Machine Learning (2024) 113:3889–3916
https://doi.org/10.1007/s10994-020-05937-w

New algorithms for trace-ratio problem with application to high-dimension and large-sample data dimensionality reduction

Wenya Shi[1] · **Gang Wu**[1]

Received: 26 May 2020 / Revised: 12 October 2020 / Accepted: 9 December 2020 / Published online: 2 March 2021

Abstract
Learning large-scale data sets with high dimensionality is a main concern in research areas including machine learning, visual recognition, information retrieval, to name a few. In many practical uses such as images, video, audio, and text processing, we have to face with high-dimension and large-sample data problems. The trace-ratio problem is a key problem for feature extraction and dimensionality reduction to circumvent the high dimensional space. However, it has been long believed that this problem has no closed-form solution, and one has to solve it by using some inner-outer iterative algorithms that are very time consuming. Therefore, efficient algorithms for high-dimension and large-sample trace-ratio problems are still lacking, especially for dense data problems. In this work, we present a closed-form solution for the trace-ratio problem, and propose two algorithms to solve it. Based on the formula and the randomized singular value decomposition, we first propose a randomized algorithm for solving high-dimension and large-sample dense trace-ratio problems. For high-dimension and large-sample sparse trace-ratio problems, we then propose an algorithm based on the closed-form solution and solving some consistent under-determined linear systems. Theoretical results are established to show the rationality and efficiency of the proposed methods. Numerical experiments are performed on some real-world data sets, which illustrate the superiority of the proposed algorithms over many state-of-the-art algorithms for high-dimension and large-sample dimensionality reduction problems.

Keywords Dimensionality reduction · Trace-ratio problem · High-dimension and large-sample data · Large-scale discriminant analysis · Randomized singular value decomposition (RSVD) · Inner-outer iterative algorithm

Editors: Tim Verdonck, Bart Baesens, María Óskarsdóttir and Seppe vanden Broucke.

This work is supported by the Fundamental Research Funds for the Central Universities under grant 2019XKQYMS89.

✉ Gang Wu
gangwu@cumt.edu.cn; gangwu76@126.com

Extended author information available on the last page of the article

1 Introduction

Feature engineering is the process of using domain knowledge of the data to create features that make machine learning algorithms work. It is a super-set of activities which include feature extraction, feature construction and feature selection, all of the them are important steps in feature engineering. Nowadays, a lot of practical applications of machine learning, visual recognition, text retrieval and bioinformatics need to deal with high-dimension and large-sample data efficiently (Alzubi and Abuarqoub 2020; Andras 2018; Chen et al. 2020; Liu et al. 2020; Vishwakarma and Singh 2019; Zhang et al. 2017). To effectively manipulate and analyze massive data, feature extraction and dimensionality reduction seem imperative in feature engineering (Eldén 2005; Fukunaga 1991; Gado et al. 2016; Hastie et al. 2001; Kokiopoulou et al. 2010; Liu et al. 2020; Park and Park 2008; Zhang et al. 2010).

Linear Discriminant Analysis (LDA) is a popular method for feature extraction, whose goal is to find a suitable linear transformation such that each high dimensional sample vector is projected into a low dimension vector, while maintaining the original cluster structure and achieving maximum class separability as much as possible (Fukunaga 1991; Hastie et al. 2001). In essence, it is also a process of dimensionality reduction. Trace-ratio problem is an important optimization problem involved in dimensionality reduction techniques such as Fisher linear discriminant analysis (LDA) (Belhumeur et al. 1997; Fukunaga 1991). More precisely, let $X = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n] \in \mathbb{R}^{d\times n}$ be a set of n training samples in a d-dimensional feature space. Assume that the data matrix is partitioned into k classes as $X = [X_1, \ldots, X_k]$, where X_j is the j-th set with n_j being the number of samples. Denote by $\mathbf{c}_j$ the centroid vector of X_j, and by $\mathbf{c}$ the global centroid vector of the training data. Let $\mathbf{1}_j = [1, 1, \ldots, 1]^T \in \mathbb{R}^j$, then the within-class scatter matrix is defined as

$$S_W = \sum_{j=1}^{k} \sum_{\mathbf{x}_i \in X_j} (\mathbf{x}_i - \mathbf{c}_j)(\mathbf{x}_i - \mathbf{c}_j)^T = H_W H_W^T,$$

where $H_W = [X_1 - \mathbf{c}_1\mathbf{1}_1^T, \ldots, X_k - \mathbf{c}_k\mathbf{1}_k^T] \in \mathbb{R}^{d\times n}$. The between-class scatter matrix is defined as

$$S_B = \sum_{j=1}^{k} n_j(\mathbf{c}_j - \mathbf{c})(\mathbf{c}_j - \mathbf{c})^T = H_B H_B^T,$$

where $H_B = [\sqrt{n_1}(\mathbf{c}_1 - \mathbf{c}), \sqrt{n_2}(\mathbf{c}_2 - \mathbf{c}), \ldots, \sqrt{n_k}(\mathbf{c}_k - \mathbf{c})] \in \mathbb{R}^{d\times k}$. The total scatter matrix is defined as

$$S_T = \sum_{j=1}^{n} (\mathbf{x}_j - \mathbf{c})(\mathbf{x}_j - \mathbf{c})^T = H_T H_T^T,$$

where $H_T = [\mathbf{x}_1 - \mathbf{c}, \mathbf{x}_2 - \mathbf{c}, \ldots, \mathbf{x}_n - \mathbf{c}] \in \mathbb{R}^{d\times n}$, moreover, it is well-known that (Park and Park 2008)

$$S_T = S_W + S_B.$$

The LDA method resorts to maximizing the between-class scatter distance while minimizing the within-class scatter distance. This gives the following *trace-ratio problem* (Fukunaga 1991; Kokiopoulou et al. 2010; Kramer et al. 2018; Ngo et al. 2012)

$$\widehat{\rho}^* = \max_{\substack{V \in \mathbb{R}^{d\times s} \\ V^T V = I}} \frac{tr(V^T S_B V)}{tr(V^T S_W V)}, \tag{1.1}$$

where tr(·) denotes the trace of a matrix, and $s \ll d$ is the reducing dimension. However, this problem is difficult to solve (Fukunaga 1991; Park and Park 2008), and in general a simpler *ratio-trace problem* is solved instead

$$\widetilde{\rho}^* = \max_{\substack{V \in \mathbb{R}^{d\times s} \\ V^T V = I}} tr\big((V^T S_W V)^{-1}(V^T S_B V)\big), \tag{1.2}$$

whose solution can be obtained from solving the following generalized eigenvalue problem

$$S_B \mathbf{v} = \lambda S_W \mathbf{v}. \tag{1.3}$$

Indeed, the trace-ratio problem and the ratio-trace problem are not mathematically equivalent (Park and Park 2008), and the trace-ratio problem (1.1) has regained great concerns in recent years (Guo et al. 2003; Jia et al. 2009; Jiang et al. 2017; Ngo et al. 2012; Nie et al. 2008; Wang et al. 2007; Zhang et al. 2010; Zhao et al. 2013). One reason is that the trace-ratio model (1.1) can yield markedly improved recognition results for supervised learning tasks compared to (1.2). However, it has been long believed that there is *no explicit solution* for the trace-ratio problem, and some commonly used techniques are inner-outer iterative algorithms (Jia et al. 2009; Ngo et al. 2012; Wang et al. 2007; Zhao et al. 2013). That is, inner iterations for solving eigenvectors and outer iterations for computing the trace-ratio value. More precisely, in the j-th outer iteration, we compute V_j from solving the eigenproblem with respect to $A - \rho_j B$ for given ρ_j, where $A = S_B$, $B = S_W$ (or S_T), and then we compute ρ_{j+1} by using V_j. The value of ρ_{j+1} is determined by different ways in different methods. For instance, the Newton-Lanczos algorithm (Ngo et al. 2012) and the method proposed by Wang et al. (2007) make use of the trace-ratio value $tr(V_j^T A V_j)/tr(V_j^T B V_j)$ as ρ_{j+1}. The GFST algorithm uses the bisection method to choose ρ_{j+1} (Guo et al. 2003), while the DNM algorithm searches ρ_{j+1} by calculating the first order expansion of the $A - \rho_j B$ (Jia et al. 2009). However, in high-dimension and large-sample problems, both the dimension and the number of the samples are very large, and the overhead of all these algorithms canbe prohibitive.

For high-dimension and large-sample data, there are many algorithms available to the ratio-trace problem. For instance, by using spectral graph analysis, the SRDA method casts discriminant analysis into a regression framework that facilitates both efficient computation and the use of regularization techniques (Cai et al. 2008). The LDADL method is designed for a new batch LDA model formulated as a regularized least squares (RLS) problem with multiple columns on the right-hand side (Zhang et al. 2017). The Rayleigh-Ritz discriminant analysis (RRDA) method is a gradient-type method based on the Rayleigh-Ritz framework for the generalized eigenvalue problem of LDA (Zhu and Huang 2014). There are also some randomized algorithms have been investigated, such as RFDA/RP (Ye et al. 2017) and FastLDA (Gado et al. 2016). However, all of the above algorithms are only designed for the *ratio-trace* problem (1.2) rather than the *trace-ratio* problem (1.1), and all of them are parameter-dependent. It is well-known that the optimal parameter is difficult to choose in practice, if there is no other information available in advance (Gui et al. 2014). Therefore, how to solve *high-dimension* and *large-sample trace-ratio* problem is an interesting topic that deserves further investigation (Jia et al. 2009; Ngo et al. 2012).

Table 1 Some notations used in this paper

Notations	Description
X	Training samples in a d-dimensional feature space
d, k	The data dimension and number of classes
s	Reducing dimension
N, n	Number of the total samples and number of training samples
$\mathbf{0}, I_i$	Zero matrix or vector, and identity matrix with dimension i
$\mathbf{1}_i$	The vector of all ones with dimension i
$rank(A), tr(A)$	Rank and trace of the matrix A
$\dim(\mathscr{W})$	Dimension of the subspace $\mathscr{W}$
$A^T, A^H, A^\dagger$	Transpose, conjugate transpose and Moore-Penrose inverse of A
$\text{span}\{W\}$	Space spanned by the columns of W
$\mathscr{N}(A), \mathscr{R}(A)$	Null space and range of the matrix A
$\mathscr{N}^\perp(A)$	The orthogonal complement space of $\mathscr{N}(A)$
$\|\cdot\|_2, \ \|\cdot\|_F$	2-norm and F-norm of a vector or matrix
$\mathscr{N}(A) \setminus \mathscr{N}(B)$	The subspace in $\mathscr{N}(A)$ but not in $\mathscr{N}(B)$
P_A	The orthogonal projector on the subspace $\text{span}\{A\}$
$\sin\angle(\mathscr{V}, \widetilde{\mathscr{V}})$	Sine of the angle between the subspaces $\mathscr{V} = span\{V\}$ and $\widetilde{\mathscr{V}} = span\{\widetilde{V}\}$
$\kappa_2(A)$	The 2-norm condition number of A
$Z \in \mathscr{W}$	Let W be a basis of the subspace $\mathscr{W}$, there is a matrix S, s.t., $Z = WS$

In this paper, we pay special attention to solving the high-dimension and large-sample trace-ratio problem efficiently. In Sect. 2, we provide an alternative way for (1.1), which does not rely on the inner-outer iterative framework. In Sect. 3, we provide a closed-form formula for this problem. Based on the formula and the randomized singular value decomposition (RSVD) (Halko et al. 2011), we propose a randomized algorithm for high-dimension and large-sample *dense* data problems. However, for large sparse data sets, RSVD will destroy the sparse structure of the original data. Thus, a method based on solving (consistent) under-determined systems is proposed for high-dimension and large-sample *sparse* data problems. Theoretical results are established to show the rationality and feasibility of the proposed methods. In Sect. 4, we perform some numerical experiments on some real-world data sets to illustrate the numerical behavior of our proposed algorithms. Concluding remarks are given in Sect. 5.

Throughout this paper, we suppose that the high-dimension and large-sample dense data matrix $X \in \mathbb{R}^{d\times n}$ is of full column rank. In this work, by high-dimension and large-sample data, we mean that both the dimensionality d and the number of samples n are very large and even in the same order, but with the assumption that $d \geq n$. Some notations used in this paper are listed in Table 1.

2 An alternative way to solve the trace-ratio problem

In this section, we present an alternative way for solving the trace-ratio problem to take the place of the inner-outer iterative framework. Let $\mathscr{W}$ be a subspace and let W be a basis of it. If Z is a matrix, in this paper, by $Z \in \mathscr{W}$, we mean there is a matrix S of appropriate size such that $Z = WS$.

We rewrite (1.1) as

$$\rho^* = \max_{\substack{V \in \mathbb{R}^{d\times s} \\ V^T V = I}} \frac{tr(V^T S_B V)}{tr(V^T S_T V)}. \tag{2.1}$$

It is seen that (1.1) and (2.1) are mathematically equivalent as $S_T = S_W + S_B$ (Guo et al. 2003). Denote by Z_1 an orthogonal basis for $\mathscr{N}^{\perp}(S_T)$, and by Z_2 an orthonormal basis for $\mathscr{N}(S_T)$, respectively. Then $Z = [Z_1, Z_2]$ is unitary, and for any matrix $V \in \mathbb{R}^{d\times s}$, there are matrices W_1, W_2 such that

$$V = Z_1 W_1 + Z_2 W_2 = V_{11} + V_{22},$$

where $V_{11} = Z_1 W_1 \in \mathscr{N}^{\perp}(S_T) = \mathscr{R}(S_T)$ as S_T is symmetric, and $V_{22} = Z_2 W_2 \in \mathscr{N}(S_T)$. Therefore,

$$S_B V = S_B Z_1 W_1 + S_B Z_2 W_2 = S_B Z_1 W_1 = S_B V_{11},$$

where we use the property that Z_2 is also in the null space of S_B. Indeed, as $S_T = S_B + S_W$ and all the three matrices S_B, S_W and S_T are symmetric positive semi-definite, the null space of S_T are in the intersection of the null spaces of S_B and S_W. Thus, we have $V^T S_B V = V_{11}^T S_B V_{11}$, and

$$tr(V^T S_B V) = tr(V_{11}^T S_B V_{11}). \tag{2.2}$$

Similarly, we can prove that

$$tr(V^T S_T V) = tr(V_{11}^T S_T V_{11}). \tag{2.3}$$

By (2.2) and (2.3),

$$\rho^* = \max_{\substack{V \in \mathbb{R}^{d\times s} \\ V^T V = I}} \frac{tr(V^T S_B V)}{tr(V^T S_T V)} = \max_{\substack{V \in \mathbb{R}^{d\times s}, V^T V = I \\ V \in \mathscr{N}^{\perp}(S_T)}} \frac{tr(V^T S_B V)}{tr(V^T S_T V)}, \tag{2.4}$$

which implies the information in $\mathscr{N}(S_T)$ has no contribution to the trace-ratio value at all. This coincides with the assertion that there is no useful information in $\mathscr{N}(S_T)$ for recognition (Park and Park 2008).

In view of (2.4), we focus on the following optimization problem in this work:

$$V^* = \arg \max_{\substack{V \in \mathbb{R}^{d\times s}, V^T V = I \\ V \in \mathscr{N}^{\perp}(S_T)}} \frac{tr(V^T S_B V)}{tr(V^T S_T V)}. \tag{2.5}$$

If the data matrix $X \in \mathbb{R}^{d\times n}$ is of full column rank, then $\dim(\mathscr{N}^{\perp}(S_T)) = \dim(\mathscr{R}(S_T)) = n - 1$ and $\dim(\mathscr{R}(S_W)) = n - k$ (Park and Park 2008). Thus,

$\dim(\mathcal{N}^{\perp}(S_T)) + \dim(\mathcal{N}(S_W)) = d + k - 1 > d$. That is, $\mathcal{N}^{\perp}(S_T)$ and $\mathcal{N}(S_W)$ have non-trivial intersection whose dimension is (at least) $k - 1$.

For any d-by-s orthonormal matrix $V \in \mathcal{N}^{\perp}(S_T)$, there holds

$$\frac{tr(V^T S_B V)}{tr(V^T S_T V)} = \frac{tr(V^T S_B V)}{tr(V^T S_B V) + tr(V^T S_W V)} \leq 1,$$

and the equality holds *if and only if* $tr(V^T S_W V) = 0$ and $tr(V^T S_B V) \neq 0$, which can be attained as $\mathcal{N}^{\perp}(S_T)$ and $\mathcal{N}(S_W)$ have non-trivial intersection. In other words, the orthonormal matrix $V^* \in \mathbb{R}^{d \times s}$ is a solution to (2.5) if $V^* \in \mathcal{N}(S_W) \cap \mathcal{N}^{\perp}(S_T)$, i.e., $V^* \in \mathcal{N}(S_W) \setminus \mathcal{N}(S_T)$. In conclusion, we have the following theorem for the solution of the trace-ratio problem (2.5).

Theorem 2.1 *The subspace $\mathcal{N}(S_W) \setminus \mathcal{N}(S_T)$ is the solution space of the trace-ratio problem* (2.5). *Let s be the reducing dimension, if $\dim(\mathcal{N}(S_W) \setminus \mathcal{N}(S_T)) \geq s$, then any orthonormal basis for an s-dimensional subspace of $\mathcal{N}(S_W) \setminus \mathcal{N}(S_T)$ is a solution to* (2.5).

Remark 2.1 Theorem 2.1 indicates that the trace-ratio problem does admit a simple solution when the dimension of the data points d is greater than or equal to the number of data points n. Note that the condition $d \geq n$ is general and can be satisfied in many real applications such as image, video, audio, and microarray data, and so on (Alzubi and Abuarqoub 2020; Andras 2018; Cai et al. 2008; Chen et al. 2020; Gado et al. 2016; Liu et al. 2020; Vishwakarma and Singh 2019; Zhang et al. 2017; Zhu and Huang 2014). Indeed, in big data era, the dimensionality or feature of the data points is huge, and it is often larger than the number of samples. Note that our strategy works when both d and n are very large and even in the same order.

Theorem 2.1 also reveals that our method is related to the null space method (Chen et al. 2000; Chu and Thye 2010; Huang et al. 2002; Lu and Wang 2012; Wu and Feng 2015). Indeed, the null space method tries to seek a solution V in the null space of S_W while maximizing $tr(V^T S_B V)$. The authors in Zhao et al. (2012) mentioned that the solution of the null space method and that of the trace-ratio LDA method are equivalent for singularity problem. We point out that our result is different from the one given in Zhao et al. (2012). First, due to (2.4), we exploit the model (2.5) as an alternative to (2.1). Second, the solution of the null space method is also in $\mathcal{N}(S_W) \setminus \mathcal{N}(S_T)$, so the conclusion in Zhao et al. (2012) is just a special case of our conclusion.

When both the dimension d and the number of training samples n are large, however, computing the null spaces of S_W and S_T directly is prohibitive in practice. Consequently, a direct application of the null space method to high-dimension and large-sample dense data sets is impractical. Based on Theorem 2.1, we aim to seek an efficient algorithm to solve (2.5) in the following work.

3 New algorithms for solving the trace-ratio problem on high-dimension and large-sample data sets

In large-scale discriminant analysis, the number of samples can be very large and even in the order of the data dimension d (Cai et al. 2008; Gado et al. 2016; Zhu and Huang 2014). In this situation, both the rows and the columns of the data matrix are very large, and a direct decomposition to the data matrix is infeasible. In this section, we give insight into fast implementations on Theorem 2.1 for large data matrices.

3.1 A closed-form solution

Consider the *centered* data matrix $\overline{X} = X(I_n - \mathbf{1}_n\mathbf{1}_n^T/n) = XL_T$, where $L_T = I_n - \mathbf{1}_n\mathbf{1}_n^T/n$. Define W as the following $n \times n$ block diagonal matrix

$$W = \begin{bmatrix} \frac{1}{n_1}\mathbf{1}_{n_1}\mathbf{1}_{n_1}^T & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \frac{1}{n_2}\mathbf{1}_{n_2}\mathbf{1}_{n_2}^T & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \frac{1}{n_k}\mathbf{1}_{n_k}\mathbf{1}_{n_k}^T \end{bmatrix},$$

then we have $S_B = \overline{X}W\overline{X}^T$ and $S_T = \overline{X}\overline{X}^T$ (Cai et al. 2008). Thus, S_B and S_T can be rewritten as

$$S_B = XL_TWL_TX^T, \quad S_T = XL_TL_TX^T = XL_TX^T, \tag{3.1}$$

where we use the fact that $L_T^2 = L_T$. Moreover, from $W\mathbf{1}_n = \mathbf{1}_n$ and $L_T\mathbf{1}_n = \mathbf{0}$, we obtain

$$S_W = S_T - S_B = X(L_T - L_TWL_T)X^T = X(I_n - W)X^T. \tag{3.2}$$

In view of Theorem 2.1, the framework of our method is composed of two steps:

(i) *First, we compute a basis for a k-dimensional subspace of $\mathcal{N}(S_W)$.*

Theorem 3.1 *Denote by Y the following $n \times k$ matrix*

$$Y = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_k] \equiv \begin{bmatrix} \mathbf{1}_{n_1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{1}_{n_2} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{1}_{n_k} \end{bmatrix} \in \mathbb{R}^{n\times k}, \tag{3.3}$$

where $\mathbf{1}_{n_i}$ is the one vector of size n_i. If X is of full column rank, then $F = (X^T)^\dagger Y$ is a basis for a k-dimension subspace of $\mathcal{N}(S_W)$, where $(X^T)^\dagger$ denotes the Moore-Penrose inverse of X^T.

Proof As X and Y are of full column rank, $F = (X^T)^\dagger Y$ is also of full column rank. Moreover, we obtain from (3.3) that $W\mathbf{y}_i = \mathbf{y}_i$, $i = 1, 2, \ldots, k$. By (3.2),

$$S_WF = X(I_n - W)X^T\left((X^T)^\dagger Y\right) = X(I_n - W)Y = \mathbf{0}, \tag{3.4}$$

which completes the proof. □

(ii) *Second, we remove some information in $\mathcal{N}(S_T)$* from *span*$\{F\}$.

Theorem 3.2 *Let* $\underline{Y} = (I_n - \mathbf{1}_n\mathbf{1}_n^T)Y$, *and let* $\overline{Y} \in \mathbb{R}^{n\times(k-1)}$ *be an orthonormal basis of* $span\{\underline{Y}\}$, *then*

$$\overline{F} = (X^T)^\dagger \overline{Y} \in \mathcal{M}(S_W) \setminus \mathcal{M}(S_T). \tag{3.5}$$

Specifically, if $F_Q \in \mathbb{R}^{d\times(k-1)}$ *is the Q-factor of the economized QR factorization of* $\overline{F}$, *then* F_Q *is a solution to* (2.5).

Proof On one hand, as $(X^T)^\dagger = X(X^TX)^{-1}$ and $W\mathbf{1}_n = \mathbf{1}_n$, we obtain from (3.4) that

$$\begin{aligned} S_W\big((X^T)^\dagger \underline{Y}\big) &= X(I_n - W)X^T\big((X^T)^\dagger \underline{Y}\big) \\ &= X(I_n - W)(X^TX)(X^TX)^{-1}(I_n - \mathbf{1}_n\mathbf{1}_n^T)Y \\ &= X(I_n - W)Y - X(I_n - W)\mathbf{1}_n\mathbf{1}_n^TY \\ &= \mathbf{0}, \end{aligned}$$

i.e., $(X^T)^\dagger \underline{Y}$ is in the null space of S_W. On the other hand, as $L_T\mathbf{1}_n = \mathbf{0}$, we get

$$\begin{aligned} S_T\big((X^T)^\dagger \underline{Y}\big) &= XL_TX^T\big((X^T)^\dagger \underline{Y}\big) \\ &= XL_T(X^TX)(X^TX)^{-1}(I_n - \mathbf{1}_n\mathbf{1}_n^T)Y \\ &= XL_TY - XL_T\mathbf{1}_n\mathbf{1}_n^TY \\ &= XL_TY. \end{aligned}$$

Since $\mathcal{M}(L_T) = span\{\mathbf{1}_n\}$, it follows from (3.3) that $L_T\mathbf{y}_i \neq \mathbf{0}$ and $XL_T\mathbf{y}_i \neq \mathbf{0}$, $i = 1, \ldots, k$. Thus, $S_T\big((X^T)^\dagger \underline{Y}\big) \neq \mathbf{0}$, and it follows that

$$(X^T)^\dagger \underline{Y} \in \mathcal{M}(S_W) \setminus \mathcal{M}(S_T).$$

Further, as $\overline{Y}$ is an orthonormal basis for $span\{\underline{Y}\}$, we have $(X^T)^\dagger \overline{Y} \in span\{(X^T)^\dagger \underline{Y}\}$, and thus the d-by-$(k-1)$ matrix $\overline{F} = (X^T)^\dagger \overline{Y} \in \mathcal{M}(S_W) \setminus \mathcal{M}(S_T)$. Since F_Q is the Q-factor of the economized QR factorization of $\overline{F}$, by Theorem 2.1, it is a solution to (2.5). □

Remark 3.1 From Sect. 2, if the data matrix $X \in \mathbb{R}^{d\times n}$ is of full column rank, we have $dim(\mathcal{M}(S_W) \setminus \mathcal{M}(S_T)) \geq k - 1$. Thus, without loss of generality, we usually set the reducing dimension $s = k - 1$. If the reducing dimension $s < k - 1$, then any s columns of F_Q is a solution to (2.5). For instance, we can choose the first s columns of F_Q, i.e., $F_Q(:, 1:s)$ as a solution.

3.2 A randomized algorithm for the trace-ratio problem on high-dimension and large-sample *dense* data sets

In this paper, we assume that the data matrix X has full column rank, however, it may be ill-conditioned in practice, and a direct computation on (3.5) is inadvisable. To deal with this problem and make the solution less sensitive to perturbations, the technique of truncated singular value decomposition (TSVD) is often utilized (Eldén 2005; Golub and Van Loan 2013). However, for high-dimension and large-sample data, both d and n are very large, thus the overhead for truncated singular value decomposition will still be prohibitive.

To overcome this difficulty, we make use of the randomized singular value algorithm (RSVD) (Gu 2015; Halko et al. 2011; Martinsson et al. 2011; Woodruff 2014) to produce a low-rank approximation $\widetilde{X}^T$ to X^T, and then compute an approximation to $\overline{F}$. Now we briefly introduce the randomized singular value decomposition, for more details on its implementation and theoretical background, we refer to Gu (2015), Halko et al. (2011). Notice that the integer r in the algorithm is a user-provided parameter whose choice is problem-dependent.

Algorithm 1 A Randomized Singular Value Decomposition (RSVD) Accelerated by the Power Iteration (Halko et al. 2011)

Input: Given a $d \times n$ matrix X, a user-provided target rank r ($r \ll n$), an over-sampling parameter p ($r + p \ll n$), and a parameter q (say, $q = 1$ or $q = 2$);
Output: An approximate rank-r approximation $\widetilde{X}^T$ to X^T;
1. Draw an $n \times (r + p)$ standard Gaussian matrix Ω;
2. Form $G = (XX^T)^q X\Omega$ by multiplying alternately with X and X^T;
3. Construct a matrix Q_1 whose columns form an orthonormal basis for the range of G;
4. Form $B = X^T Q_1$, and compute the economized singular value decomposition of $B = U_2 \Sigma_1 \widetilde{V}^T$, and let $V_1 = Q_1 \widetilde{V}$;
5. Denote $\widetilde{U}_r = U_2(:, 1:r)$, $\widetilde{V}_r = V_1(:, 1:r)$, $\widetilde{\Sigma}_r = \Sigma_1(1:r, 1:r)$, and let $\widetilde{X}^T = \widetilde{U}_r \widetilde{\Sigma}_r \widetilde{V}_r^T$.

Let $\widetilde{X}^T$ be the approximation to X^T obtained from Algorithm 1, the idea is to compute an approximation to $\overline{F}$ by solving the following optimization problem:

$$\widetilde{F} = \arg \min_{F \in \mathbb{R}^{d \times (k-1)}} \|\widetilde{X}^T F - \overline{Y}\|_F. \tag{3.6}$$

Thus, we propose the following randomized algorithm for solving the high-dimension and large-sample trace-ratio problem (2.5). The main overhead in Algorithm 2 is to compute $\widetilde{U}_r, \widetilde{V}_r$ and $\widetilde{\Sigma}_r$ by using Algorithm 1, which needs about $\mathcal{O}((d+n)r^2)$ flops, and the main storage requirement is to store a $d \times r$ matrix (Halko et al. 2011).

Algorithm 2 A Randomized Algorithm for the Trace-Ratio Problem on High-dimension and Large-Sample *Dense* Data Sets

Input: The data matrix $X \in \mathbb{R}^{d \times n}$ ($d \geq n$), a user-provided target rank r ($r \ll n$), an over-sampling parameter p (say, $p \geq 4$), and a parameter q (say, $q = 1$ or $q = 2$);
Output: The approximate solution $\widetilde{F}_Q$;
1. Use Algorithm 1 to compute a rank-r approximation to X^T: $\widetilde{X}^T \equiv \widetilde{U}_r \widetilde{\Sigma}_r \widetilde{V}_r^T$;
2. Form $\widetilde{F} = (\widetilde{X}^T)^\dagger \overline{Y} = \widetilde{V}_r \widetilde{\Sigma}_r^{-1} \widetilde{U}_r^T \overline{Y}$;
3. We orthonormalize the columns of $\widetilde{F}$, and denote by $\widetilde{F}_Q$ the resulting matrix.

Remark 3.2 We stress that Theorems 2.1, 3.1 and 3.2 hold for a general data matrix X, whether it is dense or not, and Algorithm 2 applies to both dense and sparse data sets. However, when the data matrix X is sparse, RSVD will destroy the sparse structure of the original data. Thus, Algorithm 2 is more appropriate to dense data sets.

We note that the key step in Algorithm 2 is to compute an approximation $\widetilde{X}^T$ to X^T by using Algorithm 1. Indeed, $\widetilde{X}^T$ (with rank-r) can be viewed as an approximation of X_r^T, i.e., the truncated SVD (TSVD) of X^T with rank-r. Indeed, let X_r^T be the *best* rank-r approximation to X^T in terms of 2-norm or Frobenius norm (Golub and Van Loan 2013), then $(X_r^T)^\dagger \overline{Y}$ is a reasonable choice to approximate $(X^T)^\dagger \overline{Y}$. The key problems are:

(a) *How large is the distance between the approximation* $(\widetilde{X}^T)^\dagger \overline{Y}$ *from* (3.6) *and* $(X_r^T)^\dagger \overline{Y}$ *from truncated SVD (TSVD)?*
(b) *Why the proposed randomized algorithm still works even if the solution is "inexact"?*

To answer these questions, we consider the distance between the subspace spanned by $(\widetilde{X}^T)^\dagger \overline{Y}$ and that by $(X_r^T)^\dagger \overline{Y}$. Along the line of Algorithm 2, we will divide our analysis into two procedures.

(i) *First, we establish the relationship between* X_r^T *and the approximation* $\widetilde{X}^T$.

Let P_G be the orthogonal projector onto the subspace span$\{G\}$. In (Halko et al. 2011, pp.275), Halko *et al.* derived the following deviation bound for the randomized SVD algorithm without power scheme (i.e., with $q = 0$): for all $u, t \geq 1$, there holds

$$\|(I - P_G)X\|_2 \leq \left(1 + t\sqrt{\frac{3r}{p+1}} + ut\frac{e\sqrt{r+p}}{p+1}\right)\sigma_{r+1} + t\frac{e\sqrt{r+p}}{p+1}\left(\sum_{j>r}\sigma_j^2\right)^{1/2}, \ p \geq 4, \tag{3.7}$$

with failure probability at most $2t^{-p} + e^{-u^2/2}$. Here $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n > 0$ are the (nonzero) singular values of X. For more bounds on the approximation obtained from the randomized power method, we refer to Gu (2015), Musco and Musco (2015), Woodruff (2014).

Based on (3.7), we will give a deviation bound for the approximation obtained from the randomized SVD accelerated by power iteration with $q \geq 1$. Let

$$L = (XX^T)^q X, \quad \textit{and} \quad G = L\Omega,$$

then it follows from Algorithm 1 that $Q_1 Q_1^T$ is the orthogonal projector onto the subspace span$\{G\}$. By (3.7),

$$\|(I - Q_1Q_1^T)L\|_2 \leq \left(1 + t\sqrt{\frac{3r}{p+1}} + ut\frac{e\sqrt{r+p}}{p+1}\right)\widetilde{\sigma}_{r+1} + t\frac{e\sqrt{r+p}}{p+1}\left(\sum_{j>r}\widetilde{\sigma}_j^2\right)^{1/2}, \ p \geq 4,$$

with failure probability at most $2t^{-p} + e^{-u^2/2}$, where $\widetilde{\sigma}_1 \geq \widetilde{\sigma}_2 \geq \cdots \geq \widetilde{\sigma}_n$ are the singular values of L. Let $X^T = U\Sigma P^T$ be the economized singular value decomposition of X^T, where $\Sigma = diag(\sigma_1, \sigma_2, \ldots, \sigma_n) \in \mathbb{R}^{n\times n}$ with $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n > 0$, and $U \in \mathbb{R}^{n\times n}$, $P \in \mathbb{R}^{d\times n}$ are two orthonormal matrices. Then we have $L = P\Sigma^{2q+1}U^T$ and $\widetilde{\sigma}_i = \sigma_i^{2q+1}$, $i = 1, 2, \ldots, n$. Moreover, it was shown that (Halko et al. 2011, pp.270)

$$\|(I - Q_1Q_1^T)X\|_2 \leq \|(I - Q_1Q_1^T)L\|_2^{1/(2q+1)}.$$

As a result, for all $u, t \geq 1$ and $p \geq 4$, we have

$$\begin{aligned}\|X^T - X^TQ_1Q_1^T\|_2 &= \|(I - Q_1Q_1^T)X\|_2 \\ &\leq \left[\left(1 + t\sqrt{\frac{3r}{p+1}} + ut\frac{e\sqrt{r+p}}{p+1}\right)\sigma_{r+1}^{2q+1} + t\frac{e\sqrt{r+p}}{p+1}\left(\sum_{j>r}\sigma_j^{2(2q+1)}\right)^{\frac{1}{2}}\right]^{\frac{1}{2q+1}}\end{aligned} \tag{3.8}$$

with failure probability at most $2t^{-p} + e^{-u^2/2}$.

Further, according to Steps 4–5 of Algorithm 1, we have $X^TQ_1Q_1^T = U_2\Sigma_1V_1^T$, with $\widetilde{X}^T \equiv \widetilde{U}_r\widetilde{\Sigma}_r\widetilde{V}_r^T$ being its rank-r approximation. Let $\sigma_{r+1}(X^TQ_1Q_1^T)$ be the $(r+1)$-th largest singular value of $X^TQ_1Q_1^T$, according to the interlacing theorem for singular values (Golub

and Van Loan 2013), we have $\sigma_{r+1}(X^TQ_1Q_1^T) \le \sigma_{r+1}$, where σ_{r+1} is the $(r+1)$-th largest singular value of X^T. Thus, from Steps 4–5 of Algorithm 1, we have

$$\|X^T - X^TQ_1Q_1^T\| = \|U_2\Sigma_1V_1^T - \widetilde{U}_r\widetilde{\Sigma}_r\widetilde{V}_r^T\|_2 = \sigma_{r+1}(X^TQ_1Q_1^T) \le \sigma_{r+1}. \tag{3.9}$$

According to (3.8) and (3.9), we get

$$\begin{aligned}\|X^T - \widetilde{X}^T\|_2 &= \|X^T - \widetilde{U}_r\widetilde{\Sigma}_r\widetilde{V}_r^T\|_2 \le \|X^T - X^TQ_1Q_1^T\|_2 + \|U_2\Sigma_1V_1^T - \widetilde{U}_r\widetilde{\Sigma}_r\widetilde{V}_r^T\|_2 \\ &\le \left[\left(1 + t\sqrt{\frac{3r}{p+1}} + ut\frac{e\sqrt{r+p}}{p+1}\right)\sigma_{r+1}^{2q+1} + t\frac{e\sqrt{r+p}}{p+1}\left(\sum_{j>r}\sigma_j^{2(2q+1)}\right)^{\frac{1}{2}}\right]^{\frac{1}{2q+1}} + \sigma_{r+1}\end{aligned} \tag{3.10}$$

with failure probability at most $2t^{-p} + e^{-u^2/2}$. Denote

$$\eta = \left[\left(1 + e\sqrt{\frac{3r}{p}} + e^2\sqrt{\frac{2(r+p)}{p}}\right)\sigma_{r+1}^{2q+1} + e^2\frac{\sqrt{r+p}}{p}\left(\sum_{j>r}\sigma_j^{2(2q+1)}\right)^{\frac{1}{2}}\right]^{\frac{1}{2q+1}} + \sigma_{r+1}, \tag{3.11}$$

if we take $t = e$, $u = \sqrt{2p}$, then (3.10) reduces to

$$\|X^T - \widetilde{X}^T\|_2 \le \eta = \mu\sigma_{r+1} = \mathcal{O}(\sigma_{r+1}), \tag{3.12}$$

with failure probability at most $3e^{-p}$, where $\mu = \eta/\sigma_{r+1}$. Note that the value of μ depends on the choice of q and could be in the order of $\mathcal{O}(1)$.

Recall that $X^T = U\Sigma P^T$ is the economized singular value decomposition of X^T, if we partition $U = [U_r,\ U_-]$, $\Sigma = \begin{pmatrix}\Sigma_r & \mathbf{0}\\ \mathbf{0} & \Sigma_-\end{pmatrix}$, and $P = [P_r,\ P_-]$, where $U_r \in \mathbb{R}^{n\times r}$, $\Sigma_r \in \mathbb{R}^{r\times r}$, and $P_r \in \mathbb{R}^{d\times r}$, then

$$(X^T)^\dagger = P\Sigma^\dagger U^T = [P_r, P_-]\begin{pmatrix}\Sigma_r^{-1} & \mathbf{0}\\ \mathbf{0} & \Sigma_-^{-1}\end{pmatrix}\begin{pmatrix}U_r^T\\ U_-^T\end{pmatrix} = P_r\Sigma_r^{-1}U_r^T + P_-\Sigma_-^{-1}U_-^T.$$

Notice that $X_r^T = U_r\Sigma_rP_r^T$, we have $(X_r^T)^\dagger = P_r\Sigma_r^{-1}U_r^T$, moreover,

$$\|(X_r^T)^\dagger\|_2 = \frac{1}{\sigma_r},\ \ \textit{and}\ \ \|X^T - X_r^T\|_2 = \sigma_{r+1}. \tag{3.13}$$

By (3.12) and (3.13),

$$\|X_r^T - \widetilde{X}^T\|_2 \le \|X^T - X_r^T\|_2 + \|X^T - \widetilde{X}^T\|_2 \le (1+\mu)\sigma_{r+1}. \tag{3.14}$$

(ii) *Second, we consider the angle between the subspaces* $\widetilde{\mathcal{X}} = span\{(\widetilde{X}^T)^\dagger\overline{Y}\}$ *and* $\mathcal{X} = span\{(X_r^T)^\dagger\overline{Y}\}$.

Assume that $(\widetilde{X}^T)^\dagger\overline{Y}$ and $(X_r^T)^\dagger\overline{Y}$ are of full column rank, then

$$rank\big((\widetilde{X}^T)^\dagger\overline{Y}\big) = rank\big((X_r^T)^\dagger\overline{Y}\big),$$

and we have from Wedin (1973) and (3.14) that

$$\begin{aligned}\|(X_r^T)^\dagger\overline{Y}-(\widetilde{X}^T)^\dagger\overline{Y}\|_2 &\le \left(\frac{\|(X_r^T)^\dagger-(\widetilde{X}^T)^\dagger\|_2}{\|(X_r^T)^\dagger\|_2}\right)\|(X_r^T)^\dagger\|_2\|\overline{Y}\|_2\\ &\le \sqrt{2}\|(\widetilde{X}^T)^\dagger\|_2\|X_r^T-\widetilde{X}^T\|_2\|(X_r^T)^\dagger\|_2\|\overline{Y}\|_2\\ &\le \sqrt{2}(1+\mu)\|(\widetilde{X}^T)^\dagger\|_2\frac{\sigma_{r+1}}{\sigma_r},\end{aligned} \tag{3.15}$$

where we use the fact that $\overline{Y}$ is orthonormal. On the other hand,

$$\|[(X_r^T)^\dagger\overline{Y}]^\dagger\|_2=\frac{1}{\sigma_{\min}((X_r^T)^\dagger\overline{Y})}\le\frac{1}{\sigma_{\min}((X_r^T)^\dagger)}=\sigma_{\max}(X_r^T)=\sigma_1. \tag{3.16}$$

Then from Sun (1984), (3.15) and (3.16), we have

$$\begin{aligned}\sin\angle(\mathscr{X},\widetilde{\mathscr{X}}) &= \|P_{(X_r^T)^\dagger\overline{Y}}-P_{(\widetilde{X}^T)^\dagger\overline{Y}}\|_2\\ &\le \min\{\|[(X_r^T)^\dagger\overline{Y}]^\dagger\|_2,\|[(\widetilde{X}^T)^\dagger\overline{Y}]^\dagger\|_2\}\|(X_r^T)^\dagger\overline{Y}-(\widetilde{X}^T)^\dagger\overline{Y}\|_2\\ &\le \|[(X_r^T)^\dagger\overline{Y}]^\dagger\|_2\|(X_r^T)^\dagger\overline{Y}-(\widetilde{X}^T)^\dagger\overline{Y}\|_2\\ &\le \sqrt{2}(1+\mu)\|(\widetilde{X}^T)^\dagger\|_2(\sigma_1/\sigma_r)\cdot\sigma_{r+1},\end{aligned} \tag{3.17}$$

where $P_{(X_r^T)^\dagger\overline{Y}}$ and $P_{(\widetilde{X}^T)^\dagger\overline{Y}}$ are the orthogonal projectors onto the subspace span$\{(X_r^T)^\dagger\overline{Y}\}$ and span$\{(\widetilde{X}^T)^\dagger\overline{Y}\}$, respectively.

In summary, we have the following theorem for Algorithm 2. Let $\overline{F}_Q$ be the Q-factor of the economized QR factorization of $(X_r^T)^\dagger\overline{Y}$, it provides a bound between the distance between the subspaces spanned by the "truncated" solution $\overline{F}_Q$ and the "approximate" solution $\widetilde{F}_Q$:

Theorem 3.3 *Let* $\mathscr{X}=span\{(X_r^T)^\dagger\overline{Y}\}=span\{\overline{F}_Q\}$ *and* $\widetilde{\mathscr{X}}=span\{(\widetilde{X}^T)^\dagger\overline{Y}\}=span\{\widetilde{F}_Q\}$. *If* $(X_r^T)^\dagger\overline{Y}$ *and* $(\widetilde{X}^T)^\dagger\overline{Y}$ *are of full column rank, then under the above notations, we have*

$$\sin\angle(\mathscr{X},\widetilde{\mathscr{X}})\le\left(\sqrt{2}(1+\mu)\kappa(X_r)\|(\widetilde{X}^T)^\dagger\|_2\right)\cdot\sigma_{r+1} \tag{3.18}$$

with failure probability at most $3e^{-p}$, *where* $p\ge 4$.

Remark 3.3 Recall that $\widetilde{X}^T$ is an approximation to X_r^T, as $(X_r^T)^\dagger\overline{Y}$ is of full column rank, the assumption that $rank((\widetilde{X}^T)^\dagger\overline{Y})=rank((X_r^T)^\dagger\overline{Y})$ is not strict, moreover, we have that $\|(\widetilde{X}^T)^\dagger\|_2=\mathscr{O}\frac{1}{\sigma_r}$). As a result, Theorem 3.3 indicates that the distance between the "truncated" solution space $span\{(X_r^T)^\dagger\overline{Y}\}$ and the "approximate" solution space $span\{(\widetilde{X}^T)^\dagger\overline{Y}\}$ is in the order of $\mathscr{O}\left(\kappa_2(X_r)\frac{\sigma_{r+1}}{\sigma_r}\right)$.

On the other hand, we point out that the upper bound given in (3.18) may be pessimistic in practice and even much larger than one. Our contribution is to indicate that the difference between the two subspaces $\mathscr{X}$ and $\widetilde{\mathscr{X}}$ is closely related to the condition number $\kappa(X_r)$ and the gap $\frac{\sigma_{r+1}}{\sigma_r}$ between singular values. If the singular values of X decay quickly and X_r is not too ill-conditioned, the distance between $\mathscr{X}$ and $\widetilde{\mathscr{X}}$ will be small.

Finally, we briefly interpret why our randomized algorithm works for recognition. Let $\widehat{\mathbf{a}}_i$ be a sample from the training set, and let $\widehat{\mathbf{b}}_j$ be a sample from the testing set.

In the widely used nearest neighbour classifier (NN) (Cover and Hart 1967), the class membership is from investigating the Euclidean distance as follows:

$$d_{ij} = \|\overline{F}_Q\overline{F}_Q^T(\widehat{\mathbf{a}}_i - \widehat{\mathbf{b}}_j)\|_2 = \|\overline{F}_Q^T(\widehat{\mathbf{a}}_i - \widehat{\mathbf{b}}_j)\|_2.$$

Along the line of (Wu et al. 2017, Theorem 6), we can establish the following relationship between the Euclidean distances obtained from $\overline{F}_Q$ and $\widetilde{F}_Q$.

Theorem 3.4 *Let $\overline{F}_Q, \widetilde{F}_Q \in \mathbb{R}^{d\times s}$ be orthonormal matrices whose columns span the "truncated" solution space $\overline{\mathscr{X}}$ and the "approximate" solution space $\widetilde{\mathscr{X}}$ of (2.5), respectively. Denote by $d_{ij} = \|\overline{F}_Q^T(\widehat{\mathbf{a}}_i - \widehat{\mathbf{b}}_j)\|_2$ and by $\widetilde{d}_{ij} = \|\widetilde{F}_Q^T(\widehat{\mathbf{a}}_i - \widehat{\mathbf{b}}_j)\|_2$ the "truncated" and the "approximate" Euclidean distances, respectively. If $\|\widehat{\mathbf{a}}_i\|_2, \|\widehat{\mathbf{b}}_j\|_2 = 1$ and $\cos\angle(\overline{\mathscr{X}}, \widetilde{\mathscr{X}}) \neq 0$, then*

$$\frac{\widetilde{d}_{ij} - 2\sin\angle(\overline{\mathscr{X}}, \widetilde{\mathscr{X}})}{\cos\angle(\overline{\mathscr{X}}, \widetilde{\mathscr{X}})} \leq d_{ij} \leq \widetilde{d}_{ij}\cos\angle(\overline{\mathscr{X}}, \widetilde{\mathscr{X}}) + 2\sin\angle(\overline{\mathscr{X}}, \widetilde{\mathscr{X}}). \tag{3.19}$$

Remark 3.4 Theorem 3.4 shows that if $\sin\angle(\overline{\mathscr{X}}, \widetilde{\mathscr{X}})$ is not large, then the distances $\{d_{ij}\}'s$ and $\{\widetilde{d}_{ij}\}'s$ will be close to each other. Consequently, the recognition rates obtained from the "truncated" solution and the "approximate" solution will be about the same; see (Wu et al. 2017) for more details.

3.3 An iterative algorithm for the trace-ratio problem on high-dimension and large-sample *sparse* data sets

In many high-dimension data processing tasks such as text processing, the data matrix is often large and sparse (Cai et al. 2008; Tavernier et al. 2017; Zhu and Huang 2014). In this situation, applying the randomized SVD to X are unfavorable because it will destroy the sparse structure of the original data. Thus, it is necessary to propose new technologies for solving the trace-ratio problem on large and sparse data sets. It follows from Theorem 3.2 that $\overline{F} = (X^T)^\dagger\overline{Y}$ is in the solution space $\mathscr{N}(S_W) \setminus \mathscr{N}(S_T)$. As X is of full column rank, we have

$$X^T\overline{F} = X^T(X^T)^\dagger\overline{Y} = \overline{Y}. \tag{3.20}$$

Thus, the idea is to solve the underdetermined (consistent) block system (3.20) for $\overline{F}$, with no need to form $(X^T)^\dagger$ explicitly. This avoids destroying the sparse structure of the original data matrix. We are ready to present the following algorithm for the large sparse trace-ratio problem.

Algorithm 3 An Iterative Algorithm for the Trace-Ratio Problem on Large *Sparse* Data Sets

Input: The data matrix $X \in \mathbb{R}^{d\times n}$ ($d \geq n$), and a parameter tol;
Output: The approximate solution $\widetilde{F}_Q$;
1. Form $\overline{Y}$ and solve the following underdetermined systems iteratively using tol as the convergence tolerance:

$$\widetilde{\mathbf{f}}_j = \arg\min_{\widetilde{\mathbf{f}}} \left(\sum_{i=1}^{n} (\widetilde{\mathbf{f}}^T \mathbf{x}_i - \bar{y}_i^j)^2 \right), \quad j = 1, \ldots, k-1, \tag{3.21}$$

where $\mathbf{x}_i$ is the i-th column of X and $\bar{y}_i^j$ stands for the (i, j)-th element of $\overline{Y}$;
2. Let $\widetilde{F} = [\widetilde{\mathbf{f}}_1, \widetilde{\mathbf{f}}_2, \ldots, \widetilde{\mathbf{f}}_{k-1}]$, we orthogonalize its columns and denote by $\widetilde{F}_Q$ the resulting matrix.

Remark 3.5 Unlike the methods given in Jia et al. (2009), Ngo et al. (2012), Wang et al. (2007), Zhao et al. (2013) for the trace-ratio problem, Algorithm 3 is *not* an *inner-outer iterative* method, so it is much simpler than those conventional methods. The main overhead of this algorithm is in Step 1, where we have to solve $k-1$ least squares problems iteratively. We can use the `lsrq` package provided by Cai et al. (2008) to solve (3.21), which is a modification to the LSQR algorithm due to Paige and Saunders (1982). Thus, Algorithm 3 can preserve the sparse structure of X, and the main storage requirement is to store the matrix $\widetilde{F}_Q$ of size $d \times (k-1)$.

The following theorem gives an error bound on solving (3.20) iteratively by Algorithm 3.

Theorem 3.5 *Let $\overline{F}$ be the exact solution to* (3.20)*, and let $\widetilde{F}$ be the computed solution of* (3.21) *satisfying $\|X^T\widetilde{F} - \overline{Y}\|_F / \|\overline{Y}\|_F \leq tol$, where tol is the convergence tolerance used in Algorithm* 3*. Denote by $\mathscr{F} = span\{\overline{F}\}$ and by $\widetilde{\mathscr{F}} = span\{\widetilde{F}\}$, if $\widetilde{F}$ and $\overline{F}$ are of full column rank, and the columns of $\widetilde{F} - \overline{F}$ are not in $\mathscr{N}(X^T)$, then*

$$\sin\angle(\mathscr{F}, \widetilde{\mathscr{F}}) \leq \sqrt{k-1}\kappa_2(X) \cdot tol. \tag{3.22}$$

Proof Since $X^T\overline{F} = \overline{Y}$ with $\overline{Y}$ being orthonormal, we have

$$\|X^T(\widetilde{F} - \overline{F})\|_F = \|X^T\widetilde{F} - \overline{Y}\|_F \leq \|\overline{Y}\|_F \cdot tol \leq \sqrt{k-1} \cdot tol,$$

where we used the convergence criterion $\|X^T\widetilde{F} - \overline{Y}\|_F / \|\overline{Y}\|_F \leq tol$ in Algorithm 3, as well as the fact that $\|\overline{Y}\|_F = \sqrt{k-1}$.

If the columns of $\widetilde{F} - \overline{F}$ are not in $\mathscr{N}(X^T)$, then

$$\|X^T(\widetilde{F} - \overline{F})\|_F^2 \geq \sigma_{\min}^2(X^T)\|\widetilde{F} - \overline{F}\|_F^2,$$

where $\sigma_{\min}(X^T) = \sigma_{\min}(X)$ is the smallest nonzero singular value of X^T. So it follows that

$$\|\widetilde{F} - \overline{F}\|_2 \leq \|\widetilde{F} - \overline{F}\|_F \leq \sqrt{k-1} \cdot \frac{tol}{\sigma_{\min}(X)}. \tag{3.23}$$

On the other hand, $\sigma_{\min}(\overline{F}) = \sigma_{\min}((X^T)^\dagger\overline{Y}) \geq \sigma_{\min}((X^T)^\dagger) = 1/\sigma_{\max}(X)$, where we used the fact that $\overline{Y}$ is orthonormal. Thus,

$$\|\overline{F}^{\dagger}\|_2 = \frac{1}{\sigma_{\min}(\overline{F})} \leq \sigma_{\max}(X). \tag{3.24}$$

Assume that $\widetilde{F}$ and $\overline{F}$ are of full column rank, then we have from (Sun 1984), (3.23) and (3.24) that

$$\begin{aligned}\sin\angle(\mathscr{F},\widetilde{\mathscr{F}}) &= \|P_{\widetilde{F}} - P_{\overline{F}}\|_2 \\ &\leq \min\{\|\widetilde{F}^{\dagger}\|_2, \|\overline{F}^{\dagger}\|_2\}\|\widetilde{F} - \overline{F}\|_2 \\ &\leq \|\overline{F}^{\dagger}\|_2\|\widetilde{F} - \overline{F}\|_2 \\ &\leq \sqrt{k-1}\kappa_2(X)\cdot tol,\end{aligned}$$

where $P_{\widetilde{F}}$ and $P_{\overline{F}}$ is the orthogonal projector onto span$\{\widetilde{F}\}$ and span$\{\overline{F}\}$, respectively.

□

Remark 3.6 As the exact solution $\overline{F} = (X^T)^{\dagger}\overline{Y}$ is not in $\mathscr{N}(X^T)$ and $\widetilde{F}$ is an approximation to $\overline{F}$, the assumption that the columns of $\widetilde{F} - \overline{F}$ are not in $\mathscr{N}(X^T)$ is trivial. Although the upper bound given in (3.22) can be pessimistic and even much larger than one as k is not small and the matrix X is ill-conditioned, Theorem 3.5 reveals that the distance between the computed solution and exact solution is closely related to the product $\kappa_2(X)\cdot tol$, where $\kappa_2(X)$ is the 2-norm condition number of X and *tol* is a user-described convergence threshold used in Algorithm 3. In other words, the distance will be close to zero as X is not too ill-conditioned and tol is small.

Spectral regression discriminant analysis (SRDA) (Cai et al. 2008) is a popular method for large and sparse discriminant analysis. This method combines spectral graph analysis and regression to provide an efficient approach for discriminant analysis. The main task of SRDA is to solve a set of regularized least squares problems, and there is no eigenvector computation involved. However, SRDA aims to solve the *ratio-trace* problem (1.2) rather than the *trace-ratio* problem (1.1).

Let $\widehat{X} = [X^T, \mathbf{1}_n]^T$, it was shown that the solution of SRDA satisfies (Cai et al. 2008, pp.7)

$$\widehat{F} = \widehat{X}(\widehat{X}^T\widehat{X} + \alpha I_n)^{-1}\overline{Y}, \quad \alpha \to 0, \tag{3.25}$$

where α is a regularized parameter. The following theorem establishes the relationship between (3.5) and (3.25), i.e., the solution of (2.5) and that of SRDA. It indicates that our method is different from that of SRDA in essence.

Theorem 3.6 *Let $\widehat{F}$ be defined in (3.25), and denote by $\overline{F} = (X^T)^{\dagger}\overline{Y}$ the exact solution of (2.5). If X has full column rank, we have*

$$\widehat{F} = \begin{pmatrix} I_d - \frac{1}{\beta}\mathbf{g}\mathbf{g}^T \\ \frac{1}{\beta}\mathbf{g}^T \end{pmatrix}\overline{F}, \quad \alpha \to 0,$$

where $\beta = 1 + \mathbf{1}_n^T(X^TX)^{-1}\mathbf{1}_n$, and $\mathbf{g} = (X^T)^{\dagger}\mathbf{1}_n$.

Proof We notice that

$$\widehat{X}^T\widehat{X} = \begin{pmatrix} X^T & \mathbf{1}_n \end{pmatrix}\begin{pmatrix} X \\ \mathbf{1}_n^T \end{pmatrix} = X^TX + \mathbf{1}_n\mathbf{1}_n^T$$

is a rank-1 modification to the matrix X^TX. Let $\gamma = 1 + \mathbf{1}_n^T(X^TX + \alpha I_n)^{-1}\mathbf{1}_n$ and note that $X^\dagger X = I$. From the Shermam-Morrison-Woodbury formula (Golub and Van Loan 2013), we obtain

$$\begin{aligned}
\widehat{F} &= \widehat{X}(\widehat{X}^T\widehat{X} + \alpha I_n)^{-1}\overline{Y} \\
&= \begin{pmatrix} X \\ \mathbf{1}_n^T \end{pmatrix}(X^TX + \mathbf{1}_n\mathbf{1}_n^T + \alpha I_n)^{-1}\overline{Y} \\
&= \begin{pmatrix} X \\ \mathbf{1}_n^T \end{pmatrix}\left[(X^TX + \alpha I_n)^{-1} - \frac{1}{\gamma}(X^TX + \alpha I_n)^{-1}\mathbf{1}_n\mathbf{1}_n^T(X^TX + \alpha I_n)^{-1}\right]\overline{Y} \\
&= \begin{pmatrix} X(X^TX + \alpha I_n)^{-1}\overline{Y} - \frac{1}{\gamma}X(X^TX + \alpha I_n)^{-1}\mathbf{1}_n\mathbf{1}_n^T(X^TX + \alpha I_n)^{-1}\overline{Y} \\ \mathbf{1}_n^T(X^TX + \alpha I_n)^{-1}\overline{Y} - \frac{1}{\gamma}\mathbf{1}_n^T(X^TX + \alpha I_n)^{-1}\mathbf{1}_n\mathbf{1}_n^T(X^TX + \alpha I_n)^{-1}\overline{Y} \end{pmatrix} \\
&= \begin{pmatrix} X(X^TX + \alpha I_n)^{-1}\overline{Y} - \frac{1}{\gamma}X(X^TX + \alpha I_n)^{-1}\cdot\mathbf{1}_n\mathbf{1}_n^TX^\dagger\cdot\left(X(X^TX + \alpha I_n)^{-1}\overline{Y}\right) \\ \left(1 - \frac{1}{\gamma}\mathbf{1}_n^T(X^TX + \alpha I_n)^{-1}\mathbf{1}_n\right)\cdot\mathbf{1}_n^TX^\dagger\cdot\left(X(X^TX + \alpha I_n)^{-1}\overline{Y}\right) \end{pmatrix}.
\end{aligned}$$

As $\alpha \to 0$, we see that $X(X^TX + \alpha I_n)^{-1} \to X(X^TX)^{-1} = (X^T)^\dagger$, $X(X^TX + \alpha I_n)^{-1}\overline{Y} \to \overline{F}$ and $\gamma \to 1 + \mathbf{1}_n^T(X^TX)^{-1}\mathbf{1}_n = \beta$. In conclusion,

$$\widehat{F} \to \begin{pmatrix} \overline{F} - \frac{1}{\beta}(X^T)^\dagger\mathbf{1}_n\mathbf{1}_n^TX^\dagger\overline{F} \\ \frac{1}{\beta}\mathbf{1}_n^TX^\dagger\overline{F} \end{pmatrix} = \begin{pmatrix} I_d - \frac{1}{\beta}(X^T)^\dagger\mathbf{1}_n\mathbf{1}_n^TX^\dagger \\ \frac{1}{\beta}\mathbf{1}_n^TX^\dagger \end{pmatrix}\overline{F}, \quad \alpha \to 0.$$

So we complete the proof. □

In Zhang et al. (2017), an incremental regularized least squares (LDADL) method was proposed by Zhang *et al.*. With the help of Theorem 3.6 and (Zhang et al. 2017, Lemma 4.3), we can also establish the relationship between our proposed method and LDADL. We point out that Algorithm 3 is different from SRDA and LDADL in essence. First, the three methods Algorithm 3, LDADL and SRDA are designed for different problems. More precisely, Algorithm 3 is for the *trace-ratio problem* (1.1) while SRDA and LDADL are for the *ratio-trace problem* (1.2). Second, the left-hand sides of the least-squares problems involved in LDADL and SRDA are same, while the right-hand sides are different. Indeed, the right-hand sides involved in the former have one more column than the latter, refer to Zhang et al. (2017). As a comparison, the right-hand sides involved in our method and SRDA are the same, while the left-hand sides are different. More precisely, the left-hand sides involved in SRDA have one more row than the proposed method. Third, Algorithm 3 is parameter-free while both SRDA and LDADL involve regularization parameters. It is well-known that the optimal parameter is difficult to choose in practice if there is no other information available in advance (Gui et al. 2014), so our method is preferable.

Table 2 Details of the databases: dimensionality (d), the number of total samples (N), the background and data type (sparse or dense)

Database	Dimensionality (d)	Number of total samples (N)	Background	Type
Color FERET	24576	3528	Face images	dense
AR	19800	2600	Face images	dense
Extended YaleB	10000	2432	Face images	dense
CAS-PEAL	172800	21840	Face images	dense
YouTube	102400	124819	Video data	dense
TDT2	36771	9394	Audio data	sparse
Reuters	18933	8213	Text document	sparse

4 Numerical experiments

In this section, we perform some numerical experiments on some real-world high-dimension and large-sample data dimensionality reduction problems. In Sect. 4.1, we describe the databases and some benchmark algorithms used in the experiments. In Sect. 4.2, we compare our proposed algorithms with some state-of-the-art algorithms to show the merits of the new algorithms.

4.1 Datasets, benchmark algorithms and experiment settings

Seven real-world databases, including video data, text documents, face images, audio data, are used in our experiment. The details of these data bases, such as dimensionality, the number of total samples, the background and data type are summarized in Table 2.

- The `Color FERET` dataset[1] was collected in 15 sessions between August 1993 and July 1996. The database contains 1564 sets of images for a total of 14126 images that includes 1199 individuals and 365 duplicate sets of images ranging from frontal to left and right profiles. In our experiment, a part of the FERET program containing $N = 3528$ images of $k = 269$ different people was utilized. We crop and scale the images to 192×128 pixels, and set the reducing dimension $s = 200$.
- The `AR` database[2] contains over 4000 color images corresponding to 126 people's faces (70 men and 56 women). Images feature frontal view these faces with different facial expressions, illumination conditions, and occlusions (e.g., sun glasses and scarf). The pictures were taken at the CVC under strictly controlled conditions. No restrictions on wear (clothes, glasses, etc.), make-up, hair style were imposed to participants. Each person participated in two sessions, separated by two weeks time. The same pictures were taken in both sessions. A subset of $k = 100$ with 26 images of per people, i.e., $N = 2600$ images are used in our experiment. We crop and scale the images to 120×165 pixels, and the reducing dimension s is set to be 99.

[1] https://www.face-rec.org/databases/.
[2] http://www2.ece.ohio-state.edu/~~aleix/ARdatabase.html.

- The `Extended YaleB`[3] data set contains 5760 single light source images of 10 subjects, each seen under 576 viewing conditions (9 different poses and 64 illumination conditions of each person). The images have normal, sleepy, sad and surprising expressions. A subset of $k = 38$ persons with 64 images of per people, i.e., $N = 2432$ images are used in the example. We crop and scale the images to 100×100 pixels in our experiment, and set the reduced dimension $s = k - 1 = 37$.
- The `CAS-PEAL` face database (Gao et al. 2008) was constructed under the sponsors of National Hi-Tech Program and ISVISION, by the Face Recognition Group of JDL, ICT, and CAS. It contains 99594 images of 1040 individuals (595 males and 445 females) with varying pose, expression, accessory, and lighting (PEAL). For each subject, 9 cameras spaced equally in a horizontal semicircular shelf are setup to simultaneously capture images across different poses in one shot. Each subject is also asked to look up and down to capture 18 images in another two shots. We also considered 5 kinds of expressions, 6 kinds accessories (3 glasses, and 3 caps), and 15 lighting directions. This face database is now partly made available. A subset named by CAS-PEAL-R1, contains $N = 21840$ images of 1040 subjects, and each subject across 21 different poses without any other variations are included for research purpose. We crop and scale the images to 360×480 pixels, and set the reduced dimension $s = 400$.
- The `Youtube` dataset (Wolf et al. 2011) is a large-scale video classification database. It contains 80 million YouTube video links, which are labeled as 4800 knowledge graph entities. Here we provide a links of 5020 videos with 1595 persons labels. All images have been downloaded from YouTube. In our experiments we have employed up to 90 samples of each class, resulting to a dataset of $N = 124819$ feature samples data and $k = 1595$ classes.
- The `Reuters` dataset[4] contains 21578 documents in 135 categories. The documents with multiple category labels are discarded. It left us with $N = 8213$ documents in $k = 65$ categories. After preprocessing, this corpus contains $d = 18933$ distinct terms.
- The original `TDT2` (Nist Topic Detection and Tracking) corpus[5] collects during the first half of 1998 and taken from 6 sources, including 2 newswires (APW, NYT), 2 radio programs (VOA, PRI) and 2 television programs (CNN, ABC). It consists of 11201 on-topic documents which are classified into 96 semantic categories. In this subset, those documents appearing in two or more categories were removed, and only the largest $k = 30$ categories were kept, thus leaving us with $N = 9394$ documents in total.

As the PCA+Newton-Lanczos method (PCA+NL) (Ngo et al. 2012) is a popular algorithm for the trace-ratio problem, and SRDA is a state-of-the-art algorithm for large-scale discriminant analysis (Cai et al. 2008), we take them and the proposed Algorithm 2 as the benchmark algorithms in all the experiments. The details of the PCA+Newton Lanczos method and SRDA are listed as follows:

- `PCA+NL` (Ngo et al. 2012): The PCA+Newton Lanczos method, in which the PCA stage is performed before running the Newton Lanczos method. Here we preserve 99% energy in the PCA stage. We find that infinite trace-ratio values may occur in this algo-

[3] http://cvc.yale.edu/projects/yalefacesB/yalefacesB.html.
[4] http://www.cad.zju.edu.cn/home/dengcai/.
[5] http://www.cad.zju.edu.cn/home/dengcai/.

rithm during outer iterations. In order to deal with this problem, let $V^{(j)}$, $V^{(j+1)}$ be the orthnormalized eigenvectors obtained from the j-th and the $(j+1)$-th iteration, respectively, we use the sine of the angle between the subspaces span$\{V^{(j)}\}$ and span$\{V^{(j+1)}\}$ as the stopping criterion in this algorithm. As was suggested in Ngo et al. (2012), we use the MATLAB built-in function `eigs.m` for the eigenvalue problems involved, with the stopping criterion $tol = 10^{-4}$.

- `SRDA` (Cai et al. 2008): The spectral regression discriminant analysis method based on solving regularized least squares problem. This method is one of the state-of-the art algorithms for large-scale and *sparse* discriminant analysis problems, and it is required to solve *s dense* least squares problems of size $(d+1)$-by-n. The `lsrq` package provided by Cai *et al.* is used to compute the regularized least squares problem involved, and the algorithm is stopped as soon as the residual norm is below $tol = 10^{-6}$ or the number of iterations reaches 20. The the regularized parameter is chosen as $\alpha = 10^{-2}$.
- In Algorithm 2, we set $q = 2$ and the over-sampling parameter $p = 20$. The target rank r is chosen in the following way: we set $r = 100$ if the number of training sample n is less than 1000, and set $r = 200$ if the number of training sample is in the interval (1000, 3500]; otherwise, we set $r = 400$.

All the experiments are run on a Hp workstation with 16 cores double Intel(R) Xeon(R) Platinum 8253 processors, and with CPU 2.20 GHz and RAM 256 GB. The operation system is 64-bit Windows 10. All the numerical results are obtained from running the MATLAB R2018b software.

In the examples, we randomly pick some, say, $n = 60\%N, 70\%N$ and $80\%N$ samples as the training set, and the remaining samples are used as the testing set, where N is the total number of samples. More precisely, we rearrange the original N samples data by using the MATLAB command `randperm(N)`, which returns a row vector containing a random permutation of the integers from 1 to N, then we select the samples corresponding to the first n elements of the row vector as the training set. We make use of the nearest neighbor classifier (NN) (Cover and Hart 1967) for classification in the experiments. Each experiment will be repeated 10 for times, and all the numerical results, i.e., the CPU time in seconds, the recognition rate and the standard deviation (Std-Dev), are the mean from the 10 runs.

4.2 Numerical experiments

In this subsection, we perform some numerical experiments to show the superiority of our randomized method Algorithm 2 and the iterative method Algorithm 3 over some state-of-the-art algorithms for high-dimension and large-sample data dimensionality reduction problems. As Algorithm 3 is proposed for solving high-dimension and large-sample *sparse* trace-ratio problem, we only run it on large *sparse* data sets.

Example 1 In this example, we compare Algorithm 2 with some state-of-the-art algorithms for large-scale trace-ratio problem. The test set is the *color FERET* database. Besides the three benchmark algorithms Algorithm 2, PCA+NL and SRDA, we also run the following four state-of-the-art trace-ratio algorithms for this problem:

Table 3 Example 1: numerical results on the *Color FERET* database, $d = 24576$, $N = 3528$, $s = 200$

Algorithms	CPU Time / s (Recognition rate ± Std-Dev%)		
(Methods)	$n = 60\%N$	$n = 70\%N$	$n = 80\%N$
PCA+NL	28.6 (23.20±0.90%)	43.9 (22.79±2.06%)	66.4 (22.75±1.60%)
SRDA	91.5 (27.05±1.08%)	105.5 (29.60±1.23%)	119.8 (32.68±1.59%)
Algorithm 2	1.17 (47.00±1.50%)	1.25 (49.87±0.92%)	1.37 (53.44±1.17%)
PCA+DNM	5.40 (25.44±0.84%)	6.80 (24.49±1.12%)	8.21 (23.78±1.31%)
FastITR	100.5 (25.74±2.37%)	171.1 (25.73±2.69%)	271.9 (26.30±2.87%)
PCA+iITR	6.54 (23.01±0.94%)	8.66 (22.34±1.20%)	11.7 (22.27±1.50%)
WangITR	–	–	–

Here 99% energy is preserved in the PCA-preprocessed methods, and "–" stands that the algorithm fails to converge with in 1000 seconds

- `PCA+DNM` (Jia et al. 2009): The trace-ratio linear discriminant method presented by Jia *et al*, whose MATLAB code is available from http://www.escience.cn/people/fpnie/papers.html. Here we preserve 99% energy in the PCA stage.
- `PCA+iITR` (Zhao et al. 2013): A trace-ratio linear discriminant method proposed by Zhao *et al*., in which we preserve 99% energy in the PCA stage.
- `FastITR` (Zhang et al. 2010): An algorithm advocated by Zhang *et al*.[6] for the trace-ratio problem. The MATLAB built-in function `eigs.m` is used for computing the eigenvalue problems involved. In this method, a regularized parameter is required to ensure the positive definiteness of the total scatter matrix. We set the regularized parameter to be 10^{-2} in the experiment.
- `WangITR` (Wang et al. 2007): An algorithm proposed by Wang *et al*. for the trace-ratio problem. As this algorithm solves the trace difference problem by using the eigenvalue decomposition method in each outer iteration (Wang et al. 2007, Algorithm 1), we exploit the MATLAB build-in function `eig.m` for the eigenvalue problems involved. The numerical results are list in Table 3.

We see from Table 3 that Algorithm 2 outperforms all the compared algorithms both in terms of CPU time and recognition accuracy. More precisely, it not only runs faster than the two benchmark algorithms PCA+NL and SRDA, but also beats all the algorithms for solving the trace-ratio problem. For example, Algorithm 2 is about 90 times faster than SRDA and is about 30 faster than PCA+NL. Moreover, one finds that WangITR fails to converge within 1000 seconds for this problem. This is because we have to solve trace difference problem by using eigenvalue decomposition during each outer iteration, whose cost is prohibitively large for large-scale problems.

On the other hand, it is observed that the recognition rates of Algorithm 2 are much higher than those of the other algorithms. In fact, the approximation $\widetilde{X}$ got from the randomized algorithm is a low-rank approximation to the training sample data X, and this process has the effect of denoising. Although the PCA processing can also filter noise in

[6] We thank Prof. Leihong Zhang for providing us with the MATLAB codes of this algorithm.

Table 4 Example 2: Numerical results on the *AR* database, $d = 19800$, $N = 2600$, $s = 99$

Algorithms	CPU Time/s (Recognition rate ± Std-Dev%)		
(Methods)	n = 60%N	n = 70%N	n = 80%N
PCA+NL	6.15 (93.72±1.51%)	7.08 (95.03±1.93%)	7.98 (94.46±2.56%)
SRDA	31.9 (97.86±0.34%)	36.9 (98.44±0.35%)	42.0 (98.48±0.48%)
Algorithm 2	0.63 (95.77±0.44%)	0.69 (96.70±0.30%)	0.74 (97.15±0.59%)
NLDAS	4.99 (95.82±0.55%)	6.48 (96.41±0.72%)	8.06 (96.88±0.49%)
NLDAfast	3.99 (94.54±1.51%)	5.67 (95.49±1.35%)	9.10 (96.10±1.27%)
NLDA	31.8 (95.87±0.57%)	35.7 (96.35±0.72%)	40.8 (96.98±0.54%)

In the PCA+NL algorithm, we preserve both 99% energy in the PCA stage

some sense, however, preserving 99% energy in this stage may not denoise effectively for this problem.

Example 2 As was discussed in Sect. 2, Algorithm 2 is closely related to the null space method. In this example, we compare Algorithm 2 with some null space methods including `NLDAfast` (Lu and Wang 2012), `NLDAS` (Huang et al. 2002) and `NLDA` (Chen et al. 2000). The test set is the *AR* database. Table 4 lists the numerical results.

Again, we observe from Table 4 that Algorithm 2 performs much better than the other algorithms. For example, it is about eight times faster than NLDAS, NLDAfast and PCA+NL, and is about 50 times faster than NLDA and SRDA. On the other hand, the recognition rates and the standard derivations obtained from Algorithm 2 and the null space algorithms are about the same. This because all of them seek solutions in the subspace $\mathcal{M}(S_W) \setminus \mathcal{M}(S_T)$, see Sect. 3.1, and our proposed algorithm runs much faster than the null space methods. Moreover, we notice that the recognition rates of the three algorithms are a little lower than those of SRDA, and are a little higher than those of PCA+NL. Indeed, which one is the best according to recognition rate is often problem-dependent.

Example 3 In this example, we compare Algorithm 2 with two randomized algorithms including `FastLDA` and `RFDA/RP` for large-scale discriminant analysis. The test set is the *Extended YaleB* database.

- `FastLDA` (Gado et al. 2016): A fast LDA algorithm that uses a feature extraction method based on random projection to reduce the dimensionality, and then performs LDA in the reduced space. In essence, this algorithm can be understood as a RSVD+LDA method. In this algorithm, the regularized parameter is set to be 10^{-2}, and the size of Gaussian matrix for sampling is chosen as $d \times 160$, where d is the dimension of the data.
- `RFDA/RP` (Ye et al. 2017): A fast LDA algorithm based on random projection. In this algorithm, the random projection matrix is determined by Theorem 3 of Ye et al. (2017), and the number of columns is chosen as $\frac{rank(XL_T)-\log(1/\alpha)}{\epsilon^{-2}}$, where $L_T = I_n - \mathbf{1}_n\mathbf{1}_n^T/n$, $\alpha = 0.1$ is the regularized parameter, and $\epsilon = 10^{-2}$ is related to the desired failure probability. Table 5 lists the numerical results of the five algorithms.

Table 5 Example 3: Numerical results on the *Extended YaleB* database, $d = 10000$, $N = 2432$, $s = 37$

Algorithms	CPU Time/s (Recognition rate ± Std-Dev%)		
(Methods)	n=60%N	n=70%N	n=80%N
PCA+NL	2.22 (98.82±0.26%)	2.58 (99.05±0.38%)	2.98 (99.32±0.47%)
SRDA	5.82 (99.17±0.33%)	6.79 (99.30±0.28%)	6.60 (99.59±0.33%)
Algorithm 2	0.32 (97.06±0.38%)	0.35 (97.41±0.47%)	0.38 (97.95±0.47%)
RFDA/RP	1.23 (92.77±3.89%)	1.48 (93.25±3.05%)	1.74 (96.69±3.41%)
FastLDA	0.71 (96.45±0.47%)	0.85 (97.08±0.49%)	0.93 (97.60±0.93%)

In the PCA+NL algorithm, we preserve 99% energy in the PCA stage

We see from Table 5 that the three randomized algorithms FastLDA, RFDA/RP, and Algorithm 2 run much faster than PCA+NL and SRDA, while our proposed randomized algorithm is about two times faster than FastLDA and five times faster than RFDA/RP. On the other hand, for this problem, the recognition rates of the three randomized algorithms are (a little) lower than those of SRDA and PCA+NL, while those of our proposed algorithm are the highest among the three randomized algorithms. Specifically, we find that the recognition rates of RFDA/RP are (much) lower and the standard deviations are (much) higher than those of the others, especially when n is relatively small. In fact, the success of RFDA/RP greatly relies on the random projection matrix used. However, the projection matrix requires two parameters, one is to avoid the singularity problem, and the other is to determine the success probability of a low-rank approximation. How to choose the optimal values of two parameters is a difficult problem.

Example 4 In this example, we show the superiority of Algorithm 2 over many state-of-the-art algorithms for high-dimension and large-sample *dense* data dimensionality reduction problem. The test sets are two high-dimension and large-sample dense databases *CAS-PEAL* and *YouTube*. As the number of total samples are very large in the YouTube database, we randomly pick some $n = 40\%N, 50\%N$ and $60\%N$ samples as the training set, and the remaining samples are used as the testing set, where N is the total number of samples. We compare Algorithm 2 with the above trace-ratio algorithms, two popular large-scale discriminant analysis methods `SRDA` (Cai et al. 2008) and `LDADL` (Zhang et al. 2017), two randomized algorithms `FastLDA` (Gado et al. 2016) and `RFDA/RP` (Ye et al. 2017), as well as three null space methods `NLDAfast` (Lu and Wang 2012), `NLDAS` (Huang et al. 2002) and `NLDA` (Chen et al. 2000).

In the experiments, we set the size of the Gaussian matrix for sampling in FastLDA to be $d \times 400$. For the PCA plus algorithms, we preserve both 99% and 95% energy on the PCA stage, respectively. In LDADL, we exploit the `lsrq` package provided by Cai et al. (2008) to solve the regularized least squares problem, and the algorithm is stopped as soon as the residual norm is below $tol = 10^{-6}$ or the number of iterations reaches 20. The numerical results are listed in Tables 6 and 7.

We observe from Table 6 that for the CAS-PEAL database, most of the algorithms fail to converge within 1000 seconds or suffer from heavy storage requirement. Some trace-ratio algorithms such as PCA+DNM and PCA+iITR do not work at all if we preserve

Table 6 Example 4: Numerical results on the *CAS-PEAL* database, $d = 172800$, $N = 21840$, $s = 400$

Algorithms	CPU Time / s (Recognition rate ± Std-Dev%)		
(Methods)	n=60%N	n=70%N	n=80%N
Algorithm 2	61.39 (53.83±0.79%)	67.58 (57.55±0.63%)	83.14 (58.85±0.82%)
SRDA	–	–	–
LDADL	–	–	–
PCA+NL(99%)	–	–	–
PCA+NL(95%)	–	–	–
PCA+DNM(99%)	513.3 (4.10±0.23%)	667.1 (3.81±0.23%)	893.9 (3.69±0.32%)
PCA+DNM(95%)	417.3 (52.20±0.65%)	540.8 (52.77±0.67%)	728.1 (53.78±0.67%)
PCA+iITR(99%)	981.0 (3.57±0.20%)	–	–
PCA+iITR(95%)	415.9 (52.23±0.67%)	541.0 (52.95±0.61%)	721.8 (53.95±0.69%)
FastITR	–	–	–
WangITR	O.M.	O.M.	O.M.
NLDAS	–	–	–
NLDA	–	–	–
NLDAfast	–	–	–
FastLDA	125.2 (64.05±0.63%)	185.1 (65.58±0.52%)	212.0 (67.00±0.77%)
PFDA/RP	564.4 (31.06±0.72%)	756.3 (32.54±0.53%)	990.8 (34.80±0.57%)

Here "O.M." implies that the algorithm suffers from "out of memory", and "–" denotes the CPU time exceeds 1000 seconds. In the PCA plus algorithms, we preserve both 99% and 95% energy in the PCA stage, respectively

99% energy in the PCA process, as the recognition rates are only about 4%. However, if we preserve 95% energy in the PCA process, the recognition rates raise to about 50%. The reason is that the PCA plus methods may be unstable for dimensionality reduction (Shi et al. XXX). Furthermore, Algorithm 2 is about 10 times faster than PCA+DNM and PCA+iITR.

It is seen that the recognition rates of FastLDA are the highest for the CAS-PEAL database. For this problem, the recognition rates of Algorithm 2 are about 20% higher than RFDA/RP, but are about 10% lower than those of FastLDA. On the other hand, Algorithm 2 is about 2-3 times faster than FastLDA, and is about ten times faster than RFDA/RP.

Table 7 demonstrates the advantage of randomized algorithms for high-dimension and large-sample dense data sets. More precisely, except for Algorithm 2 and FastLDA, all the algorithms fail to converge with in 1000 seconds or suffer from the difficulty of out of memory. Compared with FastLDA, Algorithm 2 is about 3 times faster than FastLDA, however, the recognition rates of our proposed algorithm are about 6% lower than those of FastLDA. This is similar to the numerical results given in Table 6 for the CAS-PEAL database. Therefore, for high-dimension and large-sample dense data sets, Algorithm 2 is a good choice if speed is more important, and it is a competitive candidate for high-dimension and large-sample *dense* data dimensionality reduction problem.

Example 5 In this example, we show the efficiency of Algorithm 2 and Algorithm 3 for high-dimension and large-sample *sparse* data dimensionality reduction problem. To this

Table 7 Example 4: Numerical results on the *YouTube* database, $d = 102400$, $N = 124819$, $s = 400$

Algorithms	CPU Time / s (Recognition rate ± Std-Dev%)		
(Methods)	n=30%N	n=40%N	n=50%N
Algorithm 2	107.6 (92.52±0.08%)	185.4 (93.02±0.21%)	270.9 (93.30±0.12%)
SRDA	–	–	O.M.
LDADL	–	–	O.M.
PCA+NL(99%)	–	–	O.M.
PCA+NL(95%)	–	–	–
PCA+DNM(99%)	–	O.M.	O.M.
PCA+DNM(95%)	–	–	–
PCA+iITR(99%)	–	–	O.M.
PCA+iITR(95%)	–	–	–
FastITR	O.M.	O.M.	O.M.
WangITR	O.M.	O.M.	O.M.
NLDAS	O.M.	O.M.	O.M.
NLDA	O.M.	O.M.	O.M.
NLDAfast	O.M.	O.M.	O.M.
FastLDA	395.1 (98.30±0.05%)	589.1 (98.36±0.06%)	893.5 (98.46±0.07%)
PFDA/RP	805.8 (91.32±0.30%)	–	–

Here "O.M." implies that the algorithm suffers from "out of memory", and "–" denotes the CPU time exceeds 1000 seconds. In the PCA plus algorithms, we preserve both 99% and 95% energy in the PCA stage, respectively

aim, we compare Algorithm 2 and Algorithm 3 with all the algorithms run before. The test sets are two high-dimension and large-sample sparse databases *TDT2* and *Reuters*. In the experiments, we set the size of the Gaussian matrix for sampling in FastLDA to be $d \times 400$. For the PCA plus algorithms, we preserve both 99% energy on the PCA stage. In LDADL, we exploit the `lsrq` package provided by Cai et al. (2008) to solve the regularized least squares problem, and the algorithm is stopped as soon as the residual norm is below $tol = 10^{-6}$ or the number of iterations reaches 20. The numerical results are listed in Tables 8 and 9.

Some comments are given. First, as both the row and the column of the training set are very large, the conventional trace-ratio algorithms PCA+NL, PCA+DNM, PCA+iITR, FastITR and WangITR are very slow or even fail to converge for these two data sets. It is obvious to see that Algorithm 2 and Algorithm 3 run much faster than these trace-ratio algorithms. These illustrate the superiority of Algorithm 2 and Algorithm 3 for high-dimension and large-sample trace-ratio problems.

Second, Algorithm 2 and Algorithm 3 runs much faster than the three null space methods and the two randomized algorithms, moreover, the recognition rates of the two proposed algorithms are much higher than the five algorithms. Third, the numerical performance of the two proposed algorithms is comparable to SRDA and LDADL for large sparse data sets. Although Algorithm 2 runs a little slower than Algorithm 3, SRDA and LDADL, the CPU time and recognition rates of Algorithm 2 and Algorithm 3 are comparable to those of SRDA and LDADL. The reason is that the sparse structure of the data

Table 8 Example 5: Numerical results on the *TDT2* database, $d = 36771$, $N = 9394$, $s = 29$

Algorithms	CPU Time / s (Recognition Rate ± Std-Dev%)		
(Methods)	n=60%N	n=70%N	n=80%N
Algorithm 2	2.826 (96.89±0.26%)	3.093 (96.88±0.38%)	3.344 (97.10±0.26%)
Algorithm 3	1.236 (96.69±0.26%)	1.327 (96.84±0.35%)	1.458 (97.06±0.52%)
SRDA	1.259 (96.73±0.25%)	1.351 (96.87±0.38%)	1.486 (97.06±0.50%)
LDADL	1.286 (96.67±0.27%)	1.380 (96.80±0.33%)	1.529 (97.04±0.53%)
PCA+NL	–	–	–
PCA+DNM	–	–	–
FastITR	–	–	–
PCA+iITR	–	–	–
WangITR	O.M.	O.M.	O.M.
NLDAS	128.7 (89.38±6.79%)	192.9 (86.90±7.80%)	276.5 (80.44±10.2%)
NLDA	230.8 (88.79±6.84%)	273.2 (86.24±7.73%)	323.1 (79.62±10.4%)
NLDAfast	97.99 (80.55±5.05%)	129.6 (80.25±5.42%)	167.5 (75.38±6.28%)
FastLDA	5.610 (95.86±0.34%)	6.349 (95.81±0.54%)	6.623 (95.94±0.40%)
PFDA/RP	10.29 (93.44±1.08%)	13.72 (92.86±1.31%)	16.18 (92.26±1.05%)

Here "O.M." implies that the algorithm suffers from "out of memory", and '–' stands for the CPU time exceeds 1000 seconds. In the PCA plus algorithms, we preserve 99% energy in the PCA stage

Table 9 Example 5: Numerical results on the *Reuters* database, $d = 18933$, $N = 8213$, $s = 64$

Algorithms	CPU Time / s (Recognition Rate ± Std-Dev%)		
(Methods)	n=60%N	n=70%N	n=80%N
Algorithm 2	1.254 (91.53±0.58%)	1.356 (91.69±0.50%)	1.489 (91.74±0.55%)
Algorithm 3	1.145 (92.12±0.46%)	1.257(92.49±0.59%)	1.335 (93.09±0.65%)
LDADL	1.176 (92.22±0.45%)	1.288 (92.65±0.56%)	1.361 (93.11±0.54%)
SRDA	1.160 (92.28±0.53%)	1.263 (92.62±0.61%)	1.345 (93.24±0.64%)
PCA+NL	957.6 (92.30±0.45%)	–	–
PCA+DNM	870.4 (92.15±0.44%)	–	–
PCA+iITR	–	–	–
FastITR	–	–	–
WangITR	O.M.	O.M.	O.M.
NLDAS	66.80 (69.92±13.9%)	103.3 (63.26±13.7%)	151.0 (56.05±11.6%)
NLDA	65.23 (69.07±14.7%)	80.26 (65.62±12.6%)	103.2 (59.66±11.1%)
NLDAfast	39.38 (63.92±11.4%)	51.32 (60.10±10.7%)	65.28 (54.12±9.85%)
FastLDA	2.653 (89.02±0.59%)	2.967 (88.91±0.53%)	3.346 (89.07±0.74%)
PFDA/RP	7.471 (87.40±4.16%)	8.501 (88.16±2.27%)	9.712 (88.57±1.75%)

Here "O.M." implies that the algorithm suffers from "out of memory", and '–' stands for the CPU time exceeds 1000 seconds. In the PCA plus algorithms, we preserve 99% energy in the PCA stage

samples is not preserved in randomized algorithms. As a result, Algorithm 2 is more suitable to high-dimension and large-sample *dense* data, while Algorithm 3 is a competitive candidate for high-dimension and large-sample *sparse* data.

5 Concluding remarks

The trace-ratio problem is crucial in high dimensionality reduction and machine learning. However, it has been long believed that this problem does not have a known explicit solution in general. A goal of this paper is to provide alternative and ideally faster algorithms for this problem. We show that the trace-ratio problem does admit a close-form solution when the dimension of the data points d is greater than or equal to the number of training samples n.

To the best of our knowledge, most of the algorithms for trace-ratio problem are based on inner-outer iterations, which are complicated and even may be prohibitive for high-dimension and large-sample data sets. Therefore, efficient algorithms for high-dimension and large-sample trace-ratio problem are still lacking, especially for dense data sets.

In this work, we pay special attention to high-dimension and large-sample trace-ratio problem, and propose two *non-inner-outer* iteration methods to solve it. With the help of the close-form solution and randomized singular decomposition, we first propose a randomized algorithm (i.e., Algorithm 2) for *dense* data sets. Theoretical results are given to show how to choose the target rank in randomized SVD, and why an inexact solution still works for recognition.

For high-dimension and large-dense sample *sparse* trace-ratio problem, we propose an iterative algorithm (i.e., Algorithm 3) based on the close-form solution. Similar to SRDA and LDADL, it does not destroy the sparse structure of the original data matrix. An advantage of the new algorithm over SRDA and LDADL is that it is parameter-free, and one only needs to solve some (consistent) under-determined linear systems rather than regularized least-squares problems. The difference and the theoretical relation between SRDA and our new method is given, and the distance between the computed solution and the exact solution is established.

Numerical experiments illustrate the numerical behavior of our proposed algorithms, and show the effectiveness of the theoretical results. They show that Algorithm 2 is superior to many state-of-the-art algorithms for dimensionality reduction on high-dimension and large-sample *dense* data sets, while Algorithm 3 is more suitable to high-dimension and large-sample *sparse* data sets. Specifically, Algorithm 2 is the best one among all the compared algorithms in overall consideration, especially when both d and n are very large and even in the same order.

Acknowledgements We would like to express our sincere thanks to the anonymous referees and our editor for insightful comments and suggestions that greatly improved the representation of this paper.

References

Alzubi, A., & Abuarqoub, A. (2020). Deep learning model with low-dimensional random projection for large-scale image search. *Engineering Science and Technology, an International Journal, 24,* 911–920.

Andras, P. (2018). High-dimensional function approximation with neural networks for large volumes of data. *IEEE Transactioins on Neural Networks and Learning Systems, 29,* 500–508.

Belhumeur, P., Hespanha, J., & kriegman, D. . (1997). Eigenfaces vs fisherface: Recognition using class-specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 19,* 711–720.

Cai, D., He, X., & Han, J. (2008). SRDA: An efficient algorithm for large-scale discriminant analysis. *IEEE Transactions on Knowledge and Data Engineering, 20,* 1–12.

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

Chen, W., Xu, Y., Yu, Z., Cao, W., Chen, C. L. P., & Han, G. (2020). Hybrid dimensionality reduction forest with pruning for high-dimensional data classification. *IEEE Access, 8,* 40138–40150.

Chu, D., & Thye, G. (2010). A new and fast implementation for null space based linear discriminant analysis. *Pattern Recognition, 43,* 1373–1379.

Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory, 13,* 21–27.

Eldén, L. (2005). *Matrix Methods in Data Mining and Pattern Recognition*. Philadelphia, PA: SIAM.

Fukunaga, K. (1991). *Introduction to Statistical Pattern Recognition* (2nd ed.). San Diego, CA: Academic Press.

Gado, N., Maes, E., Kharouf, M. (2016) Linear discriminant analysis for large-scale data: application on text and image data, *The 15th IEEE International Conference on Machine Learning and Applications*, pp. 961–964.

Gao, W., Cao, B., Shan, S., Chen, X., Zhou, D., Zhang, X., Zhao, D. (2008). The CAS-PEAL Large-Scale Chinese Face Database and Baseline Evaluations, *IEEE Transactions on System Man, and Cybernetics (Part A)*, pp. 149–161.

Golub, G. H., & Van Loan, C. F. (2013). *Matrix Computations* (4th ed.). Baltimore: The Johns Hopkins University Press.

Gu, M. (2015). Subspace iteration randomization and singular value problems. *SIAM Journal on Scientific Computing, 37,* A1139–A1173.

Gui, J., Sun, Z., Cheng, J., Ji, S., & Wu, X. (2014). How to estimate the regularization parameter for spectral regression discriminant analysis and its kernel version? *IEEE Transactions on Circuits and Systems for Video Technology, 24,* 211–223.

Guo, Y., Li, S., Yang, J., Shu, T., & Wu, L. (2003). A generalized Foley-Sammon transform based on generalized Fisher discriminant criterion and its application to face recognition. *Pattern Recognition Letter, 24,* 1447–158.

Halko, N., Martinsson, P., & Tropp, J. (2011). Finding structure with randomness: probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review, 53,* 217–288.

Hastie, T., Tibshirani, R., & Friedman, J. (2001). *The elements of statistical learning: Data mining, inference, and prediction*. New York: Springer.

Huang, R., Liu, Q., Lu, H., Ma, S. (2002). Solving the small sample size problem of LDA, the 16th International Conference on Pattern Recognition, pp. 29–32.

Jia, Y., Nie, F., & Zhang, C. (2009). Trace-ratio problem revisited. *IEEE Transactions on Neural Networks, 20,* 729–735.

Jiang, C., Xie, H., Bai, Z. (2017). Robust and efficient computation of eigenvectors in a generalized spectral method for constrained clustering, Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, 54: 757–766.

Kokiopoulou, E., Chen, J., & Saad, Y. (2010). Trace optimization and eigenproblems in dimension reduction methods. *Numerical Linear Algebra with Applications, 18,* 565–602.

Kramer, R., Young, A., & Burton, A. (2018). Understanding face familiarity. *Cognition, 172,* 46–58.

Liu, R., Ren, R., Liu, J., & Liu, J. (2020). A clustering and dimensionality reduction base devolutionary algorithm for large-scale multi-objective problems. *Applied Soft Computing Journal, 89,* 106120.

Lu, G., & Wang, Y. (2012). Feature extraction using a fast null space based linear discriminant analysis algorithm. *Information Sciences, 193,* 72–80.

Martinsson, P., Rokhlin, V., & Tygert, M. (2011). A randomized algorithm for the decomposition of matrices. *Applied and Computational Harmonic Analysis, 30,* 47–68.

Musco, C., Musco, C. (2015). Randomized block Krylov methods for stronger and faster approximate singular value decomposition, *In Advances in Neural Information Processing Systems*, pp. 1396–1404.

Ngo, T., Bellalij, M., & Saad, Y. (2012). The trace-ratio optimization problem. *SIAM Review, 54,* 545–569.

Nie, F., Xiang, S., Jia, Y., Zhang, C., & Yan, S. (2008). Trace-ratio criterion for feature selection, National Conference on. *Artificial Intelligence, 2*, 671–676.

Paige, C. C., & Saunders, M. A. (1982). LSQR: An algorithm for sparse linear equations and sparse least squares. *ACM Transactions on Mathematical Software, 8,* 43–71.

Park, C., & Park, H. (2008). A comparison of generalized linear discriminant analysis algorithms. *Pattern Recognition, 41,* 1083–1097.

Shi, W., Luo, Y., & Wu, G. (2020). On general matrix exponential discriminant analysis methods for high dimensionality reduction. *Calcolo, 57,* 1–34.

Sun, J. (1984). Stability of orthogonal projection. *Journal of University of Chinese Academy of Sciences, 1,* 123–133. ((**in Chinese**)).

Tavernier, J., Simm, J., Meerbergen, K., Kurt Wegner, J., Ceulemans, H., Moreau, Y. (2017). Fast semi-supervised discriminant analysis for binary classification of large data-sets, arXiv: 1709.04794v1.

Vishwakarma, D., & Singh, T. (2019). A visual cognizance based multi-resolution descriptor for human action recognition using key pose. *International Journal of Electronics and Communications, 107,* 157–169.

Wang, H., Yan, S., Xu, D., Huang, X. (2007). Trace-ratio vs. ratio-trace for dimensionality reduction, *IEEE Conference on Compute Vision and Pattern Recognition*, pp. 1–8.

Wedin, P. (1973). Perturbation theory for pseudoinverses, *BIT Numerical Mathematics*, pp. 217–232.

Wolf, L., Hassner, T., Maoz, I. (2011). Face recognition in unconstrained videos with matched background similarity, *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 529–534.

Woodruff, D. (2014). Sketching as a tool for numerical linear algebra. *Foundations and Trends in Theoretical Computer Science, 10,* 1–157.

Wu, G., & Feng, T. (2015). A theoretical contribution to the fast implementation of null linear discriminant analysis with random matrix multiplication. *Numerical Linear Algebra with Applications, 22,* 1180–1188.

Wu, G., Feng, T., Zhang, L., & Yang, M. (2017). Inexact implementation using Krylov subspace methods for large scale exponential discriminant analysis with applications to high dimensionality reduction problems. *Pattern Recognition, 66,* 328–341.

Ye, H., Li, Y., Chen, C., & Zhang, Z. (2017). Fast Fisher discriminant analysis with randomized algorithms. *Pattern Recognition, 72,* 82–92.

Zhang, L., Liao, L., & NG, M. .K. . (2010). Fast algorithms for the generalized Foley-Sammon discriminant analysis. *SIAM Journal on Matrix Analysis and Applications, 31,* 1584–1605.

Zhang, X., Chen, L., Chu, D., Liao, L., Ng, M., & Tan, R. (2017). Incremental regularized least squares for dimensionality reduction of large-scale data. *SIAM Journal on Scientific Computing, 38,* B414–B439.

Zhao, M., Chan, R., Tang, P., Chow, T., & Wong, S. (2013). Trace-ratio linear discriminant analysis for medical diagnosis: A case study of dementia. *IEEE Singal Processing Letters, 20,* 431–434.

Zhao, M., Zhang, Z., Chow, T., & Wu, Z. (2012). *On the theoretical and computational analysis between trace ratio LDA and null-space LDA, the 24th IEEE International Joint Conference on Neural Networks* (pp. 1–7). Australia: At Brisbane.

Zhu, L., & Huang, D. (2014). A Rayleigh-Ritz style method for large-scale discriminant analysis. *Pattern Recognition, 47,* 1698–1708.

Authors and Affiliations

Wenya Shi[1] · Gang Wu[1]

Wenya Shi
shiwenyaer@163.com

[1] School of Mathematics, China University of Mining and Technology, Xuzhou, Jiangsu 221116, People's Republic of China