Trace Ratio Problem Revisited

Yangqing Jia, Feiping Nie, and Changshui Zhang

***Abstract*—Dimensionality reduction is an important issue in many machine learning and pattern recognition applications, and the trace ratio problem is an optimization problem involved in many dimensionality reduction algorithms. Conventionally, the solution is approximated via generalized eigenvalue decomposition due to the difficulty of the original problem. However, prior works have indicated that it is more reasonable to solve it directly than the conventional way. In this paper, we propose a theoretical overview of the global optimum solution to the trace ratio problem via the equivalent trace difference problem. Eigenvalue perturbation theory is introduced to derive an efficient algorithm based on the Newton-Raphson method. Theoretical issues on the convergence and efficiency of our algorithm compared with prior literature are proposed, and are further supported by extensive empirical results.**

***Index Terms*—Dimensionality reduction, trace ratio, eigenvalue perturbation, Newton-Raphson method.**

I. Introduction

MANY machine learning and pattern recognition applications involve processing data in a high-dimensional space. For computational time, storage, de-noise and other considerations, we often reduce the dimensionality of such data in order to learn a model efficiently. Also, in many cases it has been found that the data has low-dimensional structures such as the manifold structure, so we are often interested to find a reasonable low-dimensional representation of the data.

There are a number of supervised and unsupervised dimensionality reduction algorithms such as Linear Discriminant Analysis (LDA), Kernelized LDA, Marginal Fisher Analysis, Principal Component Analysis (PCA), ISOMAP, LLE, etc. Many of these algorithms can be formulated into a fundamental optimization problem called the *trace ratio* problem, which involves searching for a transform matrix W that maximizes the trace ratio $Tr[W^T S_p W]/Tr[W^T S_l W]$ where S_p and S_l are method related positive semidefinite matrices, together with the constraint that the columns of W are unitary and orthogonal. However, the trace ratio problem does not have a closed-form global optimum solution directly. Thus, it is conventionally approximated by solving a *ratio trace* problem $\max_W Tr[(W^T S_p W)^{-1}(W^T S_l W)]$, which is essentially different and diverts from the original optimum value. Previous works, such as [1], have showed that the optimum solution to the original trace ratio problem is superior to the diverted one in performance.

However, the trace ratio problem does not have a closed-form solution. There have been some attempts to find the global optimum solution: [2] pointed out that the original trace ratio problem can be converted to a *trace difference* problem, and proposed a heuristic bisection way to find the solution. Further, Wang et al. [1] proposed an iterative algorithm called ITR that was empirically more efficient. However, except for the convergence of these methods, much of the theoretical nature is left undiscussed. In this paper, we aim to give a theoretical view of the trace ratio / trace difference problems by introducing the eigenvalue perturbation theory into the analysis. We then propose a new method that is both theoretically and empirically proved to be more efficient than the previous ones.

The following parts of the paper are organized as follows: Section II introduces the trace ratio / trace difference problems and discusses their relationship. Section III explores the property of the trace difference function, introduces the perturbation theory and proposes a new method. The convergence of the method is also provided. In Section IV, we discuss the theoretical explanation to previous algorithms and show the superiority of our method. Experiments on object recognition databases are presented in Section V. Finally, Section VI concludes the paper.

The authors are with the State Key Laboratory on Intelligent Technology and Systems, Tsinghua National Laboratory for Information Science and Technology (TNList), Department of Automation, Tsinghua University, Beijing, China.

II. The Trace Ratio Problem

A. Trace Ratio, Ratio Trace, and Trace Difference

We introduce the trace ratio problem from the notion of linear discriminant analysis (LDA). Given a set of n training data points $\mathcal{X} = \{x_i\}_{i=1}^n \subset \mathbb{R}^m$ and the corresponding label $\mathcal{Y} = \{y_i\}_{i=1}^n$, where $y_i \in \{1, \cdots, c\}$ is the label of the data point x_i, LDA tries to find a low-dimensional representation in $\mathbb{R}^d$ via a linear transformation $x' = W^T x$ (where W is a $m \times d$ matrix) that maximizes the between-class scatter and minimizes the within-class scatter at the same time:

$$W^* = \arg\max_W \frac{\sum_{i=1}^c \frac{n_i}{n} \|W^T m_i - W^T m\|^2}{\sum_{i=1}^n \frac{n_{y_i}}{n} \|W^T x_i - W^T m_{y_i}\|^2}, \quad (1)$$

where m_i and n_i are respectively the mean and the number of the data points belonging to the i-th class, and m is the mean of all the data points. By defining the between-class covariance matrix $S_b = \sum_{i=1}^c \frac{n_i}{n}(m_i - m)(m_i - m)^T$ and the within-class covariance matrix $S_w = \sum_{i=1}^n \frac{n_{y_i}}{n}(x_i - m_{y_i})(x_i - m_{y_i})^T$,the problem (1) is equivalent to the following form:

$$W^* = \arg\max_W \frac{Tr[W^T S_b W]}{Tr[W^T S_w W]}, \quad (2)$$

where $Tr[\cdot]$ denotes the matrix trace. Conventionally, we add the constraint $W^T W = I$ to get a set of orthogonal and normalized transform vectors.

Recently, Yan et al. proposed a uniform graph embedding framework for several dimensionality reduction problems [3]. Generally, we want to find a low-dimensional representation that minimizes the distance between certain pairs of data points

(x_i, x_j) with similarity weight S_{ij}, and maximizes the distance between some other pairs (x_i, x_j) with penalty weight S'_{ij}. This leads to finding a matrix W^* that satisfies:

$$W^* = \arg\max_{W^TW=I} \frac{\sum_{i,j} S'_{ij}\|W^Tx_i - W^Tx_j\|^2}{\sum_{i,j} S_{ij}\|W^Tx_i - W^Tx_j\|^2}. \quad (3)$$

Two graphs are used to model the problem. One is called similarity graph: similarity matrix S whose ij-th element is S_{ij}, diagonal matrix D whose i-th diagonal element is $D_{ii} = \sum_j S_{ij}$, and graph Laplacian $L = D - S$. The other is called penalty graph with similarly defined S', D' and L'. With such definition, the equation (3) can be written as

$$W^* = \arg\max_{W^TW=I} \frac{Tr[W^TXL'X^TW]}{Tr[W^TXLX^TW]}. \quad (4)$$

A more extensive introduction of the graph embedding framework can be found in [3].

The problems (2) and (4) are typical forms of a general trace ratio problem, defined as follows:

Definition 1 (Trace Ratio)**.** *For two $m \times m$ positive semidefinite matrices S_p and S_l, the trace ratio problem is defined as finding a $m \times d$ ($d < m$) transform matrix W^* that satisfies*

$$W^* = \arg\max_{W^TW=I} \frac{Tr[W^TS_pW]}{Tr[W^TS_lW]} \quad (5)$$

and the optimum trace ratio value

$$\lambda^* = \max_{W^TW=I} \frac{Tr[W^TS_pW]}{Tr[W^TS_lW]}. \quad (6)$$

We also assume that the rank of S_l is larger than $m - d$, *i.e.*, the null space of S_l has dimensionality less than d. This is to make the trace ratio value finite. Actually, if the rank of S_l is smaller than d, we can find a transform matrix W that satisfies $Tr[W^TS_lW] = 0$, which draws the trace ratio value to infinity[1]. For completeness, we will also discuss the case when the null space of S_l has dimensionality equal to or larger than d at the end of Section III.

The difficulty with the trace ratio problem is that it does not have a closed-form solution. Thus, the commonly-used solution is to solve an alternative *ratio trace* problem defined as follows:

Definition 2 (Ratio Trace)**.** *For two $m \times m$ positive semidefinite matrices S_p and S_l, the ratio trace problem is to find a $m \times d$ ($d < m$) transform matrix W^* that satisfies*

$$W^* = \arg\max_{W} Tr[(W^TS_lW)^{-1}(W^TS_pW)]. \quad (7)$$

This problem can be efficiently solved by generalized eigenvalue decomposition (GEVD) $S_pw_k = \beta_kS_lw_k$, where β_k is the k-th largest generalized eigenvalue. The matrix W is then constituted of the corresponding eigenvalues w_k, $k = 1, \cdots, d$. For example, LDA uses $S_p = S_b$ and $S_l = S_w$, where S_b and S_w are the interclass/intraclass covariance matrices. Another version of LDA [4] uses $S_l = S_w + S_b$.

[1]A simple way to find such W is to use the d eigenvectors corresponding to the eigenvalue 0 of S_l.

When the reduced dimensionality $d = 1$, the trace ratio and ratio trace problems are strictly equal, since W^TS_bW and W^TS_wW are both scalar values. When $d > 1$, denote the columns of W by w_i, the ratio trace problem iteratively finds the k-th column w_i that maximizes $\frac{w_i^TS_pw_i}{w_i^TS_lw_i}$. Thus it can be seen as a greedy algorithm which essentially optimizes $\sum_{i=1}^d \frac{w_i^TS_pw_i}{w_i^TS_lw_i}$. This is different from the original trace ratio problem, which can be written as:

$$\frac{Tr[W^TS_pW]}{Tr[W^TS_lW]} = \frac{\sum_{i=1}^d w_i^TS_pw_i}{\sum_{i=1}^d w_i^TS_lw_i} \quad (8)$$

As the superiority is not the main issue of this paper, we refer to prior works [1] and [2] for the comparison between different criterions. In brief, we expect that the optimum solution of the trace ratio problem should be better than the greedy GEVD solution. To solve the trace ratio problem directly, Guo et al. have indicated that one can solve an equivalent *trace difference* problem to find the global optimum of the trace ratio problem [2]. We cite the theorem without proof as follows:

Theorem 1 (Thm.2 of [2])**.** *To find the best trace ratio value λ^* and matrix W^*, it is equivalent to solve the corresponding trace difference problem: to find the zero point of the trace difference function*

$$f(\lambda) = \max_{W^TW=I} Tr[W^T(S_p - \lambda S_l)W], \quad (9)$$

i.e.*, to solve a trace difference equation $f(\lambda) = 0$. W^* can then be calculated as*

$$W^* = \arg\max_{W^TW=I} Tr[W^T(S_p - \lambda^* S_l)W]. \quad (10)$$

B. Previous Works

Several methods have been proposed to solve the trace ratio and trace difference problem. Guo et al. [2] introduced the trace ratio and trace difference problems in the notion of generalized Foley–Sammon transform. In their work, they proposed to solve the trace difference equation in an iterative way as follows:

1) Initialize λ_1 and λ_2 so that $f(\lambda_1) > 0 > f(\lambda_2)$.
2) Calculate $\lambda = (\lambda_1 + \lambda_2)/2$ and $f(\lambda)$.
3) If $f(\lambda) > 0$, let $\lambda_1 = \lambda$, else let $\lambda_2 = \lambda$.
4) Iterate until convergence.

It is easy to initialize $\lambda_1 = 0$, while λ_2 is a bit difficult to initialize, because we do not actually know the optimum trace ratio value. Thus, the method often turns to optimize an equivalent trace ratio $Tr[W^TS_pW]/Tr[W^T(S_p+S_l)W]$. The proof to the equivalency can be found in [2]. In this way, we can safely set $\lambda_2 = 1$.

Wang et al. [1] proposed an iterative algorithm called Iterative Trace Ratio (ITR) to solve the trace ratio problem. Given λ_t at each iteration t, they search for a transform matrix according to the trace difference as:

$$W_t = \arg\max_{W^TW=I} Tr[W^T(S_p - \lambda_t S_l)W], \quad (11)$$

and renew λ_{t+1} as the trace ratio given by W_t:

$$\lambda_{t+1} = \frac{Tr[W_t^TS_pW_t]}{Tr[W_t^TS_lW_t]}. \quad (12)$$

The algorithm iterates until convergence. This algorithm is empirically faster than the one in [2].

The problem of previous works is the absence of theoretical discussion. Except for the convergence of the algorithm, little theoretical foundation has been discussed before. In the next section, we will first explore several characteristics of the trace difference function $f(\lambda)$, and then derive an efficient algorithm based on the Newton-Raphson method. We will also give a theoretical explanation to the previous algorithms and show the superiority of the new algorithm.

III. Decomposed Newton's Method

A. Property of $f(\lambda)$

First, we have the following property about the trace difference function:

Lemma 1. *The function $f(\lambda)$ is monotonic decreasing and convex.*

Proof: For any value $\lambda_1 < \lambda_2 < \lambda_3$, denote $W_i = \arg\max_{W^TW=I} Tr[W^T(S_p - \lambda_i S_l)W]$, we have

$$\begin{aligned} f(\lambda_1) &= \max_{W^TW=I} Tr[W^T(S_p - \lambda_1 S_l)W] \\ &\geq Tr[W_2^T(S_p - \lambda_1 S_l)W_2] \\ &= f(\lambda_2) + (\lambda_2 - \lambda_1)Tr[W_2^T S_l W_2]. \end{aligned} \quad (13)$$

The second term is nonnegative since S_l is positive semidefinite and $\lambda_2 > \lambda_1$, so we have

$$f(\lambda_1) - f(\lambda_2) \geq (\lambda_2 - \lambda_1)Tr[W_2^T S_l W_2] \geq 0. \quad (14)$$

This proves the monotonic decreasing property of $f(\lambda)$.

Similarly, we have

$$(\lambda_3 - \lambda_2)Tr[W_2^T S_l W_2] \geq f(\lambda_2) - f(\lambda_3) \geq 0. \quad (15)$$

Combine the two inequality (14) and (15) together, we have

$$f(\lambda_2) \leq \frac{1}{\lambda_3 - \lambda_1}[(\lambda_3 - \lambda_2)f(\lambda_1) + (\lambda_2 - \lambda_1)f(\lambda_3)]. \quad (16)$$

Thus the convexity is proved. ∎

It is not difficult to verify that $f(0) \geq 0$ and $f(+\infty) \leq 0$, so Lemma 1 guarantees that the zero point of $f(\lambda)$ exists. In another word, the global optimum of the trace ratio problem can be found if we are able to solve $f(\lambda) = 0$. Moreover, if S_l is positive definite, the inequality above are all strict, and $f(\lambda)$ is strictly monotonic decreasing, thus the solution is unique[2].

Defining $f(\lambda)$ based on an optimization problem is somehow inconvenient for analysis, so we search for an analytical representation. This naturally leads us to consider the eigenvalues of the matrix $S(\lambda) = S_p - \lambda S_l$. Denote its m eigenvalues and their corresponding eigenvectors by $\{\beta_1, \cdots, \beta_m\}$ and $\{w_1, \cdots, w_m\}$, and an "index vector" $i = (i_1, \cdots, i_m)$ which is a certain permutation of $\{1, \cdots, m\}$, we have the following lemma according to the eigenvalue decomposition analysis of the matrix theory [5]:

[2]Note that it is sufficient but not necessary to require S_l to be positive definite. Actually, we can relax it to that the rank of S_l is larger than $m - d$, and similar results still holds.

Lemma 2. *The trace difference function value $f(\lambda)$ is given by the sum of the first d algebraically largest eigenvalues, and the corresponding matrix W is formed by the first d corresponding eigenvectors:*

$$f(\lambda) = \max_i \sum_{k=1}^{d} \beta_{i_k}, \quad W(\lambda) = [w_{i_1}, \cdots, w_{i_d}]. \quad (17)$$

Note that all the eigenvalues and eigenvectors are actually functions of λ, since the matrix S is dependent on λ. Thus, it may be more accurate to write them as functions of λ such as $\beta_1(\lambda)$, $w_1(\lambda)$. However, we omit the function-based representation for simplicity here.

Lemma 2 indicates that we need find a proper λ so that the sum of the first m largest eigenvalues of $S(\lambda)$ equals to zero. Still, we cannot find a direct closed-form solution, so we turn to an iterative way to solve it. First we observe how the function value and the eigenvalues change when λ varies, by introducing perturbation theory [6] to analyze the eigenvalues. Here we assume that all eigenvalues of $S(\lambda)$ are *simple eigenvalues*: in the matrix theory, a simple eigenvalue is an eigenvalue that has algebraic multiplicity 1. For such eigenvalues, we have the following result:

Lemma 3. *If $\beta(\lambda)$ is a simple eigenvalue of $S(\lambda) = S_p - \lambda S_l$ with its corresponding normalized eigenvector $w(\lambda)$ ($\|w(\lambda)\| = 1$), the derivative of the eigenvalue is given by*

$$\beta'(\lambda) = -w^T(\lambda) S_l w(\lambda) \leq 0. \quad (18)$$

Proof: Here we give a brief proof to the lemma. For more detailed discussion, see e.g. [7]. First, according to the definition of eigenvalues, we have

$$(S_p - \lambda S_l - \beta(\lambda)I)w(\lambda) \equiv 0. \quad (19)$$

Take the derivative of the right side w.r.t λ, we have

$$(-S_l - \beta'(\lambda)I)w(\lambda) + (S_p - \lambda S_l - \beta(\lambda)I)w'(\lambda) = 0. \quad (20)$$

Take the inner product of $w(\lambda)$ and the left side, we have

$$w^T(\lambda)[-S_l - \beta'(\lambda)I]w(\lambda) + w^T(\lambda)[S_p - \lambda S_l - \beta(\lambda)I]w'(\lambda) = 0. \quad (21)$$

Note that $S_p - \lambda S_l$ is symmetric, and $\|w(\lambda)\| = 1$ so $w^T(\lambda)w'(\lambda) = 0$, thus the second term of the left side equals to zero. This leads to the simple form

$$w^T(\lambda) S_l w(\lambda) + w^T(\lambda)\beta'(\lambda)w(\lambda) = 0. \quad (22)$$

Again, using $\|w(\lambda)\| = 1$ we have $w^T(\lambda)\beta'(\lambda)w(\lambda) = \beta'(\lambda)$. Thus, we have

$$\beta'(\lambda) = -w^T(\lambda) S_l w(\lambda) \leq 0. \quad (23)$$

The inequality holds because S_l is positive semidefinite. Thus the theorem is proved. ∎

One might ask whether the assumption made on the eigenvalues may be too strong. Actually, in most real-world problems, the eigenvalues of the matrices involved are generally simple eigenvalues so that Lemma 3 can be applied.

TABLE I
THE ALGORITHM OF DECOMPOSED NEWTON'S METHOD

Input: Two positive semidefinite matrices S_p and S_l
Output: Trace Ratio value λ^* and the transform matrix W^*
Procedure:
1. Initialize $\lambda_0 = 0$, $t = 0$;
2. Do eigenvalue decomposition of $S_p - \lambda_t S_l$
3. Calculate the first-order approximations using (24) and (25).
4. Renew λ_{t+1} by solving $\hat{f}_D(\lambda) = 0$.
5. If $|\lambda_{t+1} - \lambda_t| < \epsilon$, go to 6; else $t = t + 1$, go to 2
6. Output $\lambda^* = \lambda_t$, and $W^* = \max_{W^T W = I} Tr[W^T(S_p - \lambda^* S_l)]W$.

B. The Proposed Method

Consider an iterative way to solve the trace difference equation (9) given initial value λ_t, Lemma 3 enables us to use the first-order Taylor expansion to approximate the eigenvalues around λ_t as:

$$\hat{\beta}_k(\lambda) = \beta_k(\lambda_t) + \beta'_k(\lambda_t)(\lambda - \lambda_t), \tag{24}$$

for $k = 1, \cdots, m$. Using the Taylor expansion, we approximate the trace difference function $f(\lambda)$ as the sum of the largest d values in $\{\hat{\beta}_1(\lambda), \cdots, \hat{\beta}_m(\lambda)\}$:

$$\hat{f}_D(\lambda) = \max_i \sum_{k=1}^{d} \hat{\beta}_{i_k}(\lambda). \tag{25}$$

In another word, since the trace difference function $f(\lambda)$ depends on the largest d eigenvalues, we first decompose the function down to a set of eigenvalues and use their Taylor expansion to approximate the function value $f(\lambda)$, and renew λ by solving the approximated problem. Solving $\hat{f}_D(\lambda) = 0$ is not difficult since it only involves n linear functions. In practice, the time used to calculate $\hat{f}_D(\lambda) = 0$ can be neglected compared with the eigenvalue decomposition procedure. Then, we renew λ_{t+1} as the zero point of $\hat{f}_D(\lambda)$. By running such procedure iteratively, we can find the accurate solution to the trace difference equation. We call the method *Decomposed Newton's Method* (DNM). An algorithmic presentation is provided in Table I.

One may easily find the relationship between our method and the general Newton-Raphson method which is used to find zero points for differentiable functions. In Newton-Raphson method, the function is approximated by its first-order Taylor expansion, and the solution is also sought iteratively. The difference of the two and the superiority of our method is presented in the next section.

Next, we prove the convergence of our method:

Theorem 2 (convergence)**.** *Denote the optimum trace ratio value by λ^*, for any initial $\lambda_t < \lambda^*$, the renewed value λ_{t+1} satisfies (a) $\lambda_{t+1} > \lambda_t$, and (b) $\lambda_{t+1} \leq \lambda^*$.*

Proof: The first term is straightforward. Because the approximated eigenvalues $\hat{\beta}_k$ around λ_t are all monotonic decreasing functions (and recall our assumption, at least $m-d+1$ of them are strict monotonic decreasing since the null space of S_l has dimensionality smaller than d), $\hat{f}_D(\lambda)$ is strict monotonic decreasing. Using the facts $\hat{f}_D(\lambda_t) = f(\lambda_t) > 0$, $\hat{f}_D(\lambda_{t+1}) = 0$ and Lemma 1, we have $\lambda_{t+1} > \lambda_t$.

We focus on proving the second inequality. According to the definition of $\hat{f}_D(\lambda)$, we have

$$\begin{aligned}
\hat{f}_D(\lambda) &= \max_i \sum_{k=1}^{d} \hat{\beta}_{i_k}(\lambda) \\
&= \max_i \sum_{k=1}^{d} \left[\beta_{i_k}(\lambda_t) - (\lambda - \lambda_t) w_{i_k}^T(\lambda_t) S_l w_{i_k}(\lambda_t)\right] \\
&= \max_i Tr[W^T(i)(S_p - \lambda S_l)W(i)], \qquad (26)
\end{aligned}$$

where $W(i) = [w_{i_1}(\lambda_t), \cdots, w_{i_d}(\lambda_t)]$ is the matrix with columns selected from eigenvectors of $S(\lambda_t)$. Thus,

$$\hat{f}_D(\lambda) \leq \max_{W^T W = I} Tr[W^T(S_p - \lambda S_l)W] = f(\lambda). \tag{27}$$

This implies that if λ_{t+1} is the zero point of $\hat{f}_D(\lambda)$, then $f(\lambda_{t+1}) \geq 0$. Because $f(\lambda^*) = 0$ and $f(\lambda)$ is monotonic decreasing, we have $\lambda_{t+1} < \lambda^*$. ■

C. Singularity Case

The singularity case is an important issue in a large literature of dimensionality reduction, such as [8] and [9], to name just a few. Here we point out that our method does not require S_l to be full-ranked. Actually, if the null space of S_l has dimensionality smaller than d, whether S_l is singular or not is not important, since the denominator $Tr[W^T S_l W] > 0$ always holds. Also, in this way, the null space of S_l and its orthogonal complement space are simultaneously considered when finding the global optimum solution the trace ratio problem, and the singularity problem is inherently solved.

When the null space of S_l has dimensionality d' larger than d (the dimensionality to be reduced to), the optimum trace ratio value goes to infinity: note that any transform matrix W whose column vectors belong to the null space of S_l will result in the denominator $Tr[W^T S_l W]$ to be zero. In this case, a natural alternative solution is to maximize the numerator, *i.e.*, to solve $\max Tr[W^T S_p W]$ to find the appropriate transform matrix in the null space of S_l. One may easily find that this shares the same thought of the null space LDA [8].

IV. DISCUSSION

In this section, we discuss the relationship between our method and the previous methods described in Section II-B.

A. Generalized Foley–Sammon transform

One may immediately find that the Generalized Foley–Sammon transform method proposed in [2] is equivalent to the heuristic bisection method to find the zero point of a function. Actually, as has been indicated in [1], this method generally needs a large number of iterations before converging to some satisfactory solution.

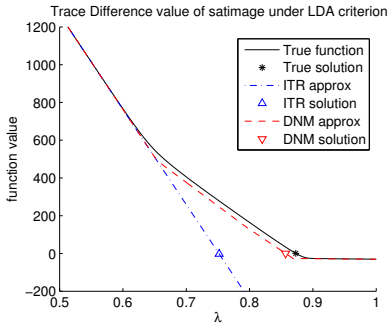

Fig. 1. A toy example of the trace difference function value $f(\lambda)$ and the two approximation methods: $\hat{f}_D(\lambda)$ and $\hat{f}_I(\lambda)$. The black solid line is the accurate trace difference function value. The blue dot-dashed line and the red dashed line are respectively ITR and DNM approximations. It can be seen that the DNM approximation value lies between the ITR approximation value and the true function value. Similar results can be found for the solutions that are marked in the figure, which supports Theorem 4.

B. Iterative Trace Ratio (ITR) algorithm

The ITR algorithm is empirically proved to be more efficient in [1], but does not have a sound theoretical explanation. Introducing perturbation analysis enables us to look into the theoretical nature of the method. Actually, we have the following theorem:

Theorem 3. *The iterative trace ratio algorithm in [1] is equivalent to the naive Newton-Raphson Method.*

Proof: Define the optimum index vector $i^{(t)}$ that sorts the n eigenvalues at λ_t from large to small, *i.e.*, $\beta_{i_1^{(t)}}(\lambda_t) \geq \beta_{i_2^{(t)}}(\lambda_t) \geq \cdots \geq \beta_{i_n^{(t)}}(\lambda_t)$. According to Lemma 2 and Lemma 3, for iteration t, the derivative of $f(\lambda)$ at point λ_t is given by

$$f'(\lambda_t) = \sum_{k=1}^{d} \beta_{i_k^{(t)}}{}'(\lambda_t) = -Tr[W_t^T S_l W_t]. \tag{28}$$

The Newton-Raphson method then renews λ_{t+1} as

$$\begin{aligned} \lambda_{t+1} &= \lambda_t - \frac{f(\lambda_t)}{f'(\lambda_t)} \\ &= \lambda_t + \frac{Tr[W_t^T(S_p - \lambda_t S_l)W_t]}{Tr[W_t^T S_l W_t]} \\ &= \frac{Tr(W_t^T S_p W_t)}{Tr(W_t^T S_l W_t)} \end{aligned} \tag{29}$$

which is exactly ITR's renew criterion. ∎

Similar with Theorem 2, the convergence of ITR can be easily proved. And due to the Newton-Raphson method nature of ITR, it is generally faster than the simple bisection method. This is also mentioned empirically in [1] without a satisfactory theoretical explanation, which we have proved here. Moreover, we can theoretically compare the speed of DNM and ITR:

Theorem 4. *For any initial $\lambda_t < \lambda^*$, the renewed value λ_{t+1} given by DNM is always no smaller than the one given by ITR.*

Proof: ITR can be viewed as finding the zero point of the direct first-order Taylor expansion of $f(\lambda)$ as

$$\hat{f}_I(\lambda) = \sum_{k=1}^{d} \hat{\beta}_{i_k^{(t)}}(\lambda). \tag{30}$$

This is also a monotonic decreasing function. To prove that the renewed value λ_{t+1} given by DNM is always no smaller than the one given by ITR, we just need to prove that $\hat{f}_I(\lambda) \leq \hat{f}_D(\lambda)$. Actually, since $\hat{f}_D(\lambda)$ searches for the optimum value over all index vectors i instead of $i^{(t)}$, we have

$$\hat{f}_D(\lambda) = \max_i \sum_{k=1}^{d} \hat{\beta}_{i_k}(\lambda) \geq \sum_{k=1}^{d} \hat{\beta}_{i_k^{(t)}}(\lambda) = \hat{f}_I(\lambda). \tag{31}$$

Thus the result is straightforward. ∎

This theorem guarantees the superiority of DNM over the ITR algorithm. In another word, the difference between ITR (*i.e.*, the naive Newton-Raphson method) and our DNM algorithm is that, ITR chooses the largest d eigenvalues at λ_t and uses the sum of their first-order Taylor expansion to approximate f. However, we notice that the largest d eigenvalues at λ_t may not remain largest when λ changes. Thus, instead of fixing on the d eigenvalues, we dynamically choose the largest d eigenvalues at λ and use their sum to approximate f. This approach assures that our approximation is always larger than that of ITR. Also, since both ITR and DNM gives a lower bound of the true $f(\lambda)$, our method guarantees to converge faster. To illustrate the relationship between $f(\lambda)$, $\hat{f}_D(\lambda)$ and $\hat{f}_I(\lambda)$, a toy example is presented in Figure 1. The experiment is performed on the UCI *satimage* data set under LDA criterion with $d = 10$.

C. Maximum Margin Criterion

A similar thought of the trace difference function is shared by Maximum Margin Criterion (MMC) [10]. In short, under MMC one optimizes the following criterion to find an appropriate transform matrix $W \in \mathbb{R}^{m \times d}$:

$$W = \arg\max_{W^T W = I} Tr[W^T(S_p - S_l)W], \tag{32}$$

where $W^T(S_p - S_l)W$ is defined as the margin between classes. However, one might raise the question: can we define the margin using $S_p - \lambda S_l$ with $\lambda \in \mathbb{R}^+$ taking values other than 1? Actually, as the trace of the interclass covariance matrix $Tr[S_p]$ is usually much larger than the trace of the intraclass covariance matrix $Tr[S_l]$, it may be better to assign a weight factor λ for balance, and λ should be larger than 1 intuitively. This is justified by the experimental result shown in Figure 3. By introducing the trace difference function, we can see that MMC may be seen as a special case of the trace ratio framework when $\lambda = 1$, and that MMC can be considered as an approximated solution to the trace ratio problem. Actually, by taking λ into account as a parameter to be optimized, our method is expected to give a better result. This also builds up a relationship between MMC and the classical dimensionality reduction algorithms.

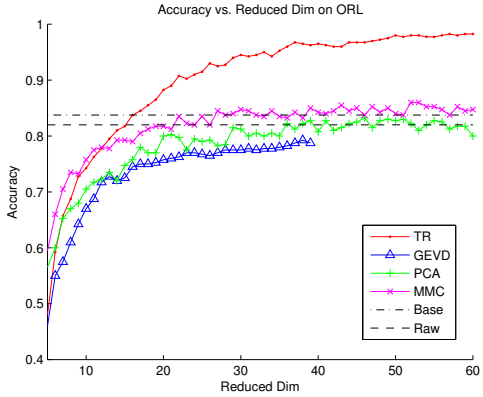

Fig. 2. Accuracy versus reduced dimensionality on the ORL face database. The line "Base" indicates the accuracy on the dimensionality after the precedent PCA process, and "Raw" indicates the accuracy on the unprocessed dimensionality.

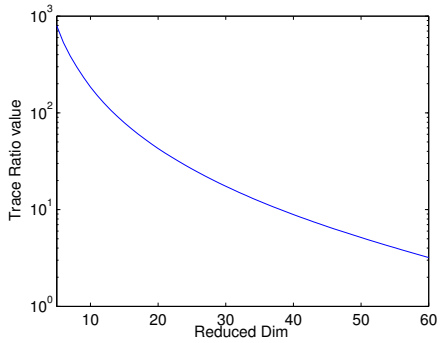

Fig. 3. The average trace ratio value of the cross validation versus reduced dimensionality.

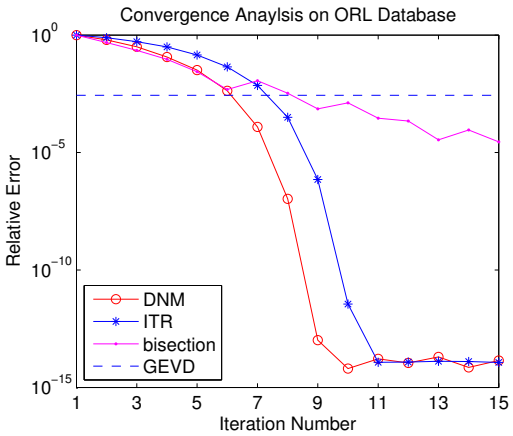

Fig. 4. Relative error $|\lambda_t - \lambda^*|/\lambda^*$ versus iteration number, which indicates the convergence of the algorithms. The errors of GEVD is also shown in the image as dashed lines for reference. Note that the final values around 10^{-15} are due to numerical accuracy.

TABLE II
ERROR RATES (IN PERCENTAGE) ON THE DATABASES USING 10-FOLD CROSS VALIDATION.

Error Rate	ORL	UMist	USPS	MNIST	Coil_{20}
Trace Ratio	96.25	98.62	88.45	87.73	98.96
Ratio Trace	76.50	98.44	89.50	87.55	97.61

V. EXPERIMENTS

To further validate the theoretical result of the previous sections, we carry out experiments on the ORL face database and other object recognition applications that usually involves a high dimensionality, and compare the results against other related methods. For all experiments, we choose the LDA criterion to generate the two matrices $S_p = S_b$ and $S_l = S_w$.

The ORL face database contains 400 images from 40 persons, with each image downsized and normalized to 56-by-46 pixels by size for computation speed consideration. In our experiment, we compare the trace ratio (TR) method against four other criterions, namely PCA, ratio trace solution to LDA via generalized eigenvalue decomposition (GEVD), and MMC. For all the dimensionality reduction algorithms, the images are treated as a vector, and a precedent PCA is used to save 98% of the input data's energy. After dimensionality reduction, a simple k-nearest-neighbor classifier is used to calculate the accuracy with neighbor number $k = 5$. We report the accuracy value of a ten-fold cross validation in Figure 2. The reduced dimensionality varies from 5 to 60. Note that for GEVD, the reduced dimensionality is at most 39 due to the rank of S_b, thus we stop at this dimensionality for it. The average trace ratio value is shown in Figure 3. It can be observed from the accuracy that the trace ratio method performs competitively against all other methods, and can get a better result than using the data without dimensionality reduction. Especially, the superiority of trace ratio increases with dimensionality. We adopted other experimental settings such as randomly choosing a proportion of the data for training and the remaining for testing. The results were similar to the cross validation, thus we omit the details here. More extensive experiments of trace ratio against other methods can be found in the related works such as [1].

To compare the convergence speed of our method against the two other trace ratio methods (bisection and ITR), we report the convergence rate of the three methods when reduced dimensionality $d = 10$. The mean value of the λ^* given by the three methods after their convergence is considered as the "true" trace ratio value λ^*, and for each iteration, the relative error $|\lambda_t - \lambda^*|/\lambda^*$ is used to monitor the convergence. The result is shown in Figure 4. Further, we perform dimensionality reduction on four other object recognition databases, namely the UMist face database, the USPS digital database, the MNIST digital database, and the COIL_{20} object database. The convergence of the trace ratio algorithms are shown in Figure 5, using the relative error as the measure. It is not surprising to see that our method converges faster in all the cases, since it has been theoretically guaranteed. The error of the trace ratio value for the GEVD solution (which corresponds to the ratio trace criterion) is also provided, showing that GEVD

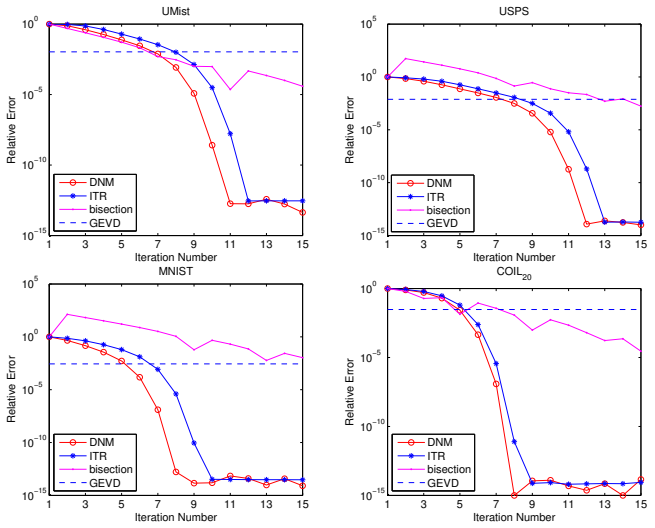

Fig. 5. Convergence analysis over the UMist, USPS, MNIST, and COIL_{20} databases.

does not give the optimum solution as we have analyzed in the previous sections. We also provide the error rate using ten-fold cross validation on the databases in Table II, where the optimum dimensionality is searched from 1 to $c-1$. According to the results and previous experiments such as in [1], we infer that the trace ratio algorithm is most effective when the number of data is small while the dimensionality is large, such as the face databases. When the dimensionality is not very high and there are a large number of data (such as handwritten digit number data sets), the trace ratio algorithm gives comparatively marginal improvement.

VI. Conclusion

Dimensionality reduction is a fundamental problem in many machine learning fields and related applications. In this paper, we focused on solving the trace ratio problem that can be used to model many dimensionality reduction algorithms. The main contribution of our paper lies in two aspects: first, we have explored the theoretical nature of the trace ratio problem, the trace difference problem and their relationship by introducing the eigenvalue perturbation theory, and have also given the theoretical explanation of the previous trace ratio algorithms. Second, we have proposed a new way to efficiently find the global optimum of the trace ratio problem, whose foundation and superiority is guaranteed by the theoretical analysis.

Acknowledgement

This work is supported by the National Natural Science Foundation of China (Grant No. 60835002, 60721003).

References

[1] H. Wang, S. Yan, D. Xu, X. Tang, and T. Huang, "Trace Ratio vs. Ratio Trace for Dimensionality Reduction," *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[2] Y. Guo, S. Li, J. Yang, T. Shu, and L. Wu, "A generalized Foley–Sammon transform based on generalized fisher discriminant criterion and its application to face recognition," *Pattern Recognition Letters*, vol. 24, no. 1-3, pp. 147–158, 2003.

[3] S. Yan, D. Xu, B. Zhang, H.-J. Zhang, Q. Yang, and S. Lin, "Graph Embedding and Extensions: A General Framework for Dimensionality Reduction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 1, pp. 40–51, 2007.

[4] J. Lu, K. Plataniotis, and A. Venetsanopoulos, "Face recognition using LDA-based algorithms," *Neural Networks, IEEE Transactions on*, vol. 14, no. 1, pp. 195–200, 2003.

[5] R. Horn and C. Johnson, *Matrix Analysis*. Cambridge University Press, 1985.

[6] T. Kato, *Perturbation Theory for Linear Operators*. Springer, 1995.

[7] K. Ngo, "An Approach of Eigenvalue Perturbation Theory," *Applied Numerical Analysis & Computational Mathematics*, vol. 2, no. 1, pp. 108–125, 2005.

[8] L. Chen, H. Liao, M. Ko, J. Lin, and G. Yu, "A new LDA-based face recognition system which can solve the small sample size problem," *Pattern Recognition*, vol. 33, no. 10, pp. 1713–1726, 2000.

[9] H. Yu and J. Yang, "A direct LDA algorithm for high-dimensional data with application to face recognition," *Pattern Recognition*, vol. 34, no. 10, pp. 2067–2070, 2001.

[10] H. Li, T. Jiang, and K. Zhang, "Efficient and Robust Feature Extraction by Maximum Margin Criterion," *Neural Networks, IEEE Transactions on*, vol. 17, no. 1, pp. 157–165, 2006.