www.elsevier.com/locate/pr

Supervised dimensionality reduction via sequential semidefinite programming

Chunhua Shen, Hongdong Li, Michael J. Brooks

PII: S0031-3203(08)00244-6
DOI: doi:10.1016/j.patcog.2008.06.015
Reference: PR 3243

To appear in: *Pattern Recognition*

Received date: 18 June 2007
Revised date: 7 May 2008
Accepted date: 16 June 2008

Cite this article as: Chunhua Shen, Hongdong Li and Michael J. Brooks, Supervised dimensionality reduction via sequential semidefinite programming, *Pattern Recognition* (2008), doi:10.1016/j.patcog.2008.06.015

Supervised Dimensionality Reduction via Sequential Semidefinite Programming

Chunhua Shen [a,b,*], Hongdong Li [a,b], Michael J. Brooks [c]

[a] *NICTA, Canberra Research Lab, Canberra, ACT 2601, Australia*
[b] *Australian National University, Canberra, ACT 0200, Australia*
[c] *University of Adelaide, Adelaide, SA 5005, Australia*

Abstract

Many dimensionality reduction problems end up with a trace quotient formulation. Since it is difficult to directly solve the trace quotient problem, traditionally the trace quotient cost function is replaced by an approximation such that generalized eigenvalue decomposition can be applied. In contrast, we directly optimize the trace quotient in this work. It is reformulated as a quasi-linear semidefinite optimization problem, which can be solved globally and efficiently using standard off-the-shelf semidefinite programming solvers. Also this optimization strategy allows one to enforce additional constraints (for example, sparseness constraints) on the projection matrix. We apply this optimization framework to a novel dimensionality reduction algorithm. The performance of the proposed algorithm is demonstrated in experiments on several UCI machine learning benchmark examples, USPS handwritten digits as well as ORL and Yale face data.

Key words:
Dimensionality reduction, Semidefinite programming, Linear discriminant analysis

1 Introduction

In pattern recognition and computer vision, techniques for dimensionality reduction have been extensively studied and utilized. Many of the dimensionality

* Corresponding author. Tel.: +61 2 6267 6282.
Email addresses: chunhua.shen@nicta.com.au (Chunhua Shen), hongdong.li@anu.edu.au (Hongdong Li), michael.brooks@adelaide.edu.au (Michael J. Brooks).

reduction methods, such as linear discriminant analysis (LDA) and its kernel version, end up with solving a trace quotient problem

$$W^\circ = \underset{W^\top W = \mathbf{I}_{d\times d}}{\operatorname{argmax}} \frac{\mathbf{Tr}(W^\top S_b W)}{\mathbf{Tr}(W^\top S_v W)}, \tag{1}$$

where S_b, S_v are two positive semidefinite (p.s.d.) matrices ($S_b \succcurlyeq 0, S_v \succcurlyeq 0$); $\mathbf{I}_{d\times d}$ is the $d \times d$ identity matrix (sometimes the dimension of $\mathbf{I}$ is omitted when it can be inferred from the context) and $\mathbf{Tr}(\cdot)$ denotes the matrix trace. $W \in \mathbb{R}^{D\times d}$ is the target projection matrix for dimensionality reduction (typically $d \ll D$). In the supervised learning framework, S_b usually encodes the distance information between different classes, while S_v encodes the distance information between data points in the same class. In the case of LDA, S_b is the inter-class scatter matrix and S_v is the intra-class scatter matrix. By formulating the problem of dimensionality reduction in a general setting and constructing S_b and S_v in different ways, we can analyze many different types of data in the above mathematical framework.

Despite the importance of the trace quotient problem, to date it lacks a direct and globally optimal solution. Usually, as an approximation, the quotient trace cost $\mathbf{Tr}((W^\top S_v W)^{-1}(W^\top S_b W))$ is instead used such that generalized eigenvalue decomposition (GEVD) can be applied and a close-form solution is readily available. It is easy to check that when $\mathbf{rank}(W) = 1$, i.e., W is a vector, then Equation (1) is actually a Rayleigh quotient problem, and can be solved by GEVD. The eigenvector corresponding to the eigenvalue of largest magnitude gives the optimal W°. Unfortunately, when $\mathbf{rank}(W) > 1$, the problem becomes much more complicated. Heuristically, the dominant eigenvectors corresponding to the largest eigenvalues are used to form the optimal W°. It is believed that the largest eigenvalue contains more useful information. Nevertheless such a GEVD approach cannot produce an optimal solution to the original optimization problem (1) [1]. Furthermore, the GEVD approach does not yield an orthogonal projection matrix. It is shown in [2,3] that orthogonal basis functions preserve the metric structure of the data better and they have more discriminating power. Orthogonal LDA (OLDA) is proposed to compute a set of orthogonal discriminant vectors via the simultaneous diagonalization of the scatter matrices [4]. In [5] it is shown that solely optimizing the Fisher criterion does not necessarily yield optimal discriminant vectors. It is better to include correlation constraints into optimization. The features produced by the classical LDA could be highly correlated (because they are not orthogonal), leading to high redundancy of information.

Recently semidefinite programming (SDP) (or more generally convex programming [6,7]) has attracted much attention in machine learning due to its flexibility and desirable global optimality [8–10]. Moreover, there exist interior-point algorithms to efficiently solve SDPs in polynomial time.

In this paper, we proffer a novel SDP based method for solving the trace quotient problem *directly*. It has the following appealing properties:

- The low target dimension is selected by the user and the algorithm guarantees a globally optimal solution using fractional programming. In other words, it is local-optima-free. Moreover, the fractional programming can be efficiently solved by a sequence of SDPs;
- The projection matrix is intrinsically orthonormal;
- Unlike the GEVD approach to LDA, using our proposed algorithm, the data are not restricted to be projected onto at most $c-1$ dimensions. (Here c is the number of classes.)

To our knowledge, this is the first attempt that directly solves the trace quotient problem, with a global optimum deterministically guaranteed. Methods are also proposed for designing appropriate S_b and S_v. The traditional LDA is only optimal when all the classes follow single Gaussian distributions that share the same covariance matrix. Our new S_b and S_v are not restricted by this assumption.

The remaining content is organized as follows. In Section 2, we describe our algorithm in detail. Section 3 applies this optimization framework to dimensionality reduction. In Section 4, we briefly review relevant work in the literature. The experimental results are presented in Section 5. We discuss new extensions in Section 6. Finally concluding remarks are discussed in Section 7.

2 Solving the Trace Quotient Problem Using SDP

In this section, we show how the trace quotient problem is reformulated into an SDP problem.

2.1 SDP formulation

By introducing an auxiliary variable δ, problem (1) is equivalent to

$$\begin{aligned} \underset{\delta,W}{\text{maximize}} \quad & \delta && \text{(2a)} \\ \text{subject to} \quad & \mathbf{Tr}(W^\top S_b W) \geq \delta \cdot \mathbf{Tr}(W^\top S_v W) && \text{(2b)} \\ & W^\top W = \mathbf{I}_{d\times d} && \text{(2c)} \\ & W \in \mathbb{R}^{D\times d}. && \text{(2d)} \end{aligned}$$

The variables we want to optimize here are δ and W. But we are only interested in W which maximizes δ. This problem is clearly not convex because

constraint (2b) is not convex, and in addition (2d) is actually a non-convex rank constraint. (2c) is quadratic in W. It is obvious that δ must be positive.

Let us define a new variable $Z \in \mathbb{R}^{D\times D}, Z = WW^\top$, constraint (2b) is then converted to $\mathbf{Tr}((S_b - \delta S_v)Z) \geq 0$ since $\mathbf{Tr}(W^\top SW) = \mathbf{Tr}(SWW^\top) = \mathbf{Tr}(SZ)$. Because Z is a matrix product of W and its transpose, it must be p.s.d. In terms of Z, cost function (1) is a linear fraction, therefore it is quasi-convex (more precisely, it is also quasi-concave, hence quasi-linear [6]). The standard technique for solving quasi-concave maximization (or quasi-convex minimization) problems is bisection search which involves solving a sequence of SDPs for our problem. The following theorem due to [11] serves as a basis for converting the non-convex constraint (2d) into a linear one.

Theorem 2.1. *Define sets* $\Omega_1 = \{WW^\top : W^\top W = \mathbf{I}_{d\times d}\}$ *and* $\Omega_2 = \{Z : Z = Z^\top, \mathbf{Tr}\,(Z) = d, 0 \preccurlyeq Z \preccurlyeq \mathbf{I}\}$. *Then* Ω_1 *is the set of extreme points of* Ω_2.

See [11] for the proof. Theorem 2.1 states that, in terms of constraint, Ω_1 is more strict than Ω_2. Therefore constraints (2c) and (2d) can be relaxed into $\mathbf{Tr}(Z) = d$ and $0 \preccurlyeq Z \preccurlyeq \mathbf{I}$, which are both convex. When the cost function is linear and it is subject to Ω_2, the solution will be at one of the extreme points [12]. Consequently, for linear cost functions, the optimization problems subject to Ω_1 and Ω_2 are exactly equivalent.

With respect to Z and δ, (2b) is still non-convex: the problem may have locally optimal points. But still the global optimum can be efficiently computed via a sequence of convex feasibility problems. By observing that the constraint is linear if δ is known, we can convert the optimization problem into a set of convex feasibility problems. A bisection search strategy is adopted to find the optimal δ. This technique is widely used in fractional programming [6,13]. Let δ° denote the unknown optimal value of the cost function. Given $\delta^* \in \mathbb{R}$, if the convex feasibility problem [1]

$$\begin{aligned} \text{find} \quad & Z && \text{(3a)} \\ \text{subject to} \quad & \mathbf{Tr}((S_b - \delta^* S_v)Z) \geq 0 && \text{(3b)} \\ & \mathbf{Tr}(Z) = d && \text{(3c)} \\ & 0 \preccurlyeq Z \preccurlyeq \mathbf{I} && \text{(3d)} \end{aligned}$$

is feasible, then we infer $\delta^\circ \geq \delta^*$. Otherwise, if the above problem is infeasible, then we infer $\delta^\circ < \delta^*$. Hence we can check whether the optimal value δ° is smaller or larger than a given value δ^*. This observation motivates a simple algorithm for solving the fractional optimization problems using bisection search, which solves an SDP feasibility problem at each step. Algorithm 1 shows how it works.

[1] A feasibility problem has no cost function. The objective is to check whether the intersection of the convex constraints is empty.

Algorithm 1 Bisection search.

Require: δ_l is lower bound of δ; δ_u is upper bound of δ and the tolerance $\sigma > 0$.

```
while δ_u − δ_l > σ do
  δ = (δ_l + δ_u)/2.
  Solve the convex feasibility problem described in (3a)–(3d).
  if feasible then
    δ_l = δ;
  else
    δ_u = δ.
  end if
end while
```

Thus far, a question remains unanswered: are constraints (3c) and (3d) equivalent to constraints (2c) and (2d) for the feasibility problem? Essentially the feasibility problem is equivalent to

$$\begin{aligned} \text{maximize} \quad & \mathbf{Tr}((S_b - \delta^* S_v)Z) && \text{(4a)} \\ \text{subject to} \quad & \mathbf{Tr}(Z) = d && \text{(4b)} \\ & 0 \preccurlyeq Z \preccurlyeq \mathbf{I}. && \text{(4c)} \end{aligned}$$

If the maximum value of the cost function is non-negative, then the feasibility problem is feasible. If the converse condition applies, it is infeasible. Because this cost function is linear, we know that Ω_1 can be replaced by Ω_2, i.e., constraints (3c) and (3d) are equivalent to (2c) and (2d) for the optimization problem.

Note that constraint (3d) is not in the standard form of SDP. It can be rewritten into the standard form as

$$\begin{aligned} \begin{bmatrix} Z & \mathbf{0} \\ \mathbf{0} & Q \end{bmatrix} \succcurlyeq 0, && \text{(5a)} \\ Z + Q = \mathbf{I}, && \text{(5b)} \end{aligned}$$

where the matrix Q acts as a slack variable. Now the problem can be solved using standard SDP packages such as CSDP [14] and SeDuMi [15]. We use CSDP in all of our experiments.

2.2 Estimating bounds of δ

The bisection search procedure requires a lower bound and an upper bound of δ. The following theorem from [11] is useful for estimating the bounds.

Theorem 2.2. *Let $S \in \mathbb{R}^{D\times D}$ be a symmetric matrix, and $\varphi_{S,1} \geq \varphi_{S,2} \geq \cdots \geq \varphi_{S,D}$ be the sorted eigenvalues of S from largest to smallest, then* $\max_{W^\top W=\mathbf{I}_{d\times d}} \mathbf{Tr}(W^\top SW) = \sum_{i=1}^{d} \varphi_{S,i}$.

Refer to [11] for the proof. This theorem can be extended to obtain the following corollary (following the proof for Theorem 2.2):

Corollary 2.1. *Let $S \in \mathbb{R}^{D\times D}$ be a symmetric matrix, and $\psi_{S,1} \leq \psi_{S,2} \leq \cdots \leq \psi_{S,D}$ be its sorted eigenvalues from smallest to largest, then*

$$\min_{W^\top W=\mathbf{I}_{d\times d}} \mathbf{Tr}(W^\top SW) = \sum_{i=1}^{d} \psi_{S,i}.$$

Therefore, we estimate the upper bound of δ:

$$\delta_u = \frac{\sum_{i=1}^{d} \varphi_{S_b,i}}{\sum_{i=1}^{d} \psi_{S_v,i}}. \tag{6}$$

In the trace quotient problem, both S_b and S_v are p.s.d. That is to say, all of their eigenvalues are non-negative. Be aware that the denominator of (6) could be zeros and $\delta_u = +\infty$. This occurs when the d smallest eigenvalues of S_v are all zeros. In this case, $\mathbf{rank}(S_v) \leq D - d$. In the case of LDA, $\mathbf{rank}(S_v) = \min(D, N)$. Here N is the number of training data. When $N \leq D - d$, which is termed the *small sample problem*, δ_u is invalid.

A principle component analysis (PCA) preprocessing can always be performed to remove the null space of the covariance matrix of the data, such that δ_u becomes valid.

A lower bound on δ is then given by

$$\delta_l = \frac{\sum_{i=1}^{d} \psi_{S_b,i}}{\sum_{i=1}^{d} \varphi_{S_v,i}}. \tag{7}$$

Clearly $\delta_l \geq 0$.

The bisection algorithm converges in $\lceil \log_2(\frac{\delta_u - \delta_l}{\sigma}) \rceil$ iterations, and obtains the global minimum within the predefined accuracy of σ. The bisection procedure is intuitive to understand. Next we describe another algorithm—Dinkelbach's algorithm—which is less intuitive but faster, for fractional programming.

Algorithm 2 Dinkelbach algorithm.

Require: An initialization $Z^{(0)}$ which satisfies constraints (4b) and (4c).

Set

$$\delta = \frac{\mathbf{Tr}(S_b Z^{(0)})}{\mathbf{Tr}(S_v Z^{(0)})}$$

and $k = 0$.

($\star$) $k = k + 1$. Solve the SDP (8) subject to constraints (4b) and (4c) to get the optimal $Z^{(k)}$, given δ.

if $\mathbf{Tr}((S_b - \delta S_v)Z^{(k)}) = 0$ **then**

stop and the optimal $Z^{\circ} = Z^{(k)}$.

else

Set

$$\delta = \frac{\mathbf{Tr}(S_b Z^{(k)})}{\mathbf{Tr}(S_v Z^{(k)})}$$

and go to step ($\star$).

end if

2.3 Dinkelbach's algorithm

Dinkelbach's algorithm [16] proposes an iterative procedure for solving the fractional program $\text{maximize}_{\boldsymbol{x}}\, f(\boldsymbol{x})/g(\boldsymbol{x})$, where $\boldsymbol{x}$ is constrained on a convex set, $f(\boldsymbol{x})$ is concave, and $g(\boldsymbol{x})$ is convex. It considers the parametric problem $\text{maximize}_{\boldsymbol{x}}\, f(\boldsymbol{x}) - \delta g(\boldsymbol{x})$ where δ is a constant. Here we need to solve

$$\text{maximize}_Z\, \mathbf{Tr}((S_b - \delta S_v)Z). \tag{8}$$

The algorithm generates a sequence of values of δ's that converge to the global optimum function value. The bisection search converges linearly while the Dinkelbach's algorithm converges super-linearly (better than linearly and worse than quadratically).

Dinkelbach's iterative algorithm for our trace quotient problem is described in Algorithm 2. We omit the convergence analysis of the algorithm which can be found in [16]. In Algorithm 2, note that: (1) A test of the form $\mathbf{Tr}((S_b - \delta S_v)Z^{(k)}) > 0$ is unnecessary since for any fixed k, $\mathbf{Tr}((S_b - \delta S_v)Z^{(k)}) = \max_Z \mathbf{Tr}((S_b - \delta S_v)Z) \geq \mathbf{Tr}((S_b - \delta S_v)Z^{(k-1)}) = 0$. However due to numerical accuracy limits, in implementation it is possible that the value of $\mathbf{Tr}((S_b - \delta S_v)Z^{(k)})$ is negative and is very close to zero; (2) To find the initialization $Z^{(0)}$ that must reside in the set defined by the constraints, in general, one might solve a feasibility problem. In our case, it is easy to show that a square matrix $Z^{(0)} \in \mathbb{R}^{D \times D}$ with d diagonal entries being 1 and all the other entries being 0 satisfies (4b) and (4c). This initialization is used in our experiments and it works well. Dinkelbach's algorithm needs no parameters and it converges faster. In contrast, bisection requires estimation of the bounds for the cost function.

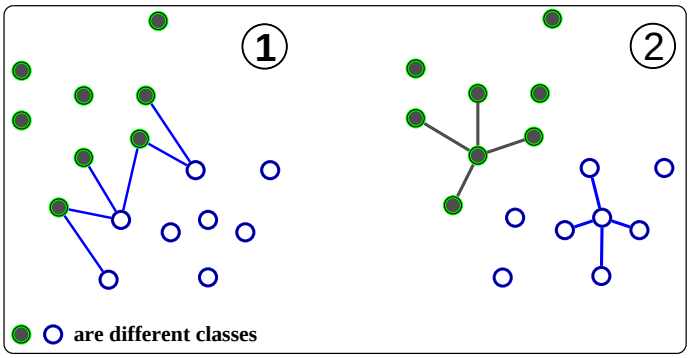

Fig. 1. The connected edges in (1) define the dissimilarity set $\mathcal{D}$ and the connections in (2) define the similarity set $\mathcal{S}$. As shown in (1), the inter-class marginal samples are connected while in (2), each sample is connected to its k' nearest neighbors in the same class. For clarity only a few connections are shown.

2.4 Computing W from Z

From the covariance matrix Z learned by SDP, we can calculate the projection matrix W by eigen-decomposition. Let V_i denote the i^{th} eigenvector, with eigenvalue λ_i. Let $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_D$ be the sorted eigenvalues. It is straightforward to see that $W = \mathbf{diag}(\sqrt{\lambda_1}, \sqrt{\lambda_2}, \cdots, \sqrt{\lambda_D})V^\top$, where $\mathbf{diag}(\cdot)$ is a square matrix with the input as its diagonal elements. To obtain a $D \times d$ projection matrix, the smallest $D - d$ eigenvalues are simply truncated. The projection matrix obtained in this way is not the same as the projection corresponding to maximizing the cost function subject to a rank constraint. However this approach is a reasonable approximation. Moreover, like PCA, dropping the eigenvectors corresponding to small eigenvalues may de-noise the input data, which is desirable in some cases.

This is the general treatment for recovering a low dimensional projection from a covariance matrix. In our case, this procedure is precise. This is obvious: λ_i, the eigenvalues of $Z = WW^\top$, are the same as the eigenvalues of $W^\top W = \mathbf{I}_{d\times d}$. That means, $\lambda_1 = \lambda_2 = \cdots = \lambda_d = 1$ and the left $D - d$ eigenvalues are all zeros. Hence in our case we can simply stack the first d leading eigenvectors to obtain W.

3 Application to Dimensionality Reduction

There are various strategies to construct the matrices S_b and S_v, which represent the inter-class and intra-class scatter matrices respectively. In general, we have a set of data $\{\boldsymbol{x}_p\}_{p=1}^M \in \mathbb{R}^{D\times M}$ and we are given a *similarity* set $\mathcal{S}$ and a *dissimilarity* set $\mathcal{D}$. Formally, $\{\mathcal{S} : (\boldsymbol{x}_p, \boldsymbol{x}_q) \in \mathcal{S} \quad \text{if } \boldsymbol{x}_p \text{ and } \boldsymbol{x}_q \text{ are similar}\}$ and $\{\mathcal{D} : (\boldsymbol{x}_p, \boldsymbol{x}_q) \in \mathcal{D} \quad \text{if } \boldsymbol{x}_p \text{ and } \boldsymbol{x}_q \text{ are dissimilar}\}$. We want to maximize

the distance

$$\begin{aligned}\sum_{(p,q)\in\mathcal{D}} \mathrm{dist}_W^2(\boldsymbol{x}_p,\boldsymbol{x}_q) &= \sum_{(p,q)\in\mathcal{D}} \|W^\top \boldsymbol{x}_p - W^\top \boldsymbol{x}_q\|^2 \\ &= \sum_{(p,q)\in\mathcal{D}} (\boldsymbol{x}_p-\boldsymbol{x}_q)^\top Z(\boldsymbol{x}_p-\boldsymbol{x}_q) \\ &= \mathbf{Tr}\,(S_b Z),\end{aligned}$$

where $S_b = \sum_{(p,q)\in\mathcal{D}}(\boldsymbol{x}_p-\boldsymbol{x}_q)(\boldsymbol{x}_p-\boldsymbol{x}_q)^\top$. This measures the inter-class distance. We also want to minimize the intra-class compactness distance:

$$\sum_{(p,q)\in\mathcal{S}} \mathrm{dist}_W^2(\boldsymbol{x}_p,\boldsymbol{x}_q) = \mathbf{Tr}\,(S_v Z),$$

where $S_v = \sum_{(p,q)\in\mathcal{S}}(\boldsymbol{x}_p-\boldsymbol{x}_q)(\boldsymbol{x}_p-\boldsymbol{x}_q)^\top$.

Inspired by the marginal fisher analysis (MFA) algorithm proposed in [17], we construct similar graphs for building S_b and S_v. Figure 1 demonstrates the basic idea. For each class, assuming $\boldsymbol{x}_p$ is in this class, and if the pair (p,q) belongs to the k closest pairs that have different labels, then $(p,q)\in\mathcal{D}$. The intra-class set $\mathcal{S}$ is easier: we connect each sample to its k' nearest neighbors in the same class. k and k' are parameters defined by the user.

This strategy avoids certain drawbacks of LDA. We do not force all the pairwise samples in the same class to be close (this might be a too strict requirement). Instead, we are more interested in driving neighboring samples as closely as possible. We do not assume any special distribution on the data. The set $\mathcal{D}$ characterizes the margin information between classes. For non-Gaussian data, it is expected to better represent the separability of different classes than the inter-class covariance of LDA. Therefore we maximize the margins while condensing individual classes simultaneously. For ease of presentation, we refer to this algorithm as SDP_1, whose S_b and S_v are calculated by the above-mentioned strategy.

In [18] a 1-nearest-neighbor margin is defined based on the concept of the nearest neighbor to a point $\boldsymbol{x}$ with the same and different label. Motivated by their work, we can slightly modify MFA's inter-class distance graph. The similarity set $\mathcal{S}$ remains unchanged as described previously. But to create the dissimilarity set $\mathcal{D}$, a simpler way is that, for each $\boldsymbol{x}_p$ we connect it to its k differently-labeled neighbors $\boldsymbol{x}_q$'s ($\boldsymbol{x}_p$ and $\boldsymbol{x}_q$ have different labels). The algorithm that implements this concept is referred as SDP_2. It is difficult to analyze which one is better. Indeed the experiments indicate for different data sets, no single method is consistently better than the other. One may also use support vector machines (SVMs) to find the boundary points and then create $\mathcal{D}$ (and then S_b) based on those boundary points [19].

4 Related Work

The closest work to ours is [1] in the sense that it also proposes a method to solve the trace quotient directly. [1] finds the projection matrix W in the Grassmann manifold. Compared with optimization in the Euclidean space, the main advantage of optimization on the Grassmann manifold is the use of fewer variables. Thus the scale of the problem is smaller. However, there are major differences between [1] and our method: Firstly, [1] optimizes $\mathbf{Tr}(W^\top S_b W - \delta \cdot W^\top S_v W)$ and has no principled way to determine the optimal value of δ. In contrast, we optimize the trace quotient function itself and a deterministic bisection search or the Dinkelbach's iteration guarantees the optimal δ; Secondly, the optimization in [1] is non-convex (difference of two quadratic functions). Therefore it is likely to become trapped into a local maximum, while our method is globally optimal.

[20] simply replaces LDA's cost function with $\mathbf{Tr}(W^\top S_b W - W^\top S_v W)$, i.e., setting $\delta = 1$. Then GEVD is used to obtain the low rank projection matrix. Obviously this optimization is not equivalent to the original problem, although it avoids the matrix inversion problem of LDA.

[21] proposes a convex programming approach to maximize the distances between classes and simultaneously to clip (but not to minimize) the distances within classes. Unlike our method, in their approach the rank constraint is not considered. Hence it is metric learning but not necessarily a dimensionality reduction method. Furthermore, although the formulation of [21] is convex, it is not an SDP. It is more computationally expensive to solve and general-purpose SDP solvers are not applicable. SDP (or general convex programming) is also used in [22,23] for learning a distance metric. [22] learns a metric that shrinks distances of neighboring similarly-labeled points and repels points in different classes by a large margin. [23] also learns a metric using convex programming.

We borrow from [17] the method of constructing the similarity and dissimilarity sets. The MFA algorithm in [17] optimizes a different cost function. It originates from graph embedding. Note that there is a kernel version of MFA. It is straightforward to kernelize our problem since it is still a trace quotient for the kernel version. We leave this topic for future research.

5 Experiments

In all our experiments, the Bisection Algorithm 1 and Dinkelbach's Algorithm 2 output almost identical results but Dinkelbach converges twice as fast as Bisection.

We observe that the direct solution indeed yields a larger trace quotient than that obtained by the quotient trace using GEVD because that is what we maximize.

Data visualization. As an intuitive demonstration, we run the proposed SDP algorithms on an artificial concentric circles data set [24], which consists of four classes (shown in different colors). The first two dimensions follow concentric circles while the left eight dimensions are all Gaussian noise. When the scale of the noise is large, PCA is distracted by the noise. LDA also fails because the data set is not linearly separable and each class' center overlaps in the same point. Both of our algorithms find the informative features (Figure 2-(3)(4)). Ideally we should optimize the projected neighborhood relationship as in [24]. Unfortunately it is difficult. [24] utilizes softmax nearest neighbors to model the neighborhood relationships before the projection is known. However the cost function is non-convex. As an approximation, one usually calculates the neighborhood relationships in the input space. Laplacian eigenmap [25] is an example. When the noise is large enough, the neighborhood obtained in this way may not *faithfully* represent the true data structure. We deliberately set the noise of the concentric data set very large, which breaks our algorithms (Figure 2-(5)(6)). Nevertheless useful prior information can be used to define a meaningful $\mathcal{D}$ and $\mathcal{S}$ whenever it is available. As an example, when we use the sets $\mathcal{D}$ and $\mathcal{S}$ of Figure 2-(3)(4) and then calculate S_b and S_v with the highly noisy data, our algorithms are still able to find the first two useful dimensions perfectly, as shown in Figure 2-(7)(8).

Classification. In the first classification experiment, we evaluate our algorithm on different data sets and compare it with PCA, LDA and large margin nearest neighbor classifier (LMNN) [2]. Note that our algorithm is much faster than [22] especially when the number of training data is large. That is because the complexity of our algorithm is independent of the number of data while in [22] more data produce more SDP constraints that slow down the SDP solver. A description of the data sets is in Table 1.

PCA is used to reduce the dimensionality of image data (USPS handwritten digits [3] and ORL face data [4]) as a preprocessing procedure for accelerating the computation. For most data sets the results are reported over 50 random 70/30 splits of the data. USPS has a predefined training and testing sets.

In the experiments, we did not carefully tune the parameters (k, k') associated with our proposed SDP approaches due to computational burden. However, we find that the parameters are not sensitive in a wide range. They can be op-

[2] The codes are obtained from the authors' website `http://www.weinbergerweb.net/Downloads/LMNN.html`

[3] `http://www.gaussianprocess.org/gpml/data/`

[4] `http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html`

	Iris	Wine	Bal	USPS1	USPS2	ORL1	ORL2
train samples	105	125	438	1459	2394	280	200
test samples	45	53	187	5832	628	120	200
classes	3	3	3	10	3	40	40
input dimensions	4	13	4	256	256	644	644
dimensions after PCA	4	13	4	55	50	42	42
parameters (k, k') for SDP_1	(100,5)	(50,3)	(220,3)	(300,3)	(300,3)	(300,3)	(200,3)
parameters (k, k') for SDP_2	(3,3)	(1,5)	(3,5)	(3,5)	(1,5)	(2,3)	(2,3)
runs	50	50	50	10	1	50	50

Table 1. Description of data sets and experimental parameters k and k'.

timally determined by cross-validation. We report a 3-NN (nearest neighbor) classifier's testing error. The result is shown in Table 2, where the baseline is obtained by directly applying 3-NN classification on the original data. We have chosen one of the simplest classifiers, k-NN, for benchmarking. Clearly, the best choice of k depends upon the data. Generally, larger values of k reduce the effect of noise on the classification, but make boundaries between classes less distinct. Thus far there is no a elegant method to determine the optimal k. Typically cross-validation is used. In most cases, one sets k to 1 or 3 for simplicity. We have set $k = 3$ in all the experiments. But the results presented here also hold for $k = 1$. Next we present details of tests.

UCI data sets: Iris, Wine and Bal. These are small data sets with only 3 classes, and are from the UCI machine learning repository [26]. Except from the Wine data, which are well separated and LDA performs best, our SDP algorithms present competitive results on the other two data sets.

USPS digit recognition. Two tests are conducted on the USPS handwriting digit data set (referred to as USPS1 and USPS2). In the first test, we use all the 10 digits. USPS has predefined training and testing subsets. The training subset has 7291 digits. We randomly split the training subset: 20% for training and the remaining 80% for testing. The dimensionality of these 16×16 images are reduced to 55D by PCA. 90.14% of the variance is preserved. LMNN gives the best result with an error rate of 4.22%. Our SDPs have similar performance. For the second test, it is only run once with the predefined training subset and test subset. The digits $1, 2$ and 3 are used. On this data set, our two SDPs deliver lowest test errors. It is worth noting that LDA performs even worse than PCA. This is likely due to the data's non-Gaussian distribution.

ORL face recognition. This data set consists of 400 faces of 40 individuals: 10 faces per individual. The image size is 56×46. We down-sample them by a factor of 2. Then PCA is applied to obtain 42D eigenfaces, which captures about 81% of the energy. Again two tests are conducted on this set. The training and testing sets are obtained by 7/3 and 5/5 sampling for each person respectively (referred to as dataset ORL1 and ORL2). In both tests, LMNN performs best, and SDP_2 is the second best. Also note that for each method, its performance on ORL1 is better than its corresponding result on ORL2. This is expected since ORL1 contains more training examples.

For all the tests, our algorithms are consistently better than PCA and LDA. The state-of-the-art LMNN outperforms ours on tasks with many classes such as USPS1, ORL1 and ORL2. This might be due to the fact that, inspired by SVM, LMNN enforces constraints on each training point. These constraints ensure that the learned metric correctly classifies as many training points as possible. The price is that LMNN's SDP optimization problem involves many

constraints. With a large amount of training data, the required computational demand could be prohibitive. Therefore as SVM, it is difficult to scale it to large size problems. In contrast, our SDP formulation is independent of the amount of training data. The complexity is entirely determined by the dimension of the input data.

Because we have observed that for the data sets with few classes, our SDP approaches usually are better than LMNN, we now verify this observation empirically. We run SDP_2 and LMNN on the data set ORL2. We vary the number of classes c from 5 to 32. The first c individuals' images are used. The parameters of SDP_2 remain unchanged: $k = 2$ and $k' = 3$. For each value of c, the experiment is run 10 times. We report the classification result in Table 3. This result confirms that our SDPs perform well for tasks with few classes. It also explains why LMNN outperforms our SDPs for data sets having many classes. It might also be possible to include constraints as LMNN does in our SDP formulation.

The second classification experiment we have conducted is to compare our methods with two LDA's variations, namely, uncorrelated linear discriminant analysis (ULDA) [27] and orthogonal linear discriminant analysis (OLDA) [4]. ULDA was proposed for extracting feature vectors with uncorrelated attributes. The crucial property of OLDA is that the discriminant vectors of OLDA are orthogonal to each other (In other words, the transformation matrix of OLDA is orthogonal).

The Yale face database [5] is used here. The Yale database contains 165 gray-scale images of 15 individuals. There are 11 images per subject. The images demonstrate variations in lighting condition, facial expression (normal, happy, sad, sleepy, surprised, and wink). The face images are manually aligned and cropped into 32×32 pixels, with 256 gray levels per pixel. The 11 faces for each individual is randomly split into training and testing sets by 4/7, 5/6 and 6/5 sampling. PCA is performed to reduce 1024D into 50D, which contains above 98% of the total energy.

An important parameter for most subspace learning based face recognition methods is dimensionality estimation. Usually the classification accuracy varies in the number of dimensions. Cross validation is often needed to estimate the best dimensionality. We simply set the dimensionality to $c - 1$, where c is the number of classes. That means, on the Yale dataset, the final dimension for all algorithms is 14. As in the first experiment, we also fix the parameters of SDP_2 to be $k = 2$ and $k' = 3$.

Table 4 summarizes the classification results. We see that ULDA performs similarly with the traditional LDA. OLDA achieves higher accuracies than ULDA

[5] `http://cvc.yale.edu/projects/yalefaces/yalefaces.html`

	Iris	Wine	Bal	USPS1	USPS2	ORL1	ORL2
baseline	4.57(2.50), 4	29.48(5.48), 13	18.10(2.02), 4	6.36(0.21), 256	2.39, 256	-	-
PCA	4.80(2.51), 2	29.28(5.53), 8	12.46(2.06), 3	5.60(0.27), 55	2.07, 50	5.11(1.91), 42	8.68(2.01), 42
LDA	4.23(2.56), 2	**1.52**(1.64), 2	13.26(2.66), 3	7.43(0.42), 9	3.03, 2	4.85(1.92), 39	7.63(1.93), 39
LMNN	4.40(2.75), 4	**4.04**(2.02), 13	**12.17**(2.37), 3	**4.22**(0.35), 55	1.91, 50	**2.23**(1.39), 42	**4.69**(1.98), 42
SDP_1	**3.02**(2.19), 3	4.83(3.05), 8	13.38(2.05), 3	5.28(0.25), 50	**1.75**, 40	3.47(1.56), 39	6.64(2.21), 39
SDP_2	**3.60**(2.61), 3	12.83(5.63), 8	**9.70**(1.99), 3	**4.73**(0.10), 50	**1.91**, 40	**3.32**(1.24), 39	**5.87**(1.98), 39

Table 2. Classification error of a 3-NN classifier on each data set (except USPS2) in the format of **mean(std)**%. The final dimension for each method is shown after the error. The best and second best results are shown in bold. Here LMNN means the large margin nearest neighbor classifier in [22]; SDP_1 and SDP_2 are our proposed methods using semidefinite programming.

# of classes	5	8	10	14	18	25	32
LMNN	0(0)	5.00(3.33)	3.80(1.99)	1.57(1.60)	3.49(2.08)	**2.34**(1.38)	**3.12**(2.05)
SDP_2	0(0)	**0.13**(0.56)	**1.20**(1.20)	**1.29**(1.05)	**3.11**(1.72)	3.20(1.77)	4.63(1.36)

Table 3. Classification error of a 3-NN classifier v.s. number of classes on the Face data (ORL2). Classification error is in the format of **mean(std)**%. Our SDP algorithm has increasing error when the number of classes increases. This might be because we optimize a cost function based on global distances summation (the trace calculation). For LMNN it is not the case. Clearly on data sets with few classes, SDP is better than LMNN. Here LMNN means the large margin nearest neighbor classifier in [22] and SDP_2 is one of our proposed methods using semidefinite programming.

	4 Train	5 Train	6 Train
baseline	47.76(4.18)	44.40(3.61)	40.87(5.12)
LDA	39.38(8.34)	24.17(2.88)	21.40(3.07)
ULDA	29.14(5.17)	25.61(2.85)	22.80(4.12)
OLDA	27.57(5.55)	24.61(3.35)	**20.53**(3.33)
SDP_2	**27.05**(5.65)	**23.17**(3.31)	20.73(2.85)

Table 4
Classification error of a 3-NN classifier on the Yale face database in the format of **mean(std)**%. Each case is run 20 times to calculate the mean and standard deviation. SDP_2 performs slightly better than OLDA. Here ULDA means uncorrelated linear discriminant analysis [27]; OLDA means orthogonal linear discriminant analysis [4]; SDP_2 is one of the proposed methods using semidefinite programming.

and LDA. The proposed SDP algorithm is slightly better than OLDA. Since both OLDA and the proposed SDP algorithm produces orthogonal transformation matrix, we may conclude that orthogonality does benefit for subspace based face recognition.

As mentioned, for the LDA algorithm and its variations, the data are restricted to be mapped to at most $c-1$ dimensions. Our SDP algorithms do not have this restriction. We have compared the final classification results on Yale when the final dimensionality varies using the SDP_2 algorithm in Table 5. It can be observed that $c-1$ is not the best dimensionality for SDP_2 in this case.

final dimensions	14	20	24	30
SDP_2	27.05(5.65)	26.43(5.14)	26.86(4.18)	28.62(5.27)

Table 5
Classification error of a 3-NN classifier on the Yale face database with 4 training examples. Each case is run 20 times.

A disadvantage of the proposed SDP algorithm is that it is computationally more expensive than spectral methods. In the above experiment, the Dinkelbach algorithm needs around 80 seconds to converge. In contrast, LDA, ULDA and OLDA need about 2 seconds. [6]

6 Extension: Explicitly Controlling Sparseness of W

In this section, we show that with the flexible optimization framework, it is straightforward to enforce additional constraints on the projection matrix. We

[6] The computation environment is: Matlab 7.4 on a desktop with a P4 3.4GHz CPU and 1G memory. The SDP solver used is CSDP 6.0.1.

consider the sparseness constraints here.

Sparseness builds one type of feature selection mechanism. It has many applications in pattern analysis and image processing [28–31]. Mathematically, we want the projection matrix W to be sparse. That is, $\mathbf{Card}(W) < \Theta$ ($0 < \Theta \leq Dd$). Here Θ is a predefined parameter. $\mathbf{Card}(W)$ denotes the cardinality of the matrix W, i.e., the number of non-zero entries in the matrix W. Since $Z = WW^\top$, we rewrite $\mathbf{Card}(W) \leq \Theta$ as $\mathbf{Card}(Z) \leq \Theta^2$. The discrete non-convex cardinality constraint can be relaxed into a weaker convex one using the technique discussed in [31].

For any $u \in \mathbb{R}^D$, $\mathbf{Card}(u) = \Theta$ means the following inequality holds: $\|u\|_1 \leq \sqrt{\Theta}\,\|u\|_2$. We can then replace the non-convex constraint $\mathbf{Card}(Z) \leq \Theta^2$ by a convex constraint: $\|Z\|_1 \leq \Theta\,\|Z\|_{\mathrm{F}}$, where $\|\cdot\|_{\mathrm{F}}$ denotes the Frobenius norm. Since $\|Z\|_{\mathrm{F}} = \left\|WW^\top\right\|_{\mathrm{F}} = \left\|W^\top W\right\|_{\mathrm{F}} = \|\mathbf{I}_{d\times d}\|_{\mathrm{F}} = \sqrt{d}$, the sparseness constraint now becomes convex (it is easy to rewrite it into a sequence of linear constraints):

$$\|Z\|_1 \leq \Theta\sqrt{d}. \tag{9}$$

By inserting the constraint (9) into Algorithm 1 or 2, we obtain a sparse projection. Note that (9) is a convex constraint,[7] which can be viewed as a convex lower bound on the function $\mathbf{Card}(Z)$. It can be decomposed into $O(D^2)$ linear constraints. For a large D, the memory requirements of Newton's method could be prohibitive.

We first run a simple experiment on artificial data to show how the sparseness of the projection matrix W changes as the value of $\Theta\sqrt{d}$ varies. For simplicity, we set $d = 1$; i.e., W is a 1D vector. We randomly generate the matrices S_b and S_v in this way: $S = U^\top U + 16w^\top w$. Here S denotes in turn S_b and S_v but with $S_b \neq S_v$. $U \in \mathbb{R}^{10\times 10}$ is a random matrix with all its elements following a uniform distribution in $[0, 1]$ and

$$w = [1, 0, 1, 0, 1, 0, 1, 0, 1, 0].$$

We sample 40 different pairs of matrices S_b and S_v. We then input S_b and S_v into the Dinkelbach algorithm with the additional sparseness constraint (9). For each Θ between 1 and 10, we solve the SDP. W is extracted by computing the first eigenvector of Z. The cardinality of W as a function of Θ is illustrated

[7] There is a standard trick from mathematical programming for expressing the ℓ_1-norm as a linear function. By decomposing the variable $Z = Z_+ - Z_-$ into positive and negative parts respectively, (9) is written into $\mathbf{1}^\top(Z_+ + Z_-)\mathbf{1} \leq \Theta\sqrt{d}$ and $Z_+ \geq 0$, $Z_- \geq 0$ (element-wise non-negative). Here $\mathbf{1}$ is a column vector with all elements being ones.

in Figure 3 [8]. We can see that Θ is indeed a good indicator of the cardinality. Note that when $\Theta = 1$, one always gets a W with a single element being one and all others being zeros in this example. We also plot an example of the obtained W with $\Theta = 3$, and W being without sparseness constraints for an intuitive comparison in Figure 4.

The second experiment is conducted on the Wine data described in Table 1. S_b and S_v are constructed using SDP_1 using the same parameters shown in Table 1. The final projected dimension is 8. We want each column of W to be sparse. In other words, only a subset of features is selected. We compare our performance against the simple thresholding method [32]. Table 6 reports the classification error. As expected, the proposed algorithm performs better than the simple thresholding method.

cardinality	4	5	6
SDP with sparseness	6.98(3.63)	5.47(2.44)	5.09(2.74)
simple thresholding	7.55(2.52)	7.55(3.72)	8.02(3.35)

Table 6
Classification error of a 3-NN classifier on the Wine dataset w.r.t. the cardinality of each row of W. Each case is run 20 times.

7 Conclusion

In this work we have presented a new supervised dimensionality reduction algorithm. It has two key components: a global optimization strategy for solving the trace quotient problem, and a new trace quotient cost function specifically designed for linear dimensionality reduction. The proposed algorithms are consistently better than LDA. Experiments show that our algorithm's performance is comparable to the LMNN algorithm but with computational advantages. Future work will be focused on the following directions. First, we have confined ourself to linear dimensionality reduction in this paper. We will explore the extension to the kernel versions. We already know that some nonlinear dimensionality reduction algorithms like kernel LDA also need to solve trace quotient problems. Second, new strategies will be devised to define an optimal discriminative set $\mathcal{D}$. [19] might prove a useful inspiration. Third, SDP's computational complexity is heavy. New efficient methods are required to scale up to large-size problems.

[8] For calculating cardinality, an element is regarded as non-zero if its absolute magnitude is larger than 10% of the vector's maximum absolute magnitude.

Acknowledgments

NICTA is funded through the Australian Government's *Backing Australia's Ability* initiative, in part through the Australian Research Council.

References

[1] S. Yan, X. Tang, Trace quotient problems revisited, in: Proc. Eur. Conf. Comp. Vis., Springer-Verlag, 2006, pp. 232–244.

[2] D. Cai, X. He, J. Han, H.-J. Zhang, Orthogonal Laplacianfaces for face recognition, IEEE Trans. Image Process. 15 (11) (2006) 3608–3614.

[3] G. Hua, P. A. Viola, S. M. Drucker, Face recognition using discriminatively trained orthogonal rank one tensor projections, in: Proc. IEEE Conf. Comp. Vis. Patt. Recogn., Minneapolis, MN, 2007.

[4] J. Ye, T. Xiong, Null space versus orthogonal linear discriminant analysis, in: Proc. Int. Conf. Mach. Learn., Pittsburgh, Pennsylvania, 2006, pp. 1073–1080.

[5] J. Yang, J.-Y. Yang, D. Zhang, What's wrong with fisher criterion?, Pattern Recogn. 35 (11) (2002) 2665–2668.

[6] S. Boyd, L. Vandenberghe, Convex Optimization, Cambridge University Press, 2004.

[7] A. Ben-Tal, A. S. Nemirovski, Lectures on modern convex optimization: analysis, algorithms, and engineering applications, SIAM, Philadelphia, PA, USA, 2001.

[8] K. Q. Weinberger, L. K. Saul, Unsupervised learning of image manifolds by semidefinite programming, Int. J. Comp. Vis. 70 (1) (2006) 77–90.

[9] J. Keuchel, C. Schnörr, C. Schellewald, D. Cremers, Binary partitioning, perceptual grouping, and restoration with semidefinite programming, IEEE Trans. Pattern Anal. Mach. Intell. 25 (11) (2003) 1364–1379.

[10] G. R. G. Lanckriet, N. Cristianini, P. Bartlett, L. E. Ghaoui, M. I. Jordan, Learning the kernel matrix with semidefinite programming, J. Mach. Learn. Res. 5 (2004) 27–72.

[11] M. L. Overton, R. S. Womersley, On the sum of the largest eigenvalues of a symmetric matrix, SIAM J. Matrix Anal. Appl. 13 (1) (1992) 41–45.

[12] M. L. Overton, R. S. Womersley, Optimality conditions and duality theory for minimizing sums of the largest eigenvalues of symmetric matrices, Math. Program. 62 (1-3) (1993) 321–357.

[13] S. Agarwal, M. Chandraker, F. Kahl, S. Belongie, D. J. Kriegman, Practical global optimization for multiview geometry, in: Proc. Eur. Conf. Comp. Vis., Vol. 1, Graz, Austria, 2006, pp. 592–605.

[14] B. Borchers, CSDP, a C library for semidefinite programming, Optim. Methods and Softw. 11 (1) (1999) 613–623.

[15] J. F. Sturm, Using SeDuMi 1.02, a matlab toolbox for optimization over symmetric cones (updated for version 1.05), Optim. Methods and Softw. 11-12 (1999) 625–653.

[16] S. Schaible, Fractional programming. II on Dinkelbach's algorithm, Management Sci. 22 (8) (1976) 868–873.

[17] S. Yan, D. Xu, B. Zhang, H. Zhang, Q. Yang, S. Lin, Graph embedding and extensions: A general framework for dimensionality reduction, IEEE Trans. Pattern Anal. Mach. Intell. 29 (1) (2007) 40–51.

[18] R. Gilad-Bachrach, A. Navot, N. Tishby, Margin based feature selection–theory and algorithms, in: Proc. Int. Conf. Mach. Learn., Banff, Alberta, Canada, 2004.

[19] R. Fransens, J. De Prins, L. Van Gool, SVM-based nonparametric discriminant analysis, an application to face detection, in: Proc. IEEE Int. Conf. Comp. Vis., Vol. 2, Nice, France, 2003, pp. 1289–1296.

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

[21] E. Xing, A. Ng, M. Jordan, S. Russell, Distance metric learning, with application to clustering with side-information, in: Proc. Adv. Neural Inf. Process. Syst., MIT Press, 2002.

[22] K. Q. Weinberger, J. Blitzer, L. K. Saul, Distance metric learning for large margin nearest neighbor classification, in: Proc. Adv. Neural Inf. Process. Syst., 2005.

[23] A. Globerson, S. Roweis, Metric learning by collapsing classes, in: Proc. Adv. Neural Inf. Process. Syst., 2005.

[24] J. Goldberger, S. Roweis, G. Hinton, R. Salakhutdinov, Neighbourhood component analysis, in: Proc. Adv. Neural Inf. Process. Syst., MIT Press, 2004.

[25] M. Belkin, P. Niyogi, Laplacian Eigenmaps for dimensionality reduction and data representation, Neural Comp. 15 (6) (2003) 1373–1396.

[26] D. Newman, S. Hettich, C. Blake, C. Merz, UCI repository of machine learning databases (1998).

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

[28] M. Heiler, C. Schnorr, Learning non-negative sparse image codes by convex programming, in: Proc. IEEE Int. Conf. Comp. Vis., Beijing, China, 2005, pp. 1667–1674.

[29] M. Heiler, C. Schnörr, Controlling sparseness in non-negative tensor factorization, in: Proc. Eur. Conf. Comp. Vis., Vol. 1, Graz, Austria, 2006, pp. 56–67.

[30] P. O. Hoyer, Non-negative matrix factorization with sparseness constraints, J. Mach. Learn. Res. 5 (2004) 1457–1469.

[31] A. d'Aspremont, L. E. Ghaoui, M. I. Jordan, G. R. G. Lanckriet, A direct formulation for sparse PCA using semidefinite programming, SIAM Review 49 (2007) 434–448.

[32] J. Cadima, I. Jolliffe, Loadings and correlations in the interpretation of principal components, Applied Statistics 22 (1995) 203–214.

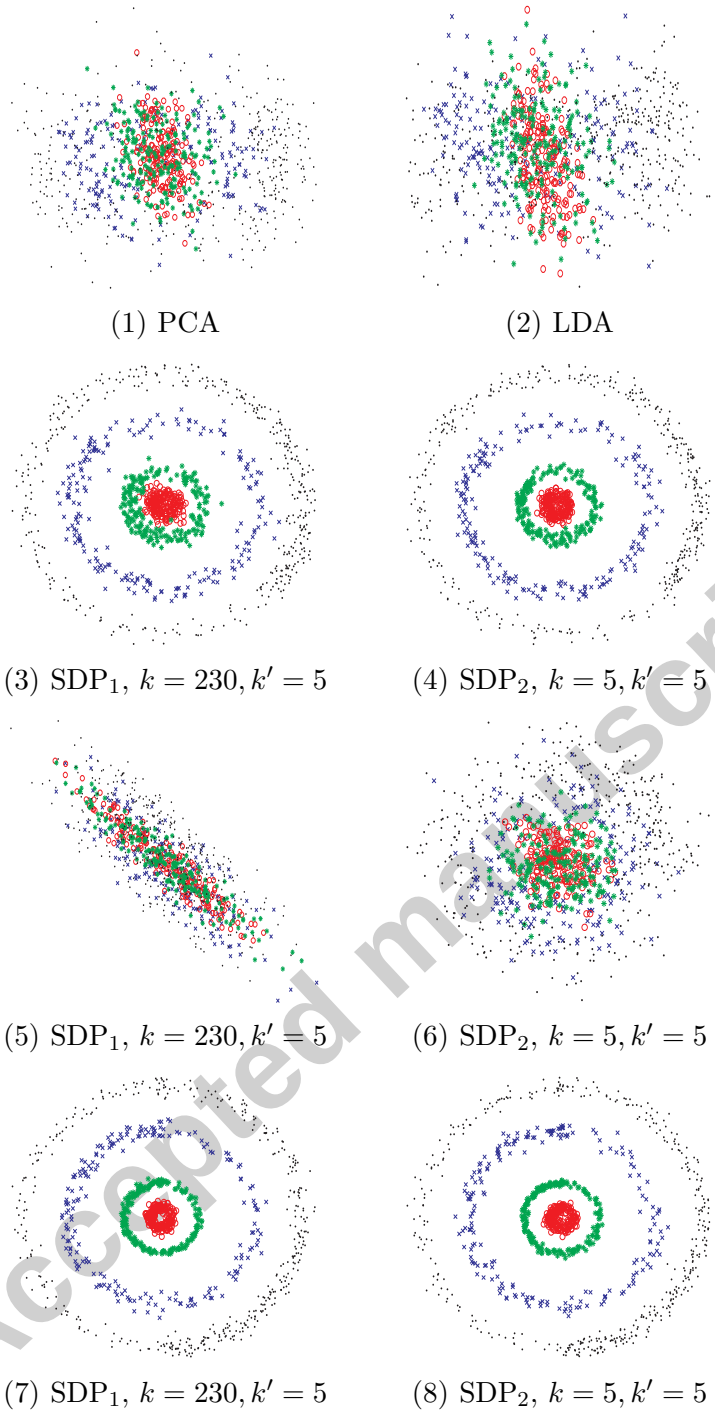

Fig. 2. Subfigures (1)(2) show the data projected into 2D using PCA and LDA. Both fail to recover the data structure. Subfigures (3)(4) show the results obtained by the two SDPs proposed in this paper. The local structure of the data is preserved after projection by SDPs. Subfigures (5)(6) are the results when the rear eight dimensions are extremely noisy. In this case the neighboring relationships based on the Euclidean distance in the input space are meaningless. Subfigures (7)(8) successfully recover the data's underlying structure given user-provided neighborhood graphs. SDP$_1$ and SDP$_2$ are the proposed methods using semidefinite programming.

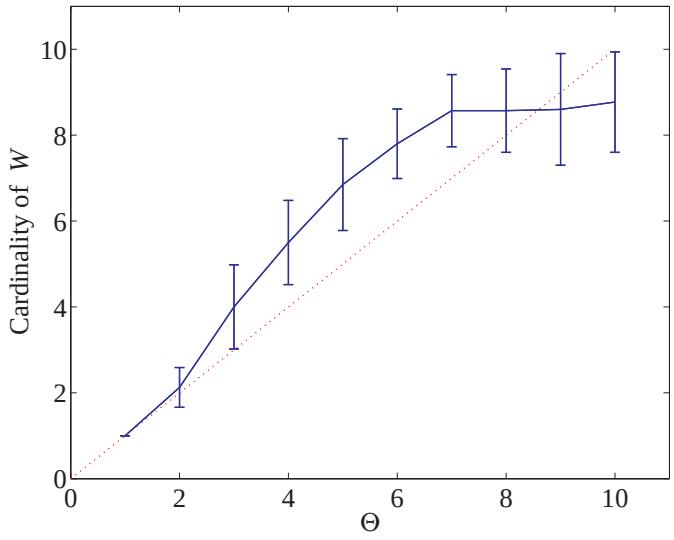

Fig. 3. Cardinality of W v.s. Θ. The error bar shows the standard deviation averaged on 40 runs.

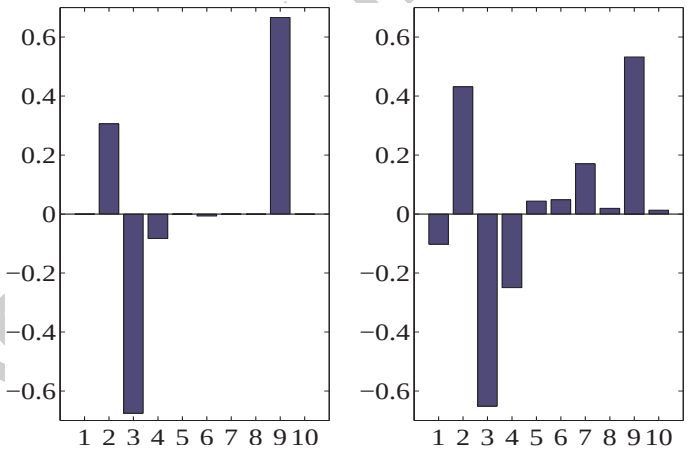

Fig. 4. The projection vector W obtained with sparseness constraints $\Theta = 3$ (left) and no sparseness constraints (right). Clearly the sparseness constraints do produce a sparse W while most of W's elements are active without sparseness constraints. Here x-axis is the index of dimensions of W and y-axis shows the corresponding values.

*** Author Biography**

Chunhua Shen received the B.Sc. and M.Sc. degrees from Nanjing University, Nanjing, China, in 1999 and 2002, respectively, and the Ph.D. degree from the School of Computer Science, University of Adelaide, Australia, in 2005.
He is currently a Researcher with the Computer Vision Program, National ICT Australia, Ltd., Canberra, and an Adjunct Research Fellow at the Australian National University. His main research interests include statistical pattern analysis and its application in computer vision.

Hongdong Li obtained his PhD degree from Zhejiang University, China, majored in Information and Electronic Engineering. He is currently a Research Fellow with the Research School of Information Sciences and Engineering (RSISE) at Australian National University (ANU). He is also a seconded Researcher to National ICT Australia Ltd.

Michael J. Brooks received the Ph.D. degree in computer science from the University of Essex, Essex, U.K., in 1983.
He joined Flinders University in 1980 and the University of Adelaide, Australia, in 1991, where he holds the Chair in Artificial Intelligence and is Head of the School of Computer Science. Since mid-2005, he has been a Nonexecutive Director of National ICT Australia, Ltd., Canberra. He has interests that include tracking, video surveillance, parameter estimation, and self-calibration. His patented surveillance work has seen worldwide application at airports, major facilities, and iconic structures around the world.