CSIAM Trans. Appl. Math.
doi: 10.4208/csiam-am.2021.nla.03

Vol. **2**, No. 2, pp. 297-312
June 2021

A Convergence Analysis on the Iterative Trace Ratio Algorithm and its Refinements

Zhongxiao Jia* and Fuhui Lai

Department of Mathematical Sciences, Tsinghua University, Beijing 100084, P.R. China.

Received 20 May 2020; Accepted 28 September 2020

Abstract. In many applications one needs to process massive data of high dimensionality. In order to make data compact and reduce computational complexity, dimensionality reduction algorithms are commonly used. Linear discriminant analysis (LDA) is one of the most used approaches, which leads to a matrix trace ratio problem, i.e., maximization of the trace ratio $\rho(V)=\mathrm{Tr}(V^TAV)/\mathrm{Tr}(V^TBV)$, where A and B are $n\times n$ real symmetric matrices with B positive definite, and V is an $n\times p$ column orthonormal matrix. In this paper, we consider a commonly used Iterative Trace Ratio (ITR) algorithm developed by Ngo et al., *The trace ratio optimization problem, SIAM Review, 54 (3) (2012), pp. 545–569*. In implementations, it is common to use the symmetric Lanczos method to compute the p eigenvectors of a certain large matrix corresponding to its p largest eigenvalues at each iteration, and the resulting algorithm is abbreviated as ITR_L. We establish the global convergence and local quadratic convergence of the trace ratio itself. We then make two improvements over ITR_L: (i) using the refined Lanczos method to compute the desired eigenvectors at each iteration and (ii) providing a better initial guess via solving a generalized eigenvalue problem of the matrix pair (A,B). The resulting algorithms are abbreviated as ITR_RL and ITR_GeigRL, respectively. Numerical experiments demonstrate that ITR_RL and ITR_GeigRL outperform ITR_L substantially.

AMS subject classifications: 65F15, 65F30, 62H30, 15A18

Key words: Dimensionality reduction, trace ratio, global convergence, eigenvalue, eigenvector, refined projection, refined Lanczos algorithm, implicitly restarted, preprocess.

1 Introduction

In many applications such as data mining, machine learning and bioinformatics, one has to process massive data. Due to excessive storage requirement and computational cost,

*Corresponding author. *Email addresses:* jiazx@tsinghua.edu.cn (Z. Jia), laifh14@163.com (F. Lai)

in order to overcome the difficulty of high dimensionality, make an effective analysis and process the data, it is necessary to use dimensionality reduction, whose aim is to reduce the high dimensionality and meanwhile to retain the useful information of data features and structures. Some dimensionality reduction approaches are from statistics and geometry, e.g., the linear discriminant analysis (LDA) [5] and the local preserving projection (LPP) [7], etc. They ultimately lead to the matrix trace ratio problem: Find the column orthonormal solution V_* to the optimization problem

$$\max_{V^TV=I_p} \frac{\mathrm{Tr}(V^TAV)}{\mathrm{Tr}(V^TBV)}, \tag{1.1}$$

where A and B are $n\times n$ real symmetric, V is $n\times p$ column orthonormal, I_p is the identity of order p with $1<p\ll n$ being the classification number of data features, and 'Tr$(\cdot)$' is the trace of a square matrix. For convenience, we suppose that B is symmetric positive definite, and define $\rho(V)=\frac{\mathrm{Tr}(V^TAV)}{\mathrm{Tr}(V^TBV)}$ to be the objective function. Therefore, the trace ratio optimization problem is to find V_* and the maximum trace ratio $\rho_*=\rho(V_*)$.

The Foley-Sammon transform (FST) method [4] is based on the Fisher discriminant criterion [3]. Exploiting this criterion, Sammon proposes an optimal set of discriminant surfaces [16], which is extended to the optimal set of discriminant vectors [4]. One can use FST method to obtain optimal sets of discriminant vectors successively. The method has received much attention in the field of pattern recognitions; see, e.g., [8]. There have been some FST based methods under different conditions, for example, Liu's method [14]. Since the discriminant vectors are successively obtained and only locally optimal, there is no guarantee that the discriminant vectors obtained ultimately are globally optimal. This is a typical shortcoming that many local optimization methods share.

There have been some methods for solving (1.1). Yan and Tang [20] propose a multiscale search algorithm, which, at each iteration, solves a symmetric matrix eigenvalue problem, but no convergence analysis is made. It appears that this method converges slowly, similar to the bisection methods proposed by Xiang et al. [19] and Guo et al. [6]. Wang et al. [18] improve these methods and propose a Newton like method for solving problem (1.1), in which they give up multiscale and bisection searches and instead solve a symmetric matrix eigenvalue problem at each iteration so as to accelerate the convergence. Ngo et al. [15] develop an Iterative Trace Ratio (ITR) method. In implementations, they suggest to use the symmetric Lanczos method at iteration k to find the p eigenvectors of $A-\rho_kB$ associated with the p largest eigenvalues where ρ_k is the current approximation to ρ_*, and we shall call the resulting ITR algorithm ITR_L. Specifically, the implicitly restarted Lanczos algorithm is used.

The convergence of ITR has been analyzed in, e.g., [1, 22, 23], where the global linear convergence of $\{\rho_k\}$ and the local quadratic convergence of $\{V_k\}$ has been proved with the columns of V_k being the unit-length eigenvectors of $A-\rho_{k-1}B$ corresponding to its p largest eigenvalues counting their multiplicities. Throughout the paper, the p largest eigenvalues are always meant to count their multiplicities. The convergence of

$\{V_k\}$ means that of the subspace spanned by the columns of V_k, and it requires that the $(p+1)$th largest eigenvalue of $A-\rho_* B$ is not equal to its pth largest one. Zhang et al. [23, Theorem 3.1] prove the local superlinear convergence of $\{\rho_k\}$. Particularly, Cai et al. [1, Theorem 5.5] prove that the ITR algorithm is exactly the Self-Consistent-Field (SCF) iteration for (1.1) when the same starting V_0 is used; see also [21]. This means that ITR and the SCF iteration for the trace ratio optimization problem produce the same results and have the same numerical performances for the same starting V_0 when the same algorithm is used to compute $\{V_k\}$. In implementations of the SCF iteration, it is also suggested to to use the implicitly restarted Lanczos algorithm to compute the p eigenvectors of $A-\rho_k B$ corresponding to its p largest eigenvalues.

In this paper, we present a new convergence result and prove a global convergence and local quadratic convergence of $\{\rho_k\}$, which improve the afore-mentioned convergence results on $\{\rho_k\}$ in [22, 23]. We then make two improvements on ITR_L, which introduce the refined Lanczos algorithm [10, 11] for computing eigenvectors more accurately and faster than the standard Lanczos algorithm and (ii) provide a better initial guess V_0, which consists of the orthonormalized p eigenvectors of (A,B) corresponding to its largest eigenvalues. Based on these two improvements, we develop two Refined ITR algorithms, called ITR_RL and ITR_GeigRL. The numerical experiments illustrate that ITR_RL and ITR_GeigRL are substantially more efficient than the ITR_L algorithm and ITR_GeigRL is the best.

The paper is organized as follows. In Section 2, we briefly review how the trace ratio problem comes and the ITR algorithm for solving it. In Section 3, we make a convergence analysis on ITR. In Section 4, we improve ITR_L in two aspects and develop the ITR_RL and ITR_GeigRL algorithms. In Section 5, we report numerical experiments to illustrate the much higher efficiency of ITR_RL than ITR_L. Finally, we conclude the paper in Section 6.

2 The trace ratio problem and the ITR algorithm

Based on the linear discriminant criterion, we require that the within-class metric, which measures the degree of aggregations of similar point sets after projection, is as small as possible and the between-class metric, which measures the degree of dispersions of different point sets is as large as possible. Let x_i, $i=1,\cdots,m$ be the points in $\mathbb{R}^n$, $X=\{x_1,x_2,\cdots,x_m\}$, and $C^{(k)}$ the index set of that all the elements in X belong to classification k, $k=1,\cdots,c$, that is, if x_i belongs to classification k, then $i\in C^{(k)}$. Define $n_k=|C^{(k)}|$ to be the cardinality of $C^{(k)}$, $\mu^{(k)}$ and μ to be the centers of the set whose indices belong to $C^{(k)}$ and the set X, respectively.

In terms of the above, we introduce the following two important matrices:

$$S_W=\sum_{k=1}^{c}\sum_{i\in C^{(k)}}(x_i-\mu^{(k)})(x_i-\mu^{(k)})^T \tag{2.1}$$

and

$$S_B=\sum_{k=1}^{c} n_k(\mu^{(k)}-\mu)(\mu^{(k)}-\mu)^T, \tag{2.2}$$

which are called the within-class and between-class discrete metric matrices, respectively. Write the transform matrix $V=[v_1,\cdots,v_p]\in\mathbb{R}^{n\times p}$. Then x_i is transformed into $y_i=V^Tx_i$. The within-class discrete metric of projection point set can be expressed as

$$\begin{aligned}\phi_W&=\sum_{k=1}^{c}\sum_{i\in C(k)}\left\|y_i-\tilde{\mu}^{(k)}\right\|_2^2\\&=\sum_{k=1}^{c}\sum_{k\in C^{(k)}}\left\|V^T(x_i-\mu^{(k)})\right\|_2^2\\&=\sum_{k=1}^{c}\sum_{i\in C^{(k)}}\mathrm{Tr}\left[V^T(x_i-\mu^{(k)})(x_i-\mu^{(k)})^TV\right]\\&=\mathrm{Tr}\left[V^TS_WV\right],\end{aligned}$$

where $\|\cdot\|_2$ is the 2-norm of a vector or matrix. Analogously, we obtain the expression of between-class discrete metric of projection point set:

$$\phi_B=\mathrm{Tr}\left[V^TS_BV\right], \tag{2.3}$$

where S_B is the matrix defined by (2.2). Clearly, S_B and S_W are real symmetric and in fact semi-positive definite.

Based on the linear discriminant criterion, the optimal dimensionality reduction is to find a column orthonormal V that maximizes the trace ratio

$$\frac{\mathrm{Tr}[V^TS_BV]}{\mathrm{Tr}[V^TS_WV]}.$$

Let $A=S_B$ and $B=S_W$. Then the above becomes the maximizing matrix trace ratio problem:

$$\max_{V^TV=I_p}\frac{\mathrm{Tr}(V^TAV)}{\mathrm{Tr}(V^TBV)}, \tag{2.4}$$

where $V\in\mathbb{R}^{n\times p}$, and A and B are symmetric. Furthermore, assume that B is positive definite, and write $\rho(V)=\frac{\mathrm{Tr}(V^TAV)}{\mathrm{Tr}(V^TBV)}$. Then such problem is called the trace ratio optimization problem of the matrices A and B.

The necessary and sufficient conditions for the optimal solution of the trace ratio problem are established in [22], which state as follows.

Theorem 2.1. *Assume that A is symmetric and B is symmetric positive definite. Then V_* is the optimal solution of the trace ratio problem if and only if the columns of V_* are the unit length eigenvectors of $A-\rho_*B$ corresponding to the p largest eigenvalues, where $\rho_*=\rho(V_*)$.*

Construct the function

$$f(\rho)=\max_{V^TV=I_p}\mathrm{Tr}(V^T(A-\rho B)V). \tag{2.5}$$

Then Theorem 2.1 shows that if $V=V^*$ then $f(\rho^*)=0$. In fact, the following result has been proved in [15, Lemma 4.1].

Theorem 2.2. *ρ_* is the maximum of the trace ratio in* (1.1) *if and only if $f(\rho_*)=0$.*

As a result, computing ρ_* is equivalent to finding the root of $f(\rho)=0$. This is a nonlinear root-finding problem and can be solved by the Newton method. To this end, we consider the derivative of $f(\rho)$. It is proved in [15, Proposition 4.4] that

$$\frac{df(\rho)}{d\rho}=-\mathrm{Tr}(V(\rho)^TBV(\rho)). \tag{2.6}$$

Since B is supposed to be symmetric positive definite, $\frac{df(\rho)}{d\rho}<0$ unconditionally, meaning that $f(\rho)$ monotonically decreases strictly with respect to ρ. In terms of the basic iterative scheme of the Newton method, one can produce iterates $\{\rho_k\}$ by

$$\begin{aligned}\rho_{k+1}&=\rho_k-\frac{\mathrm{Tr}(V(\rho_k)^T(A-\rho_kB)V(\rho_k))}{-\mathrm{Tr}(V(\rho_k)^TBV(\rho_k))}\\&=\frac{\mathrm{Tr}(V(\rho_k)^TAV(\rho_k))}{\mathrm{Tr}(V(\rho_k)^TBV(\rho_k))},\end{aligned} \tag{2.7}$$

which is the ITR algorithm for solving the trace ratio problem [15] and is summarized as Algorithm 1, where, at iteration k, the Lanczos algorithm is used to compute $V(\rho_k)$, whose columns are the p eigenvectors of the matrix $A-\rho_kB$ corresponding to its p largest eigenvalues

Cai et al. [1] have proved that Algorithm 1 is identical to the SCF iteration for the trace ratio optimization problem. Therefore, ITR and the SCF iteration produce the same $\{\rho_k\}$ and $\{V_k\}$ for the same starting V_0 and have the same numerical performances when the same algorithm is applied to the eigenvalue problem of $A-\rho_kB$ at iteration k. Clearly, a free option in the SCF iteration and the ITR algorithm is the choice of an iterative algorithm for computing the p eigenvectors of $A-\rho_kB$ corresponding to its p largest eigenvalues. When A and B are small to medium sized, one can apply the QR algorithm [2,17] to $A-\rho_kB$ directly, but if they are large scale, the QR algorithm is impractical due to the excessive storage requirement and computational cost. In this case, the symmetric Lanczos method or, specifically, the implicitly restarted Lanczos algorithm, e.g., the MATLAB function eigs, has been commonly used for this large eigenvalue problem in ITR and the SCF iteration [15,22]. In particular, we abbreviate the resulting ITR algorithm as ITR_L.

Algorithm 1 Iterative Trace Ratio (ITR) algorithm

Input: the matrices A and B, the dimension p, and the stopping tolerance tol.
Output: V_* and ρ_*.

1. Choose a column orthonormal matrix $V_0 \in \mathbb{R}^{n\times p}$ as an initial guess, compute $\rho_0 = \frac{\mathrm{Tr}(V_0^T A V_0)}{\mathrm{Tr}(V_0^T A V_0)}$.
2. For $k=0,1,\cdots$
   - Compute the eigenvectors $v_1,\cdots,v_p$ of $A-\rho_k B$ corresponding to its p largest eigenvalues, and let $V_{k+1}=[v_1,\cdots,v_p]$.
   - Compute $\rho_{k+1}=\frac{\mathrm{Tr}(V_{k+1}^T A V_{k+1})}{\mathrm{Tr}(V_{k+1}^T B V_{k+1})}$. If $|\rho_{k+1}-\rho_k|/|\rho_k| \le tol$, then stop and quit.
   - Return $V_*=V_{k+1}$ and $\rho_*=\rho_{k+1}$.

3 The convergence of ITR

We first analyze the convergence of the sequence $\{\rho_k\}$ obtained by ITR. We can establish the following global convergence result.

Theorem 3.1. *Let $\{\rho_k\}$ be the iterates computed by Algorithm 1. Then $\{\rho_k\}$ globally converges to ρ_* and locally converges quadratically.*

Proof. Note that the sequence $\{\rho_k\}$ is generated by the Newton method and ρ_* is the largest trace ratio. From (2.7) we have

$$\rho_{k+1}=\rho_k+\frac{\mathrm{Tr}(V^T(\rho_k)(A-\rho_k B)V(\rho_k))}{\mathrm{Tr}(V(\rho_k)^T B V(\rho_k))}. \tag{3.1}$$

It follows from Theorem 2.2 and the monotonic decreasing property of $f(\rho)$ that if $\rho<\rho_*$ then $f(\rho)>0$ and if $\rho>\rho_*$ then $f(\rho)<0$. Since $\rho_k \le \rho_*$, we obtain

$$f(\rho_k)=\mathrm{Tr}(V^T(\rho_k)(A-\rho_k B)V(\rho_k)) \ge 0. \tag{3.2}$$

Since $\mathrm{Tr}(V(\rho_k)^T B V(\rho_k))>0$, combining it with (3.1) and (3.2) proves that the sequence $\{\rho_k\}$ is monotonically increasing and is bounded by ρ_*. Therefore, ρ_k converges to the limit point $\tilde{\rho}\le\rho_*$ with $\tilde{\rho}$ the supremum of $\{\rho_k\}$. Then we must have

$$\tilde{\rho}=\rho_*.$$

Suppose that $\tilde{\rho}<\rho_*$. Then $f(\tilde{\rho})=\mathrm{Tr}(V(\tilde{\rho})^T(A-\tilde{\rho}B)V(\tilde{\rho}))>0$. Let Λ be the sum of eigenvalues of B. Then from the positive definiteness of B we have

$$\mathrm{Tr}(V(\rho_k)^T B V(\rho_k)) \le \Lambda. \tag{3.3}$$

Since $\rho_k \leq \tilde{\rho}$ for any k and $\rho_k \to \tilde{\rho}$, for a given constant $\varepsilon > 0$ it holds $\tilde{\rho} - \rho_k < \varepsilon$, i.e., $\rho_k + \varepsilon > \tilde{\rho}$, for k large enough. Specifically, take

$$\varepsilon = \frac{\mathrm{Tr}(V(\tilde{\rho})^T(A-\tilde{\rho}B)V(\tilde{\rho}))}{\Lambda} \leq \varepsilon_k = \frac{\mathrm{Tr}(V(\tilde{\rho})^T(A-\tilde{\rho}B)V(\tilde{\rho}))}{\mathrm{Tr}(V(\rho_k)^T B V(\rho_k))}$$

with the inequality holding from (3.3). As a result, by the monotonic decreasing property of $f(\rho) = \mathrm{Tr}(V^T(\rho)(A-\rho B)V(\rho))$ for $\rho < \rho_*$, since $\rho_k \leq \tilde{\rho} < \rho_*$, we obtain

$$\begin{aligned}\rho_{k+1} &= \rho_k + \frac{\mathrm{Tr}(V(\rho_k)^T(A-\rho_k B)V(\rho_k))}{\mathrm{Tr}(V(\rho_k)^T B V(\rho_k))} \\ &\geq \rho_k + \varepsilon_k \geq \rho_k + \varepsilon > \tilde{\rho},\end{aligned}$$

which contradicts the fact that $\tilde{\rho}$ is the supremum of the sequence $\{\rho_k\}$. Therefore, we must have $\rho_* = \tilde{\rho}$, and the sequence $\{\rho_k\}$ globally converges to ρ_*.

Finally, it is known from the positive definiteness of B and (2.6) that $f'(\rho_*) \neq 0$. Therefore, the asymptotic quadratic convergence of the Newton method means that $\{\rho_k\}$ converges quadratically. □

Zhang et al. [22, Theorem 5.1] have proved the global and linear convergence of $\{\rho_k\}$, and the superlinear convergence of $\{\rho_k\}$ has been established in [23, Theorem 3.1]. Here we have proved the global convergence and quadratic convergence of $\{\rho_k\}$, improving the existing results.

Zhang et al. [22, Theorem 5.4] have established the quadratic convergence of $\{V_k\}$, where V_k is allowed to be computed inexactly with the error satisfying a certain tolerance. Here the convergence means that of the subspace spanned by the columns of V_k. We remind that the convergence results on $\{V_k\}$ are established under the assumption that the $(p+1)$th largest eigenvalue of $A-\rho_* B$ is not equal to its pth largest one, while the convergence of $\{\rho_k\}$ does not need this assumption. Cai et al. [1, Theorem 5.5] prove that ITR is equivalent to the SCF iteration for the trace ratio problem, so that all the results on the SCF iteration are applicable to ITR.

4 The ITR algorithm using the symmetric refined Lanczos method

The key and most expensive part in ITR is the computation of the eigenvectors of the symmetric $A-\rho B$ corresponding to its p largest eigenvalues. For large matrices, as we have mentioned, ITR and the SCF iteration commonly use the symmetric Lanczos method to do this. In implementations, the algorithm used is the implicitly restarted Lanczos algorithm, which results in the ITR_L algorithm. However, the approximate eigenvectors, called Ritz vectors, obtained by the Lanczos method may be irregularly poor and even may fail to converge to the desired eigenvectors when the subspace is sufficiently good

and the approximate eigenvalues, Ritz values, converge [9, 13, 17]. In contrast, the refined Lanczos method, which is the refined Arnoldi method in the nonsymmetric case, can fix this deficiency perfectly [10, 11, 17], and it computes new approximate eigenvectors, called the refined approximate eigenvectors. Our first improvement on ITR_L is to naturally use the refined Lanczos algorithm to replace the standard Lanczos algorithm in ITR, and the resulting algorithm is called ITR_RL. Below we introduce the definition of refined approximate eigenvectors for a general projection subspace [10, 17].

Definition 4.1. *Given a projection subspace $\mathcal{K}$, let μ be an approximation to the desired eigenvalue λ of a matrix A, a refined projection method seeks the unit length vector $\hat{x}\in\mathcal{K}$ that satisfies the optimal requirement*

$$\|(A-\mu I)\hat{x}\|_2=\min_{u\in\mathcal{K},\|u\|_2=1}\|(A-\mu I)u\|_2 \tag{4.1}$$

and uses it to approximate the corresponding eigenvector x of A. The new $\hat{x}$ is called a refined approximate eigenvector, and the resulting method is also called the refined Rayleigh-Ritz method.

We comment that μ can be any available approximation to λ, e.g., the Ritz value, the harmonic Ritz value or the Rayleigh quotient of A with respect to the harmonic Ritz vector. The following results have been established in [12].

Theorem 4.1. *If $\mu\to\lambda$ and $\sin\angle(x,\mathcal{K})\to 0$, then $\hat{x}$ converges to x. Let $(\mu,\tilde{x})$ be a Ritz pair. Then provided that $\|(A-\mu I)\tilde{x}\|\neq 0$, the strict inequality holds:*

$$\|(A-\mu I)\hat{x}\|_2<\|(A-\mu I)\tilde{x}\|_2. \tag{4.2}$$

If there is another Ritz value close to μ, then generally

$$\|(A-\mu I)\hat{x}\|_2\ll\|(A-\mu I)\tilde{x}\|_2, \tag{4.3}$$

that is, $\hat{x}$ can be much more accurate than $\tilde{x}$.

When $\mathcal{K}$ is a Krylov subspace and μ is a Ritz value, the refined Rayleigh–Ritz method is the refined Lanczos method and the refined Arnoldi method for A symmetric (Hermitian) and nonsymmetric (non-Hermitian). The refined approximate eigenvector $\hat{x}$ can be computed by a small sized singular value decomposition (SVD) reliably and efficiently [10]. In practical implementations of ITR, due to the limit of storage, we shall use the implicitly restarted refined Lanczos algorithm [11] to compute the p desired eigenvectors of $A-\rho_k B$. This is our first improvement over ITR where the implicitly restarted Lanczos algorithm is used, the resulting algorithm is called the refined ITR (ITR_RL) algorithm, which is described in Algorithm 2

Our second improvement is to provide a better initial guess on V^*. Under the assumption that the p eigenvectors of $A-\rho_k B$ corresponding to its p largest eigenvalues are given exactly, although ITR globally converge quadratically, a better initial guess can definitely

Algorithm 2 Iterative Trace Ratio using the refined Lanczos algorithm (ITR_RL)

Input: the matrices A and B, the dimension p, and the stopping tolerance tol.
Output: V_* and ρ_*.

1. Choose an arbitrary column orthonormal $V_0 \in \mathbb{R}^{n\times p}$ as an initial guess, and compute $\rho_0 = \frac{\mathrm{Tr}(V_0^T A V_0]}{\mathrm{Tr}(V_0^T A V_0)}$.
2. For $k = 0,1,\cdots$
   - Use the implicitly restarted refined Lanczos algorithm to compute the eigenvectors $v_1,\cdots,v_p$ of $A-\rho_k B$ corresponding to its p largest eigenvalues, orthogonalize them, and rename them as $v_1,\cdots,v_p$. Define $V_{k+1} = [v_1,\cdots,v_p]$.
   - Compute $\rho_{k+1} = \frac{\mathrm{Tr}(V_{k+1}^T A V_{k+1})}{\mathrm{Tr}(V_{k+1}^T B V_{k+1})}$. If $|\rho_{k+1}-\rho_k|/|\rho_k| \le tol$, then stop and quit.
   - Return $V_* = V_{k+1}$ and $\rho_* = \rho_{k+1}$.

reduce iterations of ITR and improve its overall efficiency. In ITR_L and ITR_RL, the initial V_0 is supposed to be arbitrary and is chosen randomly in practice, so that it may be the case that ITR_L and ITR_RL enter the quadratic convergence stage after quite many iterations. Therefore, it is very significant to preprocess ITR_L and ITR_RL and provide a better initial guess V_0 whenever possible.

As it will turn out, it may be beneficial to compute the p (generalized) eigenvectors of the matrix pair (A,B) itself corresponding to its p largest eigenvalues and use them to form an initial V_0, as we will argue later.

Notice that in the trace ratio problem we suppose that A is symmetric and B is symmetric positive definite. Therefore, (A,B) is a symmetric positive definite (S/PD) pair. For a small to medium sized pair, the most efficient method is the QZ algorithm or the Wilkinson algorithm [17], which first computes the Cholesky decomposition $B = R^T R$, lets $C = R^{-T} A R^{-1}$, computes the eigen-decomposition of $C = U\Lambda U^T$, and recovers $X = R^{-1}U$ such that $AX = BX\Lambda$. We then orthonormalize the eigenvectors associated with the p largest eigenvalues of (A,B) and use them to form V_0. For a large matrix pair, however, these two algorithms are computationally infeasible, and, instead, only iterative projection methods are computationally viable.

Introduce the B-inner product

$$(x,y)_B = y^T B x,$$

which induces to the B-norm

$$\|x\|_B = \sqrt{(x,x)_B} = \sqrt{x^T B x}.$$

With the new inner product and the induced norm, the Lanczos algorithm and refined Lanczos problems emerge correspondingly, and the implicitly restarted Lanczos and re-

fined Lanczos algorithms are adapted naturally. They are used to compute the p eigenvectors of (A,B) corrresponding to the largest eigenvalues. We orthonormalize them to form the initial V_0. It is expected that such V_0 has some accuracy as an approximation to V_* and can be (much) better than a random V_0. The following result provides some theoretical support on our expectation and gets some insight into such a choice.

Theorem 4.2. *The eigenvector v_* of (A,B) corresponding to the largest eigenvalue belongs to the subspace spanned by the columns of the optimal solution V_*. If the largest eigenvalue is multiple, there is at least one eigenvector in this subspace.*

Proof. Write $V_*=[v_{1*},\cdots,v_{p*}]$.Then

$$\rho_*=\frac{\sum_{i=1}^p v_{i*}^TAv_{i*}}{\sum_{i=1}^p v_{i*}^TBv_{i*}}.$$

Note that there is always a vector $v_{k_1*}\in\{v_{1*},\cdots,v_{p*}\}$ with $k_1,\cdots,k_p$ being a permutation of $1,\cdots,p$ such that

$$\frac{v_{k_1*}^TAv_{k_1*}}{v_{k_1*}^TBv_{k_1*}}\le\frac{\sum_{i=2}^p v_{k_i*}^TAv_{k_{i*}}}{\sum_{i=2}^p v_{k_i*}^TBv_{k_i*}}.$$

Let

$$\begin{aligned}
&c_1=v_{k_1*}^TAv_{k_1*},\quad c_2=v_{k_1*}^TBv_{k_1*},\\
&a_1=v_*^TAv_*,\quad a_2=v_*^TBv_*,\\
&b_1=\sum_{i=2}^p v_{k_i*}^TAv_{k_i*},\quad b_2=\sum_{i=2}^p v_{k_i*}^TBv_{k_{i*}}.
\end{aligned}$$

Since $a_1/a_2\ge c_1/c_2$, $a_1/a_2\ge b_1/b_2$, $b_1/b_2\ge c_1/c_2$, we have

$$\frac{a_1+b_1}{a_2+b_2}\ge\frac{c_1+b_1}{c_2+b_2},$$

that is, $\hat{V}=[v_*,v_{k_2},\cdots,v_{k_p}]$ is such that

$$\frac{\hat{V}^TA\hat{V}}{\hat{V}^TB\hat{V}}\ge\frac{V_*^TAV_*}{V_*^TBV_*}=\rho_*$$

with the equality holding if and only if $v_*=v_{k_1*}$. Otherwise, the strict inequaliy holds, which would mean that V_* is not the optimal solution of the trace ratio problem, a contradiction. □

Theorem 4.2 indicates that the eigenvector v_* of (A,B) corresponding to its largest eigenvalue belongs to the subspace spanned by the columns of V_*. Therefore, we must benefit from the new initial V_0 as at least one of its columns favors ITR and ITR_RL, so that iterations may be reduced. Furthermore, although we are unable to make a theoretical analysis, numerical experiments demonstrate that such new initial V_0 reduces iterations substantially. We abbreviate ITR_RL with this new V_0 as ITR_GeigRL, which is summarized as Algorithm 3.

Algorithm 3 ITR_GeigRL

Input: the matrices A and B, the dimension p, and the stopping tolerance tol.
Output: **Output:** V_* and ρ_*.

1. Compute the eigenvectors $\hat{v}_1,\cdots,\hat{v}_p$ of (A,B) corresponding to its p largest eigenvalues, orthonormalize them to form the initial V_0, and compute $\rho_0=\frac{\mathrm{Tr}(V_0^TAV_0)}{\mathrm{Tr}(V_0^TAV_0)}$.
2. For $k=0,1,\cdots$
   - Use the implicitly restarted refined Lanczos algorithm to compute the eigenvectors $v_1,\cdots,v_p$ of $A-\rho_kB$ associated with its p largest eigenvalues, orthogonalize them, and rename them as $v_1,\cdots,v_p$. Define $V_{k+1}=[v_1,\cdots,v_p]$.
   - Compute $\rho_{k+1}=\frac{\mathrm{Tr}(V_{k+1}^TAV_{k+1})}{\mathrm{Tr}(V_{k+1}^TBV_{k+1})}$. If $|\rho_{k+1}-\rho_k|/|\rho_k|\leq tol$, then stop and quit.
   - Return $V_*=V_{k+1}$ and $\rho_*=\rho_{k+1}$.

5 Numerical experiments

In this section we report numerical experiments to illustrate the convergence behavior and the much higher efficiency of ITR_RL than ITR_L and that of ITR_GeigRL than ITR_RL. All the computations were preformed using Matlab R2015a 64-bit on Intel(R) celeron(R) CPU B815 1.60GHz processor and 4 GB RAM with the machine precision $\epsilon_{\mathrm{mach}}=2.22\times10^{-16}$ under the Microsoft Windows 8 64-bit system.

The seven test problems are from http://face-rec.org/databases/, where orl_face is from ORL Database and face94_female, face94_malestaff, face94_male, face95, face96 and grimace are from face recognition data. Their sizes see the tables.

We test Algorithm 1 and Algorithm 2 and demonstrate advantages of using the implicitly restarted refined Lanczos algorithm to compute desired eigenvectors. We use the MATLAB function eigs of the implicitly restarted Lanczos algorithm and the algorithm proposed in [11], abbreviated as reigs. We set Krylov subspace dimension to be $2p$ in eigs and reigs, the stopping tolerance to be 10^{-10}, and the maximum restarts to be 300. The stopping criteria for outer iterations of the three algorithms are $tol=10^{-6}$, and the maximum number of Newton (outer) iterations is 16. Table 1 lists the results obtained for p=15, where 'outIter' is the number of outer iterations, 'inIter' is the total iterations of eigs and reigs, and 'time' is the CPU time in seconds, and 'err' is the final value $|\rho_{k+1}-\rho_k|/|\rho_k|$ when the corresponding algorithm has converged or terminated when 16 outer iterations were consumed. For each problem, ITR_L and ITR_RL use the same initial V_0 generated randomly in a normal distribution.

Let us first compare ITR_RL with ITR_L. We can see from Table 1 that ITR_RL solved all the problems successfully but ITR_L failed to solve all but the last problem grimacy after maximum 16 outer iterations were used. Furthermore, we observe from Table 1

Table 1: A comparison of the algorithms on different datasets, where we take $p=15$, and orl, 94fema, 94stf, 94ma and grima represent orl_face, face94_female, face94_malestaff, face94_male,grimace, and GeigRL denotes ITR_GeigRL.

data name		orl	94fema	94stf	94ma	face95	face96	grima
size		4464	4900	4900	5600	5600	6084	4900
ITR_L	ρ_*	36.47	9.12	311.41	181.36	44.35	87.05	182.06
	outIter	16	16	16	16	16	16	10
	inIter	4406	4205	4378	4393	4495	4352	2499
	time(s)	706.9	761.3	832.9	1006.3	1071.4	1307.1	507.0
	err	1E-4	9E-6	3E-3	3E-3	1E-3	6E-4	5E-8
ITR_RL	ρ_*	36.48	9.12	312.68	182.55	44.53	87.10	182.06
	outIter	9	9	9	9	9	8	8
	inIter	1750	986	2166	2007	2004	1230	755
	time(s)	560.3	426.2	710.8	885.2	848.4	715.7	296.0
	err	1E-8	1E-11	2E-7	2E-10	3E-7	5E-8	8E-11
GeigRL	ρ_*	36.48	9.12	312.68	182.55	44.53	87.10	182.06
	outIter	5	4	4	4	5	4	4
	inIter	1471	647	1200	1200	1500	986	567
	time(s)	460.0	299.7	534.6	508.5	616.9	561.5	226.0
	err	4E-10	4E-8	3E-7	4E-7	1E-8	5E-9	7E-9

that the 'inIter' are nearly 300 times of outer iterations for eigs. This indicates that, at most of the outer iteration of ITR, eigs used almost 300 restarts, the maximum restarts allowed, and it generally did not drop below the prescribed tolerance 10^{-10}; otherwise the 'OutIter' would be (almost) the same as those used by ITR_RL. As a matter of fact, we have observed that ITR started diverging and ρ_k oscillated from the tenth outer iteration onwards because eigs did not converge and computed p eigenvectors of $A-\rho_k B$ with poor accuracy corresponding to the p largest eigenvalues. This is why we set the maximum of outer iterations to be 16 in the experiments since more outer iterations did not improve the convergence and accuracy of ρ_k in ITR_L.

In contrast, for ITR_RL, the inner iterations 'inIter' are considerably smaller than 300 times of outer iterations, which implies that at each outer iteration, reigs converged with the prescribed tolerance 10^{-10}. We can also find that the computed ρ^* by ITR and ITR_RL are obviously different for the first six problems since ITR_L failed to solve these problems, while the converged ρ^* computed by ITR and ITR_RL for grimacy are numerically the same.

Next we investigate the behavior of ITR_GeigRL and ITR_RL and show the superiority of the former. We see from Table 1 that ITR_RL and ITR_GeigRL computed the same ρ_* and solved the problems successfully. Remarkably, we observe that ITR_GeigRL was considerably superior to ITR_RL and it used roughly half of outer iterations 'outIter'

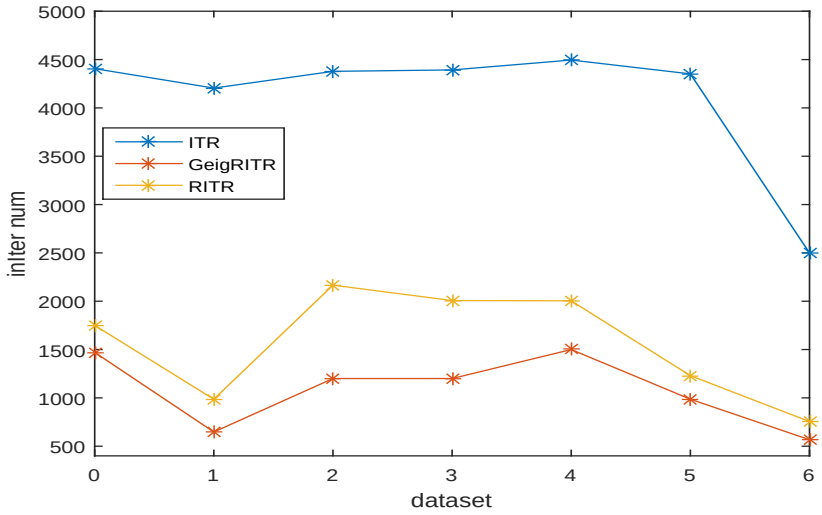

Figure 1: A comparison of inner iterations 'inIter' used by ITR_L, ITR_RL and ITR_GeigRL on different datasets, which are denoted by ITR, RITR and GeigRITR in the figure. The indices 0-6 correspond to the seven datasets face94_female, face94_malestaff, face94_male, face95, face96 and grimace.

of ITR_RL to achieve the prescribed stopping tolerance; in the meantime, we see that ITR_GeigRL consumed much less computational time than ITR_RL did for each problem.

Visually, we compare the computational efficiency by drawing inner iterations used by ITR_RL and ITR_GeigRL for all the problems; see Fig. 1, where the horizontal indices correspond to the seven test problems accordingly. Clearly, we can see from the figure that ITR_RL is substantially more efficient than ITR_L and, of the three algorithms, ITR_GeigRL is the best.

Next taking the same maximum outer iterations 16, we test the three algorithms on the problems face95, face96 by choosing $p=20$ and using variable maximum restarts 200, 300 and 400, respectively. Table 2 reports the results. As we can see, ITR failed to solve face95 for given restarts, and it also failed to solve face96 for restarts=200 but succeeded when restarts=300 and 400. By a comparison, both ITR_RL and ITR_GeigRL solved the problems very successfully and used much fewer inner and outer iterations than ITR_L did, and, of the three algorithms, ITR_GeigRL is the best and considerably more efficient than ITR_RL in terms of inner and outer iterations as well as the CPU time. We have found that similar phenomena were also be observed for the other test problems.

Finally, taking the maximum outer iterations and restarts as 16 and 300, respectively, we test the algorithms on face95, face96 for three $p=15,20,25$. Table 3 lists the results. Clearly, ITR_L failed to solve face95 for the three p's and face96 for $p=15$, and it succeeded in solving face96 for $p=20$ and 25. This indicates that ITR_L is not necessarily better for p smaller. In contrast, ITR_RL and ITR_GeigRL performed well and solved the two problems satisfactorily, but ITR_GeigRL was considerably more efficient than ITR_RL.

Table 2: A comparison of the algorithms on different maxInIter's.

data name		face95			face96		
size		5600			6084		
maxInIter		200	300	400	200	300	400
ITR_L	ρ_*	33.51	33.54	33.54	76.85	76.86	76.86
	outIter	16	16	16	16	11	13
	inIter	3029	4444	5867	2870	2710	4304
	time(s)	952.0	1274.9	1566.5	1313.7	1054.3	1543.4
	err	9E-4	2E-4	3E-4	1E-4	9E-7	3E-7
ITR_RL	ρ_*	33.56	33.56	33.56	76.86	76.86	76.86
	outIter	9	9	9	8	8	8
	inIter	1366	1978	2269	871	936	1063
	time(s)	743.9	1110.3	1295.3	733.5	799.3	883.6
	err	2E-7	2E-8	2E-10	7E-9	6E-9	6E-9
ITR_GeigRL	ρ_*	33.56	33.56	33.56	76.86	76.86	76.86
	outIter	5	5	5	4	4	4
	inIter	877	1376	1488	623	723	736
	time(s)	573.8	760.7	847.0	747.0	671.1	618.8
	err	5E-7	2E-7	2E-7	1E-8	2E-8	1E-8

Table 3: A comparison of the algorithms on different p's.

data name		face95			face96		
size		5600			6084		
p		15	20	25	15	20	25
ITR	ρ_*	44.35	33.54	29.86	87.05	76.86	69.28
	outIter	16	16	16	16	11	8
	inIter	4495	4444	4405	4352	2710	1723
	time(s)	1071.4	1274.9	1717.0	1307.1	1054.3	1144.6
	err	2E-3	2E-4	2E-4	6E-4	9E-7	1E-9
ITR_RL	ρ_*	44.53	33.56	29.88	87.10	76.86	69.28
	outIter	9	9	9	8	8	8
	inIter	2004	1978	1733	1230	936	709
	time(s)	848.4	1110.3	1353.7	715.7	799.3	832.0
	err	3E-7	2E-8	6E-12	5E-8	6E-9	1E-9
GRITR	ρ_*	44.53	33.56	29.88	87.10	76.86	69.28
	outIter	5	5	5	4	4	4
	inIter	1500	1376	1160	986	723	515
	time(s)	616.9	760.7	950.3	561.5	671.1	616.9
	err	1E-8	2E-7	5E-7	5E-9	2E-8	2E-8

Summarizing all the experiments, we conclude that ITR_RL is more reliable and performs much better than ITR_L and one can benefit much from preprocessing, so that the resulting ITR_GeigRL further improves the performance of ITR_RL.

6 Conclusions

We have made a convergence analysis on the ITR algorithm and established a global and asymptotic quadratic convergence result on $\{\rho_k\}$. In practical implementations, it has been common to use ITR_L. We have made two improvements on ITR_L. The first improvement has replaced the inner solver the implicitly restarted Lanczos algorithm, i.e., the Matlab code eigs, by the more efficient implicitly restarted refined Lanczos algorithm which leads to the ITR_RL algorithm. The second improvement has provided a better initial V_0 for ITR_RL by computing the p eigenvectors of the matrix pair (A,B) associated with its largest eigenvalues and orthonormalizing them. We have given some theoretical support on such replacement.

The numerical experiments have confirmed the considerable higher efficiency of ITR_RL than ITR and that of GeigRITR than ITR_RL. Actually, ITR_L fails to solve six ones of the seven practical problems in the first set of experiments and does not perform well in the second and third sets of experiments, but ITR_RL works reliably and efficiently for all the test problems. ITR_GeigRL further improves the overall efficiency of ITR_RL substantially, and it saves about half of outer iterations used by ITR_RL and consumes much fewer inner iterations than ITR_RL does.

Acknowledgments

This work was supported in part by the National Science Foundation of China (No. 11771249). We would like to thank two referees for their careful reading of our paper and for their valuable suggestions and comments.

References

[1] Y. Cai, L. H. Zhang, Z. Bai, and R. C. Li, On an eigenvector-dependent nonlinear eigenvalue problem, SIAM J. Matrix Anal. Appl., 339 (2018), 1360-1382.
[2] J. W. Demmel, Applied Numerical Linear Algebra, SIAM, Philadelphia, PA, 1997.
[3] R. A., Fisher, The use of multiple measurements in taxonomic problems, Annals of Eugenics, 7 (1936), 79-188.
[4] D. H. Foley and J. W. Sammon, An optimal set of discriminant vectors, IEEE Trans. Computers, 100 (1975), 281-289.
[5] K. Fukunaga, Introduction to Statistical Pattern Recognition, Academic Press, 2013.
[6] Y. F. Guo, S. J. Li and J. Y. Yang, A generalized Foley–Sammon transform based on generalized fisher discriminant criterion and its application to face recognition, Pattern Recognition Letters, 24 (2003), 147-158.

[7] X. He, S. Yan and Y. Hu, Face recognition using Laplacian faces, IEEE Trans. Pattern Anal. Machine Intell., 27 (2005), 328-340.
[8] Z. Q. Hong, Algebraic feature extraction of image for recognition, Pattern Recognition, 24 (1991), 211-219.
[9] Z. Jia, The convergence of generalized Lanczos methods for large unsymmetric eigenproblems, SIAM J. Matrix Anal. Appl., 16 (1995), 843-862.
[10] Z. Jia, Refined iterative algorithms based on Arnoldi's process for large unsymmetric eigenproblems, Linear Algebra Appl., 259 (1997), 1-23.
[11] Z. Jia, Polynomial characterizations of the approximate eigenvectors by the refined Arnoldi method and an implicitly restarted refined Arnoldi algorithm, Linear Algebra Appl., 287 (1999), 191-214.
[12] Z. Jia, Some theoretical comparisons of refined Ritz vectors and Ritz vectors, Sci. China Math., 47 (2004), 222-233.
[13] Z. Jia and G. W. Stewart, An analysis of the Rayleigh–Ritz method for approximating eigenspaces, Math. Comput., 70 (2001), 637-647.
[14] K. Liu, Y. Q. Cheng and J. Y. Yang, An efficient algorithm for Foley–Sammon optimal set of discriminant vectors by algebraic method, Intern J. Pattern Recogn. Art. Intell., 6 (5) (1992), 817-829.
[15] T. T. Ngo, M. Bellalij and Y. Saad, The trace ratio optimization problem, SIAM Rev., 54 (2012), 545-569.
[16] J. W. Sammon, An optimal discriminant plane, IEEE Trans. Computers, 100 (1970), 826-829.
[17] G. W. Stewart, Matrix Algorithms II: Eigensystem, SIAM, Philadelphia, PA, 2001.
[18] H. Wang, S. Yan and D. Xu, Trace ratio vs. ratio trace for dimensionality reduction, Proceedings of Computer Vision and Pattern Recognition, 2007. CVPR'07, IEEE Conf. IEEE, 2007, 1–8.
[19] S. Xiang, F. Nie and C. Zhang, Learning a Mahalanobis distance metric for data clustering and classification, Pattern Recogn., 41 (2008), 3600-3612.
[20] S. Yan and X. Tang, Trace quotient problems revisited, Computer Vision–ECCV, (2006), 232–244.
[21] L. H. Zhang and R. C. Li, Maximization of the sum of ratio trace on the Stiefel manifold, II: Computation, Sci. China Math., 58 (2015), 1549-1566.
[22] L. H. Zhang, L. Z. Liao and M. K. Ng, Fast algorithms for the generalized Foley–Sammon discriminant analysis, SIAM J. Matrix Anal. Appl., 31 (2010), 1584-1605.
[23] L. H. Zhang, L. Z. Liao and M. K. Ng, Superliner convergence of a general algorithm for the generalized Foley-Sammon discriminant analysis, J. Optim. Theory Appl., 157 (2013), 853-865.