Multi-Task Learning in Heterogeneous Feature Spaces

Yu Zhang & Dit-Yan Yeung
Department of Computer Science and Engineering
Hong Kong University of Science and Technology
Clear Water Bay, Kowloon, Hong Kong, China
{zhangyu,dyyeung}@cse.ust.hk

Abstract

Multi-task learning aims at improving the generalization performance of a learning task with the help of some other related tasks. Although many multi-task learning methods have been proposed, they are all based on the assumption that all tasks share the same data representation. This assumption is too restrictive for general applications. In this paper, we propose a multi-task extension of linear discriminant analysis (LDA), called multi-task discriminant analysis (MTDA), which can deal with learning tasks with different data representations. For each task, MTDA learns a separate transformation which consists of two parts, one specific to the task and one common to all tasks. A by-product of MTDA is that it can alleviate the labeled data deficiency problem of LDA. Moreover, unlike many existing multi-task learning methods, MTDA can handle binary and multi-class problems for each task in a generic way. Experimental results on face recognition show that MTDA consistently outperforms related methods.

Introduction

Multi-task learning (Caruana 1997; Baxter 1997; Thrun 1996) is a machine learning paradigm which aims at improving the generalization performance of a learning task with the help of some other related tasks. Early attempts were strongly inspired by human learning activities in that people often apply the knowledge gained from previous learning tasks to help learn a new task. Besides transferring the learning experience sequentially, learning experience can also be leveraged when multiple tasks are learned simultaneously. For example, a baby learning to recognize human faces also gains experience in recognizing other objects. Over the past decade, many multi-task learning methods have been proposed. Multi-task neural network (Caruana 1997) learns the hidden layer representation as a common data representation for all tasks. Multi-task feature learning (Argyriou, Evgeniou, and Pontil 2008) also learns a common data representation but under the regularization framework. Regularized multi-task support vector machine (SVM) (Evgeniou and Pontil 2004) assumes that all tasks are similar and incorporates this assumption into the objective function of conventional SVM as a regularization term. Task clustering methods (Thrun and O'Sullivan 1996; Bakker and Heskes 2003) partition all tasks into clusters and learn a common (or similar) data or model representation for all tasks in each cluster. More recently, some methods such as (Zhang and Yeung 2010) have been proposed to learn the task relationships under the regularization framework.

An underlying assumption shared by all multi-task learning methods proposed thus far is that different tasks use the same data representation, i.e., same feature space. While this assumption is valid for some applications, it is too restricted for other applications. For example, in some face recognition and object recognition applications, there are image databases collected under different environmental conditions and their data representations are also different. In the situation that each database contains only limited labeled data, it is desirable to utilize all databases to improve the generalization performance since all databases are about the same application. This thinking is in line with the spirit of multi-task learning. Unfortunately, existing multi-task learning methods cannot be applied directly to this scenario because different tasks have different data representations.

In view of this limitation of existing methods, we propose a new multi-task learning method, called multi-task discriminant analysis (MTDA), which can be seen as a multi-task extension of a widely used supervised dimensionality reduction technique called linear discriminant analysis (LDA) (Fukunaga 1991). Unlike simply pooling the data for multiple learning tasks together and learning a common transformation for all tasks, MTDA learns a separate transformation for each task. Each transformation consists of two parts, one specific to the corresponding task and one common to all tasks. The learning of MTDA is based on an objective function which is similar to that of the single-task LDA. The optimization problem can be solved by an alternating method in which each subproblem can guarantee global optimality. While most existing multi-task learning methods can only handle learning tasks with data sharing the same feature space, MTDA can naturally deal with heterogeneous feature spaces. A by-product of MTDA is that it can alleviate the labeled data deficiency problem of LDA (Chen et al. 2000) by exploiting the label information from other tasks to help improve the performance of LDA for one task. Moreover, while most existing multi-task classification methods are formulated directly for binary clas-

sification problems in each task and require nontrivial extension in order for them to handle multi-class problems, MTDA, like LDA on which it is based, can be applied to binary and multi-class problems for each task in a generic way. Experiments on face recognition applications demonstrate the effectiveness of our proposed method.

Background

We first give a quick review of LDA which also serves to introduce some definitions that will be used subsequently. Suppose we are given a training set of N labeled data points, $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$, where $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \{1, \ldots, C\}$. $\mathcal{D}$ consists of N points from C classes with N_k points from the kth class, i.e., $\sum_{k=1}^C N_k = N$. The between-class scatter matrix $\mathbf{S}_b$, within-class scatter matrix $\mathbf{S}_w$ and total scatter matrix $\mathbf{S}_t$ are defined as $\mathbf{S}_b = \sum_{k=1}^C \frac{N_k}{N}(\bar{\mathbf{m}}_k - \bar{\mathbf{m}})(\bar{\mathbf{m}}_k - \bar{\mathbf{m}})^T$, $\mathbf{S}_w = \sum_{k=1}^C \sum_{y_i=k} \frac{1}{N}(\mathbf{x}_i - \bar{\mathbf{m}}_k)(\mathbf{x}_i - \bar{\mathbf{m}}_k)^T$, $\mathbf{S}_t = \sum_{i=1}^N \frac{1}{N}(\mathbf{x}_i - \bar{\mathbf{m}})(\mathbf{x}_i - \bar{\mathbf{m}})^T$ where $\bar{\mathbf{m}} = (\sum_{i=1}^N \mathbf{x}_i)/N$ is the sample mean of the whole data set $\mathcal{D}$ and $\bar{\mathbf{m}}_k = (\sum_{y_i=k} \mathbf{x}_i)/N_k$ is the class mean of the kth class. It is easy to show that $\mathbf{S}_t = \mathbf{S}_b + \mathbf{S}_w$. Two objective functions have been widely used for LDA. The first one is in the ratio trace form (Fukunaga 1991):

$$\mathbf{W}^\star = \arg\max_{\mathbf{W}} \operatorname{tr}\Big((\mathbf{W}^T\mathbf{S}_w\mathbf{W})^{-1}\mathbf{W}^T\mathbf{S}_b\mathbf{W}\Big), \qquad (1)$$

and the second one is in the trace ratio form (Wang et al. 2007):

$$\mathbf{W}^\star = \arg\max_{\mathbf{W}^T\mathbf{W}=\mathbf{I}_l} \frac{\operatorname{tr}(\mathbf{W}^T\mathbf{S}_b\mathbf{W})}{\operatorname{tr}(\mathbf{W}^T\mathbf{S}_t\mathbf{W})}, \qquad (2)$$

where $\operatorname{tr}(\cdot)$ denotes the trace of a square matrix, $\mathbf{I}_l$ denotes the $l \times l$ identity matrix, l is the reduced dimensionality of the trace ratio form, and $\mathbf{W} \in \mathbb{R}^{d\times l}$ is the transformation matrix for dimensionality reduction. The solution of the ratio trace form can be obtained from the eigenvectors of $\mathbf{S}_w^{-1}\mathbf{S}_b$ corresponding to the largest eigenvalues. On the other hand, the trace ratio form has no analytical solution and has to resort to an iterative method to obtain the optimal solution. Specifically, if $\mathbf{W}^{(k)}$ denotes the solution at the kth iteration, then at the $(k+1)$th iteration, $\mathbf{W}^{(k+1)}$ can be obtained from the top eigenvectors of $\mathbf{S}_b - \lambda_k\mathbf{S}_t$ where $\lambda_k = \frac{\operatorname{tr}((\mathbf{W}^{(k)})^T\mathbf{S}_b\mathbf{W}^{(k)})}{\operatorname{tr}((\mathbf{W}^{(k)})^T\mathbf{S}_t\mathbf{W}^{(k)})}$. This procedure can be proved to converge to the globally optimal solution.

The two objective functions have advantages and disadvantages. The ratio trace form is computationally more efficient than the trace ratio form. On the other hand, the physical meaning of the trace ratio form is clearer than that of the ratio trace form because the numerator and denominator of the objective function in the trace ratio form represent the average between-class distance and average total distance in the low-dimensional space, respectively.

Multi-Task Discriminant Analysis

Suppose we are given m tasks $\{\mathcal{T}_i\}_{i=1}^m$. The training set $\mathcal{D}_i = \{(\mathbf{x}_j^i, y_j^i)\}_{j=1}^{n_i}$ for $\mathcal{T}_i$ contains n_i data points with $\mathbf{x}_j^i \in \mathbb{R}^{d_i}$ and its corresponding output $y_j^i \in \{1, \ldots, c_i\}$, where d_i is the dimensionality of the data in $\mathcal{D}_i$ and c_i is the number of classes in $\mathcal{D}_i$.

We do not assume that the data sets from different tasks share the same feature space and hence the feature spaces can be of different dimensionality. This makes MTDA applicable under more general settings than most existing multi-task learning methods.

Objective Function

We represent the transformation applied to $\mathcal{D}_i$ as $\mathbf{U}_i = \mathbf{W}_i\mathbf{P}$, where $\mathbf{U}_i \in \mathbb{R}^{d_i\times d}$, $\mathbf{W}_i \in \mathbb{R}^{d_i\times d'}$, $\mathbf{P} \in \mathbb{R}^{d'\times d}$ with $d' > d$. Here d' is the intermediate dimensionality. $\mathbf{P}$ is the common structure shared by all tasks representing some characteristics of the application itself in the common lower-dimensional space, and $\mathbf{W}_i$ captures the characteristics specific to $\mathcal{D}_i$. To a certain extent, this is similar to multi-task structure learning in (Ando and Zhang 2005). However, in our case, a subspace is shared in the latent space after transforming each data set $\mathcal{D}_i$ by $\mathbf{W}_i$, but for (Ando and Zhang 2005) which still assumes different tasks lie in the same feature space, a common subspace is directly shared in the original data representation.

The total, between-class and within-class scatter matrices for $\mathcal{D}_i$ are defined as $\mathbf{S}_t^i = \sum_{j=1}^{n_i} \frac{1}{n_i}(\mathbf{x}_j^i - \bar{\mathbf{m}}^i)(\mathbf{x}_j^i - \bar{\mathbf{m}}^i)^T$, $\mathbf{S}_b^i = \sum_{k=1}^{c_i} \frac{n_{ik}}{n_i}(\bar{\mathbf{m}}_k^i - \bar{\mathbf{m}}^i)(\bar{\mathbf{m}}_k^i - \bar{\mathbf{m}}^i)^T$ and $\mathbf{S}_w^i = \sum_{k=1}^{c_i}\sum_{y_j^i=k} \frac{1}{n_i}(\mathbf{x}_j^i - \bar{\mathbf{m}}_k^i)(\mathbf{x}_j^i - \bar{\mathbf{m}}_k^i)^T$, respectively, where $\bar{\mathbf{m}}^i$ is the sample mean of all data points in $\mathcal{D}_i$, n_{ik} is the number of data points belonging to the kth class in $\mathcal{D}_i$, and $\bar{\mathbf{m}}_k^i$ is the class mean of the kth class in $\mathcal{D}_i$. It is easy to verify that $\mathbf{S}_t^i = \mathbf{S}_b^i + \mathbf{S}_w^i$.

The optimization problem for MTDA is formulated as

$$\max_{\{\mathbf{W}_i\},\mathbf{P}} \quad \frac{\operatorname{tr}(\sum_{i=1}^m \mathbf{P}^T\mathbf{W}_i^T\mathbf{S}_b^i\mathbf{W}_i\mathbf{P})}{\operatorname{tr}(\sum_{i=1}^m \mathbf{P}^T\mathbf{W}_i^T\mathbf{S}_t^i\mathbf{W}_i\mathbf{P})}$$
$$\text{s.t.} \quad \mathbf{P}^T\mathbf{P} = \mathbf{I}_d, \mathbf{W}_i^T\mathbf{W}_i = \mathbf{I}_{d'} \text{ for } i = 1, \ldots, m, \qquad (3)$$

where $\{\mathbf{W}_i\}$ denotes the set of all $\mathbf{W}_i$. Since $\mathbf{S}_t^i = \mathbf{S}_b^i + \mathbf{S}_w^i$ and $\mathbf{S}_t^i$, $\mathbf{S}_b^i$ and $\mathbf{S}_w^i$ are all positive semidefinite, the optimal value of the objective function in (3) lies in $[0, 1]$.

Optimization Procedure

Note that it is difficult to solve the optimization problem (3) with respect to $\{\mathbf{W}_i\}$ and $\mathbf{P}$ jointly. Here we adopt an alternating method. More specifically, we first optimize the objective function with respect to each of the m matrices $\mathbf{W}_i$ when $\mathbf{W}^{-i} = \{\mathbf{W}_1, \ldots, \mathbf{W}_{i-1}, \mathbf{W}_{i+1}, \ldots, \mathbf{W}_m\}$ and $\mathbf{P}$ are fixed, and then optimize it with respect to $\mathbf{P}$ when $\{\mathbf{W}_i\}$ are fixed. This procedure is repeated until convergence. In what follows, we will present these two steps of the optimization procedure separately.

Optimizing w.r.t. $\mathbf{W}_i$ with fixed $\mathbf{W}^{-i}$ and $\mathbf{P}$

When $\mathbf{W}^{-i}$ and $\mathbf{P}$ are fixed, the optimization problem (3) becomes

$$\max_{\mathbf{W}_i} \frac{\operatorname{tr}(\mathbf{P}^T\mathbf{W}_i^T\mathbf{S}_b^i\mathbf{W}_i\mathbf{P}) + a^i d}{\operatorname{tr}(\mathbf{P}^T\mathbf{W}_i^T\mathbf{S}_t^i\mathbf{W}_i\mathbf{P}) + b^i d} \quad \text{s.t. } \mathbf{W}_i^T\mathbf{W}_i = \mathbf{I}_{d'}, \qquad (4)$$

where $a^i = \frac{1}{d}\operatorname{tr}(\sum_{j=1,j\neq i}^m \mathbf{P}^T\mathbf{W}_j^T\mathbf{S}_b^j\mathbf{W}_j\mathbf{P})$ and $b^i = \frac{1}{d}\operatorname{tr}(\sum_{j=1,j\neq i}^m \mathbf{P}^T\mathbf{W}_j^T\mathbf{S}_t^j\mathbf{W}_j\mathbf{P})$ are two constants. Since $\mathbf{W}_i^T\mathbf{W}_i = \mathbf{I}_{d'}$ and $\mathbf{P}^T\mathbf{P} = \mathbf{I}_d$ according to the constraints in (3), we have $\mathbf{P}^T\mathbf{W}_i^T\mathbf{W}_i\mathbf{P} = \mathbf{P}^T\mathbf{P} = \mathbf{I}_d$ and

so $\mathrm{tr}(\mathbf{P}^T\mathbf{W}_i^T\mathbf{W}_i\mathbf{P}) = d$. We plug this into (4) and obtain the following problem:

$$\max_{\mathbf{W}_i} \frac{\mathrm{tr}(\mathbf{P}^T\mathbf{W}_i^T\tilde{\mathbf{S}}_b^i\mathbf{W}_i\mathbf{P})}{\mathrm{tr}(\mathbf{P}^T\mathbf{W}_i^T\tilde{\mathbf{S}}_t^i\mathbf{W}_i\mathbf{P})} \quad \text{s.t. } \mathbf{W}_i^T\mathbf{W}_i = \mathbf{I}_{d'}, \tag{5}$$

where $\tilde{\mathbf{S}}_b^i = \mathbf{S}_b^i + a^i\mathbf{I}_{d_i}$ and $\tilde{\mathbf{S}}_t^i = \mathbf{S}_t^i + b^i\mathbf{I}_{d_i}$. Since this problem is different from the optimization problem for conventional LDA, we defer its discussion to the next subsection. From the analysis there, we can find the (globally) optimal solution for problem (5).

Optimizing w.r.t. $\mathbf{P}$ with fixed $\{\mathbf{W}_i\}$

When $\{\mathbf{W}_i\}$ are fixed, the problem (3) with respect to $\mathbf{P}$ becomes

$$\max_{\mathbf{P}} \frac{\mathrm{tr}\left[\mathbf{P}^T(\sum_{i=1}^m \mathbf{W}_i^T\mathbf{S}_b^i\mathbf{W}_i)\mathbf{P}\right]}{\mathrm{tr}\left[\mathbf{P}^T(\sum_{i=1}^m \mathbf{W}_i^T\mathbf{S}_t^i\mathbf{W}_i)\mathbf{P}\right]} \quad \text{s.t. } \mathbf{P}^T\mathbf{P} = \mathbf{I}_d. \tag{6}$$

Now problem (6) is also in the trace ratio form of LDA, so we can use the iterative method in (Wang et al. 2007) to find the (globally) optimal solution.

Since we can find the (globally) optimal solution in each step of the alternating method, the method can be guaranteed to find a local maximum for problem (3) (Bertsekas 1999).

Detailed Results

In this subsection, we provide details on solving the optimization problem (5). We first rewrite (5) as

$$\max_{\mathbf{W}_i} \frac{\mathrm{tr}(\mathbf{W}_i^T\tilde{\mathbf{S}}_b^i\mathbf{W}_i\mathbf{M})}{\mathrm{tr}(\mathbf{W}_i^T\tilde{\mathbf{S}}_t^i\mathbf{W}_i\mathbf{M})} \quad \text{s.t. } \mathbf{W}_i^T\mathbf{W}_i = \mathbf{I}_{d'}, \tag{7}$$

where $\mathbf{M} = \mathbf{P}\mathbf{P}^T$. The trace ratio form of the conventional LDA in (2) can be seen as a special case of (7) when $\mathbf{M}$ is an identity matrix. In summary, Table 1 shows an iterative algorithm for solving (7).

Table 1: Algorithm for solving optimization problem (7)

Input: $\tilde{\mathbf{S}}_b^i$, $\tilde{\mathbf{S}}_t^i$ and $\mathbf{M}$
1: Initialize $\mathbf{W}_i^{(0)}$; 2: For $k = 1, \ldots, N_{iter}$ 2.1: Compute the ratio α_k from $\mathbf{W}_i^{(k-1)}$ as: $\alpha_k = \frac{\mathrm{tr}((\mathbf{W}_i^{(k-1)})^T\tilde{\mathbf{S}}_b^i\mathbf{W}_i^{(k-1)}\mathbf{M})}{\mathrm{tr}((\mathbf{W}_i^{(k-1)})^T\tilde{\mathbf{S}}_t^i\mathbf{W}_i^{(k-1)}\mathbf{M})}$; 2.2: Solve the optimization problem $\mathbf{W}_i^{(k)} = \arg\max_{\mathbf{W}^T\mathbf{W}=\mathbf{I}_{d'}} \mathrm{tr}(\mathbf{W}^T(\tilde{\mathbf{S}}_b^i - \alpha_k\tilde{\mathbf{S}}_t^i)\mathbf{W}\mathbf{M})$; 2.3: Let $\mathbf{S} = \mathbf{W}_i^{(k)}(\mathbf{W}_i^{(k)})^T\tilde{\mathbf{S}}_t^i\mathbf{W}_i^{(k)}(\mathbf{W}_i^{(k)})^T$; 2.4: Let $\mathbf{W}_i^{(k)}$ be the eigenvector matrix of $\mathbf{S}$ corresponding to the top d' eigenvalues; 2.5: If $\|\mathbf{W}_i^{(k)} - \mathbf{W}_i^{(k-1)}\|_F \leq \varepsilon$ (here we set $\varepsilon = 10^{-4}$) break;
Output: $\mathbf{W}_i$

Before analyzing this algorithm, we first solve the optimization problem in step 2.2 of the algorithm. The following lemma is useful here.

Lemma 1 ((Anderson 2003), pp. 645) *Let $\mathbf{A}$ and $\mathbf{B}$ be real $p \times p$ symmetric matrices and $\mathbf{W}$ be a $p \times p$ orthogonal matrix. Then*

$$\max_{\mathbf{W}^T\mathbf{W}=\mathbf{I}_p} \mathrm{tr}(\mathbf{W}\mathbf{A}\mathbf{W}^T\mathbf{B}) = \sum_{i=1}^p \lambda_i(\mathbf{A})\lambda_i(\mathbf{B}),$$

where $\lambda_i(\mathbf{M})$ denotes the ith largest eigenvalue of a matrix $\mathbf{M}$.

Although Lemma 1 does not explicitly tell us what the optimal solution of $\mathbf{W}$ is, it is easy to see that one optimal solution $\mathbf{W}^\star$ satisfies $\mathbf{W}^\star = \mathbf{U}_b\mathbf{U}_a^T$, where $\mathbf{U}_a$ and $\mathbf{U}_b$ are the eigenvector matrices of $\mathbf{A}$ and $\mathbf{B}$ in descending order of the eigenvalues.

We now present the solution of the optimization problem in step 2.2.

Theorem 1 *Let $\mathbf{A}$ be a real $p \times p$ symmetric matrix and $\mathbf{B}$ be a real $q \times q$ positive semidefinite matrix where $p > q$. Then*

$$\max_{\mathbf{W}\in\mathbb{R}^{p\times q}, \mathbf{W}^T\mathbf{W}=\mathbf{I}_q} \mathrm{tr}(\mathbf{W}^T\mathbf{A}\mathbf{W}\mathbf{B}) = \sum_{i=1}^q \lambda_i(\mathbf{A})\lambda_i(\mathbf{B}),$$

and the optimal solution $\mathbf{W}^\star$ satisfies $\mathbf{W}^\star = \mathbf{U}_{a1}\mathbf{U}_b^T\mathbf{Q}$, where $\mathbf{U}_{a1}$ is the eigenvector matrix of $\mathbf{A}$ corresponding to the top q eigenvalues, $\mathbf{U}_b$ is the eigenvector matrix of $\mathbf{B}$, and $\mathbf{Q}$ is any $q \times q$ orthogonal matrix.

Proof: We let $\tilde{\mathbf{B}}$ be $\begin{pmatrix} \mathbf{B} & \mathbf{0}_{q\times(p-q)} \\ \mathbf{0}_{(p-q)\times q} & \mathbf{0}_{(p-q)\times(p-q)} \end{pmatrix}$ and $\mathbf{W}' \in \mathbb{R}^{p\times(p-q)}$ be the orthogonal basis of the null space of $\mathbf{W}$, where $\mathbf{0}_{m\times n}$ denotes the $m \times n$ zero matrix. Then we denote $\tilde{\mathbf{W}}$ as $\tilde{\mathbf{W}} = [\mathbf{W}, \mathbf{W}']$ and so $\tilde{\mathbf{W}}$ and $\tilde{\mathbf{W}}^T$ are $p \times p$ orthogonal matrices. Since $\mathbf{W}\mathbf{B}\mathbf{W}^T = \tilde{\mathbf{W}}\tilde{\mathbf{B}}\tilde{\mathbf{W}}^T$, we have

$$\begin{aligned}\max_{\mathbf{W}^T\mathbf{W}=\mathbf{I}_q} \mathrm{tr}(\mathbf{W}^T\mathbf{A}\mathbf{W}\mathbf{B}) &= \max_{\tilde{\mathbf{W}}^T\tilde{\mathbf{W}}=\mathbf{I}_p} \mathrm{tr}(\tilde{\mathbf{W}}^T\mathbf{A}\tilde{\mathbf{W}}\tilde{\mathbf{B}}) \\ &= \sum_{i=1}^p \lambda_i(\mathbf{A})\lambda_i(\tilde{\mathbf{B}})\end{aligned}$$

by applying Lemma 1. Because of the relationship between $\mathbf{B}$ and $\tilde{\mathbf{B}}$ and the positive semidefiniteness of $\mathbf{B}$, it is easy to show that $\lambda_i(\mathbf{B}) = \lambda_i(\tilde{\mathbf{B}}) \geq 0$ for all $i \leq q$ and $\lambda_i(\tilde{\mathbf{B}}) = 0$ for all $i > q$. Plugging these into the above equation, we can get

$$\max_{\mathbf{W}^T\mathbf{W}=\mathbf{I}_q} \mathrm{tr}(\mathbf{W}^T\mathbf{A}\mathbf{W}\mathbf{B}) = \sum_{i=1}^q \lambda_i(\mathbf{A})\lambda_i(\mathbf{B}).$$

Moreover, the optimal solution $\tilde{\mathbf{W}}^\star$ satisfies $(\tilde{\mathbf{W}}^\star)^T = \mathbf{U}_{\tilde{b}}\mathbf{U}_a^T$ where $\mathbf{U}_a^T$ and $\mathbf{U}_{\tilde{b}}$ are the eigenvector matrices of $\mathbf{A}$ and $\tilde{\mathbf{B}}$. Considering the structure of $\tilde{\mathbf{B}}$, we have $\mathbf{U}_{\tilde{b}} = \begin{pmatrix} \mathbf{U}_b & \mathbf{0}_{q\times(p-q)} \\ \mathbf{0}_{(p-q)\times q} & \mathbf{R} \end{pmatrix}$ where $\mathbf{R} \in \mathbb{R}^{(p-q)\times(p-q)}$ is an orthogonal matrix. Then we can get the optimal solution $\mathbf{W}^\star$ for $\mathbf{W}$ as $\mathbf{W}^\star = \mathbf{U}_{a1}\mathbf{U}_b^T$. Since $\lambda_i(\mathbf{B}) = \lambda_i(\mathbf{Q}\mathbf{B}\mathbf{Q}^T)$ for all i when $\mathbf{Q}$ is a $q \times q$ orthogonal matrix, the optimal solution satisfies $\mathbf{W}^\star = \mathbf{U}_{a1}\mathbf{U}_b^T\mathbf{Q}$. □

Algorithm Analysis

In this subsection, we analyze the algorithm presented in Table 1. Let the objective function of problem (7) be denoted as $J(\mathbf{W}) = \frac{\mathrm{tr}(\mathbf{W}^T\tilde{\mathbf{S}}_b^i\mathbf{W}\mathbf{M})}{\mathrm{tr}(\mathbf{W}^T\tilde{\mathbf{S}}_t^i\mathbf{W}\mathbf{M})}$.

Lemma 2 *For the algorithm in Table 1, we have $J(\mathbf{W}_i^{(k)}) \geq J(\mathbf{W}_i^{(k-1)})$.*

Proof: We define $g(\mathbf{W}) = \mathrm{tr}(\mathbf{W}^T(\tilde{\mathbf{S}}_b^i - \alpha_k\tilde{\mathbf{S}}_t^i)\mathbf{W}\mathbf{M})$. Then $g(\mathbf{W}_i^{(k-1)}) = 0$ since $\alpha_k = \frac{\mathrm{tr}((\mathbf{W}_i^{(k-1)})^T\tilde{\mathbf{S}}_b^i\mathbf{W}_i^{(k-1)}\mathbf{M})}{\mathrm{tr}(\mathbf{W}_i^{(k-1)})^T\tilde{\mathbf{S}}_t^i\mathbf{W}_i^{(k-1)}\mathbf{M})}$. Because $\mathbf{W}_i^{(k)} = \arg\max_{\mathbf{W}^T\mathbf{W}=\mathbf{I}_{d'}} g(\mathbf{W})$ and $(\mathbf{W}_i^{(k-1)})^T\mathbf{W}_i^{(k-1)} = \mathbf{I}_{d'}$, we have $g(\mathbf{W}_i^{(k)}) \geq g(\mathbf{W}_i^{(k-1)}) = 0$. This means $\frac{\mathrm{tr}((\mathbf{W}_i^{(k)})^T\tilde{\mathbf{S}}_b^i\mathbf{W}_i^{(k)}\mathbf{M})}{\mathrm{tr}(\mathbf{W}_i^{(k)})^T\tilde{\mathbf{S}}_t^i\mathbf{W}_i^{(k)}\mathbf{M})} \geq \alpha_k$, which implies that $J(\mathbf{W}_i^{(k)}) \geq J(\mathbf{W}_i^{(k-1)})$. □

To prove the convergence of $\mathbf{W}_i^{(k)}$, we introduce the concepts of point-to-set mapping (Hogan 1973) and strict monotonicity (Meyer 1976).

A point-to-set mapping Ω is a function mapping $\mathcal{X}$ to $2^{\mathcal{X}}$ where $2^{\mathcal{X}}$ denotes the power set of a set $\mathcal{X}$. In our algorithm in Table 1, the change from $\mathbf{W}_i^{(k-1)}$ to $\mathbf{W}_i^{(k)}$ can be viewed as a point-to-set mapping where the set contains $\mathbf{W}_i^{(k)}$ with any orthogonal transformation.

An algorithm can be viewed as a point-to-set mapping $\Omega : \mathcal{X} \rightarrow 2^{\mathcal{X}}$ and, given an initial point $\mathcal{X}_1$, the algorithm will generate a sequence of points via the rule $\mathcal{X}_k \in \Omega(\mathcal{X}_{k-1})$. Let $f : \mathcal{X} \rightarrow \mathbb{R}_+$ be a nonnegative continuous function. An algorithm is strictly monotonic with respect to f if (i) $Y \in \Omega(X)$ implies that $f(Y) \geq f(X)$; and (ii) $Y \in \Omega(X)$ and $f(Y) = f(X)$ imply that $Y = X$.

Lemma 3 *The iterative algorithm in Table 1 is strictly monotonic with respect to $f = J(\mathbf{W})$.*

Proof: It is obvious that $J(\mathbf{W})$ is a nonnegative continuous function. From Lemma 2, which says $J(\mathbf{W}_i^{(k)}) \geq J(\mathbf{W}_i^{(k-1)})$, the first condition of strict monotonicity holds for our algorithm. For the second condition, if $J(\mathbf{W}_i^{(k)}) = J(\mathbf{W}_i^{(k-1)})$, so $\alpha_{k+1} = \alpha_k$, and $\mathbf{W}_i^{(k-1)}$ and $\mathbf{W}_i^{(k)}$ are the optimal solution of the problem

$$\arg\max_{\mathbf{W}^T\mathbf{W}=\mathbf{I}_{d'}} \mathrm{tr}(\mathbf{W}^T(\tilde{\mathbf{S}}_b^i - \alpha_k\tilde{\mathbf{S}}_t^i)\mathbf{W}\mathbf{M}).$$

Then from Theorem 1, we can see that there only exists one orthogonal transformation difference between $\mathbf{W}_i^{(k-1)}$ and $\mathbf{W}_i^{(k)}$, that is, $\mathbf{W}_i^{(k)} = \mathbf{W}_i^{(k-1)}\mathbf{Q}$ where $\mathbf{Q}$ is some orthogonal matrix. Moreover, from Theorem 1, we can see that $\mathbf{W}_i^{(k-1)}$ and $\mathbf{W}_i^{(k)}$ lie in the subspace spanned by $\mathbf{U}_1\mathbf{U}_2^T$ where $\mathbf{U}_1$ is the eigenvector matrix of $(\tilde{\mathbf{S}}_b^i - \alpha_k\tilde{\mathbf{S}}_t^i)$ corresponding to the top d' eigenvalues and $\mathbf{U}_2$ is the eigenvector matrix of $\mathbf{M} = \mathbf{P}\mathbf{P}^T$. After steps 2.3 and 2.4 in the algorithm, $\mathbf{W}_i^{(k-1)}$ and $\mathbf{W}_i^{(k)}$ become orthogonal transformation invariant and thus we have $\mathbf{W}_i^{(k-1)} = \mathbf{W}_i^{(k)}$. So the second condition of strict monotonicity holds for our algorithm. Finally, we can reach the conclusion that the iterative algorithm in Table 1 is strictly monotonic with respect to $f = J(\mathbf{W})$. □

Theorem 2 *For the algorithm in Table 1, α_k will monotonically increase and converge to the global optimum.*

Proof: Using Lemma 2 and Lemma 3, the proof is similar to that of Theorem 1 in (Wang et al. 2007). □

Related Work

To the best of our knowledge, no existing multi-task learning method can handle different feature representations in different tasks. The only related work is the so-called translated learning (Dai et al. 2008), which generalizes transfer learning (Pan and Yang 2010) across different feature spaces. However, the goal of translated learning is to only improve the performance of a target task with the help of some source tasks. In our case, however, there is no distinction between all tasks and the goal is to improve the performance of all tasks simultaneously. Moreover, translated learning is only applicable to some specific applications, such as using textual information to help image classification as studied in (Dai et al. 2008).

A sparse multi-task discriminant analysis method has been proposed in (Han et al. 2010). However, this method still requires that different tasks share the same feature space. Another restriction of this method is that different tasks must have some overlapping classes. However, our method has no such requirement.

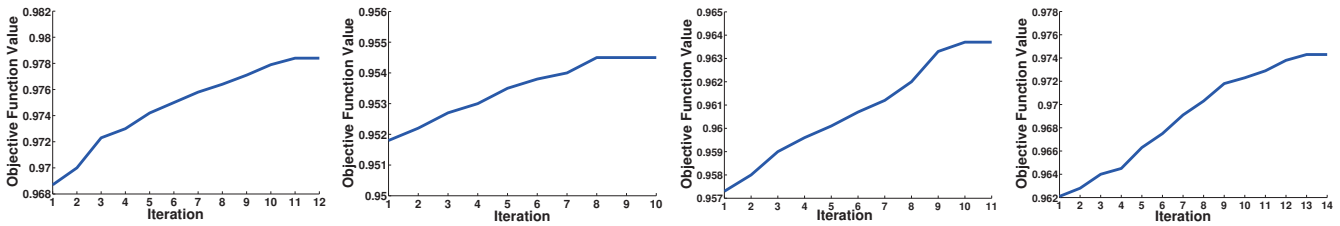
There exist some methods for transfer dimensionality reduction, e.g., (Wang, Song, and Zhang 2008), which utilize information in the source tasks to help dimensionality reduction in the target task, but they cannot handle heterogeneous feature spaces and they, like translated learning, only improve the performance of the target task but not all tasks.

Experiment

In this section, we report some experimental results on face recognition to assess the performance of MTDA.

Experimental Setup

Subspace methods are widely used in many face recognition and object recognition applications, with Eigenface (Turk and Pentland 1991) (based on principal component analysis, or PCA) and Fisherface (Belhumeur, Hespanha, and Kriegman 1997) (based on LDA) being two of the most popular subspace methods. For our experiment, we use three face databases: AR (Martínez and Benavente 1998), ORL (Belhumeur, Hespanha, and Kriegman 1997), and PIE (Sim, Baker, and Bsat 2003). The face images in the AR face database are all frontal view images with differences in expression, illumination and occlusion. There are 26 images for each person taken in two sessions, each having 13 images. In our experiment, 2,600 images of 100 persons (50 men and 50 women) are used. Before the experiment, each image is converted to gray scale and normalized to a size of 33×24 pixels. The ORL face database contains 400 face images of 40 persons, each having 10 images. These face images have significant variations in pose and scale. Each image is preprocessed to a size of 28×23 pixels. The PIE face database contains facial images for 68 persons, and in our experiment, we choose the frontal pose from the PIE database with varying lighting and illumination conditions. There are about 49 images for each subject. Before the experiment, we resize each image to a resolution of 32×32 pixels. From the sample images, we can see that the characteristics of the three databases are very different.



(a) $p = 2$ (b) $p = 3$ (c) $p = 4$ (d) $p = 5$

Figure 1: Convergence of objective function for different values of p

Table 2: Recognition error rates (in mean±std-dev) on three databases for different values of p. 1ST TABLE: $p = 2$; 2ND TABLE: $p = 3$; 3RD TABLE: $p = 4$; 4TH TABLE: $p = 5$. A result marked with $\star$ means that it is significantly better than the other methods.

Method	AR	ORL	PIE
PCA	0.8571±0.0082	0.1950±0.0174	0.7786±0.0068
LDA-rt	0.5198±0.0166	0.1906±0.0193	0.3915±0.0187
LDA-tr	0.4828±0.0119	0.1875±0.0164	0.3427±0.0149
Aggregate	0.9894±0.0022	0.9700±0.0120	0.9841±0.0062
MTDA	**0.4463±0.0183***	**0.1547±0.0178***	**0.3289±0.0150***

Method	AR	ORL	PIE
PCA	0.8150±0.0075	0.1121±0.0190	0.6991±0.0117
LDA-rt	0.4696±0.0234	0.1325±0.0209	0.3124±0.0124
LDA-tr	0.2766±0.0212	**0.0882±0.0234***	0.2456±0.0153
Aggregate	0.9907±0.0026	0.9775±0.0086	0.9858±0.0025
MTDA	**0.2532±0.0216***	**0.0757±0.0183***	**0.2116±0.0163***

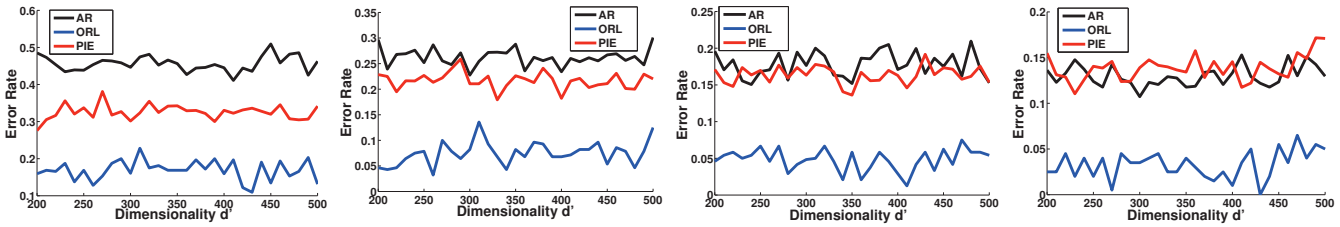
Method	AR	ORL	PIE
PCA	0.7780±0.0095	0.0879±0.0137	0.6389±0.0098
LDA-rt	0.2990±0.0128	0.0917±0.0275	0.2063±0.0119
LDA-tr	0.2158±0.0108	0.0662±0.0146	0.2022±0.0219
Aggregate	0.2993±0.0142	0.3396±0.0378	0.2126±0.0187
MTDA	**0.1746±0.0108***	**0.0429±0.0127***	**0.1702±0.0105***

Method	AR	ORL	PIE
PCA	0.7405±0.0093	0.0600±0.0183	0.5832±0.0104
LDA-rt	0.3292±0.0136	0.0535±0.0189	0.2026±0.0180
LDA-tr	0.1904±0.0088	0.0505±0.0176	0.1709±0.0191
Aggregate	0.2213±0.0134	0.2415±0.0215	0.1912±0.0139
MTDA	**0.1306±0.0084***	**0.0320±0.0151***	**0.1312±0.0112***

In our experiment, learning from each database is treated as one task and so there are three tasks in total. Each task corresponds to a multi-class classification problem where the number of classes in each task is equal to the number of subjects (persons) in each database. We compare our method with single-task PCA and LDA using the ratio trace and trace ratio forms (denoted by LDA-rt and LDA-tr, respectively) which just use PCA and LDA for each database. Since different tasks have different data representations, existing multi-task learning methods cannot be applied directly. Moreover, most multi-task classification methods assume that each task is a binary classification problem and thus they again cannot be applied directly to our face recognition problem. So we compare MTDA with a baseline multi-task learning method, called Aggregate method, which first applies PCA to project data from different databases to a common space in $\mathbb{R}^{d'}$ and then applies LDA on all the data points in that space. After performing dimensionality reduction, we use a simple nearest neighbor classifier to perform classification in the lower-dimensional space.

Experimental Results

To see the effect of varying the size of the training set, we randomly select $p \in \{2, 3, 4, 5\}$ images from each subject in each database for the training set and the rest for the test set. We fix d' as 300. For each configuration and each method, we perform 20 random trials and report the average error rate as well as the standard deviation for those methods in Table 2. For each configuration, the lowest classification error is shown in bold. From Table 2, we can see that MTDA outperforms single-task PCA, single-task LDA with the ratio trace form, single-task LDA with the trace ratio form and Aggregate for all tasks consistently. It is interesting to see



(a) $p = 2$ (b) $p = 3$ (c) $p = 4$ (d) $p = 5$

Figure 2: Prediction error against d' for different values of p

that the performance of Aggregate has a large improvement when the training set increases in size.

Convergence Analysis

To test the convergence of our method, we plot the value of the objective function in problem (3) in Figures 1(a) to 1(d) with different values of p. We find that the objective function value increases and then levels off, showing the convergence of the algorithm. Moreover, convergence is very fast taking only about 10 iterations.

Sensitivity Analysis

To see the effect of varying the intermediate dimensionality d' on the performance, we vary d' from 200 to 500 at an interval of 10. Results on the three tasks are shown in Figures 2(a) to 2(d) with different values of p. From the results, we can see that the performance of MTDA does not change too much when d' varies from 200 to 500, which shows that MTDA is not very sensitive to the parameter d'. Since the best results often occur when d' is between 400 and 500, we prefer using larger values of d' in real applications.

Conclusion

We have proposed in this paper a novel extension of LDA to the multi-task setting, making it possible to perform multi-task discriminant analysis. By exploiting multiple data sets from multiple tasks for the same application, MTDA offers a different solution to overcome the limitation of LDA under situations when labeled data for each learning task is scarce.

Among the possible extensions of MTDA, one interesting direction is to extend it to handle tensors for 2D or higher-order data. Moreover, in case unlabeled data is also available for each task, we can further extend MTDA to the semi-supervised setting.

Acknowledgment

This research has been supported by General Research Fund 622209 from the Research Grants Council of Hong Kong.

References

Anderson, T. W. 2003. *An Introduction to Multivariate Statistical Analysis*.

Ando, R. K., and Zhang, T. 2005. A framework for learning predictive structures from multiple tasks and unlabeled data. *JMLR*.

Argyriou, A.; Evgeniou, T.; and Pontil, M. 2008. Convex multi-task feature learning. *MLJ*.

Bakker, B., and Heskes, T. 2003. Task clustering and gating for Bayesian multitask learning. *JMLR*.

Baxter, J. 1997. A Bayesian/information theoretic model of learning to learn via multiple task sampling. *MLJ*.

Belhumeur, P. N.; Hespanha, J. P.; and Kriegman, D. J. 1997. Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. *TPAMI*.

Bertsekas, D. P. 1999. *Nonlinear Programming*.

Caruana, R. 1997. Multitask learning. *MLJ*.

Chen, L.; Liao, H.; Ko, M.; Lin, J.; and Yu, G. 2000. A new LDA-based face recognition system which can solve the small sample size problem. *PR*.

Dai, W.; Chen, Y.; Xue, G.-R.; Yang, Q.; and Yu, Y. 2008. Translated learning: Transfer learning across different feature spaces. In *NIPS 21*.

Evgeniou, T., and Pontil, M. 2004. Regularized multi-task learning. In *SIGKDD*.

Fukunaga, K. 1991. *Introduction to Statistical Pattern Recognition*.

Han, Y.; Wu, F.; Jia, J.; Zhuang, Y.; and Yu, B. 2010. Multi-task sparse discriminant analysis (MtSDA) with overlapping categories. In *AAAI*.

Hogan, W. 1973. Point-to-set maps in mathematical programming. *SIAM Review*.

Martínez, A. M., and Benavente, R. 1998. The AR-face database. Technical report.

Meyer, R. 1976. Sufficient conditions for the convergence of monotonic mathematical programming algorithms. *Journal of Computer and System Science*.

Pan, S. J., and Yang, Q. 2010. A survey on transfer learning. *TKDE*.

Sim, T.; Baker, S.; and Bsat, M. 2003. The CMU pose, illumination and expression database. *TPAMI*.

Thrun, S., and O'Sullivan, J. 1996. Discovering structure in multiple learning tasks: The TC algorithm. In *ICML*.

Thrun, S. 1996. Is learning the n-th thing any easier than learning the first? In *NIPS 8*.

Turk, M., and Pentland, A. 1991. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*.

Wang, H.; Yan, S.; Xu, D.; Tang, X.; and Huang, T. 2007. Trace ratio vs. ratio trace for dimensionality reduction. In *CVPR*.

Wang, Z.; Song, Y.; and Zhang, C. 2008. Transferred dimensionality reduction. In *ECMLPKDD*.

Zhang, Y., and Yeung, D.-Y. 2010. A convex formulation for learning task relationships in multi-task learning. In *UAI*.