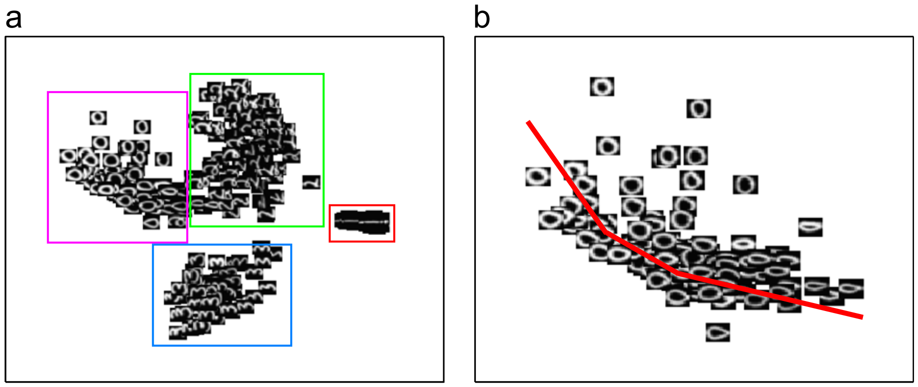

Fig. 7. 2*d* sub-manifold learned by *TR-SDA* and zoom in for digit *0*: (a) TR-SDA of four digits 0-4 and (b) Zooming in digit *0*. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

6.3. Classification

In this section, we used six datasets to compare the classification performance between our proposed *TR-SDA* algorithm and other algorithms such as *PCA*, *LPP*, *MMC*, *CCA*, *LDA* and *SDA*. The six datasets include the *UMIST* face dataset [37], *ORL* face dataset [38], *USPS* handwritten digit dataset [39], *MNIST* handwritten digit dataset [40], COIL100 dataset [41] and AR dataset [42]. The details of data information and simulation settings are list in Table 5.

It is noted that a variant version of *CCA* based on $c-1$ label coding [18] was used, because it can solve multi-class classification problem, while the original *CCA* [16] only deals with binary classification. In this comparative study, we randomly split each dataset into training set and test set. We also randomly selected data points as the training set to form labeled and unlabeled set. The training set in all datasets are preliminarily processed with *PCA* operator to eliminate the null space before performing dimensionality reduction. For unsupervised method such as *PCA* and *LPP*, we used the training set to train the learner. For supervised method such as *MMC*, *CCA*, *LDA* and *TR-LDA*, we used only labeled set to train the learner. For semi-supervised method such as *SDA* and *TR-SDA*, we used all the training set with both labeled and unlabeled set to train the learner. All algorithms used labeled set in the output reduced space to train a nearest neighborhood classifier for evaluating the accuracies of test set.

Table 5
Data information and simulation settings.

Dataset	*# Classes*	*# Images*	*# Dim*	*# Training*	*# Test*
UMIST	20	564	32 × 32	15 × 20	*remains*
ORL	40	400	32 × 32	8 × 40	2 × 40
USPS	10	9298	16 × 16	100 × 10	100 × 10
MNIST	10	60,000	28 × 28	100 × 10	100 × 10
COIL100	100	7200	32 × 32	20 × 100	20 × 100
AR	100	2600	32 × 32	20 × 100	*remains*

6.3.1. Face recognition

For face recognition, we use the *UMIST* and *ORL* face dataset to evaluate the performance of algorithms. The simulation settings are as follows: We randomly selected 15 data points per class to form training set for *UMIST* dataset, and 8 data points per class for *ORL* dataset. The remaining dataset is as test dataset. In the training set, we randomly selected 4, 7, 10 and 2, 5, 8 data points per class as labeled set and the remains as unlabeled set in the *UMIST* and *ORL* dataset, respectively. For manifold regularized term in *SDA* and *TR-SDA*, the regularized parameter λ_m is set as $0.1\mu_0$ both in *UMIST* and *ORL* dataset, where $\mu_0 = (Tr(XL_mX^T))/\ (Tr(XL_wX^T))$. For *LPP*, *SDA* and *TR-SDA*, we set the neighborhood number as 8. We then employ Gaussian function to construct the weight matrix in *LPP*, *SDA* and *TR-SDA*. The parameter σ in Gaussian function is determined as follows: We first calculated all the pairwise distances among data points of the whole training set. We then set σ equivalent to half the median of those distances. This can provide a reasonable estimation for the value σ [30]. The above two parameters are set the same for the *UMIST* and *ORL* dataset.

For simulation, we first fixed the labeled number in the training set as 4, 7, 10 for the *UMIST* dataset and 2, 5, 8 for the *ORL* dataset. The average accuracies over 20 randomly split with the above parameters under different dimensionality are shown in Figs. 8 and 9 for *UMIST* and *ORL* dataset, respectively. Tables 6 and 7

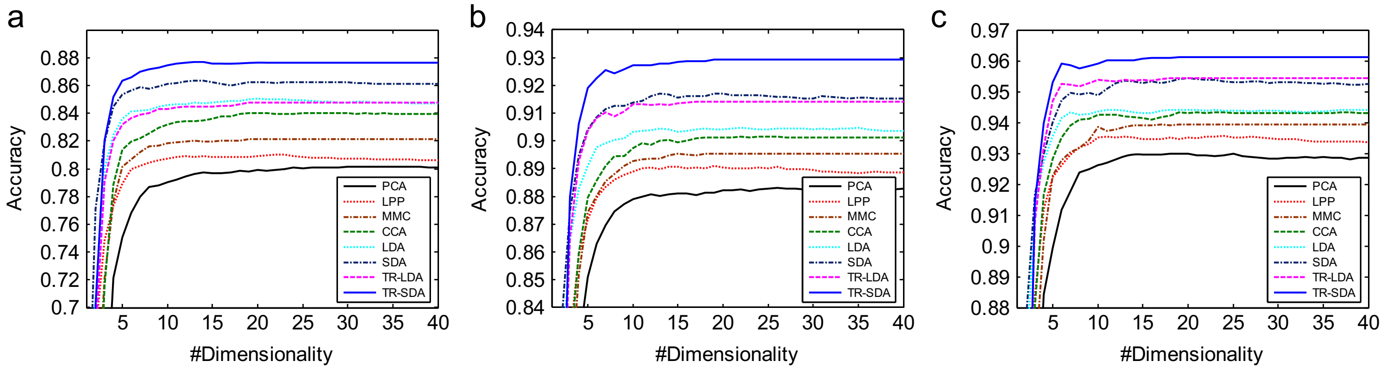

Fig. 8. Average accuracy under different dimensionality: *UMIST* dataset (a) 4 labels, (b) 7 labels and (c) 10 labels.

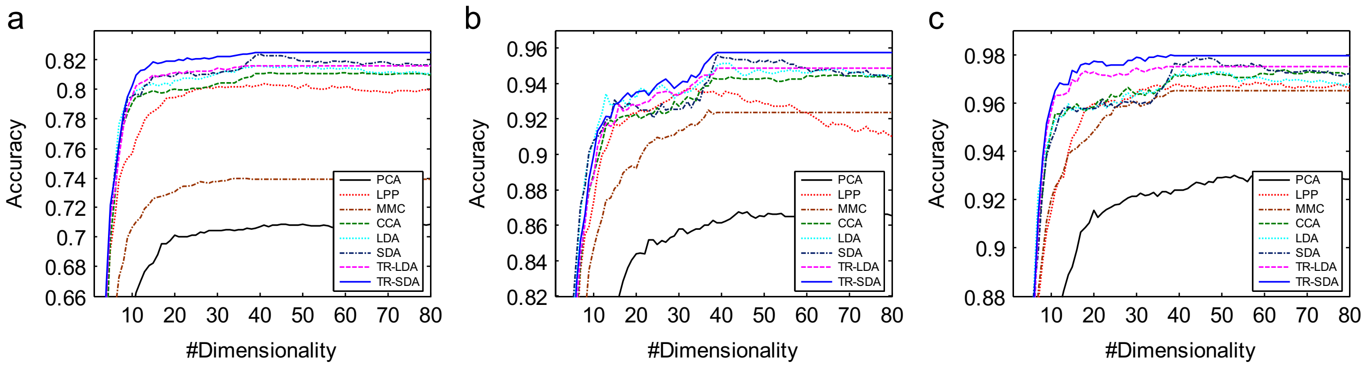

Fig. 9. Average accuracy under different dimensionality: *ORL* dataset (a) 2 labels, (b) 5 labels and (c) 8 labels.

Table 6
Average accuracy on the test set: *UMIST* dataset.

Dataset	*Method*	4 labeled			7 labeled			10 labeled		
		Mean (%)	Var (%)	Dim	Mean (%)	Var (%)	Dim	Mean (%)	Var (%)	Dim
UMIST	*PCA*	80.16	0.97	32	88.34	0.69	35	93.00	0.65	18
	LPP	81.00	0.84	23	89.09	0.58	19	93.56	0.49	24
	MMC	82.13	0.83	19	89.54	0.68	19	93.93	0.53	19
	CCA	84.02	0.68	28	90.15	0.53	24	94.32	0.51	19
	LDA	85.03	0.71	19	90.47	0.49	19	94.40	0.53	19
	SDA	86.34	0.74	13	91.71	0.40	13	95.43	0.50	19
	TR-LDA	84.75	0.67	19	91.42	0.44	18	94.43	0.48	19
	TR-SDA	87.69	0.53	13	92.92	0.42	19	96.13	0.51	19

Table 7
Average accuracy on the test set: *ORL* dataset.

Dataset	*Method*	2 labeled			5 labeled			8 labeled		
		Mean (%)	Var (%)	Dim	Mean (%)	Var (%)	Dim	Mean (%)	Var (%)	Dim
ORL	*PCA*	71.00	1.95	69	86.75	3.02	46	93.12	2.73	61
	LPP	80.37	1.47	41	93.75	2.69	33	96.87	1.31	58
	MMC	74.00	2.02	34	92.50	3.52	37	96.50	1.90	39
	CCA	81.12	1.54	60	94.50	2.79	64	97.37	0.97	65
	LDA	81.62	1.42	39	95.12	2.52	39	97.37	1.16	39
	SDA	82.37	1.66	39	95.62	3.49	39	97.87	1.97	39
	TR-LDA	81.62	1.43	39	94.87	2.75	39	97.50	1.40	39
	TR-SDA	82.50	1.95	39	95.75	2.99	39	98.00	1.39	38

show the average accuracy with the best dimensionality for the two datasets. From the results shown in Fig. 8 and Table 6, we can observe that for the *UMIST* dataset the semi-supervised methods outperform the corresponding supervised methods by 2–3% improvements, i.e. *SDA* and *TR-SDA* are superior to *LDA* and *TR-LDA*, respectively. This indicates that by incorporating the unlabeled set into the training procedure, the classification performance can be markedly improved, because manifold structure embedded in the dataset is preserved. In addition, *TR-SDA* achieves better results than *SDA*. This is mainly due to the orthogonal property of the projection matrix. We further compare two supervised algorithms namely *MMC* and multi-class *CCA*. The results show that our proposed method outperformed *MMC* and *CCA* by about 5% and 3%, respectively. These improvements are believed to be due to the fact that our *TR-SDA* is a kind of improved *MMC*. Given a certain class indicator matrix [18], the multi-class *CCA* can be equivalent to *LDA*, thus *TR-SDA* can certainly outperform *CCA*. All supervised and semi-supervised methods are better than unsupervised methods such as *PCA* and *LPP*, which means the labeled information is of great importance for discriminative learning, for instance the *TR-SDA* outperformed *PCA* and *LPP* by approximately 7% and 6%, respectively. It is noticed that the classification accuracy of all algorithms change when the number of labeled set increases, for instance the accuracy of *TR-SDA* increased from about 87% to 96% when the number of labeled data increased from 4 to 10. We can also observe from Fig. 8 that the accuracy of all algorithms varies when the number of reduced dimensionality increased. For *LDA* and *SDA*, their accuracy remained unchanged beyond the bound of $c-1$ dimensionality. For other methods such as *PCA*, *LPP*, *MMC*, *CCA* and *TR-SDA*, their accuracies increased only until a certain dimensionality. Another observation from Fig. 8 is that our proposed method *TR-SDA* converges more efficient than other methods, i.e.*TR-SDA* can reach the highest accuracy using fewest number of dimensionality. This shows a great superiority of our proposed algorithm over other methods.

For the *ORL* dataset, the following observations from Fig. 9 and Table 7 can be obtained. (1) *SDA* and *TR-SDA* are superior to its corresponding supervised methods of *LDA* and *TR-LDA*, for instance *SDA* and *TR-SDA* outperformed *LDA* and *TR-LDA* by about 1–2%.(2) Our proposed method *TR-SDA* is better than *SDA* due to the orthogonal property. (3) Our proposed method *TR-SDA* can deliver about 7% and 2% improvements compared with the two supervised methods of *MMC* and *CCA*. (4) All supervised and semi-supervised algorithms are better than unsupervised algorithms such as *PCA* and *LPP*, e.g. *TR-SDA* can achieve 8% and 2% improvements compared to *PCA* and *LPP*, respectively. (5) The accuracies of all algorithms will change significantly when the labeled number increased, i.e. the accuracy of *TR-SDA* increased from 82% to 98% when the number of labeled data increased from 2 to 8. (6) Our proposed method *TR-SDA* can reach the highest accuracy using the fewest number of dimensionality.

6.3.2. Handwritten digit recognition

For handwritten digit recognition, we used the *USPS* and *MNIST* dataset to evaluate the performance. Since the *MNIST* dataset has a training set of 60,000 data points and a testing set of 10,000 data points, we only selected the first 2000 data points from the original training set and testing set. The simulation settings are as follows: We randomly selected 100 data points per class as training set and 100 data points as test dataset from the *USPS* and *MNIST* dataset. In the training set, we randomly selected 20, 50, 80 data points per class as labeled set and the remains as unlabeled set. For the manifold regularized term in *SDA* and *TR-SDA*, the regularized parameter λ_m was set as $0.1\mu_0$ for the *USPS* and *MNIST* dataset. For *LPP*, *SDA* and *TR-SDA*, we set the neighbor number as 8. The Gaussian function is used to construct the weight matrix in *LPP*, *SDA* and *TR-SDA*, for which the parameter σ in Gaussian function is determined using the same strategy in face recognition.

We first fixed the number of labeled data in the training set as 20, 50 and 100 to train the learners. The average accuracy over 20 randomly splits with the above parameters under different dimensionality are shown in Figs. 10 and 11 for the *USPS* and *MNIST* dataset, respectively. Tables 8 and 9 show the average accuracy with the best dimensionality for the two datasets. From the results in Fig. 10 and Table 8, we can observe that for the *USPS* dataset, the semi-supervised algorithms such as *SDA* and *TR-SDA* outperformed the corresponding supervised algorithms of *LDA* and *TR-LDA* by about 2%. Our proposed *TR-SDA* is slightly better than *SDA* due to the orthogonal property of the projection matrix of *TR-SDA*. In addition, compared with some other state-of-art supervised algorithms such as *MMC* and *CCA*, *TR-SDA* is also superior to the two algorithms. The improvements, compared with *MMC* and *CCA*, can reach to about 3% and 1%, respectively. All supervised and semi-supervised algorithms outperformed the unsupervised algorithms of *PCA* and *LPP*, e.g. *TR-SDA* can provide a 4% improvements compared to *PCA* and *LPP*. Another observation shown in Fig. 10 and Table 8 is that the accuracy of all algorithms change as the number of labeled data increased, e.g. the accuracies of *TR-SDA* can reach 89%, 93% and 94% when the number of labeled set are 20, 50 and 80, respectively. But it is noticed that accuracy settles at a certain level even when the number of labeled data continuously increases. Apparently, the labeled information is sufficient for discriminative learning when the labeled data reaches certain level and further increases of labeled data will not have noticeable effect. We can also observe from Fig. 10 that the accuracy of all algorithms varies as the number of reduced dimensionality increases. It is found that the accuracy of *LDA* and *SDA* maintain unchanged to a bound of $c-1$, when the ranks of the projection matrix are at most $c-1$. For other algorithms such as *PCA*, *LPP*, *MMC*, *CCA*, *TR-LDA* and *TR-SDA*, their accuracies continuously increase until a certain dimensionality is reached. But we must

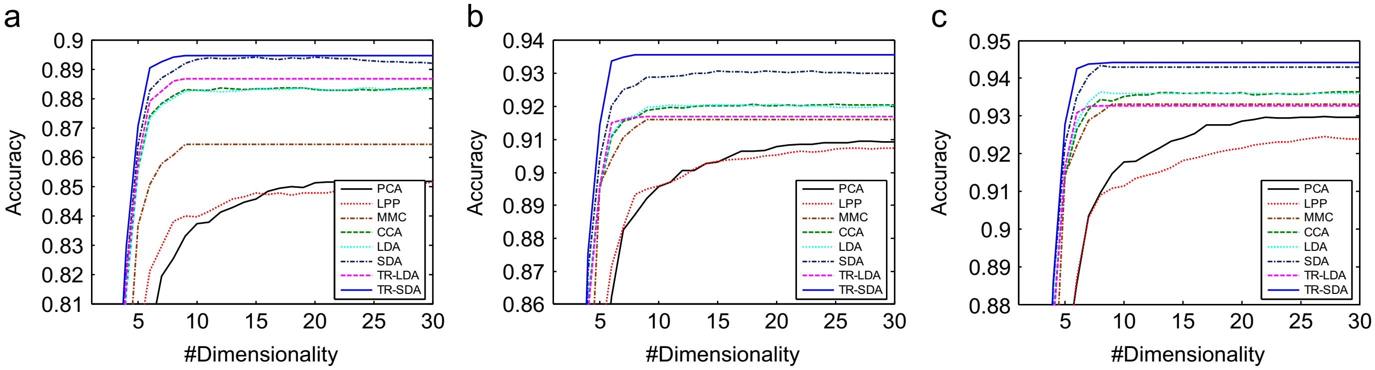

Fig. 10. Average accuracy under different dimensionality: *USPS* dataset (a) 20 labels, (b) 50 labels and (c) 80 labels.

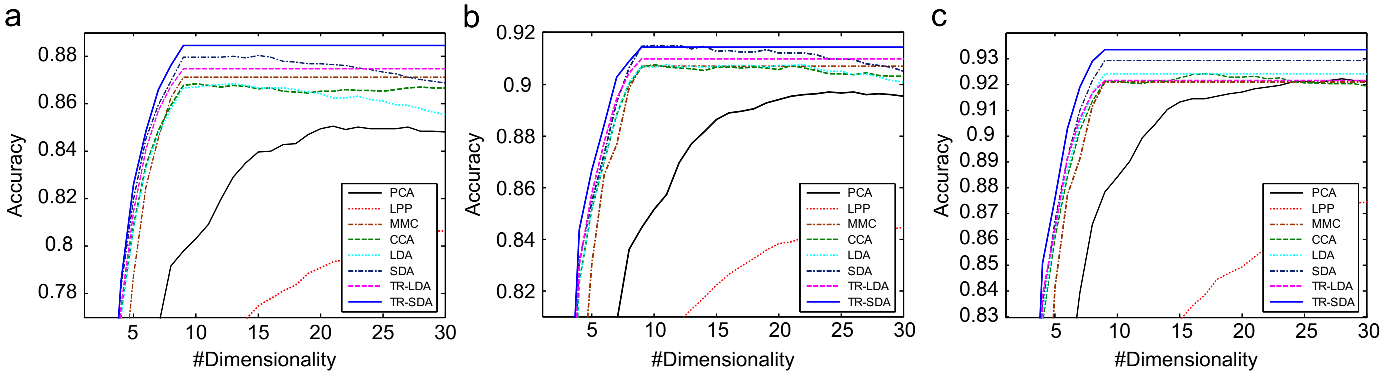

Fig. 11. Average accuracy under different dimensionality: *MNIST* dataset (a) 20 labels, (b) 50 labels and (c) 80 labels.

Table 8
Average accuracy on the test set: *USPS* dataset.

Dataset	*Method*	20 labeled			50 labeled			80 labeled		
		Mean (%)	Var (%)	Dim	Mean (%)	Var (%)	Dim	Mean (%)	Var (%)	Dim
USPS	*PCA*	85.23	0.27	27	90.94	0.66	26	92.98	0.29	27
	LPP	85.17	0.29	29	90.74	0.59	30	92.45	0.46	27
	MMC	86.47	0.46	9	91.60	0.70	9	93.32	0.26	9
	CCA	88.39	0.37	18	92.07	0.65	29	93.64	0.32	29
	LDA	88.38	0.38	9	92.07	0.68	9	93.64	0.34	8
	SDA	89.43	0.40	9	93.08	0.93	9	94.33	0.35	8
	TR-LDA	88.70	0.43	9	91.70	0.52	9	93.28	0.40	9
	TR-SDA	89.49	0.61	9	93.57	0.67	9	94.41	0.38	9

Table 9
Average accuracy on the test set: *MNIST* dataset.

Dataset	*Method*	20 labeled			50 labeled			80 labeled		
		Mean (%)	Var (%)	Dim	Mean (%)	Var (%)	Dim	Mean (%)	Var (%)	Dim
MNIST	*PCA*	85.04	0.83	24	89.70	1.36	21	91.21	0.60	28
	LPP	80.64	0.86	27	84.50	1.30	29	87.46	1.11	29
	MMC	87.12	0.97	9	90.71	1.44	9	92.10	0.60	9
	CCA	86.84	1.07	10	90.76	1.42	10	92.42	0.61	17
	LDA	86.84	0.85	9	90.76	1.30	9	92.42	0.60	9
	SDA	88.04	1.33	9	91.51	1.36	9	92.92	0.58	9
	TR-LDA	87.47	0.86	9	90.49	1.66	9	92.14	0.90	9
	TR-SDA	88.46	1.09	9	91.44	1.56	9	93.33	0.88	9

say that the determination of the best reduced dimensionality still remains an open issue. In Fig. 7 it shows that our proposed *TR-SDA* can converge more efficiently than other algorithms, i.e. *TR-SDA* requires fewer dimensionalities to reach the same level of accuracy compared with other algorithms.

For the *MNIST* dataset, the following observations from Fig. 11 and Table 9 are found: (1) The semi-supervised algorithms of *SDA* and *TR-SDA* perform better than the supervised algorithms of *LDA* and *TR-LDA*, i.e. *SDA* and *TR-SDA* can achieve 2% and 1% improvements over *LDA* and *TR-LDA*, respectively. (2) Our proposed *TR-SDA* is better than *SDA* due to orthogonal property of the projection matrix learned by *TR-SDA*. (3) Our proposed *TR-SDA* is superior to other state-of-art supervised algorithms such as *MMC* and *CCA* with about 2% and 3% improvements, respectively. (4) All supervised and semi-supervised algorithms are better than unsupervised algorithms, e.g. *TR-SDA* can reach approximately 3% and 8% improvements over *PCA* and *LPP*, respectively. (5) The accuracies of all algorithms change with the number of labeled data. But the accuracy will not change dramatically when the number of labeled data reached certain level. (6) The performances of all algorithms vary as the reduced dimensionality increase. (7) Our proposed method *TR-SDA* converges more efficiently, i.e. *TR-SDA* can reach higher level of accuracy using the same dimensionality.

6.3.3. *Large scale dataset*

For large scale dataset, we used the *COIL100* and *AR* dataset to evaluate the performance. The simulation settings are as follows: We randomly selected 20 data points per class as training set both for the *COIL100* and *AR* dataset. The remaining dataset is as test dataset. In the training set, we randomly selected 4, 7, 10 data points per class as labeled set and the remains as unlabeled set both for *COIL100* and *AR* dataset. For the manifold regularized term in *SDA* and *TR-SDA*, the regularized parameter λ_m was set as $0.01\mu_0$. For *LPP*, *SDA* and *TR-SDA*, we set the neighbor number as 8. The Gaussian function is used to construct the weight matrix in

LPP, *SDA* and *TR-SDA*, for which the parameter σ in Gaussian function is determined using the same strategy in face recognition.

We first fixed the number of labeled data in the training set as 4, 7 and 10 to train the learners. The average accuracy over 20 randomly splits with the above parameters under different dimensionality are shown in Figs. 12 and 13 for the *COIL100* and *AR* dataset, respectively. Tables 10 and 11 show the average accuracy with the best dimensionality for the two datasets. From the results in Fig. 12 and Table 10, we can observe that for the *COIL100* dataset, the semi-supervised algorithms such as *SDA* and *TR-SDA* outperformed the corresponding supervised algorithms of *LDA* and *TR-LDA* by about 2%. In addition, we observe a consistent superiority in the performance of the orthogonal algorithms, i.e. *PCA*, *MMC*, *TR-LDA* and *TR-SDA* outperform the non-orthogonal algorithms such as *LPP*, *LDA* and *SDA*, e.g. *TR-LDA* and *TR-SDA* can provide 8–18% improvements compared to *LDA* and *SDA*. Another observation shown in Fig. 12 and Table 10 is that the accuracy of all algorithms change as the number of labeled data increased, e.g. the accuracies of *TR-SDA* can reach 77%, 86% and 90% when the number of labeled set are 4, 7 and 10, respectively. We can also observe from Fig. 10 that the accuracy of all algorithms varies as the number of reduced dimensionality increases. It is found that the accuracies of all datasets firstly increase to a certain dimensionality and then start to decrease. In Fig. 10 it shows that our proposed *TR-SDA* can converge more efficiently than other algorithms, i.e. *TR-SDA* requires fewest dimensionalities to reach the same level of accuracy compared with other algorithms.

For the *AR* dataset, from Fig. 13 and Table 11, we can observe similar results as *COIL100* dataset. Other observations include: (1) *SDA* and *TR-SDA* are slightly superior to its corresponding supervised methods of *LDA* and *TR-LDA*, this is mainly because the *AR*

Table 10
Average accuracy on the test set: *COIL100* dataset.

Dataset	*Method*	4 labeled			7 labeled			10 labeled		
		Mean (%)	Var (%)	Dim	Mean (%)	Var (%)	Dim	Mean (%)	Var (%)	Dim
COIL100	*PCA*	71.51	1.29	120	80.27	0.96	95	84.74	1.28	85
	LPP	69.18	0.90	30	77.28	0.77	25	82.54	1.18	35
	MMC	72.38	1.18	30	80.96	0.80	30	85.87	0.76	35
	CCA	59.21	1.38	30	75.89	1.26	25	83.17	1.29	30
	LDA	59.21	1.38	30	75.89	1.26	25	83.17	1.29	30
	SDA	60.95	1.29	35	76.20	0.77	25	83.27	0.79	35
	TR-LDA	76.37	1.17	20	85.20	0.77	30	89.87	0.70	30
	TR-SDA	77.85	1.17	20	86.30	0.77	30	90.98	0.70	30

Table 11
Average accuracy on the test set: *AR* dataset.

Dataset	*Method*	4 labeled			7 labeled			10 labeled		
		Mean (%)	Var (%)	Dim	Mean (%)	Var (%)	Dim	Mean (%)	Var (%)	Dim
AR	*PCA*	48.06	1.71	180	57.93	1.73	180	64.42	2.27	180
	LPP	47.38	2.03	190	56.97	1.36	185	64.16	2.47	190
	MMC	64.66	3.29	135	80.89	2.28	120	87.35	1.67	130
	CCA	81.32	1.96	65	92.16	1.11	65	94.30	0.80	70
	LDA	81.32	1.98	65	92.16	1.11	65	94.30	0.80	70
	SDA	82.94	1.90	70	92.50	1.75	60	94.35	0.81	60
	TR-LDA	83.69	1.77	75	94.54	1.11	85	96.25	1.35	85
	TR-SDA	85.92	1.97	80	94.87	1.05	90	96.87	1.12	75

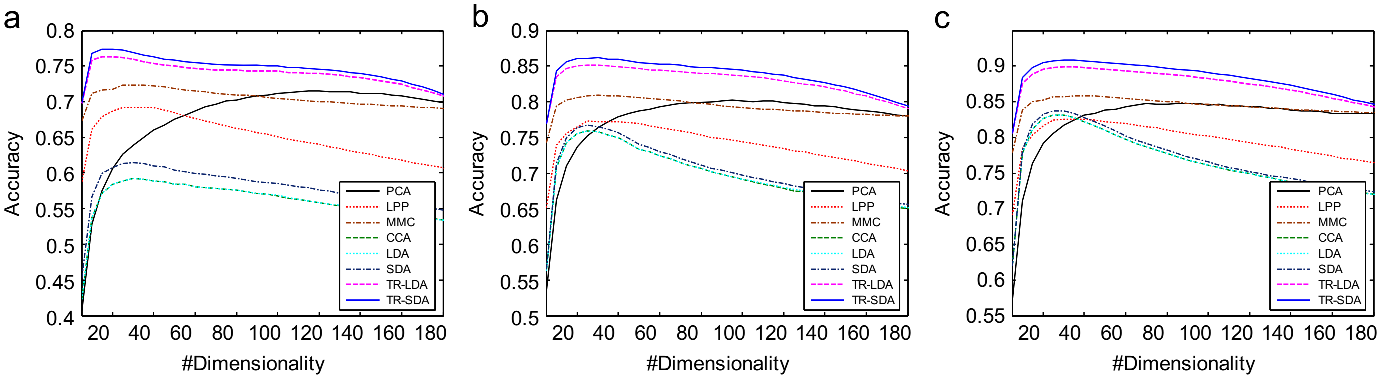

Fig. 12. Average accuracy under different dimensionality: *COIL100* dataset (a) 4 labels, (b) 7 labels and (c) 10 labels.

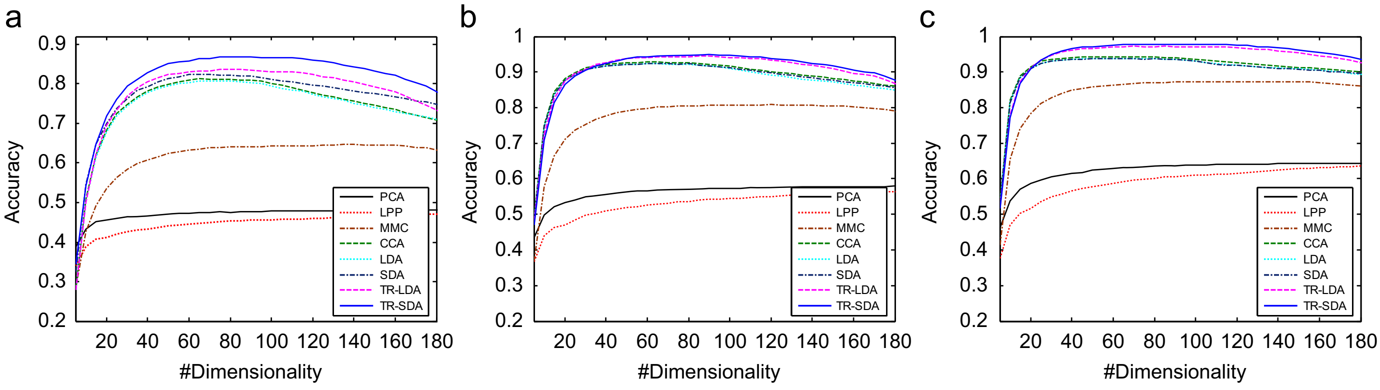

Fig. 13. Average accuracy under different dimensionality: *AR* dataset (a) 4 labels, (b) 7 labels and (c) 10 labels.

dataset does not have a clear low-dimensional manifold structure for the high-dimensional dataset, the manifold term does not play a key role to enhance the performance. (2) *PCA* and *LPP* have poor performances. (3) The accuracies of all algorithms vary as the number of reduced dimensionality increases. It is found that the accuracies of the supervised and semi-supervised algorithms firstly increase to a certain dimensionality and then start to decrease. For other algorithms such as *PCA* and *LPP*, their accuracies increased until to a certain dimensionality.

6.4. Convergent analysis

We compared the convergent speed between *ITR* and *ITR-Score* algorithms for solving *TR* problem of Eq. (4). In this study, six datasets were chosen for comparison including *UMIST*, *ORL*, *AR*, *USPS*, *MNIST* and *COIL100* dataset. The simulation settings is as follows: we randomly chose 7 data points as training set for *UMIST*, *ORL*, *AR*, *COIL100* datasets and 50 data points for *USPS* and *MNIST* datasets. The reduced dimensionality is set to 10 for *UMIST*, *ORL*, *USPS*, *MNIST* datasets and 30 for *AR* and *COIL100* datasets. In the iterative process, we chose the trace difference value $g(t)=g(\lambda_t)=\max_{W^TW=I}Tr[W^T(S_b-\lambda_t S_w)W]$ to evaluate the convergence. As described in Theorem 1, the global optimal λ^* results in $\max_{W^TW=I}Tr[W^T(S_b-\lambda^* S_w)W]=0$, hence this evaluation measures the convergent speed of the trace ratio value to the global optimum.

Fig. 14 shows the convergent results for different datasets. From the results we can observe that both *ITR* and *ITR-Score*

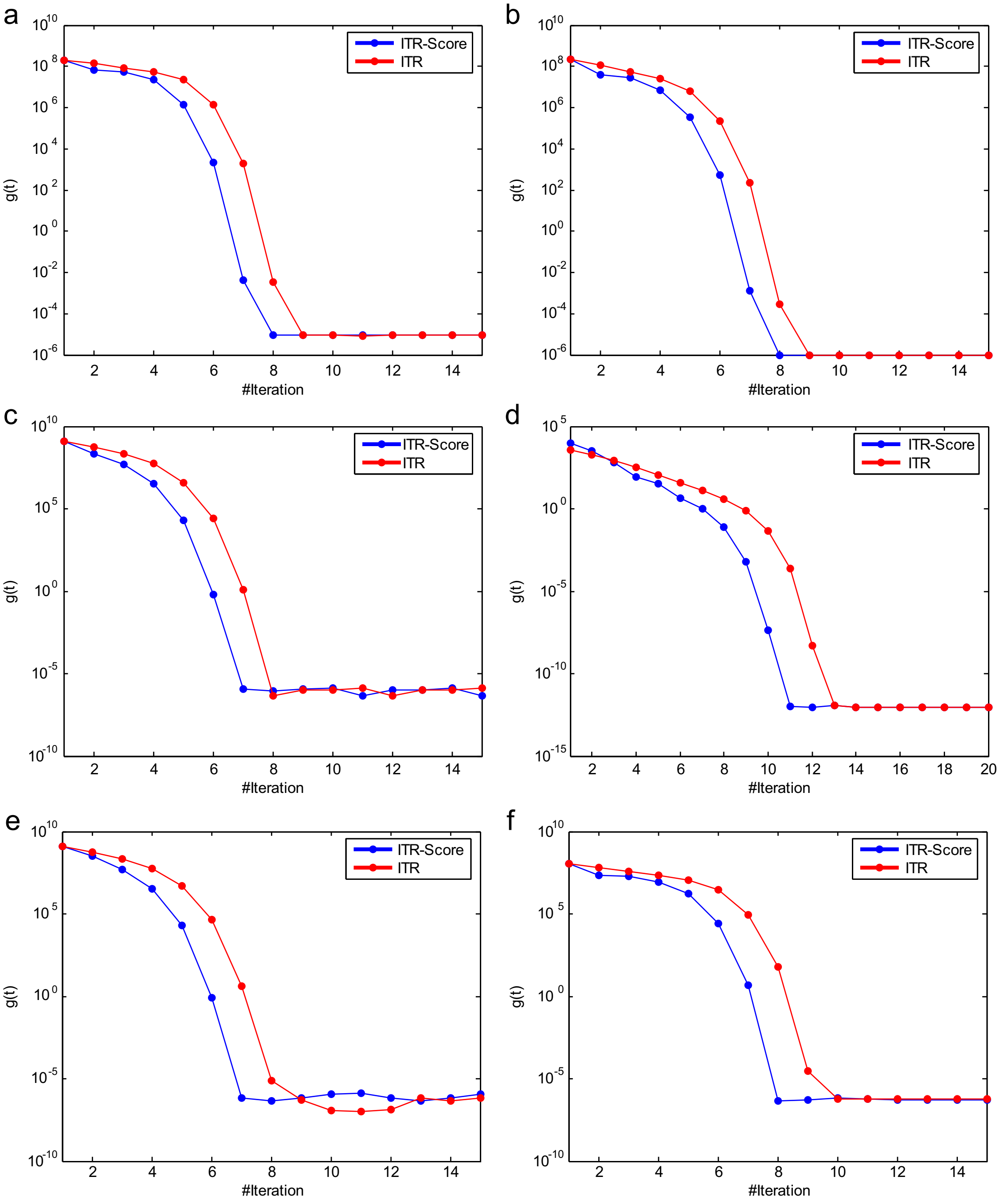

Fig. 14. Convergent analysis: comparative study between *ITR* and *ITR-Score* algorithms: (a) *UMIST* dataset, (b) *ORL* dataset, (c) *USPS* dataset, (d) *MNIST* dataset, (e) *COIL100* dataset and (f) *AR* dataset.

algorithms can converge to the optimal trace ratio value hence causing $g(\lambda^*)=0$. This can be true as it has been theoretically guaranteed in Section 2.2.3. In addition, it can be easily observed that our proposed *ITR-Score* algorithm is able to converge faster than *ITR* algorithm in all datasets. This improvement is believed to be due to the reason that for any initial $\lambda_t<\lambda^*$, the updated λ_{t+1} of the proposed *ITR-Score* algorithm is larger than that of *ITR* algorithm.

6.5. Kernel validation

In this section, we choose four *UCI* datasets to evaluate the kernel version of our algorithms and compare them with other algorithms. The datasets include *Iris*, *Wine*, *Balance* and *Synthetic Control Chart Time Series* (*SCCTS*). The details of data information are listed in Table 12.

For comparative study, we randomly chose 70% data points from each dataset as training set and the rest 30% as test set. We also randomly chose 30% data points from training set as labeled set and the rest 70% as unlabeled set. For unsupervised method such as *PCA* and *LPP*, we used the training set to train the learner. For supervised method such as *MMC*, *CCA*, *LDA* and *TR-LDA*, we used only labeled set to train the learner. For semi-supervised method such as *SDA* and *TR-SDA*, we used all the training set with both labeled and unlabeled set to train the learner. All algorithms used labeled set in the output reduced space to train a nearest neighborhood classifier for evaluating the accuracies of test set. In this study, we use radial basis function (*RBF*) as kernel function. The reduced dimensionality is set 3.

The average accuracies over 20 randomly split are shown in Fig. 15 for different *UCI* datasets. From the results in Fig. 15, we can observe that (1) *SDA* and *TR-SDA* are superior to its corresponding supervised methods of *LDA* and *TR-LDA*, for instance *SDA* and *TR-SDA* outperformed *LDA* and *TR-LDA* by about 1–2%. (2) Our proposed method *TR-SDA* is better than *SDA*, especially in the *Balance* datasets (by about 3%). (3) All supervised and semi-supervised algorithms are better than unsupervised algorithms, e.g. *TR-SDA* can achieve 3%, 20%, 22% and 6% improvements compared to *PCA* and *LPP* in *Iris*, *Wine*, *Balance* and *SCCTS* dataset, respectively.

Table 12
Data information of *UCI* dataset.

Dataset	# *Class*	# *Num.*	# *Dim.*
Iris	3	150	4
Wine	3	178	13
Balance	3	625	4
SCCTS	6	600	60

6.6. Image segmentation

We demonstrated the image segmentation of our proposed algorithms and compared them with other algorithms. In this study, we chose an image in the *COREL* dataset for segmentation [43] (see Fig. 16a). The image can be divided into five classes including hill, boat, sky, sea and beach and our goal is to segment all these classes. In the image, we describe each pixel as a 5-dimensional vector, i.e. $x_p=[r,g,b,x,y]^T$, where (r,g,b) are the *R*, *G*, *B* values of the pixel *p* and (x,y) are its spatial coordinates. We next chose the specified pixels in the color lines (see Fig. 16b, each color represents a class in the image) as labeled set and divide the remaining pixels into unlabeled and test set. Finally, all

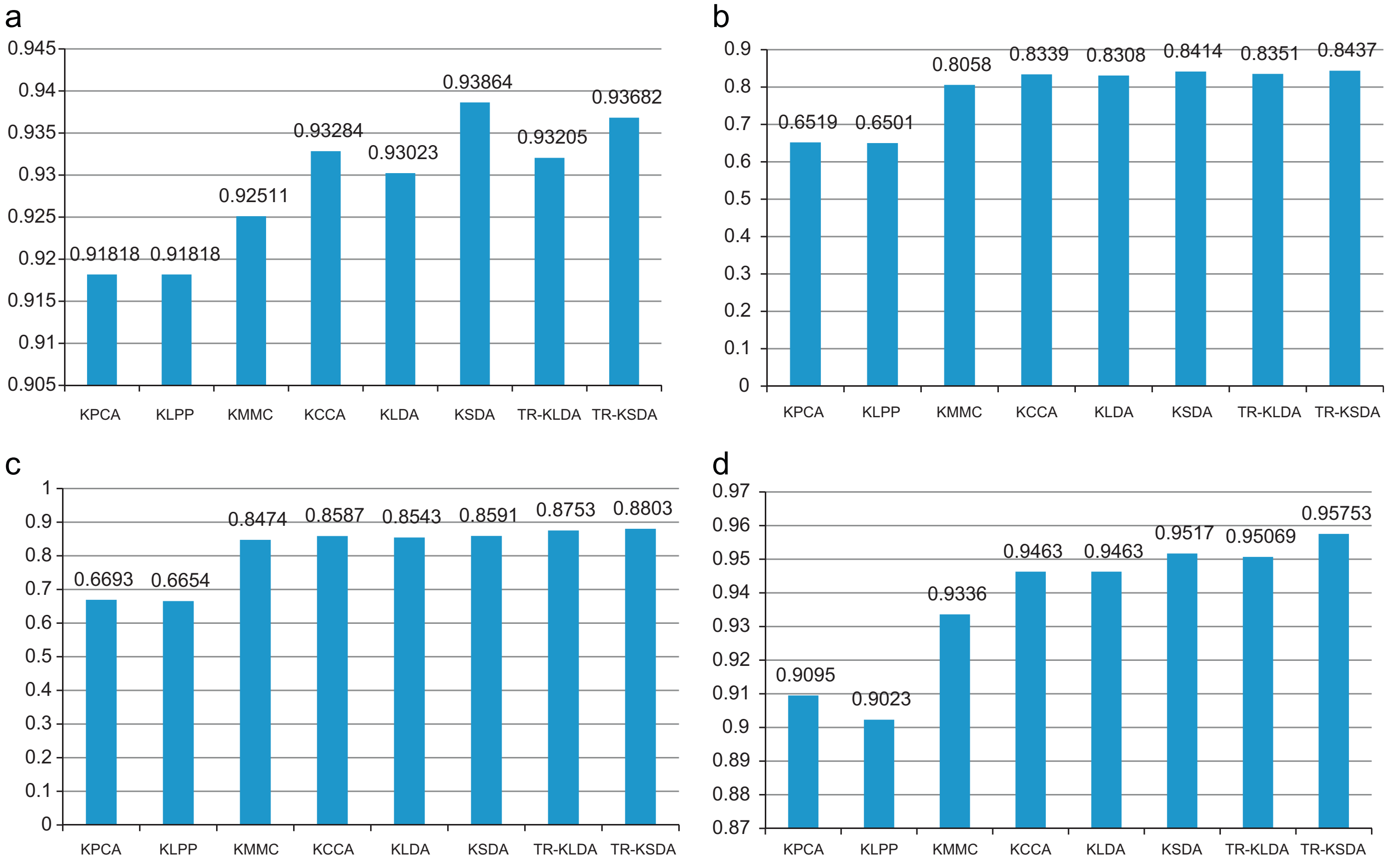

Fig. 15. Kernel validation: average accuracy on the test set (a) *Iris* dataset, (b) *Wine* dataset, (c) *Balance* dataset and (d) *Synthetic Control Chart Time Series* dataset (*SCCTS*).

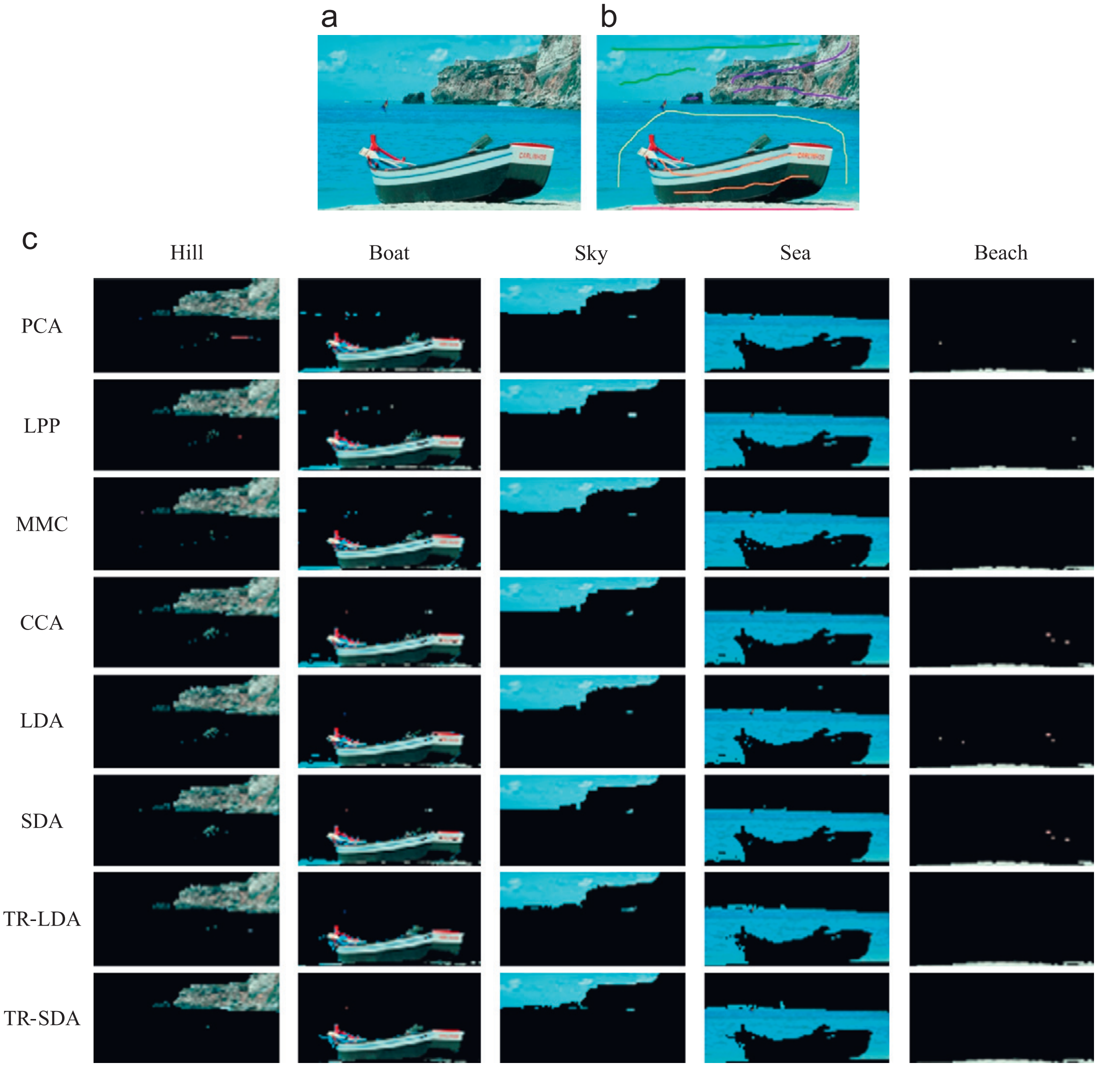

Fig. 16. Image segmentation: *COREL* dataset (a) Original image, (b) partially labeled image, the pixels in each color line represent an object and (c) the segmentation results of different algorithms.

algorithms used the label set in the output reduced space to train a nearest neighborhood classifier for evaluating the class label of unlabeled and test set. In this example, the reduced dimensionality is set 3.

Fig. 16c shows the image segmentation results of different algorithms. From the results we can see that the proposed *TR-LDA* and *TR-SDA* algorithms are better than other algorithms. Taking the boat as an example, it demonstrates that there are less miss-classified pixels in our algorithms than in other algorithms. This can be observed that in our algorithms the pixels belonging to the boat are precisely extracted. However in other algorithms, part of pixels belonging to the sea is miss-classified to the boat. The similar performance can also be observed in the segmentation results of other classes. This enhanced performance is mainly because that optimal projection obtained by our algorithms is orthogonal, which results in preserving the similarities (Euclidean distance) between pixels.

7. Conclusions

A new efficient algorithm for finding optimal solution of trace ratio problem is proposed. Based on this algorithm, we derive an orthogonal constrained semi-supervised learning framework. We show that the algorithm can be extended for solving corresponding semi-supervised problems. The essence of the proposed algorithm is that it is able to incorporate unlabeled set into a learning procedure for preserving the geometrical structure embedded in both labeled and unlabeled set. Also, the algorithm is able to preserve the discriminative structure embedded in labeled set so that the data points in different classes can be separated. It is important to note that under such a framework many existing semi-supervised dimensionality reduction methods such as *SDA*, *SSDR*, *SSMMC* can be improved by incorporating our proposed framework. The framework can also formulate a corresponding kernel version for handling nonlinear problems. Theoretical analysis presented in this paper indicates that there are certain relationships between linear and nonlinear algorithms. It is worth noting that *TR-LDA*, *TR-SDA* and their corresponding kernel version can be connected in a unified form. Finally, extensive simulations on synthetic dataset and real world dataset have been conducted. The results demonstrate that our proposed *TR-SDA* is effective and is able to deliver significantly improved performance compared with other state-of-art algorithms.

Appendix A

In order to prove Theorem 2, we first give two lemmas:

Lemma 1. *If* $\forall i$, $a_i \geq 0$, $b_i > 0$ *satisfying* $(a_1/b_1) \geq (a_2/b_2) \geq \cdots (a_k/b_k)$, *then* $(a_1/b_1) \geq ((a_1+a_2+\cdots a_k)/(b_1+b_2+\cdots b_k)) \geq (a_k/b_k)$ *.*

Proof. of Lemma 1:

Let $(a_1/b_1)=p$, $\forall i \neq 1$, $a_i \geq 0$, $b_i > 0$, we have $a_i < pb_i$. Hence,

$$\frac{a_1+a_2+\cdots a_k}{b_1+b_2+\cdots b_k} \leq \frac{p(b_1+b_2+\cdots b_k)}{b_1+b_2+\cdots b_k} \leq \frac{a_1}{b_1}.$$

Let $(a_k/b_k)=q$, $\forall i \neq k$, $a_i \geq 0$, $b_i > 0$, we have $a_i > qb_i$. Hence

$$\frac{a_1+a_2+\cdots a_k}{b_1+b_2+\cdots b_k} \geq \frac{q(b_1+b_2+\cdots b_k)}{b_1+b_2+\cdots b_k} \leq \frac{a_k}{b_k}.$$

Thus, we have

$$\frac{a_1}{b_1} \geq \frac{a_1+a_2+\cdots a_k}{b_1+b_2+\cdots b_k} \geq \frac{a_k}{b_k}.$$

Lemma 2. *If* $\forall\ i$, $a_i \geq 0$, $b_i > 0$ *satisfying*

$$\frac{a_1}{b_1} \geq \frac{a_2}{b_2} \geq \cdots \frac{a_{m_1}}{b_{m_1}} \geq \frac{a_{m_1+1}}{b_{m_1+1}} \geq \cdots \frac{a_{m_2}}{b_{m_2}}, \quad \textit{then}$$
$$\frac{a_1+a_2+\cdots a_{m_1}}{b_1+b_2+\cdots b_{m_1}} \geq \frac{a_1+a_2+\cdots a_{m_2}}{b_1+b_2+\cdots b_{m_2}}.$$

Proof. of Lemma 2:

According to Lemma 1, we have

$$\frac{a_1+a_2+\cdots a_{m_1}}{b_1+b_2+\cdots b_{m_1}} \geq \frac{a_{m_1}}{b_{m_1}} \geq \frac{a_{m_1+1}}{b_{m_1+1}} \geq \frac{a_{m_1+1}+a_{m_1+2}+\cdots a_{m_2}}{b_{m_1+1}+b_{m_1+2}+\cdots b_{m_2}},$$

hence we have

$$\frac{a_1+a_2+\cdots a_{m_1}}{b_1+b_2+\cdots b_{m_1}} \geq \frac{a_{m_1+1}+a_{m_1+2}+\cdots a_{m_2}}{b_{m_1+1}+b_{m_1+2}+\cdots b_{m_2}}.$$

According to Lemma 1 again, we have

$$\frac{a_1+a_2+\cdots a_{m_1}}{b_1+b_2+\cdots b_{m_1}} \geq \frac{a_1+a_2+\cdots a_{m_2}}{b_1+b_2+\cdots b_{m_2}}.$$

Proof. of Theorem 2:

Let $W_D=[w_1,w_2\ldots w_d,w_{d+1},\ldots,w_D] \in \mathbb{R}^{D\times D}$ be an orthogonal square matrix with column vectors satisfying $W_D^T W_D = W_D W_D^T = I$. If we assume

$$\frac{w_1^T X L_b X^T w_1}{w_1^T X L_w X^T w_1} \geq \frac{w_2^T X L_b X^T w_2}{w_2^T X L_w X^T w_2} \geq \cdots \geq \frac{w_d^T X L_b X^T w_d}{w_d^T X L_w X^T w_d}$$
$$\geq \frac{w_{d+1}^T X L_b X^T w_{d+1}}{w_{d+1}^T X L_w X^T w_{d+1}} \geq \cdots \geq \frac{w_D^T X L_b X^T w_D}{w_D^T X L_w X^T w_D},$$

then according to Lemma 2, we have

$$\frac{\sum_{i=1}^{d} w_i^T X L_b X^T w_i}{\sum_{i=1}^{d} w_i^T X L_w X^T w_i} \geq \frac{\sum_{i=1}^{D} w_i^T X L_b X^T w_i}{\sum_{i=1}^{D} w_i^T X L_w X^T w_i}.$$

Let $W_d=[x_1,x_2,\ldots,x_d]$, according to the trace property $Tr(AB)=Tr(BA)$, the above inequality can be rewritten as

$$\frac{Tr(W_d^T X L_b X^T W_d)}{Tr(W_d^T X L_w X^T W_d)} = \frac{\sum_{i=1}^{d} w_i^T X L_b X^T w_i}{\sum_{i=1}^{d} w_i^T X L_w X^T w_i} \geq \frac{\sum_{i=1}^{D} w_i^T X L_b X^T w_i}{\sum_{i=1}^{D} w_i^T X L_w X^T w_i}$$
$$= \frac{Tr(W_D^T X L_b X^T W_D)}{Tr(W_D^T X L_w X^T W_D)} \quad = \frac{Tr(X L_b X^T W_D W_D^T)}{Tr(X L_w X^T W_D W_D^T)}$$
$$= \frac{Tr(X L_b X^T)}{Tr(X L_w X^T)}$$

Based on the object function in Eq. (5), we have

$$\lambda^* = \max_{W^T W = I} \frac{Tr(W^T X L_b X^T W)}{Tr(W^T X L_w X^T W)} \geq \frac{Tr(W_d^T X L_b X^T W_d)}{Tr(W_d^T X L_w X^T W_d)}$$
$$\geq \frac{Tr(W_D^T X L_b X^T W_D)}{Tr(W_D^T X L_w X^T W_D)} = \frac{Tr(X L_b X^T)}{Tr(X L_w X^T)}.$$

Thus the lower bound of λ^* is $(Tr(X^T L_b X))/(Tr(X^T L_w X))$.

References

[1] J.B. Tenenbaum, V. de Silva, J.C. Langford, A global geometric framework for nonlinear dimensionality reduction, Science 290 (5500) (2000) 2319–2323.

[2] S. Roweis, L. Saul, Nonlinear dimensionality reduction by locally linear embedding, Science 290 (5500) (2000) 2323–2326.

[3] Y. Guo, S. Li, J. Yang, T. Shu, L. Wu, A generalized Foley–Sammon transform based on generalized fisher discriminant criterion and its application to face recognition, Pattern Recognition Letter 24 (1–3) (2003) 147–158.

[4] H. Wang, S. Yan, d. Xu, X. Tang, T. Huang, Trace ratio vs. ratio trace for dimensionality reduction, in: Proceedings of CVPR, 2007.

[5] Y. Jia, F. Nie, C. Zhang, Trace ratio problem revisited, IEEE Transactions on Neural Network 20 (4) (2009) 729–735.

[6] S. Xiang, F. Nie, C. Zhang, Learning a Mahalanobis distance metric for data clustering and classification, Pattern Recognition 41 (12) (2008) 3600–3612.

[7] S. Yan, X. Tang, Trace quotient problem revisited, in: Proceedings of ECCV, 2006, pp. 232–244.

[8] C. Shen, H. Li, M.J. Brooks, Supervised dimensionality reduction via sequential semi-definite programming, Pattern Recognition 41 (12) (2008) 3644–3652.

[10] J. Ye, Least square linear discriminant analysis, in: Proceedings of ICML, 2007.

[11] R.A. Fisher, The use of multiple measurements in taxonomic problems, Annals of Eugenics 7 (1936) 179–188.

[12] P.N. Belhumeur., J.P. Hespanha, D.J. Kriegman, Eigenfaces vs. fisherfaces: recognition using class specific linear projection, IEEE Transactions on Pattern analysis and Machine Intelligence 19 (7) (1997) 711–720.

[13] X. He, S. Yan, Y. Hu, P. Niyogi, H. Zhang, Face recognition using Laplacianfaces, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (3) (2005) 328–340.

[14] M. Belkin, P. Niyoqi, Laplacian eigenmaps for dimensionality reduction and data representation, Neural Computation 15 (2003) 1373–1396.

[15] H. Li, T. Jiang, K. Zhang, Efficient and robust feature extraction by maximum margin criterion, IEEE Transactions on Neural Networks 17 (1) (2006) 157–165.

[16] H. Hotelling, Relations between two sets of variates, Biometrika 28 (3–4) (1936) 321–377.

[17] D.R. Hardoon, S. Szedmak, J.S. Taylor, Canonical correlation analysis: An overview with application to learning method, Neural Computation 16 (12) (2004) 2639–2664.

[18] T. Sun, S. Chen, Class label versus sample label-based *CCA*, Applied Mathematics and Computation 185 (1) (2007) 272–283.

[19] D. Cai, X. He, J. Han, Semi-supervised discriminant analysis, in: Proceedings of ICCV, 2007.

[20] D. Zhang, Z. Zhou, S. Chen, Semi-supervised dimensionality reduction, in: Proceedings of SDM, 2007.

[21] J. Chen, J. Ye, Q. Li, Integrating global and local structures: a least squares framework for dimensionality reduction, in: Proceedings of CVPR, 2007.

[22] Y. Song, F. Nie, C. Zhang, S. Xiang, A unified framework for semi-supervised dimensionality reduction, Pattern Recognition 41 (9) (2008) 2789–2799.

[23] M. B. Blaschko, C. H. Lampert, A. Gretton, Semi-supervised Laplacian regularization of kernel canonical correlation analysis, in: Proceedings of the ECML PKDD, 2008.

[24] F. Nie, S. Xiang, Y. Jia, C. Zhang, Semi-supervised orthogonal discriminant analysis via label propagation, Pattern Recognition 42 (11) (2009) 2515–2627.

[25] 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) (2000) 1713–1726.

[26] Z. Jin, J.Y. Yang, Z.S. Hu, Z. Lou, Face recognition based on uncorrelated discriminant transformation, Pattern Recognition 34 (7) (2001) 1405–1416.

[27] P. Howland, J. Wang, H. Park, Solving the small sample size problem in face recognition using generalized discriminant analysis, Pattern Recognition 39 (2) (2006) 277–287.

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

[29] H. Hu, Orthogonal neighborhood preserving discriminant analysis for face recognition, Pattern Recognition 41 (6) (2008) 2045–2054.

[30] E. Kokiopoulou, Y. Saad, Orthogonal neighborhood preserving projections: a projection-based dimensionality reduction technique, IEEE Trans. on Pattern Analysis and Machine Intelligence 29 (12) (2007) 2143–2156.

[31] D. Cai, X. He, J. Han, H. Zhang, Orthogonal Laplacian faces for face recognition, IEEE Transactions on Image Processing 15 (11) (2006) 3608–3614.

[32] J. Yang, A.F. Frangi, D. Zhang, J.Y. Yang, J. Zhong, KPCA plus LDA: a complete kernel fisher discriminant Framework for feature extraction and recognition,

IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (2) (2005) 230–244.

[33] K.R. Muller, S. Mika, G. Ratsch, K. Tsuda, B. Scholkopf, An introduction to Kernel-based learning algorithms, IEEE Transactions on Neural Networks 12 (2) (2001) 181–201.

[34] K. Fukuaga, Introduction to Statistical Pattern classification, Academic Press, USA, 1990.

[35] R.A. Horn, C.R. Johnson, Matrix analysis, Cambridge University Press, Cambridge, 1990.

[36] F.R.K. Chung, Spectral Graph Theory, American Mathematical Society, 1997.

[37] D.B. Graham, N.M. Allinson, Characterizing virtual eigensignatures for general purpose face recognition in face recognition: from theory to application, NATO ASI Series F, computer and Systems Sciences 163 (1998) 446–456.

[38] F.S. Samaria, A.C. Harter, Parameterization of a stochastic model for human face identification, IEEE Workshop on Applications of Computer Vision (1994) 138–142.

[39] J. Hull, A database for handwritten text recognition research, IEEE Transactions on Pattern Recognition and Machine Intelligence 16 (5) (1994) 550–554.

[40] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proceedings of IEEE 86 (11) (1998) 2278–2324.

[41] S. A. Nene, S. K. Nayar, H. Murase, Columbia Object Image Library (COIL-100), Technical Report CUCS-006-96, 1996.

[42] A.M. Martinez, A.C. Kak, PCA versus LDA, IEEE Transactions on Pattern Recognition and Machine Intelligence 23 (2) (2001) 228–233.

[43] J.Z. Wang, J. Li, G. Wiederhold, Simplicity: semantics sensitive integrated matching for picture libraries, IEEE Transactions on Pattern Recognition and Machine Intelligence 23 (9) (2001) 947–963.

[44] J. Yang, J.Y. Yang, Why can LDA be performed in PCA transformed space? Pattern Recognition 36 (2) (2003) 563–566.

[45] K. Liu, Y.Q. Cheng, J.Y. Yang, X. Liu, An efficient algorithm for Foley–Sammon optimal set of discriminant vectors by algebraic method, International Journal of Pattern Recognition and Artificial Intelligence 6 (5) (1992) 817–829.

[46] Y. Pang, Y. Yuan, Outlier-resisting graph embedding, Neurocomputing 73 (4–6) (2010) 968–974.

[47] X. Wang, X. Gao, Y. Yuan, D. Tao, J. Li, Semi-supervised Gaussian process latent variable model with pairwise constraints, Neurocomputing 73 (10–12) (2010) 2186–2195.

Mingbo Zhao is currently pursuing his Ph.D. degree at the Department of Electronic Engineering, City University of Hong Kong, Hong Kong. He received his B.Eng. degree and master degree from the Department of Electronic Engineering, Shanxi University, Shanxi, PR China in 2005 and 2008, respectively. His current interests include data mining, machine learning, pattern recognition, and their applications.

Zhao Zhang is currently working towards his Ph.D. degree at the Department of Electronic Engineering, City University of Hong Kong, Hong Kong. He received his B.Eng. (First Hons.) and master degrees from the Department of Computer Science and Technology, Nanjing Forestry University, Nanjing, PR China in 2008 and 2010, respectively. His current interests include machine learning, pattern recognition and computational intelligence.

Tommy W. S. Chow (IEEE M'93–SM'03) received his B.Sc. (First Hons.) and Ph.D. degrees from the University of Sunderland, Sunderland, UK. He joined the City University of Hong Kong, Hong Kong, as a Lecturer in 1988. He is currently a Professor in the Electronic Engineering Department. His research interests are in the area of Machine learning including Supervised and unsupervised learning, Data mining, Pattern recognition and fault diagnostic. He worked for NEI Reyrolle Technology at Hebburn, England developing digital simulator for transient network analyser. He then worked on a research project involving high current density current collection system for superconducting direct current machines, in collaboration with the Ministry of Defense (Navy) at Bath, England and the International Research and Development at Newcastle upon Tyne. He has authored or coauthored of over 120 technical papers in international journals, 5 book chapters, and over 60 technical papers in international conference proceedings.