Pattern Recognition 41 (2008) 3600 – 3612

Contents lists available at ScienceDirect

Pattern Recognition

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

Learning a Mahalanobis distance metric for data clustering and classification

Shiming Xiang*, Feiping Nie, Changshui Zhang

Tsinghua National Laboratory for Information Science and Technology (TNList), Department of Automation, Tsinghua University, Beijing 100084, PR China

ARTICLE INFO

Article history:
Received 7 October 2007
Received in revised form 27 February 2008
Accepted 16 May 2008

Keywords:
Distance metric learning
Mahalanobis distance
Global optimization
Data clustering
Interactive image segmentation
Face pose estimation

ABSTRACT

Distance metric is a key issue in many machine learning algorithms. This paper considers a general problem of learning from pairwise constraints in the form of must-links and cannot-links. As one kind of *side information*, a must-link indicates the pair of the two data points must be in a same class, while a cannot-link indicates that the two data points must be in two different classes. Given must-link and cannot-link information, our goal is to learn a Mahalanobis distance metric. Under this metric, we hope the distances of point pairs in must-links are as small as possible and those of point pairs in cannot-links are as large as possible. This task is formulated as a constrained optimization problem, in which the global optimum can be obtained effectively and efficiently. Finally, some applications in data clustering, interactive natural image segmentation and face pose estimation are given in this paper. Experimental results illustrate the effectiveness of our algorithm.

1. Introduction

Distance metric is a key issue in many machine learning algorithms. For example, Kmeans and K-nearest neighbor (KNN) classifier need to be supplied a suitable distance metric, through which neighboring data points can be identified. The commonly used Euclidean distance metric assumes that each feature of data point is equally important and independent from others. This assumption may not be always satisfied in real applications, especially when dealing with high dimensional data where some features may not be tightly related to the topic of interest. In contrast, a distance metric with good quality should identify important features and discriminate relevant and irrelevant features. Thus, supplying such a distance metric is highly problem-specific and determines the success or failure of the learning algorithm or the developed system [1–13].

There has been considerable research on distance metric learning over the past few years [14]. One family of algorithms are developed with known class labels of training data points. Algorithms in this family include the neighboring component analysis [15], large margin nearest neighbor classification [16], large margin component analysis [17], class collapse [18], and other extension work [19,20]. The success in a variety of problems shows that the learned distance metric yields substantial improvements over the commonly used Euclidean distance metric [15–18]. However, class label may be strong information from the users and cannot be easily obtained in some real-world situations. In contrast, it is more natural to specify which pairs of data points are similar or dissimilar. Such pairwise constraints appear popularly in many applications. For example, in image retrieval the similar and dissimilar images to the query one are labeled by the user and such image pairs can be used to learn a distance metric [21]. Accordingly, another family of distance metric learning algorithms are developed to make use of such pairwise constraints [14,21–29]. Pairwise constraint is a kind of side information [22]. One popular form of side information is must-links and cannot-links [22,30–35]. A must-link indicates the pair of data points must be in a same class, while a cannot-link indicates that the two data points must be in two different classes. Another popular form is the relative comparison with "*A is closer to B than A is to C*" [26]. The utility of pairwise constraints has been demonstrated in many applications, indicating that significantly improvement of the algorithm can be achieved [21–27].

The two families of distance learning algorithms are extended in many aspects. Based on the class labels of training data points, Weinberger and Tesauro proposed to learn distance metric for kernel regression [36]. Based on labeled training data, Hertz et al. maximized the margin with boosting to obtain distance functions for clustering [37]. Bilenko et al. integrated the pairwise constraints (must-links and cannot-links) and metric learning into a semi-supervised clustering [38]. Clustering on many data sets shows that the performance of Kmeans algorithm has been substantially improved. Also based on must-links and cannot-links, Davis et al. developed an information theory-based framework [39]. Compared with most existing methods, their framework need not perform complex computation, such as eigenvalue decomposition and semi-definite

* Corresponding author. Tel.: +86 10 627 96 872; fax: +86 10 627 86 911.
E-mail addresses: smxiang@gmail.com (S. Xiang), zcs@mail.tsinghua.edu.cn (C. Zhang).

0031-3203/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2008.05.018

programming [15,16]. Yang et al. presented a Bayesian framework in which a posterior distribution for the distance metric is estimated from the labeled pairwise constraints [40]. Kumar et al. used the relative comparisons to develop a new clustering algorithm in a semi-supervised clustering setting [41]. Formulating the problem as a linear programming, Rosales and Fung proposed to learn a sparse metric with relative comparison constraints. The sparsity of the learned metric can help to reduce the distance computation [42]. In addition, the distance metric learning algorithms are also extended with kernel tricks [11,21,43–45]. Nonlinear adaptive metric learning algorithm has also been developed [46]. Furthermore, some online distance metrics learning algorithms [39,47] have been proposed recently for the situations where the data points are collected sequentially. The use of the learned distance metrics has been demonstrated in many real-word applications, including speech processing [48], visual representation [49], word categorization [12], face verification [50], medical image processing [51], video object classification [52], biological data processing [53], image retrieval [21,54], and so on.

In this paper we focus on learning a Mahalanobis distance metric from must-links and cannot-links. The Mahalanobis distance is a measure between two data points in the space defined by relevant features. Since it accounts for unequal variances as well as correlations between features, it will adequately evaluate the distance by assigning different weights or importance factors to the features of data points. Only when the features are uncorrelated, the distance under a Mahalanobis distance metric is identical to that under the Euclidean distance metric. In addition, geometrically, a Mahalanobis distance metric can adjust the geometrical distribution of data so that the distance between similar data points is small [22]. Thus it can enhance the performance of clustering or classification algorithms, such as KNN classifier. Such advantages can be used to perform special tasks on a given data set, if given a suitable Mahalanobis distance metric. It is natural to learn it from some prior knowledge supplied by the user according to her/his own task. One easy way to supply prior knowledge is to supply some instances of similar/dissimilar data pairs (must-links/cannot-links). We hope a Mahalanobis distance metric can be learned by forcing it to adjust the distances of the given instances and then applied to new data.

The basic idea in this paper is to minimize the distances of point pairs in must-links and maximize those of point pairs in cannot-links. To this end, we formulate this task as a constrained optimization problem. Since the formulated problem cannot be analytically solved, an iterative framework is developed to find the optimum in way of binary search. A lower bound and an upper bound including the optimum are explicitly estimated and then used to control the initial value. This will benefit the initialization of the iterative algorithm. The globally optimal Mahalanobis distance matrix is finally obtained effectively and efficiently. In addition, the computation is also fast, up to exponential convergence. Comparative experiments on data clustering, interactive natural image segmentation and face pose estimation show the validity of our algorithm.

The remainder of this paper is organized as follows. Section 2 will briefly introduce the related work and our method. We address our problem and develop the algorithm in Section 3. The experimental results and applications in data clustering, interactive image segmentation and face pose estimation are reported in Section 4. Section 5 concludes this paper.

2. Related work and our method

Given two data points $\mathbf{x}_1 \in \mathbb{R}^n$ and $\mathbf{x}_2 \in \mathbb{R}^n$, their Mahalanobis distance can be calculated as follows:

$$d_{\mathbf{A}}(\mathbf{x}_1,\mathbf{x}_2)=\sqrt{(\mathbf{x}_1-\mathbf{x}_2)^{\mathrm{T}}\mathbf{A}(\mathbf{x}_1-\mathbf{x}_2)} \tag{1}$$

where $\mathbf{A} \in \mathbb{R}^{n\times n}$ is positively semi-definite. Using the eigenvalue decomposition, $\mathbf{A}$ can be decomposed into $\mathbf{A}=\mathbf{W}\mathbf{W}^{\mathrm{T}}$. Thus, it is also feasible to learn the matrix $\mathbf{W}$. Then, we have

$$d_{\mathbf{A}}(\mathbf{x}_1,\mathbf{x}_2)=\sqrt{(\mathbf{x}_1-\mathbf{x}_2)^{\mathrm{T}}\cdot(\mathbf{W}\mathbf{W}^{\mathrm{T}})\cdot(\mathbf{x}_1-\mathbf{x}_2)} \tag{2}$$

Typically, Xing et al. studied the problem of learning a Mahalanobis matrix from must-links and cannot-links [22]. In their framework, the sum of the Mahalanobis distances of the point pairs in the must-links is used as the objective function, which is minimized under the constraints developed from the point pairs in the cannot-links. Gradient ascent and iterative projection are used to solve the optimization problem. The algorithm is effective, but it is time consuming when dealing with high dimensional data. Bar-Hillel et al. proposed the algorithm of relevance component analysis (RCA) [23]. RCA needs to solve the inverse matrix of the covariance matrix of the point pairs in the chunklets (must-links), which may not exist in the case of high dimensionality [55–57]. Such a drawback may lead the algorithm difficult to be performed.

Hoi et al. proposed the discriminative component analysis (DCA) [21]. They use the *ratio of determinants* as the objective function to learn a matrix $\mathbf{W}^*$:

$$\mathbf{W}^*=\arg\max_{\mathbf{W}}\frac{|\mathbf{W}^{\mathrm{T}}\hat{\mathbf{S}}_b\mathbf{W}|}{|\mathbf{W}^{\mathrm{T}}\hat{\mathbf{S}}_w\mathbf{W}|} \tag{3}$$

where $\hat{\mathbf{S}}_b$ and $\hat{\mathbf{S}}_w$ are the covariance matrices calculated from the point pairs in the discriminative chunklets (cannot-links) and those in the must-links [21]. After $\mathbf{W}^*$ is obtained, a Mahalanobis matrix $\mathbf{A}$ can be constructed as $\mathbf{A}=\mathbf{W}^*(\mathbf{W}^*)^{\mathrm{T}}$. Problem (3) has been well discussed in subspace learning [58] and can be analytically solved. Actually, $\mathbf{W}^*$ can be calculated via the eigenvalue decomposition of matrix $\hat{\mathbf{S}}_w^{-1}\hat{\mathbf{S}}_b$. However, singularity problem may also occur since we need to calculate $\hat{\mathbf{S}}_w^{-1}$. To avoid the singular problem, DCA selects to diagonalize the covariance matrices $\hat{\mathbf{S}}_b$ and $\hat{\mathbf{S}}_w$ simultaneously and discards the eigenvectors corresponding to the zero eigenvalue.

Formally, the objective function used in this paper can be given as follows:

$$\mathbf{W}^*=\arg\max_{\mathbf{W}^{\mathrm{T}}\mathbf{W}=\mathbf{I}}\frac{\mathrm{tr}(\mathbf{W}^{\mathrm{T}}\hat{\mathbf{S}}_b\mathbf{W})}{\mathrm{tr}(\mathbf{W}^{\mathrm{T}}\hat{\mathbf{S}}_w\mathbf{W})} \tag{4}$$

where tr is the trace operator of matrix, $\hat{\mathbf{S}}_w$ is calculated from the must-links and $\hat{\mathbf{S}}_b$ is calculated from the cannot-links. The final Mahalanobis matrix $\mathbf{A}$ is also constructed as $\mathbf{A}=\mathbf{W}^*(\mathbf{W}^*)^{\mathrm{T}}$.

In contrast, RCA is developed via $\hat{\mathbf{S}}_w$, while DCA and our method are developed via $\hat{\mathbf{S}}_b$ and $\hat{\mathbf{S}}_w$. But they have different objective functions to be optimized. RCA constructs the objective function in terms of information theory. DCA takes the ratio of two determinants as its objective function. This paper uses the *ratio of distances* (expressed as traces in the form of matrices in Problem (4)) as the objective function. In addition, we introduce an orthogonality constraint $\mathbf{W}^{\mathrm{T}}\mathbf{W}=\mathbf{I}$ to avoid degenerate solutions. However, our problem cannot be directly solved by eigenvalue decomposition approaches. We construct an iterative framework, in which a lower bound and an upper bound including the optimum are estimated for initialization. Our algorithm need not calculate the inverse matrix of $\hat{S}_w$ and thus the singularity problem is avoided.

Compared with the seminal method proposed by Xing et al. (where gradient ascent approach is used) [22], our method uses a nice heuristic (iterative) search approach to solve the optimization problem. Much time can be saved when dealing with high dimensional data.

As mentioned before, the task of this paper is to learn a distance metric from the given sets of must-links and cannot-links. Mathematically, our formulation about the problem by maximizing the ratio of distances yields just the same form of objective function used

in literature [60,65,66]. These algorithms are developed for supervised subspace learning or linear dimensionality reduction, in which the covariance matrices $\hat{\mathbf{S}}_b$ and $\hat{\mathbf{S}}_w$ are calculated via the class labels of all the training data points. A comprehensive comparison about these three previous methods is given in Refs. [66]. The main differences between our algorithm and that proposed by Wang et al. [66] can be summarized as follows:

(1) These two algorithms have different goals. The goal of the algorithm proposed by Wang et al. [66] is to reduce the dimensionality of data with given class label information, while our algorithm is to learn a distance metric with side information. Each data point and its class label will be considered in Wang's algorithm when constructing $\hat{\mathbf{S}}_b$ and $\hat{\mathbf{S}}_w$. In contrast, our algorithm only needs to consider those pairs of must-links and cannot-links.
(2) Two cases are discussed in our algorithm. The reason is that the denominator of the objective function in Problem (4) may be zero. The iterative algorithm developed by Wang et al. [66] works in the case that the denominator of the objective function is not zero. Such discussions are introduced in this paper.
(3) We show a new property in our paper, namely, the monotonicity of the objective function. In the case of non-zero denominator, the objective value monotonously decreases with the increase of the dimensionality of the subspace we use. Such a property guides us to find a bound including the optimum for initialization and iterations.
(4) When the denominator of the objective function is not zero, there exists a unique globally optimal solution [66]. For the same $\hat{\mathbf{S}}_b$ and $\hat{\mathbf{S}}_w$ and with the parameter d, the algorithms will yield the same solution. To speed up the search of our approach, we give a lower bound and an upper bound including the optimum.
(5) The heuristic search approach proposed by Wang et al. [66] is developed by utilizing the previously estimated transformation matrix $\mathbf{W}$. Intrinsically, it is exactly one of the Newton's methods, while our method is a binary search method. Given the initial value for iterations, it is slightly faster than our algorithm.

3. Algorithm

3.1. Problem formulation

Suppose we are given a data set $\mathscr{X} = \{\mathbf{x}_i\}_{i=1}^N \subset \mathbb{R}^n$ and two sets of pairwise constraints including must-links: $\mathscr{S} = \{(\mathbf{x}_i, \mathbf{x}_j) | \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ are in a same class}\}$, and cannot-links: $\mathscr{D} = \{(\mathbf{x}_i, \mathbf{x}_j) | \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ are in two different classes}\}$. Our goal is to learn a Mahalanobis matrix $\mathbf{A}$ such that the distances of point pairs in $\mathscr{S}$ are as small as possible, while those in $\mathscr{D}$ are as large as possible.

According to Eq. (2), equivalently, we can select to optimize the matrix $\mathbf{W} \in \mathbb{R}^{n\times d}$, with $d \leqslant n$. To this end, we introduce a transformation:

$$\mathbf{y} = \mathbf{W}^{\mathrm{T}}\mathbf{x}. \tag{5}$$

Under this transformation, the sum of the squared distances of the point pairs in $\mathscr{S}$ can be calculated as follows:

$$\begin{aligned} d_w &= \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathscr{S}} (\mathbf{W}^{\mathrm{T}}\mathbf{x}_i - \mathbf{W}^{\mathrm{T}}\mathbf{x}_j)^{\mathrm{T}}(\mathbf{W}^{\mathrm{T}}\mathbf{x}_i - \mathbf{W}^{\mathrm{T}}\mathbf{x}_j) \\ &= \mathrm{tr}(\mathbf{W}^{\mathrm{T}}\hat{\mathbf{S}}_w\mathbf{W}) \end{aligned} \tag{6}$$

here tr is a trace operator, and $\hat{S}_w$ is the covariance matrix of the point pairs in $\mathscr{S}$:

$$\hat{\mathbf{S}}_w = \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathscr{S}} (\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^{\mathrm{T}} \tag{7}$$

Correspondingly, for the point pairs in $\mathscr{D}$, we have

$$d_b = \mathrm{tr}(\mathbf{W}^{\mathrm{T}}\hat{\mathbf{S}}_b\mathbf{W}) \tag{8}$$

where $\hat{\mathbf{S}}_b \in \mathbb{R}^{n\times n}$ and $\hat{\mathbf{S}}_b = \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathscr{D}} (\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^{\mathrm{T}}$.

We try to minimize d_w and maximize d_b. This formulation yields the follow optimization problem:

$$\mathbf{W}^* = \arg\max_{\mathbf{W}^{\mathrm{T}}\mathbf{W}=\mathbf{I}} \frac{\mathrm{tr}(\mathbf{W}^{\mathrm{T}}\hat{\mathbf{S}}_b\mathbf{W})}{\mathrm{tr}(\mathbf{W}^{\mathrm{T}}\hat{\mathbf{S}}_w\mathbf{W})} \tag{9}$$

here $\mathbf{W} \in \mathbb{R}^{n\times d}$, the constraint $\mathbf{W}^{\mathrm{T}}\mathbf{W}=\mathbf{I}$ is introduced to avoid degenerate solutions, and $\mathbf{I}$ is an $d \times d$ identity matrix. Note that $\mathbf{W}$ is not a square matrix if $d < n$. In this case, $\mathbf{W}\mathbf{W}^{\mathrm{T}}$ will not equal to an identity matrix. However, in the case of $d = n$, we have $\mathbf{W}\mathbf{W}^{\mathrm{T}} = \mathbf{W}^{\mathrm{T}}\mathbf{W} = \mathbf{I}$. This case will generate the standard Euclidean distance and thus will not be considered in this paper.

After the optimum $\mathbf{W}^*$ is obtained, a Mahalanobis matrix $\mathbf{A}$ can be constructed as follows:

$$\mathbf{A} = \begin{cases} \mathbf{W}^*(\mathbf{W}^*)^{\mathrm{T}} & \text{if } d < n \\ \mathbf{I} & \text{if } d = n \end{cases} \tag{10}$$

3.2. Solving the optimization problem

To solve Problem (9), we first consider the denominator $d_w = \mathrm{tr}(\mathbf{W}^{\mathrm{T}}\hat{\mathbf{S}}_w\mathbf{W})$ in two cases. We have the following theorem:

Theorem 1. *Suppose* $\mathbf{W} \in \mathbb{R}^{n\times d}$, $\mathbf{W}^{\mathrm{T}}\mathbf{W} = \mathbf{I}$, *and* r ($\leqslant n$) *is the rank of matrix* $\hat{\mathbf{S}}_w$. *If* $d > n - r$, *then* $\mathrm{tr}(\mathbf{W}^{\mathrm{T}}\hat{\mathbf{S}}_w\mathbf{W}) > 0$. *If* $d \leqslant n - r$, *then* $\mathrm{tr}(\mathbf{W}^{\mathrm{T}}\hat{\mathbf{S}}_w\mathbf{W})$ *may be equal to zero.*

Proof. Based on Rayleigh quotient theory [59], $\min \mathrm{tr}(\mathbf{W}^{\mathrm{T}}\hat{\mathbf{S}}_w\mathbf{W}) = \sum_{i=1}^{d}\beta_i$ holds if $\mathbf{W}^{\mathrm{T}}\mathbf{W} = \mathbf{I}$. Here $\beta_1, \ldots, \beta_d$ are the first d smallest eigenvalues of $\hat{\mathbf{S}}_w$. According to Eq. (7), we can easily justify that $\hat{\mathbf{S}}_w$ is positive semi-definite and thus all of its eigenvalues are non-negative. Since its rank equals to r, it has r positive eigenvalues and $n - r$ zero eigenvalues. If $d > n - r$, there exists at least one positive eigenvalue among $\beta_1, \ldots, \beta_d$. This indicates that $\mathrm{tr}(\mathbf{W}^{\mathrm{T}}\hat{\mathbf{S}}_w\mathbf{W}) \geqslant \min \mathrm{tr}(\mathbf{W}^{\mathrm{T}}\hat{\mathbf{S}}_w\mathbf{W}) > 0$ holds. In the case of $d \leqslant n - r$, however, each β_i may be equal to zero. Thus $\mathrm{tr}(\mathbf{W}^{\mathrm{T}}\hat{\mathbf{S}}_w\mathbf{W})$ may be zero. □

This theorem implies that it is necessary for us to discuss the problem in two cases.

Case 1: $d > n - r$.

Let λ^* be the optimal value of Problem (9), namely,

$$\lambda^* = \max_{\mathbf{W}^{\mathrm{T}}\mathbf{W}=\mathbf{I}} \frac{\mathrm{tr}(\mathbf{W}^{\mathrm{T}}\hat{\mathbf{S}}_b\mathbf{W})}{\mathrm{tr}(\mathbf{W}^{\mathrm{T}}\hat{\mathbf{S}}_w\mathbf{W})}.$$

According to the work by Guo et al. [60], it follows:

$$\max_{\mathbf{W}^{\mathrm{T}}\mathbf{W}=\mathbf{I}} \mathrm{tr}(\mathbf{W}^{\mathrm{T}}(\hat{\mathbf{S}}_b - \lambda^*\hat{\mathbf{S}}_w)\mathbf{W}) = 0. \tag{11}$$

Inspired by Eq. (11), we introduce a function about λ:

$$g(\lambda) = \max_{\mathbf{W}^{\mathrm{T}}\mathbf{W}=\mathbf{I}} \mathrm{tr}(\mathbf{W}^{\mathrm{T}}(\hat{\mathbf{S}}_b - \lambda\hat{\mathbf{S}}_w)\mathbf{W}) \tag{12}$$

The value of $g(\lambda)$ can be easily calculated. According to the theories of matrix [59], it equals to the sum of the first d largest eigenvalues of $(\hat{\mathbf{S}}_b - \lambda\hat{\mathbf{S}}_w)$. Based on Eq. (11), now our task is to find a λ such that $g(\lambda) = 0$.

Note that in this case $\mathrm{tr}(\mathbf{W}^{\mathrm{T}}\hat{\mathbf{S}}_w\mathbf{W}) > 0$, then the following two propositions hold naturally:

$g(\lambda) < 0 \Rightarrow \lambda > \lambda^*$

$g(\lambda)>0 \Rightarrow \lambda<\lambda^*$

This indicates that we can iteratively find λ^* according to the sign of $g(\lambda)$. After λ^* is determined, the optimal $\mathbf{W}^*$ can be obtained by performing eigenvalue decomposition of $(\hat{\mathbf{S}}_b-\lambda^*\hat{\mathbf{S}}_w)$. In this way, we avoid calculating the inverse matrix of $\hat{\mathbf{S}}_w$.

To give an initial value for iteratively finding the optimum λ^*, now we determine a lower bound and an upper bound for λ^*. Actually, we have the following theorem:

Theorem 2. *Let r be the rank of* $\hat{\mathbf{S}}_w$. *If* $d>n-r$, *then*

$$\frac{\mathrm{tr}(\hat{\mathbf{S}}_b)}{\mathrm{tr}(\hat{\mathbf{S}}_w)}\leqslant\lambda^*\leqslant\frac{\sum_{i=1}^{d}\alpha_i}{\sum_{i=1}^{d}\beta_i} \tag{13}$$

where $\alpha_1,\ldots,\alpha_d$ *are the first d largest eigenvalues of* $\hat{\mathbf{S}}_b$, *and* $\beta_1,\ldots,\beta_d$ *are the first d smallest eigenvalues of* $\hat{\mathbf{S}}_w$.

To prove this theorem, first we give the following two lemmas:

Lemma 1. $\forall i, a_i\geqslant 0, b_i>0$, *if* $a_1/b_1\leqslant a_2/b_2\leqslant\cdots\leqslant a_p/b_p$, *then* $\sum_{i=1}^{p}a_i/\sum_{i=1}^{p}b_i\leqslant a_p/b_p$.

Proof. Let $a_p/b_p=q$. $\forall i$, we have $a_i\leqslant qb_i$. Thus it follows $\sum_{i=1}^{p}a_i/\sum_{i=1}^{p}b_i \leqslant a_p/b_p$. □

Lemma 2. *Let r be the rank of* $\hat{\mathbf{S}}_w$, $\mathbf{W}_1\in\mathbb{R}^{n\times d_1}$ *and* $\mathbf{W}_2\in\mathbb{R}^{n\times d_2}$. *If* $d_1>d_2>n-r$, *then*

$$\max_{\mathbf{W}_1^{\mathrm{T}}\mathbf{W}_1=\mathbf{I}}\frac{\mathrm{tr}(\mathbf{W}_1^{\mathrm{T}}\hat{\mathbf{S}}_b\mathbf{W}_1)}{\mathrm{tr}(\mathbf{W}_1^{\mathrm{T}}\hat{\mathbf{S}}_w\mathbf{W}_1)}\leqslant\max_{\mathbf{W}_2^{\mathrm{T}}\mathbf{W}_2=\mathbf{I}}\frac{\mathrm{tr}(\mathbf{W}_2^{\mathrm{T}}\hat{\mathbf{S}}_b\mathbf{W}_2)}{\mathrm{tr}(\mathbf{W}_2^{\mathrm{T}}\hat{\mathbf{S}}_w\mathbf{W}_2)}.$$

Proof. Let

$$\mathbf{W}_1^*=\arg\max_{\mathbf{W}_1^{\mathrm{T}}\mathbf{W}_1=\mathbf{I}}\frac{\mathrm{tr}(\mathbf{W}_1^{\mathrm{T}}\hat{\mathbf{S}}_b\mathbf{W}_1)}{\mathrm{tr}(\mathbf{W}_1^{\mathrm{T}}\hat{\mathbf{S}}_w\mathbf{W}_1)}.$$

We can get $C_{d_1}^{d_2}$ sub-matrices, each of which contains d_2 column vectors of $\mathbf{W}_1^*$. Let $p=C_{d_1}^{d_2}$ and denote them by $\mathbf{W}_{(i)}\in\mathbb{R}^{n\times d_2}$, $i=1,\ldots,p$. Without loss of generality, suppose

$$\frac{\mathrm{tr}(\mathbf{W}_{(1)}^{\mathrm{T}}\hat{\mathbf{S}}_b\mathbf{W}_{(1)})}{\mathrm{tr}(\mathbf{W}_{(1)}^{\mathrm{T}}\hat{\mathbf{S}}_w\mathbf{W}_{(1)})}\leqslant\cdots\leqslant\frac{\mathrm{tr}(\mathbf{W}_{(p)}^{\mathrm{T}}\hat{\mathbf{S}}_b\mathbf{W}_{(p)})}{\mathrm{tr}(\mathbf{W}_{(p)}^{\mathrm{T}}\hat{\mathbf{S}}_w\mathbf{W}_{(p)})}.$$

Note that each column vector of $\mathbf{W}_1^*$ will appear $C_{d_1-1}^{d_2-1}$ times in these p sub-matrices. Then we have

$$\begin{aligned}\max_{\mathbf{W}_1^{\mathrm{T}}\mathbf{W}_1=\mathbf{I}}\frac{\mathrm{tr}(\mathbf{W}_1^{\mathrm{T}}\hat{\mathbf{S}}_b\mathbf{W}_1)}{\mathrm{tr}(\mathbf{W}_1^{\mathrm{T}}\hat{\mathbf{S}}_w\mathbf{W}_1)}&=\frac{C_{d_1-1}^{d_2-1}\cdot\mathrm{tr}((\mathbf{W}_1^*)^{\mathrm{T}}\hat{\mathbf{S}}_b\mathbf{W}_1^*)}{C_{d_1-1}^{d_2-1}\cdot\mathrm{tr}((\mathbf{W}_1^*)^{\mathrm{T}}\hat{\mathbf{S}}_w(\mathbf{W}_1^*)^{\mathrm{T}})}\\&=\frac{\sum_{i=1}^{p}\mathrm{tr}(\mathbf{W}_{(i)}^{\mathrm{T}}\hat{\mathbf{S}}_b\mathbf{W}_{(i)})}{\sum_{i=1}^{p}\mathrm{tr}(\mathbf{W}_{(i)}^{\mathrm{T}}\hat{\mathbf{S}}_w\mathbf{W}_{(i)})}\\&\leqslant\frac{\mathrm{tr}(\mathbf{W}_{(p)}^{\mathrm{T}}\hat{\mathbf{S}}_b\mathbf{W}_{(p)})}{\mathrm{tr}(\mathbf{W}_{(p)}^{\mathrm{T}}\hat{\mathbf{S}}_w\mathbf{W}_{(p)})}\\&\leqslant\max_{\mathbf{W}_2^{\mathrm{T}}\mathbf{W}_2=\mathbf{I}}\frac{\mathrm{tr}(\mathbf{W}_2^{\mathrm{T}}\hat{\mathbf{S}}_b\mathbf{W}_2)}{\mathrm{tr}(\mathbf{W}_2^{\mathrm{T}}\hat{\mathbf{S}}_w\mathbf{W}_2)}\end{aligned}$$

The first and the second equalities hold naturally, according to the rules of trace operator of matrix. The first inequality holds according to Lemma 1, while the second inequality holds since

$$\max_{\mathbf{W}_2^{\mathrm{T}}\mathbf{W}_2=\mathbf{I}}\frac{\mathrm{tr}(\mathbf{W}_2^{\mathrm{T}}\hat{\mathbf{S}}_b\mathbf{W}_2)}{\mathrm{tr}(\mathbf{W}_2^{\mathrm{T}}\hat{\mathbf{S}}_w\mathbf{W}_2)}$$

can serve as an upper bound. Thus we finish the proof. □

Table 1
Binary search for solving the optimization problem

Input: $\hat{\mathbf{S}}_w$, $\hat{\mathbf{S}}_b\in\mathbb{R}^{n\times n}$, the lower dimensionality d, and an error constant ε.
Output: A matrix $\mathbf{W}^*\in\mathbb{R}^{n\times d}$.
1. Calculate the rank r of the matrix $\hat{\mathbf{S}}_w$
Case 1: $d>n-r$
2. $\lambda_1\leftarrow\mathrm{tr}(\hat{\mathbf{S}}_b)/\mathrm{tr}(\hat{\mathbf{S}}_w)$, $\lambda_2\leftarrow(\sum_{i=1}^{d}\alpha_i)/(\sum_{i=1}^{d}\beta_i)$, $\lambda\leftarrow(\lambda_1+\lambda_2)/2$.
3. While $\lambda_2-\lambda_1>\varepsilon$, do
(a) Calculate $g(\lambda)$ by solving Problem (12).
(b) If $g(\lambda)>0$, then $\lambda_1\leftarrow\lambda$; else $\lambda_2\leftarrow\lambda$.
(c) $\lambda\leftarrow(\lambda_1+\lambda_2)/2$.
EndWhile.
4. $\mathbf{W}^*=[\boldsymbol{\mu}_1,\ldots,\boldsymbol{\mu}_d]$, where $\boldsymbol{\mu}_1,\ldots,\boldsymbol{\mu}_d$ are the d eigenvectors, corresponding to the d largest eigenvalues of $\hat{\mathbf{S}}_b-\lambda\hat{\mathbf{S}}_w$.
Case 2: $d\leqslant n-r$
$\mathbf{W}^*=\mathbf{Z}\cdot[\boldsymbol{v}_1,\ldots,\boldsymbol{v}_d]$. Here $\boldsymbol{v}_1,\ldots,\boldsymbol{v}_d$ are d eigenvectors corresponding to the d largest eigenvalues of $\mathbf{Z}^{\mathrm{T}}\hat{\mathbf{S}}_b\mathbf{Z}$, and $\mathbf{Z}=[\boldsymbol{z}_1,\ldots,\boldsymbol{z}_{n-r}]$ are the eigenvectors corresponding to $n-r$ zero eigenvalues of $\hat{\mathbf{S}}_w$.

Here we show a new property, namely, the monotonicity of the objective function. In the case of non-zero denominator, the objective value monotonously decreases with the increase of the dimensionality of the subspace we use.

Now we can give the proof of Theorem 2 as follows:

Proof of Theorem 2. Lemma 2 indicates that the optimal value monotonously decreases with the increasing of d. Thus we can find a lower bound for λ^* when $d=n$. In this case, $\mathbf{W}\in\mathbb{R}^{n\times n}$ is a square matrix and $\mathbf{W}\mathbf{W}^{\mathrm{T}}=\mathbf{I}$ also holds. According to the rule of trace operator (here, $\mathrm{tr}(\mathbf{AB})=\mathrm{tr}(\mathbf{BA})$), it follows:

$$\lambda^*\geqslant\frac{\mathrm{tr}(\mathbf{W}^{\mathrm{T}}\hat{\mathbf{S}}_b\mathbf{W})}{\mathrm{tr}(\mathbf{W}^{\mathrm{T}}\hat{\mathbf{S}}_w\mathbf{W})}=\frac{\mathrm{tr}(\hat{\mathbf{S}}_b\mathbf{W}\mathbf{W}^{\mathrm{T}})}{\mathrm{tr}(\hat{\mathbf{S}}_w\mathbf{W}\mathbf{W}^{\mathrm{T}})}=\frac{\mathrm{tr}(\hat{\mathbf{S}}_b)}{\mathrm{tr}(\hat{\mathbf{S}}_w)} \tag{14}$$

According to Rayleigh quotient theory [59], for symmetric matrices $\hat{\mathbf{S}}_b$ and $\hat{\mathbf{S}}_w$, we have

$$\max_{\mathbf{W}^{\mathrm{T}}\mathbf{W}=\mathbf{I}}\mathrm{tr}(\mathbf{W}^{\mathrm{T}}\hat{\mathbf{W}}_b\mathbf{W})=\sum_{i=1}^{d}\alpha_i$$

and

$$\min_{\mathbf{W}^{\mathrm{T}}\mathbf{W}=\mathbf{I}}\mathrm{tr}(\mathbf{W}^{\mathrm{T}}\hat{\mathbf{S}}_w\mathbf{W})=\sum_{i=1}^{d}\beta_i$$

here $\alpha_1,\ldots,\alpha_d$ are the first d largest eigenvalues of $\hat{\mathbf{S}}_b$, and $\beta_1,\ldots,\beta_d$ are the first d smallest eigenvalues of $\hat{\mathbf{S}}_w$. Then $\sum_{i=1}^{d}\alpha_i/\sum_{i=1}^{d}\beta_i$ is an upper bound of

$$\max_{\mathbf{W}^{\mathrm{T}}\mathbf{W}=\mathbf{I}}\frac{\mathrm{tr}(\mathbf{W}^{\mathrm{T}}\hat{\mathbf{S}}_b\mathbf{W})}{\mathrm{tr}(\mathbf{W}^{\mathrm{T}}\hat{\mathbf{S}}_w\mathbf{W})}$$

Thus, the second inequality holds. □

Given the lower bound and the upper bound, λ^* can be reached in way of binary search. The steps are listed in Table 1. The optimal W^* is finally obtained by performing the eigenvalue decomposition of $\hat{S}_b-\lambda^*\hat{S}_w$. From the performance steps, we can see that the singularity problem can be naturally avoided.

Case 2: $d\leqslant n-r$.

If $\mathbf{W}$ is in the null space[1] of $\hat{\mathbf{S}}_w$, then $\mathrm{tr}(\mathbf{W}^{\mathrm{T}}\hat{\mathbf{S}}_w\mathbf{W})=0$ and λ^* will be infinite. Thus it is feasible to maximize the numerator $\mathrm{tr}(\mathbf{W}^{\mathrm{T}}\hat{\mathbf{S}}_b\mathbf{W})$ after performing a null-space transformation $\mathbf{y}=\mathbf{Z}^{\mathrm{T}}\mathbf{x}$:

$$\mathbf{V}^* = \arg\max_{\mathbf{V}^{\mathrm{T}}\mathbf{V}=\mathbf{I}} (\mathbf{V}^{\mathrm{T}}(\mathbf{Z}^{\mathrm{T}}\hat{\mathbf{S}}_b\mathbf{Z})\mathbf{V}), \tag{15}$$

where $\mathbf{Z}\in\mathbb{R}^{n\times(n-r)}$ is a matrix whose column vectors are the eigenvectors corresponding to $n-r$ zero eigenvalues of $\hat{\mathbf{S}}_w$, and $\mathbf{V}\in\mathbb{R}^{(n-r)\times d}$ is a matrix to be optimized. After $\mathbf{V}^*$ is obtained, we can get $\mathbf{W}^*=\mathbf{Z}\mathbf{V}^*$. The algorithm is also given in Table 1.

3.3. Algorithm

The algorithm in Table 1 needs to perform eigenvalue decomposition of $\hat{\mathbf{S}}_b-\lambda\hat{\mathbf{S}}_w\in\mathbb{R}^{n\times n}$. When n is very large, saying $n>5000$, usually current PCs have difficulties in finishing this task. Reducing the dimensionality is desired when facing such high dimensional data. For Problem (9), we can first eliminate the null space of $\hat{\mathbf{S}}_b+\hat{\mathbf{S}}_w$. Actually, we have the following theorem:

Theorem 3. *Problem* (9) *can be solved in the orthogonal complement space of the null space of* $\hat{\mathbf{S}}_b+\hat{\mathbf{S}}_w$, *without loss of any information.*

To be concision, the proof about Theorem 3 is given in Appendix A. Finally, the algorithm for learning a Mahalanobis distance metric from pairwise constraints is given in Table 2.

To calculate the null space of matrix $\hat{\mathbf{S}}_b+\hat{\mathbf{S}}_w$, we need to perform an eigenvalue decomposition of it. If the dimensionality (n) is larger than the number of data points (N), the rank of $\hat{\mathbf{S}}_b+\hat{\mathbf{S}}_w$ will not be greater than N. In this case, we need not to perform the eigenvalue decomposition on the original scale of $n\times n$. We have the following Theorem:

Theorem 4. *Given two matrix* $\mathbf{A}\in\mathbb{R}^{n\times N}$ *and* $\mathbf{B}\in\mathbb{R}^{N\times n}$, *then* $\mathbf{AB}$ *and* $\mathbf{BA}$ *have the same non-zero eigenvalues. For each non-zero eigenvalue of* $\mathbf{AB}$, *if the corresponding eigenvector of* $\mathbf{AB}$ *is* $\mathbf{v}$, *then the corresponding eigenvector of* $\mathbf{BA}$ *is* $\mathbf{u}=\mathbf{B}\mathbf{v}$.

The proof about Theorem 4 is also given in Appendix B.

Now let $\mathbf{X}$ be the data matrix containing N data points, namely, $\mathbf{X}=[\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_N]\in\mathbb{R}^{n\times N}$. Based on the must-links, a symmetrical indicator matrix $\mathbf{L}_S\in\mathbb{R}^{N\times N}$ with element $L_S(i,j)$ can be defined as follows:

$$\begin{cases} L_S(i,j)=L_S(j,i)=1; & (\mathbf{x}_i,\mathbf{x}_j)\in\mathscr{S} \\ L_S(i,j)=L_S(j,i)=0; & (\mathbf{x}_i,\mathbf{x}_j)\notin\mathscr{S} \end{cases}$$

Furthermore, based on the cannot-links, a symmetrical indicator matrix $\mathbf{L}_d\in\mathbb{R}^{N\times N}$ with element $L_d(i,j)$ can be defined as follows:

$$\begin{cases} L_d(i,j)=L_d(j,i)=1; & (\mathbf{x}_i,\mathbf{x}_j)\in\mathscr{D} \\ L_d(i,j)=L_d(j,i)=0; & (\mathbf{x}_i,\mathbf{x}_j)\notin\mathscr{D} \end{cases}$$

Let $\mathbf{L}_w=diag(sum(\mathbf{L}_S))-\mathbf{L}_S$ and $\mathbf{L}_b=diag(sum(\mathbf{L}_d))-\mathbf{L}_d$, where $sum(\cdot)$ is an N-dimensional vector which records the sum of each row of the matrix. Now it can be easily justified that

$$\hat{\mathbf{S}}_w=\tfrac{1}{2}\mathbf{X}\mathbf{L}_w\mathbf{X}^{\mathrm{T}} \tag{16}$$

and

$$\hat{\mathbf{S}}_b=\tfrac{1}{2}\mathbf{X}\mathbf{L}_b\mathbf{X}^{\mathrm{T}} \tag{17}$$

[1] The null space of $\mathbf{A}\in\mathbb{R}^{n\times n}$ is the set of column vectors $\mathbf{X}$ such that $\mathbf{AX}=0$. This space can be span by the eigenvectors corresponding to zero eigenvalues of $\mathbf{A}$.

Table 2
Algorithm of learning a Mahalanobis distance metric

0. Preprocess:
   a) Eliminate the null space of $\hat{\mathbf{S}}_w+\hat{\mathbf{S}}_b$ and obtain a linear transformation $\mathbf{y}=\mathbf{W}_1^{\mathrm{T}}\mathbf{x}$. Here $\mathbf{W}_1$ only contains the eigenvectors corresponding to the non-zero eigenvalues of $\hat{\mathbf{S}}_w+\hat{\mathbf{S}}_b$ and $\mathbf{W}_1^{\mathrm{T}}\mathbf{W}_1=\mathbf{I}$.
   b) Obtain the new matrices $\tilde{\mathbf{S}}_w=\mathbf{W}_1^{\mathrm{T}}\hat{\mathbf{S}}_w\mathbf{W}_1$ and $\tilde{\mathbf{S}}_b=\mathbf{W}_1^{\mathrm{T}}\hat{\mathbf{S}}_b\mathbf{W}_1$.
1. Input d, ε, $\tilde{\mathbf{S}}_w$ and $\tilde{\mathbf{S}}_b$.
2. Learn a $\mathbf{W}^*$ according to the algorithm in Table 1.
3. Output a Mahalanobis matrix for the original data points: $\mathbf{A}=\mathbf{W}_1\mathbf{W}^*(\mathbf{W}^*)^{\mathrm{T}}\mathbf{W}_1^{\mathrm{T}}$.

Thus, we have

$$\hat{\mathbf{S}}_w+\hat{\mathbf{S}}_b=\mathbf{X}(\tfrac{1}{2}\mathbf{L}_w+\tfrac{1}{2}\mathbf{L}_b)\mathbf{X}^{\mathrm{T}} \tag{18}$$

Let $\mathbf{L}=\mathbf{X}^{\mathrm{T}}\mathbf{X}(\frac{1}{2}\mathbf{L}_w+\frac{1}{2}\mathbf{L}_b)\in\mathbb{R}^{N\times N}$. In the case of $N<n$, we can calculate the non-zero eigenvalues of $\mathbf{L}$ and their corresponding eigenvectors.

Let the rank of $\mathbf{L}$ be r ($\leqslant N$). Since the rank of matrix equals to the number of non-zero eigenvalues, then $\mathbf{L}$ has r non-zero eigenvalues. Denote their corresponding eigenvectors by $\{\mathbf{v}_1,\mathbf{v}_2,\ldots,\mathbf{v}_r\}$. Then according to Theorem 4, we can get r eigenvectors of $\hat{\mathbf{S}}_w+\hat{\mathbf{S}}_b$:

$$\mathbf{u}_i=\mathbf{X}(\tfrac{1}{2}\mathbf{L}_w+\tfrac{1}{2}\mathbf{L}_b)\mathbf{v}_i,\quad i=1,2,\ldots,r \tag{19}$$

Note that the eigenvectors of $\hat{\mathbf{S}}_w+\hat{\mathbf{S}}_b$ obtained by Eq. (19) are orthogonal to each other, namely, $\mathbf{u}_i^{\mathrm{T}}\mathbf{u}_j=0$; $i\neq j$. But the length of each vector may not be one. Thus we should normalize each $\mathbf{u}_i$ such that it has unit length. This can be easily obtained by performing $\mathbf{u}_i\leftarrow(1/\|\mathbf{u}_i\|)\mathbf{u}_i$.

These r vectors $\{\mathbf{u}_1,\mathbf{u}_2,\ldots,\mathbf{u}_r\}$ constitute a base of the orthogonal complement space of the null space of $\hat{\mathbf{S}}_w+\hat{\mathbf{S}}_b$. Let $\mathbf{W}_1=[\mathbf{u}_1,\mathbf{u}_2,\ldots,\mathbf{u}_r]\in\mathbb{R}^{n\times r}$ (note that it is just the $\mathbf{W}_1$ in Proof of Theorem 3). Then for each source data point $\mathbf{x}_i$, we can project it onto the orthogonal complement space of the null space of $\hat{\mathbf{S}}_w+\hat{\mathbf{S}}_b$ by performing $\mathbf{x}_i\leftarrow\mathbf{W}_1^{\mathrm{T}}\mathbf{x}_i$. In this way, we eliminate the null space of $\hat{\mathbf{S}}_w+\hat{\mathbf{S}}_b$ as well as reduce the source dimensionality. In the case of $n>N$, the newly dimensionality-reduced $\{\mathbf{x}_i\}_{i=1}^N$ will be supplied to Algorithm 1 in Table 1.

4. Experiments

We evaluated our algorithm on several data sets, and compared it with RCA, DCA and Xing's method. We show the experimental results and the applications to data clustering, interactive natural image segmentation and face pose estimation.

4.1. Experiment on Toy data set

Fig. 1(a) shows three classes, each of which contains 100 data points in $\mathbb{R}^3$. Totally, there are $3\times100\times99/2=14\,850$ point pairs which can be used as must-links and $3\times100\times100=30\,000$ point pairs which can be used as cannot-links. In experiments, we randomly select 5, 10 and 25 point pairs from each class to construct three sets of must-links $\mathscr{S}_1$, $\mathscr{S}_2$ and $\mathscr{S}_3$. Thus, $\mathscr{S}_1$, $\mathscr{S}_2$ and $\mathscr{S}_3$ only contain 15, 30 and 75 pairwise constraints. Then we take transitive closures[2] over the constraints in $\mathscr{S}_1$, $\mathscr{S}_2$ and $\mathscr{S}_3$, respectively.

Three sets of cannot-links, $\mathscr{D}_1$, $\mathscr{D}_2$ and $\mathscr{D}_3$ are also randomly generated, which contain 75, 300 and 600 cannot-links, respectively. We also take transitive closures[3] over the constraints in $\mathscr{D}_1$, $\mathscr{D}_2$ and $\mathscr{D}_3$.

[2] Suppose $(\mathbf{x}_i,\mathbf{x}_j)$ and $(\mathbf{x}_j,\mathbf{x}_k)$ are two must-links. Then $(\mathbf{x}_i,\mathbf{x}_k)$ is also a must-link. It is added automatically into $\mathscr{S}$.

[3] Suppose $(\mathbf{x}_i,\mathbf{x}_j)$ is a must-link and $(\mathbf{x}_j,\mathbf{x}_k)$ is a cannot-link. Then $(\mathbf{x}_i,\mathbf{x}_k)$ is also a cannot-link. It is added automatically into $\mathscr{D}$.

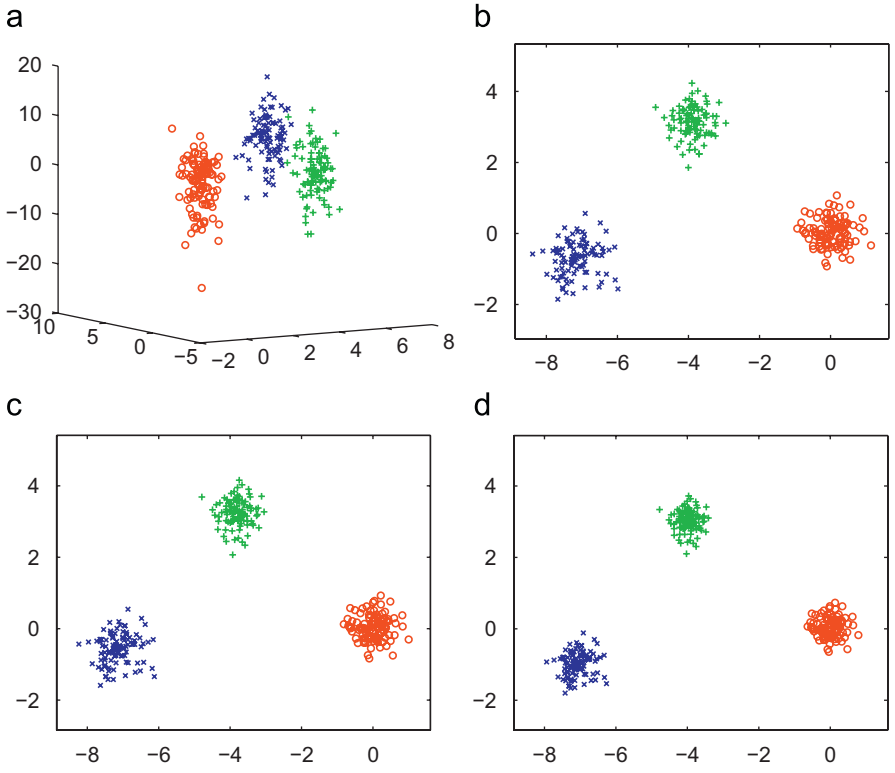

Fig. 1. (a) The 3×100 original data points. (b), (c) and (d) the data points transformed from the learned linear transforms with $\mathscr{S}_i$ and $\mathscr{D}_i$ $(i = 1, 2, 3)$.

The algorithm in Table 1 with $d = 2$ is used to learn the three linear transformations, respectively, from $\mathscr{S}_i$ and $\mathscr{D}_i$ $(i = 1, 2, 3)$. Fig. 1(b)–(d) shows the data points transformed from the learned transformations (i.e., $\mathbf{y} = (\mathbf{W}^*)^{\mathrm{T}}\mathbf{x}$ and $\mathbf{W}^* \in \mathbb{R}^{3\times 2}$). We can see that the data points within a class are all pulled together. They tend to be tightly close to each other with the increasing of the number of pairwise constraints. Meanwhile, the data points in different classes are separated very well with a small number of pairwise constraints.

4.2. Application to data clustering

Kmeans is a classical clustering algorithm, which is popularly used in many applications. During iterations, it needs a distance metric to calculate the distances between data points and cluster centers. In the absence of prior knowledge, the Euclidean distance metric is often employed in Kmeans algorithm. Here we use a learned Mahalanobis distance metric to replace it.

We use a normalized accuracy score to evaluate the clustering algorithms [22]. For two-cluster data, the accuracy measure is evaluated as follows

$$\text{accuracy} = \sum_{i>j} \frac{\delta\{\delta\{c_i = c_j\} = \delta\{\hat{c}_i = \hat{c}_j\}\}}{0.5n(n-1)} \tag{20}$$

where $\delta\{\cdot\}$ is an indicator (δ(true)=1 and δ(false)=0), $\hat{c}_i$ is the cluster to which $\mathbf{x}_i$ is assigned by the clustering algorithm, and c_i is the "correct" assignment. The score above is equivalent to calculating the probability that for $\mathbf{x}_i$ and $\mathbf{x}_j$ drawn randomly from the data set, their assignment $(\hat{c}_i, \hat{c}_j)$ by the clustering algorithm agrees with their true assignment (c_i, c_j).

As described in Refs. [22], this score should be normalized when the number of the clusters is greater than 2. Normalization can be achieved by selecting the point pairs from the same cluster (as determined by $\hat{c}$) and from the different clusters with equal probability. As a result, the "matches" and the "dis-matches" are given the same weight.

Table 3
A brief description of the data sets

	Breast	Diabetes	Iris	Protein	ORL	COIL
Number of samples (N)	683	768	150	116	400	1440
Input dimensionality (n)	10	8	4	20	10 304	256
Number of clusters (C)	2	2	3	6	40	20
Dimensionality (d)	5	4	2	8	60	60
K_c (small must-link set $\mathscr{S}$)	612	694	133	92	280	1008
K_c (large must-link set $\mathscr{S}$)	470	611	116	61	200	720

The data sets we used are described as follows (see Table 3):

The UCI data sets: We performed our algorithm on four data sets: *breast*, *diabetes*, *iris*, and *protein*.

The ORL database: It includes 40 distinct individuals and each individual has 10 gray images with different expressions and facial details [61]. The size of each image is 112×92. The source dimensionality of data points is 10 304.

The COIL-20 database: It includes 20 objects [62], each of which has 72 gray images, which are taken from different view directions. In experiments, each image is down-sampled to be one with 16×16 pixels. Thus, the input dimensionality is 256.

In experiments, the "true" clustering is given by the class labels of the data points. The must-links in $\mathscr{S}$ are randomly selected from the sets of point pairs within the same classes. A "small" must-link subset and a "large" must-link subset are generated for comparison. Here, "small" and "large" are evaluated via the number of connected

[4] Available at http://www.ics.uci.edu/mlearn/MLRepository.html.

Table 4
Clustering accuracy and standard deviation of accuracy on six data sets

Data set	Method	Accuracy (%)	Std. (%)	Accuracy (%)	Std. (%)
Breast	Kmeans	94.2	–	94.2	–
	Xing's	94.2	0.3	94.3	0.3
	RCA	93.3	0.3	94.3	0.7
	DCA	92.0	1.9	93.5	0.9
	Our	**94.4**	0.3	**94.5**	0.2
Diabetes	Kmeans	55.8	–	55.8	–
	Xing's	56.6	2.8	60.1	2.3
	RCA	58.3	3.0	60.5	3.1
	DCA	57.5	4.2	60.3	2.9
	Our	**60.9**	2.3	**62.5**	2.2
Iris	Kmeans	85.5	–	85.5	–
	Xing's	92.1	0.2	93.2	0.1
	RCA	95.9	2.3	97.0	1.4
	DCA	95.5	2.5	96.6	2.2
	Our	**96.6**	1.4	**97.1**	1.3
Protein	Kmeans	66.2	–	66.2	–
	Xing's	68.1	2.6	71.0	2.4
	RCA	68.2	2.2	**81.3**	2.3
	DCA	62.4	2.5	65.1	5.8
	Our	**73.6**	2.3	77.8	2.4
COIL	Kmeans	82.5	–	82.5	–
	Xing's	87.1	3.7	89.2	3.6
	RCA	93.6	0.8	**94.5**	0.5
	DCA	93.4	0.9	94.2	1.1
	Our	**93.9**	0.6	94.1	0.6
ORL	Kmeans	84.1	–	84.1	–
	Xing's	85.0	1.0	86.1	1.5
	RCA	61.5	0.7	68.0	1.3
	DCA	85.0	1.3	86.5	1.8
	Our	**94.7**	1.0	**96.3**	0.7

components K_c [22].[5] For the UCI data sets, the "small" $\mathscr{S}$ is randomly chosen so that the resulting number of connected components K_c is equal to about 90% of the size of the original data sets. In the case of "large" $\mathscr{S}$, this number is changed to be about 70%. For COIL, ORL data sets, these two numbers are changed to be about 70% and 50%, respectively. Table 3 lists the number of K_c. Note that here only a small number of pairwise constraints are employed to learn the distance metric, compared with all the pairwise constraints we can select. Finally, the cannot-links in $\mathscr{D}$ are generated based on the data points in $\mathscr{S}$, but with different clusters.

RCA, DCA, Xing's method and our method are used to learn distance metrics for comparisons. In each experiment, the null space of $\hat{\mathbf{S}}_w + \hat{\mathbf{S}}_b$ is eliminated. The results obtained by standard Kmeans, Kmeans+Xing's method, Kmeans+RCA, Kmeans+DCA and Kmeans+our-method are reported in Table 4. Two group of experimental results are given by averaging 20 trials. The left group corresponds to the "small" $\mathscr{S}$, while the right group corresponds to the "large" $\mathscr{S}$.

With a learned distance metric, the performance of Kmeans is significantly improved. Compared with DCA and Xing's method, in most cases our method achieves higher accuracy, especially when applied to high dimensional data. The experimental results also indicate that our method is competitive with RCA. It is more robust than RCA as it avoids the singularity problem. Actually in experiments, with RCA the performance may be stopped due to singularity problem. Additionally, it may generate very low accuracy of clustering. For example, when we test ORL data set with RCA, the accuracy is very low, even lower than that of Kmeans algorithm. Actually, it is difficult to accurately estimate the information entropy in RCA only from a small number of samples in the case of high-dimensionality.

[5] Note that the larger K_c is, the smaller the number of must-links we can obtain, and thus the smaller the size of $\mathscr{S}$ is.

Table 5
Computation time (second) of learning the Mahalanobis matrix from the "small" $\mathscr{S}$ and $\mathscr{D}$ on a PC with 1.7 GHz CPU and 512 RAM, using Matlab 6.5

	Breast	Diabetes	Iris	Protein	ORL	COIL
Xing's	7.201	10.11	1.261	2.594	333.2	7443.5
RCA	0.003	0.002	0.002	0.015	1.291	1.472
DCA	0.001	0.001	0.007	0.012	1.491	1.403
Our	0.004	0.010	0.008	0.013	4.290	6.391

Table 5 lists the computation time. Our method is much faster than Xing's method. It is slightly slower than RCA and DCA, due to the iterative algorithm.

4.3. Application to interactive natural image segmentation

Extracting the foreground objects in natural images is one of the most fundamental tasks in image understanding. In spite of many thoughtful efforts, it is still a very challenging problem. Recently, some interactive segmentation frameworks are developed to reduce the complexity of segmentation (more references can be obtained through Refs. [63,64]). In interactive segmentation frameworks, an important issue is to compute the likelihood values of each pixel to the user specified strokes. These values are usually obtained with Euclidean distance metric. Here we use a learned Mahalanobis distance metric to calculate them. We demonstrate that with a learned distance metric even a simple classifier as KNN classifier could generate satisfactory segmentation results.

The steps of learning a distance metric are as follows: (1) collect the user specified pixels about the background and foreground; (2) encode all possible labeled pixel pairs to get the must-links $\mathscr{S}$ and cannot-links $\mathscr{D}$; (3) learn a Mahalanobis distance metric according to the algorithm described in Table 1.

In experiments, each pixel p is described as a 5-dimensional vector, i.e., $\mathbf{x}_p = [r, g, b, x, y]^T$, in which (r, g, b) is the normalized color of pixel p and (x, y) is its spatial coordinate normalized with image width and height. The learned distance metric with $d=3$ is employed to replace the Euclidean distance metric when using KNN classifier ($K = 1$ in experiments) to infer the class labels of the pixels.

Fig. 2 shows some experimental results. The first row shows the four source images with the user specified pixels about the background and foreground. The labeled pixels are grouped as pairwise constraints of must-links and cannot-links to learn a distance metric with Xing's method, RCA, DCA and our method. From the second to the sixth row are the segmented results by KNN classifier with standard Euclidean distance metric, KNN classifier with the learned distance metric by Xing's method, RCA, DCA and our method, respectively. We can see that with the standard Euclidean distance metric, KNN classifier fails to generate satisfactory segmentation results. Actually, in Euclidean distance metric, color and coordinate are given equal weight. If the pixels are far from the labeled region, the spatial distance will be greater than the color distance, and these pixels may be classified incorrectly, for example, those pixels near the circle in the pyramid image (see the third column in Fig. 2). However, color and coordinate may have different weights for segmentation. These weights are learned into the Mahalanobis matrix A. We can see that with the learned distance metric, the performance of KNN classifier is significantly improved with RCA, DCA and our method. In contrast with the standard Euclidean distance metric, Xing's method generates similar segmented results.

Compared with RCA and DCA, our method generate more accurate results. Taking the flower image as an example, Fig. 3 shows two segmented regions with original image resolution for comparison.

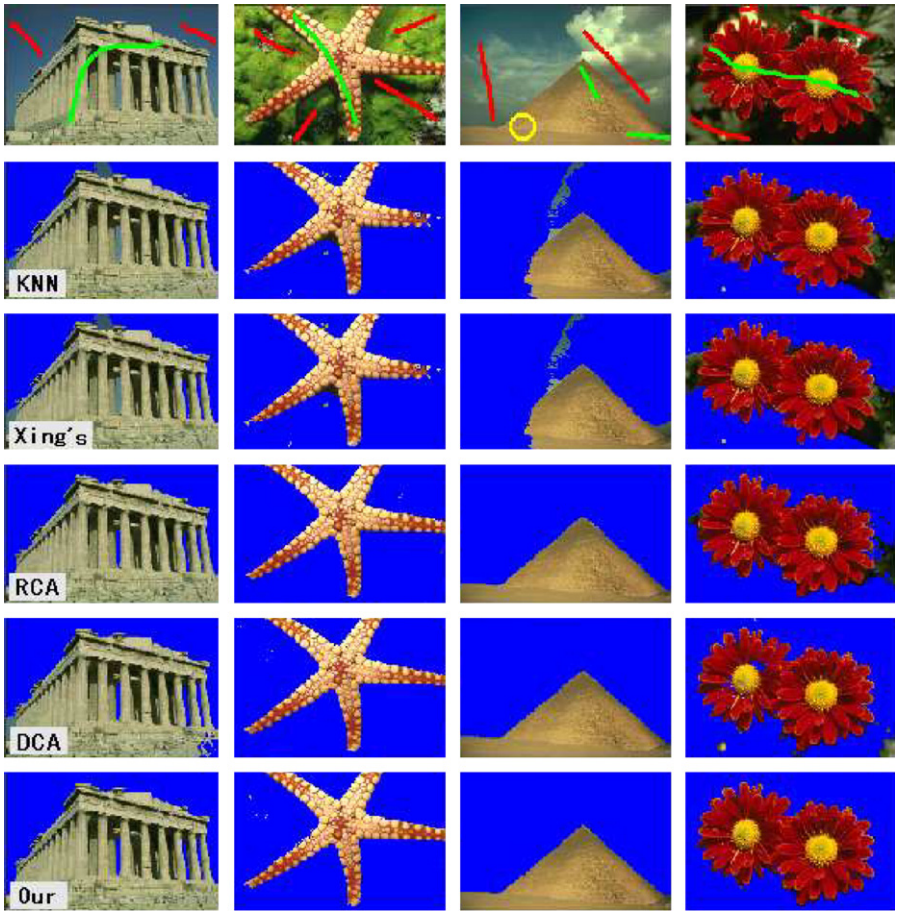

Fig. 2. Four image segmentation experiments. The first row shows the original image with user strokes. From the second to the sixth row are the segmented results with KNN classifier with standard Euclidean distance metric, KNN + Xing's method, KNN + RCA, KNN + DCA and KNN + our method.

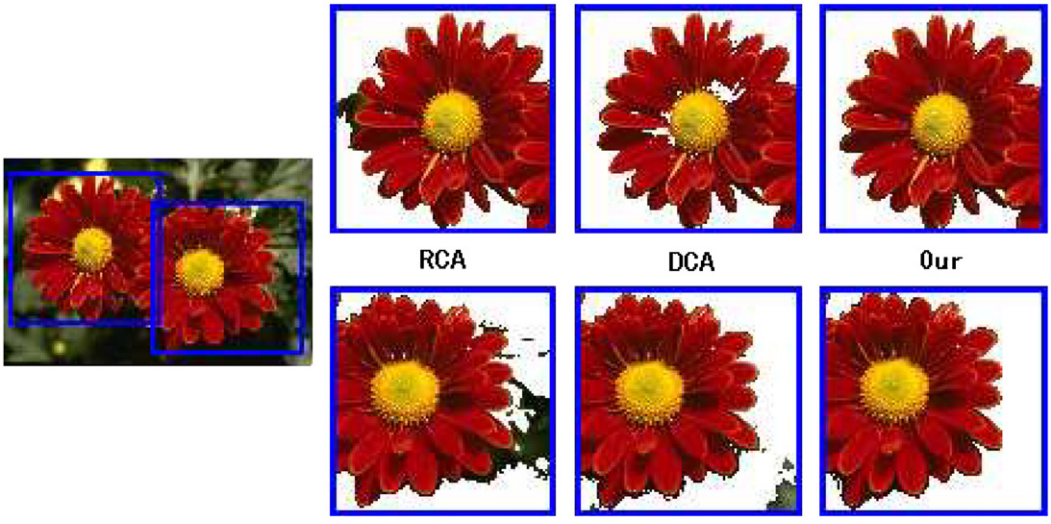

Fig. 3. Details in two segmented regions with original image resolution. In the right panel, the first and the second column show the results by RCA and DCA, and the third column reports the results by our algorithm.

In the right panel, the first and the second columns show the results by RCA and DCA, and the third column reports the results by our algorithm. From the details we can see better segmentation is achieved by our method.

4.4. Application to face pose estimation

Face recognition is a challenging research direction in pattern recognition. Many existing face recognition systems can generate

Fig. 4. The face images of 10 subjects used to construct the must-links and cannot-links.

higher recognition accuracy from frontal face images. However, in most real-world applications, the subject is free of the camera and the system may receive face images with different poses. Thus, estimating the face pose is an important preprocess in face recognition to improve the robustness of the face recognition system. Here we show an experiment in which a Mahalanobis distance metric is learned from a small number of instances about similar poses and dissimilar poses to help estimate the poses of new subjects, which are not included in the training database.

The images of 15 subjects are used from the pose database [67]. For each subject with zero vertical pose angle, we use 13 horizontal pose angles varying from -90° to 90° (every 15° a pose) to conduct the experiment. Totally, we have 195 face images. We use 10 subjects in the database to supply the instances of must-links and cannot-links. The images of the rest five subjects are used as query samples whose face poses are to be estimated. Thus the training data set does not include the test data set. To be clear, we show the images of the 10 subjects for training in Fig. 4 and the images of the rest five subjects for test in Fig. 5.

In this experiment, we do not consider the identification of the face images, but consider the similar/dissimilar face poses. Here, a must-link is defined to connect a pair of face images with the angle difference of poses not greater than 15°, while a cannot-link is defined to connect a pair of face images with the angle difference of poses greater than 45°. In each trial, we randomly select 100 must-links to construct the subset $\mathscr{S}$. This number equals to about 17% (100/585) of the total eligible candidates. We also randomly select 1000 cannot-links to construct the subset $\mathscr{D}$. This number equals to about 23% (1000/4400) of the total eligible candidates. In this experiment, 20 trials are conducted to evaluate the performance.

To run the algorithm, all the images are resized to be 48×36 pixels. The source dimensionality is 1728 and it is reduced to 120 by performing principal component analysis. In computation, we set the parameter d to be 60. When the optimal Mahalanobis distance matrix $\mathbf{A}$ is learned, we use Eq. (1) to calculate the distance between the new images in Fig. 5 and those in Fig. 4. Thus, for each image in Fig. 5, we can get 130 distances. We sort them in ascending order and use the first 10 ranks to estimate the pose of the new images. This treatment is just as the same as image retrieval from database. Fig. 6 shows an example obtained in one trial. The query image is the last image in the fourth row in Fig. 5. Compared with Xing's method, RCA and DCA, we see that the poses of the images obtained with our method are closer to that of the query image.

To give a comprehensive evaluation, the errors of the estimated pose angles are calculated. They are first calculated on each trial, and then further averaged on all of the 20 trails.

Fig. 5. The face images of five subjects whose poses are estimated.

Fig. 6. The first 10 images with the most similar poses to the query image, which are estimated from the images shown in Fig. 4, obtained by Xing's method, RCA, DCA and our method.

Specifically, in each trial and for each image in Fig. 5, we use the average of the poses angles of the first 10 ranked images as its estimated pose angle. This can be done since the pose angles of the images in Fig. 4 are all known. Then, the absolute error is calculated as the difference between the estimated pose angle and the true pose angle. Thus, we obtain an error matrix with five rows and 13 columns. That is, each row of this matrix corresponds to a new subject shown in Fig. 5, and records the angle errors of its 13 poses. We further average these errors column by column, and then get a row vector of average errors for 13 poses. In this way, we finish the computation in this trial.

Finally, we further average the error vectors obtained via 20 trials. Fig. 7 shows the final error curves. As can be seen, the average errors of the estimated pose angles by our method are less than those obtained by the other methods. The largest error in our method is only up to 18.8°, the smallest is 7.3°, and most errors are located near about 8.5°.

5. Conclusion

In summary, this paper addresses a general problem of learning a Mahalanobis distance metric from side information. It is formulated as a constrained optimization problem, in which the *ratio of distances* (in terms of ratio of matrix traces) is used as the objective function. An optimization algorithm is proposed to solve this problem, in which a lower bound and an upper bound including the optimum are explicitly estimated and then used to stipulate the initial value for iterations. Experimental results show that with a small number of pairwise constraints our algorithm can provide a good distance metric for performances.

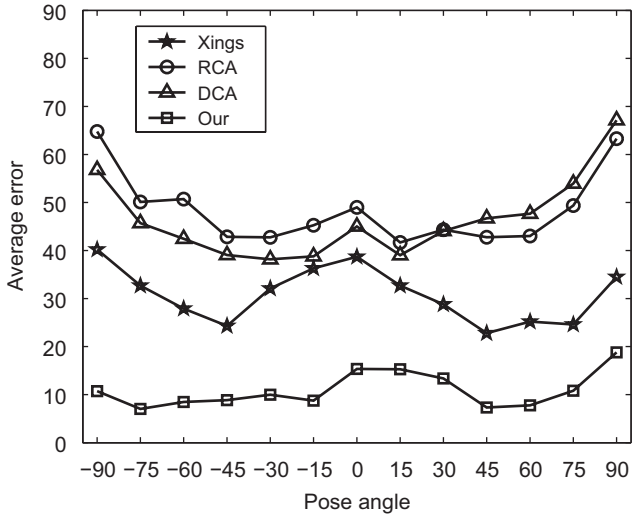

Fig. 7. Error curves of 13 face poses in Xing's method, RCA, DCA and our method.

Except for the significant improvement of the learned distance metric over the Euclidean distance metric, there still exist a few aspects to be researched. Intrinsically, our algorithm adopts a binary search approach to find the optimum. More fast iteration algorithms will be investigated in the future. We will also develop incremental learning version of our algorithm for online data processing.

Acknowledgments

This work is supported by the projects (Grant No. 60721003 and 60675009) of the National Natural Science Foundation of China. The anonymous reviewers have helped to improve the quality and representation of this paper.

Appendix A. Proof of Theorem 3

Lemma 3. *If* $\mathbf{A}$ *is positive semi-definite, then* $\forall \mathbf{x}$, $\mathbf{x}^{\mathrm{T}}\mathbf{A}\mathbf{x}=0 \Leftrightarrow A\mathbf{x}=0$.

Proof. Since $\mathbf{A}$ is semi-definite, then there exists a matrix $\mathbf{B}$ such that $\mathbf{A}=\mathbf{B}^{\mathrm{T}}\mathbf{B}$ [59]. On the one hand, $\forall \mathbf{x}$, $\mathbf{x}^{\mathrm{T}}A\mathbf{x}=0 \Rightarrow \mathbf{x}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}\mathbf{B}\mathbf{x}=0 \Rightarrow (\mathbf{B}\mathbf{x})^{\mathrm{T}}\mathbf{B}\mathbf{x}=0 \Rightarrow \mathbf{B}\mathbf{x}=0 \Rightarrow \mathbf{B}^{\mathrm{T}}\mathbf{B}\mathbf{x}=0 \Rightarrow \mathbf{A}\mathbf{x}=0$. On the other hand, $\mathbf{A}\mathbf{x}=0 \Rightarrow \mathbf{x}^{\mathrm{T}}\mathbf{A}\mathbf{x}=0$ holds naturally. Thus we have $\mathbf{x}^{\mathrm{T}}\mathbf{A}\mathbf{x}=0 \Leftrightarrow \mathbf{A}\mathbf{x}=0$. □

Lemma 4. *Let null*$(\cdot)$ *denote the null space of a matrix. If* $\mathbf{A}\in\mathbb{R}^{n\times n}$ *and* $\mathbf{B}\in\mathbb{R}^{n\times n}$ *are positive semi-definite, then null*$(\mathbf{A}+\mathbf{B})=null(\mathbf{A})\cap null(\mathbf{B})$.

Proof. Note that $\mathbf{A}+\mathbf{B}$ is also positive semi-definite. According to Lemma 3, $\forall \mathbf{x}\in null(\mathbf{A}+\mathbf{B}) \Rightarrow (\mathbf{A}+\mathbf{B})\mathbf{x}=0 \Rightarrow \mathbf{x}^{\mathrm{T}}(\mathbf{A}+\mathbf{B})\mathbf{x}=0 \Rightarrow \mathbf{x}^{\mathrm{T}}\mathbf{A}\mathbf{x}+\mathbf{x}^{\mathrm{T}}\mathbf{B}\mathbf{x}=0 \Rightarrow \mathbf{x}^{\mathrm{T}}\mathbf{A}\mathbf{x}=0 \wedge \mathbf{x}^{\mathrm{T}}\mathbf{B}\mathbf{x}=0 \Rightarrow \mathbf{A}\mathbf{x}=0 \wedge \mathbf{B}\mathbf{x}=0$. Thus, $\mathbf{x}\in null(\mathbf{A})$ and $\mathbf{x}\in null(\mathbf{B})$.

On the other hand, $\forall \mathbf{x}\in(null(\mathbf{A})\cap null(\mathbf{B})) \Rightarrow \mathbf{x}\in null(\mathbf{A}+\mathbf{B})$ can be easily justified. Finally, we obtain $null(\mathbf{A}+\mathbf{B})=null(\mathbf{A})\cap null(\mathbf{B})$. □

Lemma 5. *Let* $\mathbf{A}\in\mathbb{R}^{n\times n}$, $\mathbf{B}\in\mathbb{R}^{n\times n}$ *and* $W\in\mathbb{R}^{n\times d}$. *Eliminating the null space of* $\mathbf{A}+\mathbf{B}$ *will not affect the value of* $\mathrm{tr}(\mathbf{W}^{\mathrm{T}}\mathbf{A}\mathbf{W})/\mathrm{tr}(\mathbf{W}^{\mathrm{T}}\mathbf{B}\mathbf{W})$.

Proof. Let $\mathbf{W}_0\in\mathbb{R}^{n\times k_0}$, $\mathbf{W}_1\in\mathbb{R}^{n\times k_1}$, and $k_0+k_1=n$. Suppose the column vectors of $\mathbf{W}_0$ are a base of $null(\mathbf{A}+\mathbf{B})$ and those of $\mathbf{W}_1$ are a base of its orthogonal complement space. According to linear algebra, for $\mathbf{W}\in\mathbb{R}^{n\times d}$, there exist two coefficient matrices $\alpha_0\in R^{k_0\times d}$ and $\alpha_1\in R^{k_1\times d}$ such that $\mathbf{W}$ can be linearly represented:

$$\mathbf{W}=\mathbf{W}_0\cdot\alpha_0+\mathbf{W}_1\cdot\alpha_1. \tag{21}$$

Based on Lemma 4 and Eq. (21), then

$$\frac{\mathrm{tr}(\mathbf{W}^{\mathrm{T}}\mathbf{A}\mathbf{W})}{\mathrm{tr}(\mathbf{W}^{\mathrm{T}}\mathbf{B}\mathbf{W})}=\frac{\mathrm{tr}((\mathbf{W}_1\alpha_1)^{\mathrm{T}}\mathbf{A}(\mathbf{W}_1\alpha_1))}{\mathrm{tr}((\mathbf{W}_1\alpha_1)^{\mathrm{T}}\mathbf{B}(\mathbf{W}_1\alpha_1))}$$

This indicates that the null space of $\mathbf{A}+\mathbf{B}$ will not affect the value of $\mathrm{tr}(\mathbf{W}^{\mathrm{T}}\mathbf{A}\mathbf{W})/\mathrm{tr}(\mathbf{W}^{\mathrm{T}}\mathbf{B}\mathbf{W})$. □

Proof of Theorem 3. Let π be the orthogonal complement space of $null(\hat{\mathbf{S}}_b+\hat{\mathbf{S}}_w)$. Lemma 5 indicates that we can consider $\mathrm{tr}(\mathbf{W}^{\mathrm{T}}\hat{\mathbf{W}}_b\mathbf{A}\mathbf{W})/\mathrm{tr}(\mathbf{W}^{\mathrm{T}}\hat{\mathbf{S}}_w\mathbf{W})$ in this space.

Suppose the column vectors of $\mathbf{W}_1\in\mathbb{R}^{n\times k_1}$ consist of a base of π and $\mathbf{W}_1^{\mathrm{T}}\mathbf{W}_1=\mathbf{I}$. $\forall \mathbf{W}\in R^{n\times d}\subset\pi$ and $\mathbf{W}^{\mathrm{T}}\mathbf{W}=\mathbf{I}$, there exists a coefficient matrix α_1 such that $\mathbf{W}=\mathbf{W}_1\cdot\alpha_1$. Here $\alpha_1\in\mathbb{R}^{k_1\times d}$ and $\alpha_1^{\mathrm{T}}\alpha_1=\mathbf{I}$. Then

$$\begin{aligned}\max_{\mathbf{W}^{\mathrm{T}}\mathbf{W}=\mathbf{I}}\frac{\mathrm{tr}(\mathbf{W}^{\mathrm{T}}\hat{\mathbf{S}}_b\mathbf{W})}{\mathrm{tr}(\mathbf{W}^{\mathrm{T}}\hat{\mathbf{S}}_w\mathbf{W})}&=\max_{\substack{\mathbf{W}^{\mathrm{T}}\mathbf{W}=\mathbf{I}\\ W\subset\pi}}\frac{\mathrm{tr}(\mathbf{W}^{\mathrm{T}}\hat{\mathbf{S}}_b\mathbf{W})}{\mathrm{tr}(\mathbf{W}^{\mathrm{T}}\hat{\mathbf{S}}_w\mathbf{W})}\\&=\max_{\alpha_1^{\mathrm{T}}\alpha_1=\mathbf{I}}\frac{\mathrm{tr}(\alpha_1^{\mathrm{T}}\mathbf{W}_1^{\mathrm{T}}\hat{\mathbf{S}}_b\mathbf{W}_1\alpha_1)}{\mathrm{tr}(\alpha_1^{\mathrm{T}}\mathbf{W}_1^{\mathrm{T}}\hat{\mathbf{S}}_w\mathbf{W}_1\alpha_1)}.\end{aligned} \tag{22}$$

Now we introduce a linear transformation $\mathbf{y}=\mathbf{W}_1^{\mathrm{T}}\mathbf{x}$ and denote the covariance matrices of the transformed point pairs in $\mathscr{S}$ and $\mathscr{D}$ by $\tilde{\mathbf{S}}_w$ and $\tilde{\mathbf{S}}_b$.

We can see that $\tilde{\mathbf{S}}_w=\mathbf{W}_1^{\mathrm{T}}\hat{\mathbf{S}}_w\mathbf{W}_1$ and $\tilde{\mathbf{S}}_b=\mathbf{W}_1^{\mathrm{T}}\hat{\mathbf{S}}_b\mathbf{W}_1$. Introducing a new notation $\tilde{\mathbf{W}}$, Eq. (22) is re-written as follows:

$$\max_{\mathbf{W}^{\mathrm{T}}\mathbf{W}=\mathbf{I}}\frac{\mathrm{tr}(\mathbf{W}^{\mathrm{T}}\hat{\mathbf{S}}_b\mathbf{W})}{\mathrm{tr}(\mathbf{W}^{\mathrm{T}}\hat{\mathbf{S}}_w\mathbf{W})}=\max_{\tilde{\mathbf{W}}^{\mathrm{T}}\tilde{\mathbf{W}}=\mathbf{I}}\frac{\mathrm{tr}(\tilde{\mathbf{W}}^{\mathrm{T}}\tilde{\mathbf{S}}_b\tilde{\mathbf{W}})}{\mathrm{tr}(\tilde{\mathbf{W}}^{\mathrm{T}}\tilde{\mathbf{S}}_w\tilde{\mathbf{W}})}, \tag{23}$$

here $\tilde{\mathbf{W}}\in\mathbb{R}^{k_1\times d}$. Thus we finish the proof. □

Appendix B. Proof of Theorem 4

Proof. Suppose λ is a non-zero eigenvalue of $\mathbf{AB}$ and $\mathbf{v}$ is its corresponding eigenvector. Thus, $\mathbf{ABv}=\lambda\mathbf{v}\neq\mathbf{0} \Rightarrow \mathbf{Bv}\neq\mathbf{0}$ and $\mathbf{ABv}=\lambda\mathbf{v} \Rightarrow \mathbf{BABv}=\lambda\mathbf{Bv}$. Therefore, $\mathbf{Bv}$ is an eigenvector of $\mathbf{BA}$ corresponding to the same non-zero eigenvalue λ.

On the other hand, suppose λ is a non-zero eigenvalue of $\mathbf{BA}$ and $\mathbf{u}$ is its corresponding eigenvector. We can also justify that $\mathbf{Au}$ is an eigenvector of $\mathbf{B}$ corresponding to the same non-zero eigenvalue λ. Therefore, $\mathbf{AB}$ and $\mathbf{BA}$ have the same non-zero eigenvalues, and for each non-zero eigenvalue, if the corresponding eigenvector of $\mathbf{AB}$ is $\mathbf{v}$, then the corresponding eigenvector of $\mathbf{BA}$ is $\mathbf{u}=\mathbf{Bv}$. □

References

[1] T. Hastie, R. Tibshirani, Discriminant adaptive nearest neighbor classification, IEEE Trans. Pattern Anal. Mach. Intell. 18 (6) (1996) 607–615.
[2] H. Muller, T. Pun, D. Squire, Learning from user behavior in image retrieval: application of market basket analysis, Int. J. Comput. Vision 56 (1–2) (2004) 65–77.
[3] X. He, O. King, W.Y. Ma, M. Li, H.J. Zhang, Learning a semantic space from users relevance feedback for image retrieval, IEEE Trans. Circuits Systems Video Technol. 13 (1) (2003) 39–48.
[4] J. Peltonen, A. Klami, S. Kaski, Learning more accurate metrics for self-organizing maps, in: International Conference on Artificial Neural Networks, Madrid, Spain, 2002, pp. 999–1004.
[5] C. Domeniconi, D. Gunopulos, Adaptive nearest neighbor classification using support vector machines, in: Advances in Neural Information Processing Systems, vol. 14, 2002.
[6] J. Peng, D. Heisterkamp, H. Dai, Adaptive kernel metric nearest neighbor classification, in: International Conference on Pattern Recognition, Quebec City, Canada, 2002, pp. 33–36.

[7] R. Yan, A. Hauptmann, R. Jin, Negative pseudo-relevance feedback in content-based video retrieval, in: Proceedings of ACM on Multimedia, Berkeley, CA, USA, 2003, pp. 343–346.
[8] X. He, W.Y. Ma, H.J. Zhang, Learning an image manifold for retrieval, in: Proceedings of ACM on Multimedia, New York, USA, 2004, pp. 17–23.
[9] J.R. He, M.J. Li, H.J. Zhang, H.H. Tong, C.S. Zhang, Manifold ranking based image retrieval, in: Proceedings of ACM on Multimedia, New York, USA, 2004, pp. 9–16.
[10] A.S. Varde, E.A. Rundensteiner, C. Ruiz, M. Maniruzzaman, R. Jr., Learning semantics-preserving distance metrics for clustering graphical data, in: SIGKDD Workshop on Multimedia Data Mining: Mining Integrated Media and Complex Data, Chicago, IL, USA, 2005, pp. 107–112.
[11] G. Wu, E.Y. Chang, N. Panda, Formulating context-dependent similarity functions, in: Proceedings of ACM on Multimedia, Chicago, IL, USA, 2005, pp. 725–734.
[12] E.E. Korkmaz, G. Ucoluk, Choosing a distance metric for automatic word categorization, in: Proceedings of the Joint Conference on New Methods in Language Processing and Computational Natural Language Learning, Sydney, Australia, 1998, pp. 111–120.
[13] F. Li, J. Yang, J. Wang, A transductive framework of distance metric learning by spectral dimensionality reduction, in: Proceedings of International Conference on Machine Learning, Corvallis, Oregon, USA, 2007, pp. 513–520.
[14] L. Yang, R. Jin, Distance metric learning: a comprehensive survey, Technical Report, Michigan State University ⟨http://www.cse.msu.edu/~yangliu1/frame_survey_v2.pdf⟩, 2006.
[15] J. Goldberger, S. Roweis, G. Hinton, R. Salakhutdinov, Neighbourhood components analysis, in: Advances in NIPS, MIT Press, Cambridge, MA, USA, 2004, pp. 513–520.
[16] K. Weinberger, J. Blitzer, L. Saul, Distance metric learning for large margin nearest neighbor classification, in: Advances in NIPS, MIT Press, Cambridge, MA, USA, 2006, pp. 1473–1480.
[17] L. Torresani, K.C. Lee, Large margin component analysis, in: Advances in NIPS, MIT Press, Cambridge, MA, USA, 2007, pp. 505–512.
[18] A. Globerson, S. Roweis, Metric learning by collapsing classes, in: Advances in NIPS, MIT Press, Cambridge, MA, USA, 2006, pp. 451–458.
[19] Z.H. Zhang, J.T. Kwok, D.Y. Yeung, Parametric distance metric learning with label information, in: IJCAI, Acapulco, Mexico, 2003, pp. 1450–1452.
[20] G. Lebanon, Flexible metric nearest neighbor classification, Technical Report, Statistics Department, Stanford University, 1994.
[21] C.H. Hoi, W. Liu, M.R. Lyu, W.Y. Ma, Learning distance metrics with contextual constraints for image retrieval, in: Proceedings of Conference on Computer Vision and Pattern Recognition, vol. 2, New York, USA, 2006, pp. 2072–2078.
[22] E.P. Xing, A.Y. Ng, M.I. Jordan, S. Russell, Distance metric learning, with application to clustering with side-information, in: Advances in NIPS, MIT Press, Cambridge, MA, USA, 2003, pp. 505–512.
[23] A. Bar-hillel, T. Hertz, N. Shental, D. Weinshall, Learning distance functions using equivalence relations, J. Mach. Learn. Res. 6 (2005) 11–18.
[24] I.W. Tsang, J.T. Kwok, Distance metric learning with kernels, in: Proceedings of the International Conference on Artificial Neural Networks (ICANN), Istanbul, Turkey, 2003, pp. 126–129.
[25] R. Rosales, G. Fung, Learning sparse metrics via linear programming, in: SIGKDD, New York, USA, 2006, pp. 367–373.
[26] M. Schultz, T. Joachims, Learning a distance metric from relative comparisons, in: Advances in NIPS, MIT Press, Cambridge, MA, USA, 2004.
[27] L. Yang, R. Jin, R. Sukthankar, Y. Liu, An efficient algorithm for local distance metric learning, in: AAAI, Boston, USA, 2006.
[28] D. Mochihashi, G. Kikui, K. Kita, Learning nonstructural distance metric by minimum cluster distortions, in: Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, Barcelona, Spain, 2004, pp. 341–348.
[29] W. Tang, S. Zhong, Pairwise constraints-guided dimensionality reduction, in: SDM Workshop on Feature Selection for Data Mining, 2006.
[30] T.D. Bie, M. Momma, N. Cristianini, Efficiently learning the metric using side-information, in: International Conference on Algorithmic Learning Theory, Sapporo, Japan, 2003, pp. 175–189.
[31] N. Shental, A. Bar-Hillel, T. Hertz, D. Weinshall, Computing Gaussian mixture models with em using side-information, in: ICML Workshop on The Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining, 2003.
[32] 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 (1) (2004) 27–72.
[33] Z. Lu, T. Leen, Semi-supervised learning with penalized probabilistic clustering, in: Advances in NIPS, MIT Press, Cambridge, MA, USA, 2005, pp. 849–856.
[34] D.-Y. Yeung, H. Chang, Extending the relevant component analysis algorithm for metric learning using both positive and negative equivalence constraints, Pattern Recognition 39 (5) (2006) 1007–1010.
[35] J. Zhang, R. Yan, On the value of pairwise constraints in classification and consistency, in: Proceedings of International Conference on Machine Learning, Corvallis, Oregon, USA, 2007, pp. 1111–1118.
[36] K.Q. Weinberger, G. Tesauro, Metric learning for kernel regression, in: Proceedings of International Workshop on Artificial Intelligence and Statistics, Puerto Rico, 2007, pp. 608–615.
[37] T. Hertz, A. Bar-Hillel, D. Weinshall, Boosting margin based distance functions for clustering, in: Proceedings of International Conference on Machine Learning, Banff, Alberta, Canada, 2004, pp. 393–400.
[38] M. Bilenko, S. Basu, R.J. Mooney, Integrating constraints and metric learning in semi-supervised clustering, in: Proceedings of International Conference on Machine Learning, Banff, Alberta, Canada, 2004, pp. 81–88.
[39] J.V. Davis, B. Kulis, P. Jain, S. Sra, I.S. Dhillon, Information-theoretic metric learning, in: Proceedings of International Conference on Machine Learning, Corvallis, Oregon, USA, 2007, pp. 209–216.
[40] L. Yang, R. Jin, R. Sukthankar, Bayesian active distance metric learning, in: Proceedings of International Conference on Uncertainty in Artificial Intelligence, 2007.
[41] N. Kumar, K. Kummamuru, D. Paranjpe, Semi-supervised clustering with metric learning using relative comparisons, in: IEEE International Conference on Data Mining, New Orleans, Louisiana, USA, 2005, pp. 693–696.
[42] R. Rosales, G. Fung, Learning sparse metrics via linear programming, in: Proceedings of the International Conference on Knowledge Discovery and Data Mining, Chicago, IL, USA, 2005, pp. 367–373.
[43] I.W. Tsang, P.M. Cheung, J.T. Kwok, Kernel relevant component analysis for distance metric learning, in: Proceedings of the International Joint Conference on Neural Networks (IJCNN), Montreal, Canada, 2005, pp. 954–959.
[44] J. Kwok, I. Tsang, Learning with idealized kernels, in: Proceedings of International Conference on Machine Learning, Washington, DC, USA, 2003, pp. 400–407.
[45] Z. Zhang, Learning metrics via discriminant kernels and multidimensional scaling: toward expected Euclidean representation, in: Proceedings of International Conference on Machine Learning, Washington, DC, USA, 2003, pp. 872–879.
[46] J. Chen, Z. Zhao, J. Ye, H. Liu, Nonlinear adaptive distance metric learning for clustering, in: Conference on Knowledge Discovery and Data Mining, San Jose, USA, 2007, pp. 123–132.
[47] S. Shalev-Shwartz, Y. Singer, A.Y. Ng, Online and batch learning of pseudo-metrics, in: Proceedings of International Conference on Machine Learning, Banff, Alberta, Canada, 2004, pp. 94–101.
[48] F.R. Bach, M.I. Jordan, Learning spectral clustering, with application to speech separation, J. Mach. Learn. Res. 7 (2006) 1963–2001.
[49] E.-J. Ong, R. Bowden, Learning distances for arbitrary visual features, in: Proceedings of British Machine Vision Conference, vol. 2, Edinburgh, England, 2006, pp. 749–758.
[50] S. Chopra, R. Hadsell, Y. LeCun, Learning a similarity metric discriminatively, with application to face verification, in: Proceedings of International Conference on Computer Vision and Pattern Recognition, San Diego, USA, 2005, pp. 539–546.
[51] L. Yang, R. Jin, R. Sukthankar, B. Zheng, L. Mummert, M. Satyanarayanan, M. Chen, D. Jukic, Learning distance metrics for interactive search-assisted diagnosis of mammograms, in: SPIE Symposium on Medical Imaging: Computer-Aided Diagnosis, vol. 6514, 2007.
[52] R. Yan, J. Zhang, J. Yang, A. Hauptmann, A discriminative learning framework with pairwise constraints for video object classification, in: Proceedings of International Conference on Computer Vision and Pattern Recognition, Washington, DC, USA, 2004, pp. 284–291.
[53] C.J. Langmead, A randomized algorithm for learning mahalanobis metrics: application to classification and regression of biological data, in: Asia Pacific Bioinformatics Conference, Taiwan, China, 2006, pp. 217–226.
[54] E. Chang, B. Li, On learning perceptual distance function for image retrieval, in: Asia Pacific Bioinformatics Conference, Orlando, Florida, USA, 2002, pp. 4092–4095.
[55] P.N. Belhumeur, J.P. Hespanha, D.J. Kriegman, Eigenfaces vs. fisherfaces: recognition using class specific linear projection, IEEE Trans. Pattern Anal. Mach. Intelligence. 19 (7) (1997) 711–720.
[56] L. Chen, H. Liao, M. Ko, J. Lin, G. Yu, A new lda based face recognition system which can solve the small sample size problem, Pattern Recognition 33 (10) (2000) 1713–1726.
[57] H. Yu, J. Yang, A direct lda algorithm for high-dimensional data—with application to face recognition, Pattern Recognition 34 (10) (2001) 2067–2070.
[58] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, second ed., Wiley, New York, USA, 2000.
[59] G.H. Golub, C.F. van Loan, Matrix Computations, third ed., The Johns Hopkins University Press, Baltimore, MD, USA, 1996.
[60] Y.F. Guo, S.J. Li, J.Y. Yang, T.T. Shu, L.D. Wu, A generalized Foley–Sammon transform based on generalized fisher discriminant criterion and its application to face recognition, Pattern Recognition Lett. 24 (1) (2003) 147–158.
[61] F.S. Samaria, A.C. Harter, Parameterisation of a stochastic model for human face identification, in: Proceedings of Second IEEE Workshop on Applications of Computer Vision, 1994, pp. 138–142.
[62] S. Nene, S. Nayar, H. Murase, Columbia object image library (coil-20), Technical Report, Columbia University, 1996.
[63] Y.Y. Boykov, M.P. Jolly, Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images, in: International Conference on Computer Vision, Vancouver, Canada, 2001, pp. 105–112.
[64] Y. Li, J. Sun, C.K. Tang, H.Y. Shum, Lazy snapping, in: SIGGRAPH, Los Angeles, USA, 2004, pp. 303–307.
[65] S. Yan, X. Tang, Trace quotient problems revisited, in: European Conference on Computer Vision, vol. 2, Graz, Austria, 2006, pp. 232–244.

[66] H. Wang, S. Yan, D. Xu, X. Tang, T. Huang, Trace ratio vs. ratio trace for dimensionality reduction, in: Proceedings of Conference on Computer Vision and Pattern Recognition, 2007.

[67] N. Gourier, D. Hall, J. Crowley, Estimating face orientation from robust detection of salient facial features, in: ICPR Workshop on Visual Observation of Deictic Gestures, Cambridge, UK, 2004.

About the Author—SHIMING XIANG received his B.S. degree from Chongqing Normal University, China in 1993 and M.S. degree from Chongqing University, China in 1996, and Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences, China in 2004. From 1996 to 2001, he was a lecturer of Huazhong University of Science and Technology, Wuhan, China. He was a Post Doctor in the Department of Automation, Tsinghua University until 2006. He is currently an Associate Professor in the Institute of Automation, Chinese Academy of Science. His interests include computer vision, pattern recognition, machine learning, etc.

About the Author—FEIPING NIE received his B.S. degree from the Department of Computer Science, North China University of Water Conservancy and Electric Power, China in 2000, and M.S. degree from the Department of Computer Science, Lanzhou University, China in 2003. He is currently a Ph.D. candidate in the Department of Automation, Tsinghua University. His research interests focus on machine learning and its applications.

About the Author—CHANGSHUI ZHANG received his B.S. degree from the Department of Mathematics of Peking University, China in 1986, and Ph.D. degree from the Department of Automation, Tsinghua University in 1992. He is currently a Professor of Department of Automation, Tsinghua University. He is an Associate Editor of the Journal of Pattern Recognition. His interests include artificial intelligence, image processing, pattern recognition, machine learning, evolutionary computation and complex system analysis, etc.