Neural Comput & Applic (2013) 22:1337–1347
DOI 10.1007/s00521-012-0956-8

ICONIP 2011

A multi-task framework for metric learning with common subspace

Peipei Yang · Kaizhu Huang · Cheng-Lin Liu

Received: 10 March 2012 / Accepted: 14 May 2012 / Published online: 3 June 2012

Abstract Metric learning has been widely studied in machine learning due to its capability to improve the performance of various algorithms. Meanwhile, multi-task learning usually leads to better performance by exploiting the shared information across all tasks. In this paper, we propose a novel framework to make metric learning benefit from jointly training all tasks. Based on the assumption that discriminative information is retained in a common subspace for all tasks, our framework can be readily used to extend many current metric learning methods. In particular, we apply our framework on the widely used Large Margin Component Analysis (LMCA) and yield a new model called multi-task LMCA. It performs remarkably well compared to many competitive methods. Besides, this method is able to learn a low-rank metric directly, which effects as feature reduction and enables noise compression and low storage. A series of experiments demonstrate the superiority of our method against three other comparison algorithms on both synthetic and real data.

Keywords Multi-task learning · Metric learning · Low rank · Subspace

P. Yang · K. Huang (✉) · C.-L. Liu
National Laboratory of Pattern Recognition,
Institute of Automation, Chinese Academy of Sciences,
Beijing 100190, China
e-mail: kzhuang@nlpr.ia.ac.cn

P. Yang
e-mail: ppyang@nlpr.ia.ac.cn

C.-L. Liu
e-mail: liucl@nlpr.ia.ac.cn

1 Introduction

As an important topic in machine learning, metric learning has been widely studied by many researchers [1–6]. A metric is in general a measure that indicates the similarity between any pair of data points. The purpose of metric learning is then to learn a more proper measure from data by incorporating certain side-information. On the other hand, multi-task learning (MTL) has recently received considerable attention [7–11] due to its ability to enhance the performance of many supervised and unsupervised machine learning problems. The basic idea of MTL is to train multiple related problems jointly and benefit from the propagation of discriminative information among tasks.

One example to illustrate MTL can be seen in speech recognition [12]. Even when reading the same words, different persons pronounce differently depending on their gender, accent, nationality or other characteristics. Each individual speaker can then be viewed as an individual task and they are closely related. The generalization performance is better when they are jointly trained. This method proves particularly effective especially when few samples can be obtained for certain problems.

However, there are very few attempts to combine these two methods and thus most of existing metric learning methods are incapable of taking advantages of multi-task learning. When the number of training samples is small, traditional metric learning, that is, the single-task learning usually fails to learn a good metric and hence cannot deliver better classification or clustering performance.

One of such attempts called mtLMNN is presented in [13], which is a multi-task extension for the Large Margin Metric Learning (LMNN) model. Similar to the multi-task SVM model [8], mtLMNN assumes that the distance metric for each task is composed with a common

component and a task-specific component. Then all tasks are coupled with the common component, and this method proves effective with experiments on two data sets. However, mtLMNN suffers from two shortcomings. (1) A low-rank metric is unavailable with mtLMNN, which is critical for denoising and resisting overfitting. If the low-rank estimation of the metric with principal component analysis (PCA) [14] is used instead, the performance is noticeably decreased as observed in our experiments. (2) It is computationally more complicated, especially in the case of high dimension. Using t and D to denote the task number and the data dimensionality, respectively, there are $(t+1)D^2$ parameters to be optimized in mtLMNN.

In this paper, we propose a general framework to combine multi-task learning and metric learning based on the assumption that the discriminative information across all the tasks is retained in a low-dimensional common subspace. The basic idea is to learn a common subspace for all tasks and an individual metric for each task simultaneously, where each individual metric is restricted in the common subspace. Then all tasks are coupled with the help of common subspace and estimated more accurately.

The framework can be readily used to extend many current metric learning algorithms to multi-task learning. As an illustration, in this paper, we apply it on a popular metric learning method called Large Margin Component Analysis (LMCA) [5] and yield a new model multi-task LMCA (mtLMCA). In addition to learning an appropriate metric, this model optimizes directly on the transformation matrix and demonstrates surprisingly good performance compared with many competitive methods. One appealing feature of the proposed mtLMCA is that we can learn a metric of low rank, which can suppress noise effectively and hence be more resistant to over-fitting. Besides, since we optimize the transformation matrix instead of Mahalanobis matrix, our framework also benefits from fewer parameters due to the low-dimensional assumption. In contrast to $D(D+tD)$ parameters for mtLMNN, there are merely $d(D+td)$ parameters in our method. Here $d \ll D$ represents the dimensionality of the common subspace. Finally, later experimental results show that our proposed method consistently outperforms mtLMNN in many data sets.

The rest of this paper is organized as follows. In Sect. 2, after some necessary definitions, we show how the common subspace helps to combine different tasks with an intuitive example and then introduce our novel framework in details. In Sect. 3, we present the method and result of our experiment evaluated on four data sets. Finally, we set out the conclusion in Sect. 4. A short version has earlier appeared in [15], while it is significantly expanded in this paper.

2 Multi-task low-rank metric learning

In this section, we first present the notation and the problem definition. Then the assumption of common subspace is illustrated with an example. After that we propose our multi-task metric learning framework and its optimization algorithm in detail.

2.1 Notation

For convenience, we firstly summarize all the notation and symbols used in this papers. Assume that there are T related tasks. $\mathcal{X}_t$ denotes the training data set of the t-th task $\{\mathbf{x}_{tk} \in \mathbb{R}^D, k = 1, 2, \ldots, N_t\}$, where D and N_t are dimension and the number of training samples, respectively. The function $f_t : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$ denotes the distance metric of task-t. In the context of low-rank metric learning, f_t is assumed to be defined based on a linear transformation $L_t : \mathbb{R}^D \to \mathbb{R}^d$ (with $d \ll D$ for obtaining a low rank) as

$$f_{L_t}(\mathbf{x}_{ti}, \mathbf{x}_{tj}) = \mathbf{x}_{t,ij}^\top L_t^\top L_t \mathbf{x}_{t,ij} \triangleq f_{t,ij} \tag{1}$$

where $\mathbf{x}_{t,ij} = \mathbf{x}_{ti} - \mathbf{x}_{tj}$.

The sets of all the similar and dissimilar pairs in $\mathcal{X}_t$ are denoted as $\mathcal{S}_t$ and $\mathcal{D}_t$, respectively, and a set of triplets $\mathcal{T}_t = \{(i,j,k) | (i,j) \in \mathcal{S}_t, (i,k) \in \mathcal{D}_t\}$ are used to define the *side-information* [1]. For example, a simple kind of side-information is $f_t(\mathbf{x}_{ti}, \mathbf{x}_{tj}) \leq f_t(\mathbf{x}_{ti}, \mathbf{x}_{tk})\ \forall (i,j,k) \in \mathcal{T}_t$, which enforces similar data pairs $(\mathbf{x}_{ti}, \mathbf{x}_{tj}) | (i,j) \in \mathcal{S}_t$ to stay closer than dissimilar pairs $(\mathbf{x}_{ti}, \mathbf{x}_{tk}) | (i,k) \in \mathcal{D}_t$ with the new distance metric f_t [16].

2.2 Problem definition

The basic target of multi-task metric learning is to learn an appropriate distance metric f_t for each task-t utilizing all the side-information from the joint training set $\{\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_T\}$.

The loss involved in task-t (defined as l_t) is determined by the distance function f_t (or transformation L_t) and the pairs appearing in $\mathcal{S}_t$ and $\mathcal{D}_t$:

$$l_t = \epsilon_t(L_t) = \epsilon_t(\{f_{t,ij}(L_t)\}), \text{ with } (i,j) \in \mathcal{S}_t \bigcup \mathcal{D}_t,$$

where ϵ_t is any available loss function. Hence, the overall loss involved in all the tasks can be written as

$$l(\{L_t\}) = \sum_t l_t = \sum_t \epsilon_t(L_t). \tag{2}$$

In order to utilize the correlation information among tasks, we assume that the discriminative information embedded in L_t can be retained in a common subspace L_0. We will introduce the concept of common subspace and illustrate its utilization with an example in the following.

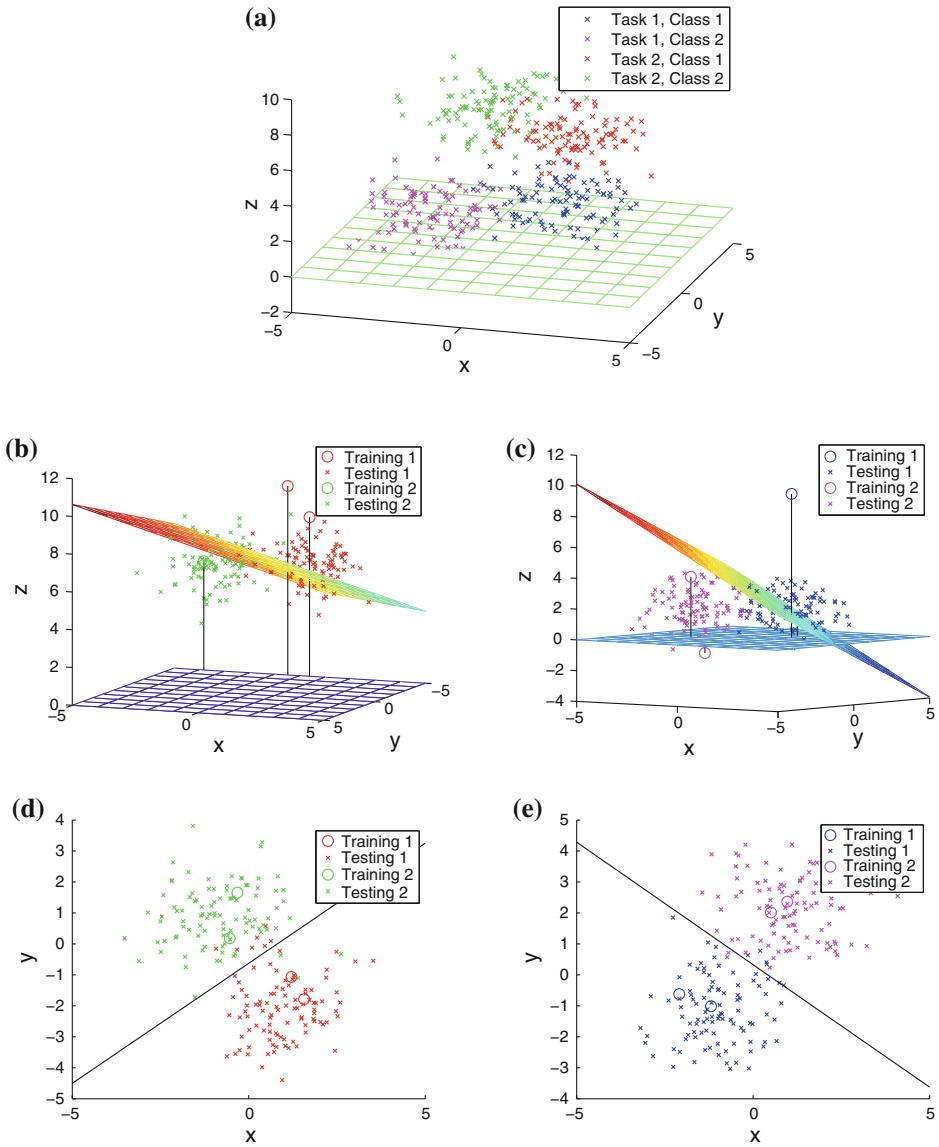

Fig. 1 Benefit of jointly training multiple tasks. **a** Two tasks share a common informative subspace; **b** task-1 in original space (accuracy = 79 %); **c** task-2 in original space (accuracy = 74 %); **d** task-1 in common informative subspace (accuracy = 96 %); **e** task-2 in common informative subspace (accuracy = 95 %)

2.3 Coupling multiple tasks with common subspace

In this section, we will show how to utilize the common subspace to couple multiple tasks and improve the performance. This also motivates our multi-task metric learning framework. Note that when we have several related tasks to be learned, although each task has an individual classifier, the discriminative information across them is often retained in a unique low-dimensional subspace. This phenomenon is illustrated with an example in Fig. 1, and we will show how to make use of this property to improve the performance of all tasks.

Assume that there are two classification tasks shown in Fig. 1a where task-1 is to separate red and green points while task-2 to separate blue and magenta points. Figure 1b, c shows the separating hyperplane learned with training samples of the single task-1/2 using the nearest class mean algorithm, where circle and cross-points represent training samples and test samples, respectively. Because there are too few training samples for each class to represent its distribution, the result is worse than expected where the accuracy is 79 % for task-1 and 74 % for task-2. In another aspect, it is noticeable from Fig. 1a that for both tasks, z-axis indeed contains nothing but noise and all discriminative information is contained in the informative subspace xy-plane. Thus, if the samples are projected into this informative subspace, the noise on z-axis is removed and we can get a better separating hyperplane as shown in Fig. 1d, e. With such separating hyperplanes, the test accuracy for task-1/2 is improved to 96 and 95 %.

However, such an informative subspace is unavailable with only the training samples of any single task. Having noticed that the informative subspace of two tasks are identical, in spite of their different classification hyperplanes, we can learn the informative subspace more accurately using samples of both the two tasks. Then, each task benefits from the low-dimensional informative subspace with less noise and better generalization.

2.4 Multi-task framework for low-rank metric learning

In this subsection, we present the mathematical description of our multi-task metric learning framework that applies the common subspace assumption to the basic framework (2). In order to formulate the problem mathematically, we first propose Theorem 1.

Theorem 1 *Use $f_{L_t}(\mathbf{x}_{ti}, \mathbf{x}_{tj})$ to denote the distance of $\mathbf{x}_{ti}, \mathbf{x}_{tj} \in \mathbb{R}^D$ defined by transformation matrix L_t as (1). Then, for any $L_t \in \mathbb{R}^{d\times D}$ where $d < D$, there exists a d-dimensional subspace $\mathbb{S}_t$ spanned by orthonormal basis $\{\mathbf{p}_{t1}, \ldots, \mathbf{p}_{td}\}$ with a metric defined by $R_t \in \mathbb{R}^{d\times d}$ so that*

$$f_{L_t}(\mathbf{x}_{ti}, \mathbf{x}_{tj}) = f_{R_t}(\hat{\mathbf{x}}_{ti}, \hat{\mathbf{x}}_{tj}),$$

where $\hat{\mathbf{x}}_{ti} = P_t^\top \mathbf{x}_{ti} = [\mathbf{p}_{t1} \ldots \mathbf{p}_{td}]^\top \mathbf{x}_{ti} \in \mathbb{R}^d$ is the coordinate of the projection of $\mathbf{x}_{ti}$ in $\mathbb{S}_t$ with respect to basis P_t.

Proof Consider the singular value decomposition (SVD) of L_t, we have

$$\begin{aligned} L_t &= U_t S_t V_t^\top \\ &= [\mathbf{u}_{t1} \quad \ldots \quad \mathbf{u}_{td}] \begin{bmatrix} \sigma_{t1} & \ldots & 0 & \ldots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & \ldots & \sigma_{td} & \ldots & 0 \end{bmatrix} \begin{bmatrix} \mathbf{v}_{t1}^\top \\ \vdots \\ \mathbf{v}_{tD}^\top \end{bmatrix} \\ &= [\mathbf{u}_{t1} \quad \ldots \quad \mathbf{u}_{td}] \begin{bmatrix} \sigma_{t1} & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & \sigma_{td} \end{bmatrix} \begin{bmatrix} \mathbf{v}_{t1}^\top \\ \vdots \\ \mathbf{v}_{td}^\top \end{bmatrix} \\ &\triangleq U_t \tilde{S}_t P_t^\top \end{aligned} \tag{3}$$

where $\tilde{S}_t$ and P_t are the submatrices composed with the left d columns of S_t and V_t, respectively.

Denote $P_t = [\mathbf{v}_{t1} \ldots \mathbf{v}_{td}] = [\mathbf{p}_{t1} \ldots \mathbf{p}_{td}]$ and $\mathbb{S}_t = \text{span}\{\mathbf{p}_{t1} \ldots \mathbf{p}_{td}\}$. It is obvious that $\{\mathbf{p}_{ti}\}$ comprises an orthonormal basis of $\mathbb{S}_t$ due to the orthogonality of V_t. Thus, $\hat{\mathbf{x}}_{ti} = P_t^\top \mathbf{x}_{ti}$ gives the coordinate of the projection of $\mathbf{x}_{ti}$ in $\mathbb{S}_t$ with respect to basis P_t.

Then denoting $R_t = U_t \tilde{S}_t$ and substituting (3) into (1), we obtain

$$L_t = R_t P_t^\top \tag{4}$$

and

$$\begin{aligned} f_{L_t}(\mathbf{x}_{ti}, \mathbf{x}_{tj}) &= \mathbf{x}_{t,ij}^\top P_t \tilde{S}_t^\top U_t^\top U_t \tilde{S}_t P_t^\top \mathbf{x}_{t,ij} \\ &= \hat{\mathbf{x}}_{t,ij}^\top R_t^\top R_t \hat{\mathbf{x}}_{t,ij} \\ &= f_{R_t}(\hat{\mathbf{x}}_{ti}, \hat{\mathbf{x}}_{tj}) \end{aligned} \tag{5}$$

This completes the proof. □

Theorem 1 proves that for any metric defined in $\mathbb{R}^D$ by transformation matrix $L_t \in \mathbb{R}^{d\times D}$, there exists a d-dimensional subspace $\mathbb{S}_t$ with a metric defined by $R_t \in \mathbb{R}^{d\times d}$ so that the distance of any pair of points in $\mathbb{R}^D$ remains constant if they are projected to $\mathbb{S}_t$. Thus, a metric defined by L_t has an equivalent formulation with explicitly decomposed to a *low-dimensional metric part* R_t and a *subspace part* $\mathbb{S}_t$, or equivalently, P_t. Then our multi-task framework based on common subspace assumption can be simply described as to learn an individual metric R_t for each task in an common subspace $P_t = P, \forall t$. Using (4), it is expressed as $L_t = R_t P^\top, \forall t$.

There is still a problem that the columns of P are assumed to be orthonormal. However, we show that the orthonormal assumption can be approximately discarded. Assume that the SVD of P is

$$P = U_P S_P V_P^\top = \tilde{U}_P \tilde{S}_P V_P^\top \tag{6}$$

where $\tilde{U}_P \in \mathbb{R}^{D\times d}$ is the first d columns of U_P and $\tilde{S}_P \in \mathbb{R}^{d\times d}$ is the first d rows of S_P. Substitute (6) into (4), we obtain $L_t = R_t V_P \tilde{S}_P^\top \tilde{U}_P^\top$.

Then simply denote $\tilde{R}_t = R_t V_P \tilde{S}_P^\top$ and $\tilde{P} = \tilde{U}_P$. We can reformulate $L_t = \tilde{R}_t \tilde{P}^\top$ where the columns of $\tilde{P}$ are orthonormal. Thus, we can formulate our multi-task metric learning constraint as $L_t = R_t L_0$ where L_0 is a $d \times D$ matrix without additional constraints.

With the discussion above, we then would like to minimize the overall loss l defined in Eq. (2). The final optimization problem of multi-task low-rank metric learning can be written as follows:

$$\begin{aligned} \min_{L_0, \{R_t\}} l(L_0, \{R_t\}) &= \sum_t \epsilon_t(R_t L_0) \\ &= \sum_t \epsilon_t(\{f_{t,ij}(R_t L_0)\}), (i,j) \in \mathcal{S}_t \bigcup \mathcal{D}_t, \end{aligned} \tag{7}$$

where $f_{t,ij}(R_t L_0) = \mathbf{x}_{t,ij}^\top L_0^\top R_t^\top R_t L_0 \mathbf{x}_{t,ij}$.

2.5 Relation with multi-task feature learning

It is notable that our method is different from simply learning a metric in the low-dimensional space learned

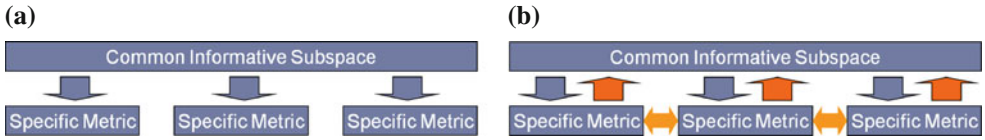

Fig. 2 Difference between metric learning with multi-task feature learning and our method. **a** Multi-task feature learning + metric learning; **b** multi-task metric learning in common subspace

with multi-task feature learning method (denote as MTFL+ML). The difference between them is shown in Fig. 2. For MTFL+ML, only the common informative subspace is used to learn a better metric for each task, while the individual metrics have no effect on the informative subspace. In contrast, since we focus on the problem of defining a metric in sample space that naturally determines the feature space, it is possible to entangle the concept of common subspace into metric learning process and obtain our framework.

In our framework, as shown in Fig. 2b, a low-dimensional informative subspace is learned by jointly training all tasks and used to help each task to learn a better metric. Then the learned metrics furthermore help to find the informative subspace more accurately. The estimation of common subspace and individual metrics are mutually interacted, and this implicitly strengthens the information propagation between the individual metrics.

2.6 Optimization

In the following, we try to adopt the gradient descent method to solve the optimization problem (7). The gradient of ϵ_t with respect to L_t is

$$\frac{\partial \epsilon_t}{\partial L_t} = \sum_{i,j}\left(\frac{\partial \epsilon_t}{\partial f_{t,ij}} \cdot \frac{\partial f_{t,ij}}{\partial L_t}\right) = \sum_{i,j}\left(\frac{\partial \epsilon_t}{\partial f_{t,ij}} \cdot 2L_t \mathbf{x}_{t,ij}\mathbf{x}_{t,ij}^\top\right) = 2L_t \sum_{i,j}\left(\frac{\partial \epsilon_t}{\partial f_{t,ij}} \cdot \mathbf{x}_{t,ij}\mathbf{x}_{t,ij}^\top\right). \tag{8}$$

Since $\frac{\partial f_{t,ij}}{\partial L_0} = 2R_t^\top R_t L_0 \mathbf{x}_{t,ij}\mathbf{x}_{t,ij}^\top$, the gradient can then be calculated using (8) as

$$\frac{\partial l}{\partial L_0} = \sum_t \frac{\partial \epsilon_t}{\partial L_0} = \sum_t \left(2R_t^\top R_t L_0 \sum_i \left(\frac{\partial \epsilon_t}{\partial f_{t,ij}} \cdot \mathbf{x}_{t,ij}\mathbf{x}_{t,ij}^\top\right)\right) = \sum_t \left(2R_t^\top R_t L_0 \Delta_t\right)$$

$$\frac{\partial l}{\partial R_t} = \frac{\partial \epsilon_t}{\partial R_t} = 2R_t \sum_{i,j}\left(\frac{\partial \epsilon_t}{\partial f_{t,ij}} \cdot \left(L_0 \mathbf{x}_{t,ij}\right)\left(L_0 \mathbf{x}_{t,ij}\right)^\top\right) = 2R_t L_0 \Delta_t L_0^\top, \tag{9}$$

where

$$\Delta_t = \sum_{i,j}\left(\frac{\partial \epsilon_t}{\partial f_{t,ij}} \cdot \mathbf{x}_{t,ij}\mathbf{x}_{t,ij}^\top\right). \tag{10}$$

With (8–10), we can easily use the gradient descend method to optimize the L_0 and R_t and hence obtain the final low-rank metric for each task.

2.7 Special case

In this section, we show how to apply our multi-task low-rank metric learning framework to a specific metric learning method. We take the LMCA [5] as a typical example and develop a multi-task LMCA model.[1]

In LMCA, for each sample, some nearest neighbors with the same label are defined as *target neighbors*, which are assumed to have established a perimeter such that differently labeled samples should not invade. Those differently labeled samples invading this perimeter are referred to as *impostors* and the goal of learning is to minimize the number of impostors. The difference between LMCA and LMNN is that LMCA optimizes the transformation matrix L_t while LMNN optimizes the Mahalanobis matrix $M_t = L_t^\top L_t$. Given n input examples $\mathbf{x}_{t1}, \ldots, \mathbf{x}_{tn}$ in $\mathbb{R}^D$ and their corresponding class labels $y_{t1}, \ldots, y_{tn}$, the loss function with respect to transformation matrix L_t is

$$\epsilon_t(L_t) = (1-\mu)\sum_{(i,j)\in\mathcal{S}_t}\left\|L_t(\mathbf{x}_{ti}-\mathbf{x}_{tj})\right\|^2 + \mu\sum_{(i,j,k)\in\mathcal{T}_t}\mathbf{h}\left(\left\|L_t(\mathbf{x}_{ti}-\mathbf{x}_{tj})\right\|^2 - \left\|L_t(\mathbf{x}_{ti}-\mathbf{x}_{tk})\right\|^2 + 1\right), \tag{11}$$

where $\mathbf{h}(s) = \max(s,0)$ is the hinge function.

Minimizing $\epsilon_t(L_t)$ can be implemented using the gradient-based method. Define $\tilde{\mathcal{T}}_t$ as the set of triples which trigger the hinge loss:

$$(i,j,k) \in \tilde{\mathcal{T}}_t \quad \textit{iff} \left\|L_t(\mathbf{x}_{ti}-\mathbf{x}_{tj})\right\|^2 - \left\|L_t(\mathbf{x}_{ti}-\mathbf{x}_{tk})\right\|^2 + 1 > 0.$$

The gradient can then be calculated with

[1] Note that it is straightforward to extend our framework to other metric learning models that optimize the objective function with the transformation matrix.

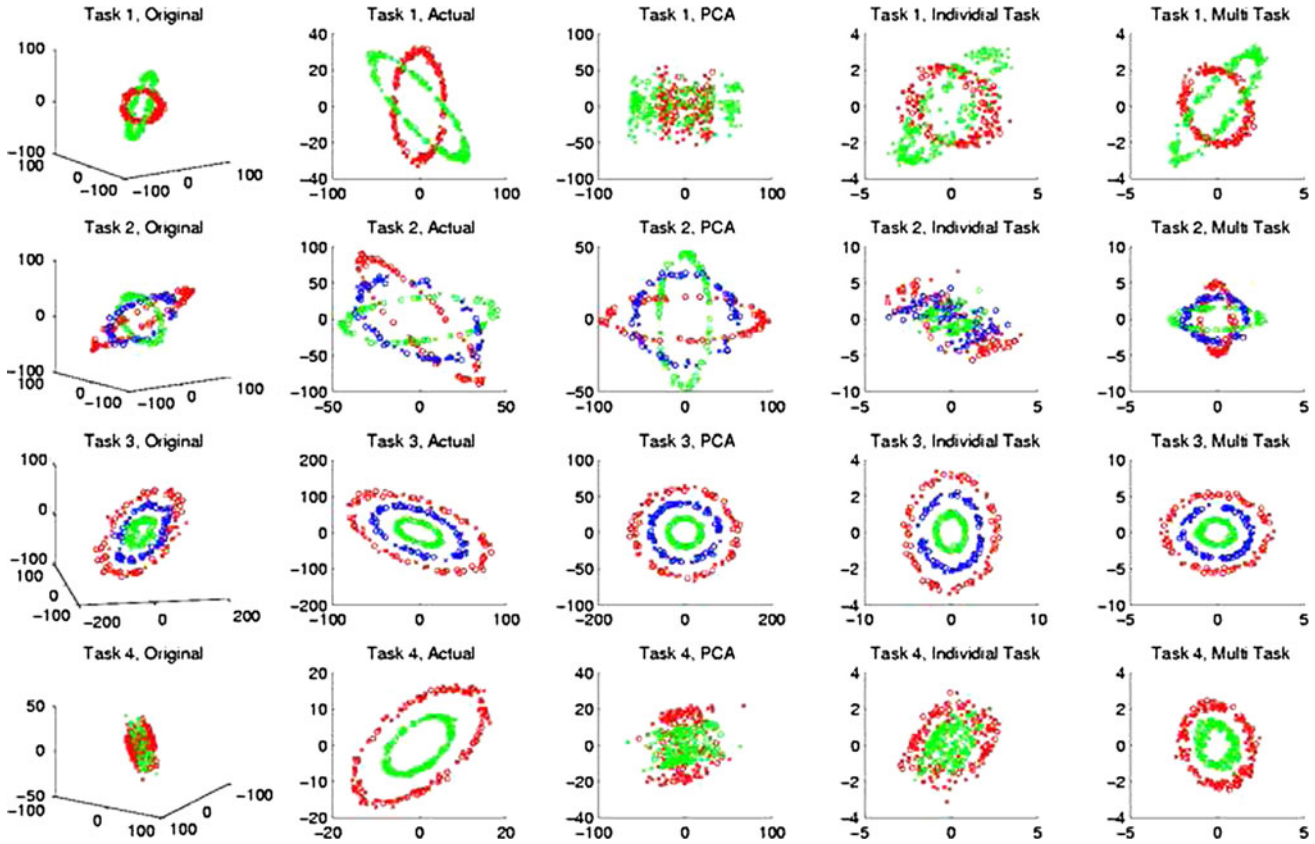

Fig. 3 Illustration of the proposed multi-task low-rank metric learning method. The figure is best viewed in color

$$\frac{\partial \epsilon_t(L_t)}{\partial L_t} = 2(1-\mu)L_t \sum_{(i,j)\in\mathcal{S}_t} (\mathbf{x}_{ti}-\mathbf{x}_{tj})(\mathbf{x}_{ti}-\mathbf{x}_{tj})^\top + 2\mu L_t \sum_{(i,j,k)\in\tilde{\mathcal{T}}_t} \Big[(\mathbf{x}_{ti}-\mathbf{x}_{tj})(\mathbf{x}_{ti}-\mathbf{x}_{tj})^\top - (\mathbf{x}_{ti}-\mathbf{x}_{tk})(\mathbf{x}_{ti}-\mathbf{x}_{tk})^\top\Big] \quad (12)$$

Substituting the transformation matrix of task-t with $L_t = R_t L_0$ and the optimization item ϵ_t in (7) with (11), we can obtain the objective function of multi-task LMCA as

$$l(L_0,\{R_t\}) = \sum_t \epsilon_t(R_t L_0) = \sum_t \Bigg\{ (1-\mu) \sum_{(i,j)\in\mathcal{S}_t} \left\|R_t L_0(\mathbf{x}_{ti}-\mathbf{x}_{tj})\right\|^2 + \mu \sum_{(i,j,k)\in\tilde{\mathcal{T}}_t} \mathbf{h}\Big(\left\|R_t L_0(\mathbf{x}_{ti}-\mathbf{x}_{tj})\right\|^2 - \|R_t L_0(\mathbf{x}_{ti}-\mathbf{x}_{tk})\|^2 + 1\Big)\Bigg\}.$$

The calculation of Δ_t is

$$\Delta_t = (1-\mu) \sum_{(i,j)\in\mathcal{S}_t} (\mathbf{x}_{ti}-\mathbf{x}_{tj})(\mathbf{x}_{ti}-\mathbf{x}_{tj})^\top + \mu \sum_{(i,j,k)\in\tilde{\mathcal{T}}_t} \Big[(\mathbf{x}_{ti}-\mathbf{x}_{tj})(\mathbf{x}_{ti}-\mathbf{x}_{tj})^\top - (\mathbf{x}_{ti}-\mathbf{x}_{tk})(\mathbf{x}_{ti}-\mathbf{x}_{tk})^\top\Big].$$

With Δ_t, the gradient can be calculated with Eq. (9).

3 Experiments

In this section, we first illustrate our proposed multi-task method on a synthetic data set. We then conduct extensive evaluations on four real data sets in comparison with three competitive methods.

3.1 Illustration on synthetic data

In this section, we take the example of concentric circles in [2] to illustrate the effect of our multi-task framework. Assume there are T classification tasks where the samples are distributed in the three-dimensional space and there are

c_t classes in the t-th task. For all the tasks, there exists a **common** two-dimensional subspace (plane) in which the samples of each class are distributed in an elliptical ring centered at zero. The third-dimension orthogonal to this plane contains merely Gaussian noise. The samples of randomly generated 4 tasks were shown in the first column of Fig. 3. In this example, there are 2, 3, 3, 2 classes in the 4 tasks, respectively, and each color corresponds to one class. The circle points and the dot points are, respectively, training samples and test samples with the same distribution. Moreover, as the Gaussian noise will largely degrade the distance calculation in the original space, we should try to search a low-rank metric defined in a low-dimensional subspace.

We apply our proposed mtLMCA on the synthetic data and try to find an appropriate metric by unitizing the correlation information across all the tasks. We project all the points to the subspace which is defined by the learned metric. We visualize the results in Fig. 3. For comparison, we also show the results obtained by the traditional PCA, the individual LMCA (applied individually on each task). Clearly, we can see that for task 1 and task 4, PCA (column 3) found improper metrics due to the large Gaussian noise. For individual LMCA (column 4), the samples are mixed in task 2 because the training samples are not enough. This leads to an improper metric in task 2. In comparison, our proposed mtLMCA (column 5) perfectly found the best metric for each task by exploiting the shared information across all the tasks.

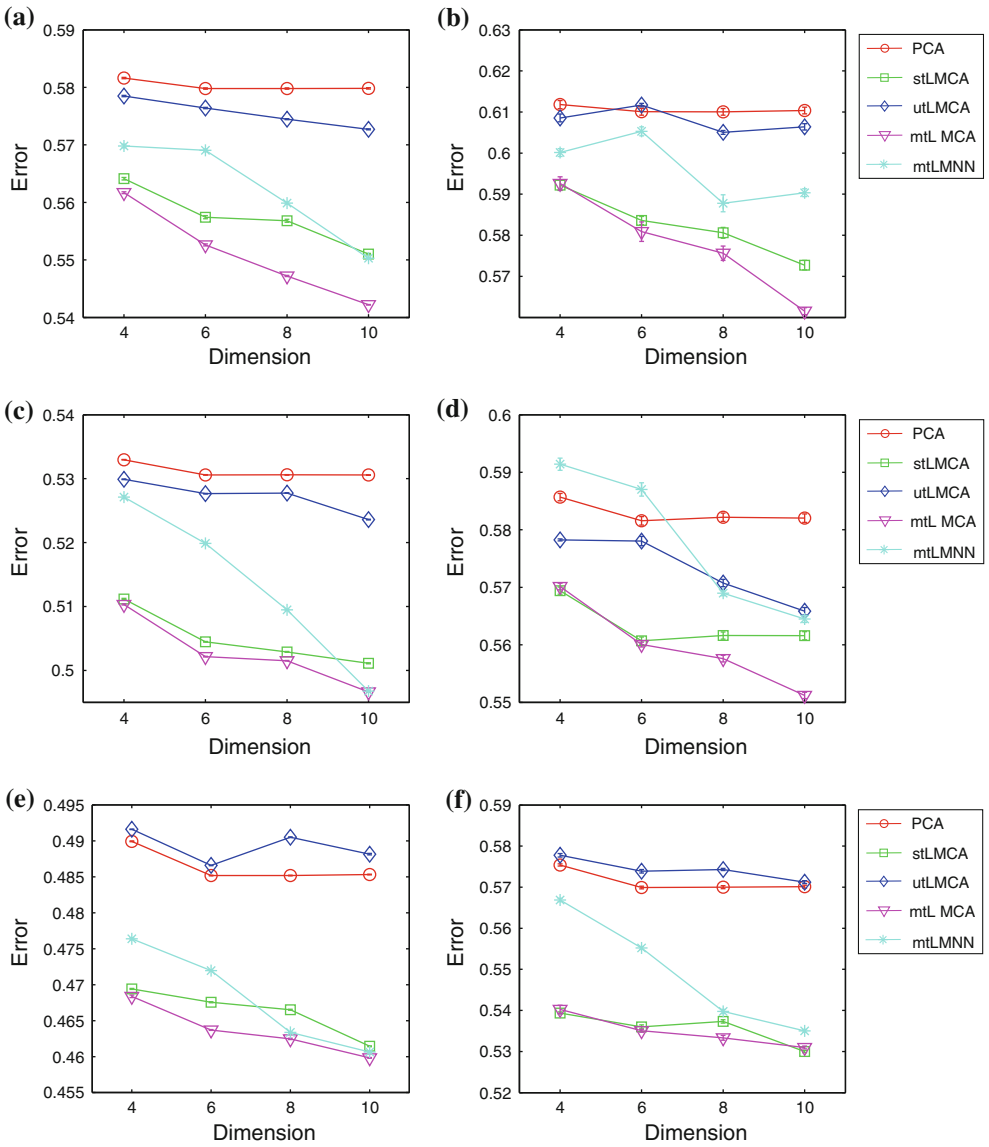

Fig. 4 Experiment results on wine quality data set. **a** on test samples: 5 % training samples used; **b** on training samples: 5 % training samples used; **c** on test samples: 10 % training samples used; **d** on training samples: 10 % training samples used; **e** on test samples: 15 % training samples used; **f** on training samples: 15 % training samples used

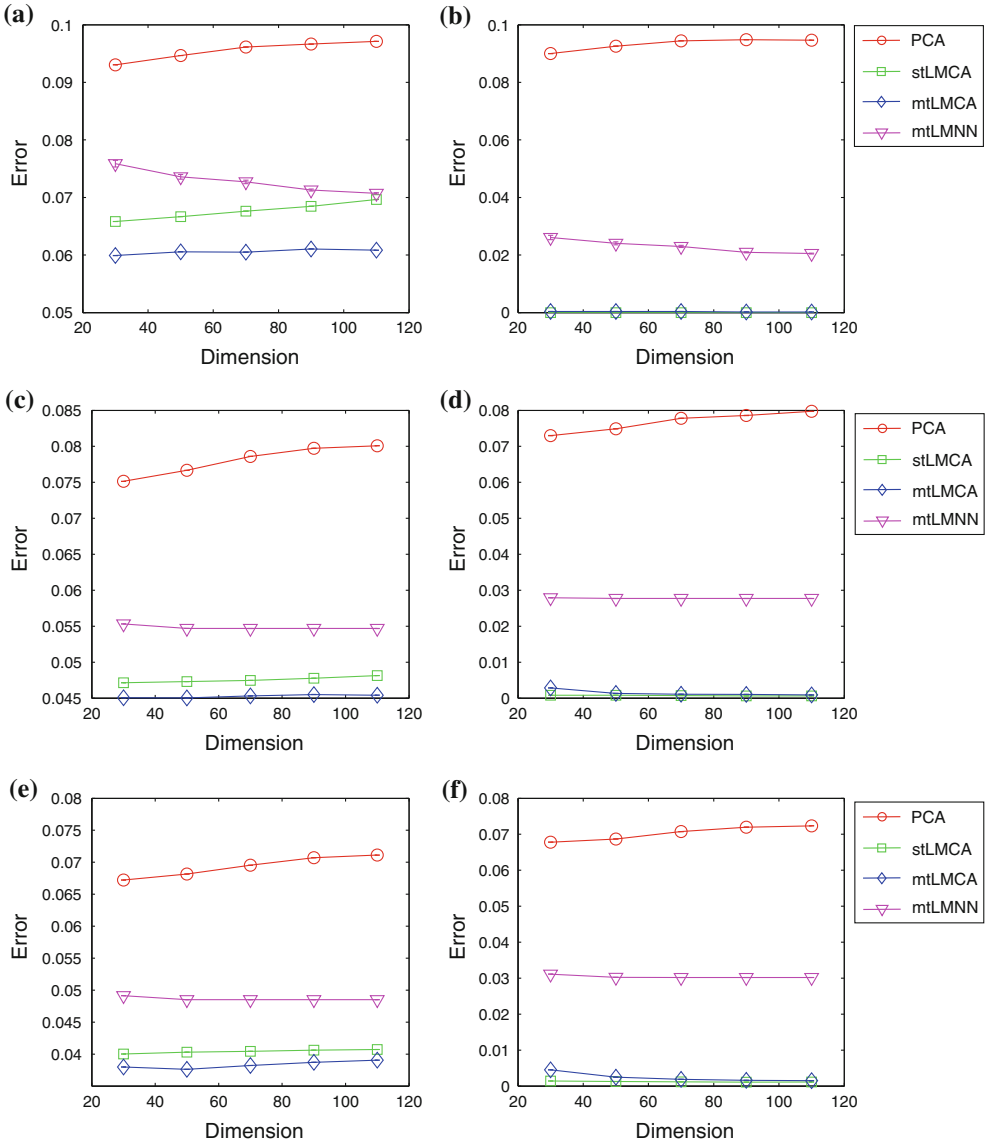

Fig. 5 Experiment results on handwritten letter data set. **a** on test samples: 5 % training samples used; **b** on training samples: 5 % training samples used; **c** on test samples: 10 % training samples used; **d** on training samples: 10 % training samples used; **e** on test samples: 15 % training samples used; **f** on training samples: 15 % training samples used

3.2 Experiment on real data

We evaluate our proposed mtLMCA method on four multi-task data sets. Following many previous metric learning methods, we use the category information to generate relative similarity pairs. For each sample, the nearest 2 neighbors in terms of Euclidean distance are chosen as target neighbors, while the samples sharing different labels and staying closer than any target neighbor are chosen as imposers.

For each data set, we compare our proposed model with PCA, single-task LMCA (stLMCA), and mtLMNN [13]. If all tasks share a common label space, we furthermore compare with uniform-task LMCA (utLMCA), which means to gather the samples in all tasks together and learns a uniform metric for all tasks.

In the experiment, we apply these algorithms to learn a metric of different ranks with the training samples and then compare the classification error rates on both the test samples and training samples using the nearest neighbor classifier. Since mtLMNN is unable to learn a low-rank metric directly, we implement an eigenvalue decomposition on the learned Mahalanobis matrix and use the eigenvectors corresponding to the d largest eigenvalues to generate a low-rank transformation matrix. The parameter μ in the objective function is set to 0.5 empirically in our experiment. The optimization is initialized with $L_0 = I_{d \times D}$ and $R_t = I_d, t = 1, \ldots, T$, where $I_{d \times D}$ is a

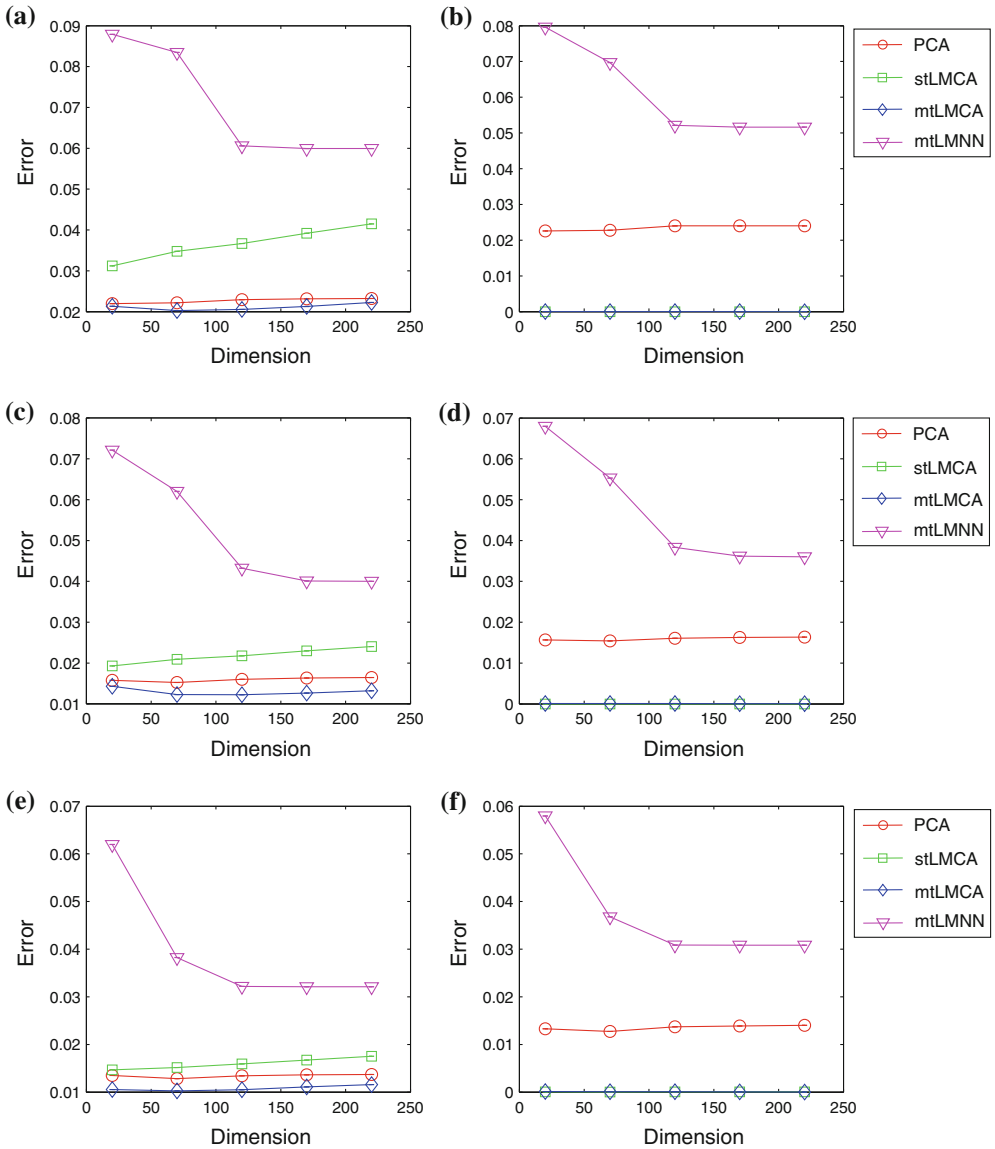

Fig. 6 Experiment results on USPS digit data set. **a** on test samples: 5 % training samples used; **b** on training samples: 5 % training samples used; **c** on test samples: 10 % training samples used; **d** on training samples: 10 % training samples used; **e** on test samples: 15 % training samples used; **f** on training samples: 15 % training samples used

matrix with all the diagonal elements set to 1 and other elements to 0. The optimization process is terminated if the relative difference of the objective function is less than η, which is set to 10^{-5} in our experiment. We run each experiments five times and plot the average error, the maximum error, and the minimum error for each data set.

3.2.1 Wine quality classification

The wine data set[2] is about wine quality including 1,599 red wine samples and 4,898 white wine samples. The labels are given by experts with grades between 0 and 10. Tasks to predict the grade of these two kinds of wine are assumed to be related. For each task, we randomly select 5, 10 and 15 % samples and learn a metric with them. Then the remaining samples are used to test, and the error rates on both test samples and training samples with different dimensions of common subspace are shown in Fig. 4.

3.2.2 Handwritten letter classification

This data set[3] contains handwritten words. It consists of 8 binary classification problems: c/e, g/y, m/n, a/g, i/j, a/o, f/t, h/n. The features are the bitmap of the images of written

[2] http://archive.ics.uci.edu/ml/datasets/Wine+Quality.

[3] http://multitask.cs.berkeley.edu/.

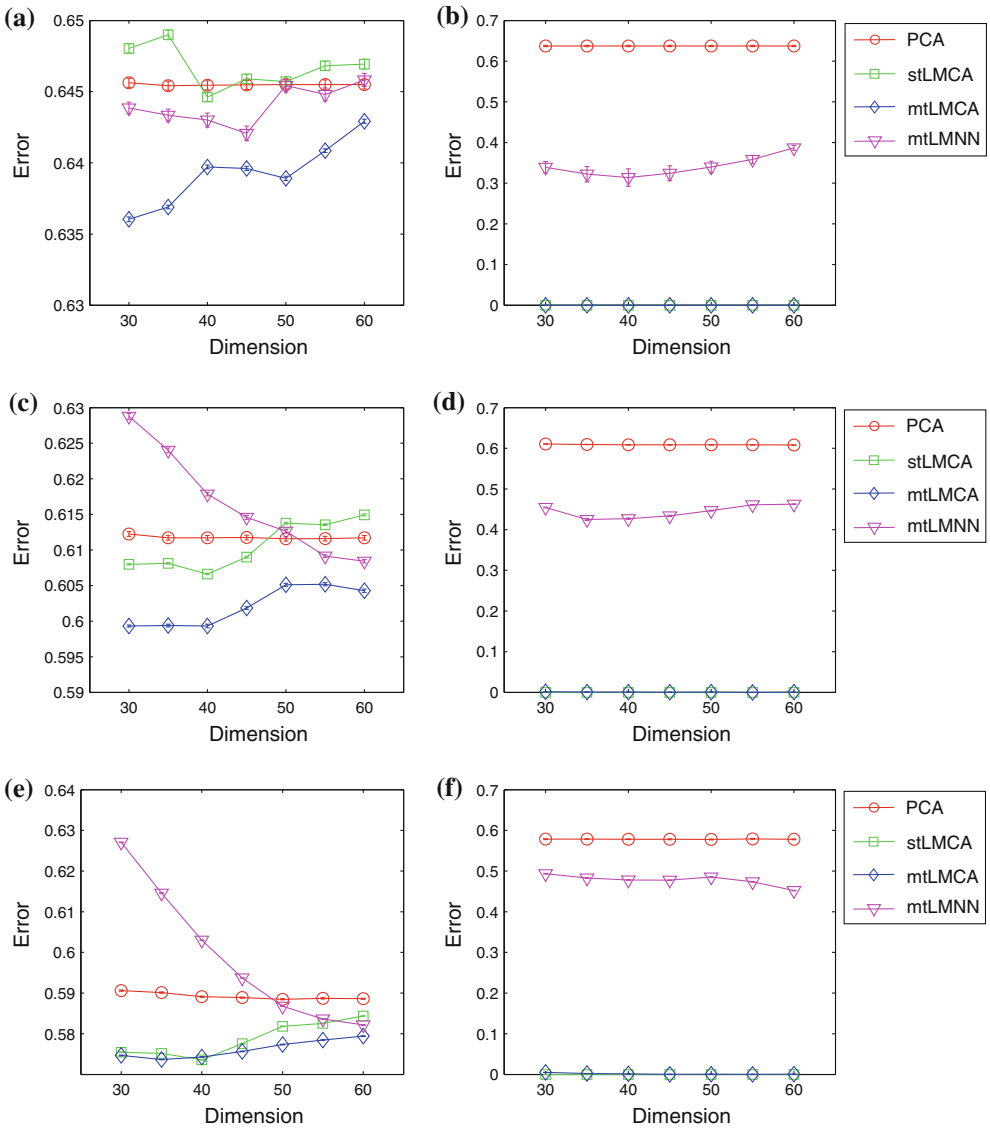

Fig. 7 Experiment results on coIL data set. **a** on test samples: 5 ‰ training samples used; **b** on training samples: 5 ‰ training samples used; **c** on test samples: 10 ‰ training samples used; **d** on training samples: 10 ‰ training samples used; **e** on test samples: 15 ‰ training samples used; **f** on training samples: 15 ‰ training samples used

letters. Each classification problem is regarded as one task and 5, 10, 15 % of randomly selected samples are used to train a metric while the remaining for test. The results on both test samples and training samples are shown in Fig. 5.

3.2.3 USPS digit classification

The USPS digit data set[4] consists of 7,291 16 × 16 grayscale images of digits 0–9 automatically scanned from envelopes by the US Postal Service. The features are then the 256 grayscale values. For each digit, we can get a two-class classification task in which the samples of this digit represent the positive patterns and the others negative patterns. Therefore, there are 10 tasks in total and 5, 10, 15 % of randomly selected samples are used to train a metric while the remaining for test. The results on both test samples and training samples are shown in Fig. 6.

3.2.4 Insurance company benchmark data set

The insurance company benchmark (CoIL) data set[5] contains information on customers of an insurance company. The data consist of 86 variables including product usage data and socio-demographic data derived from zip area codes. There are totally 5,822 training samples and 4000 test samples. We select out the 37, 38, 39, 40, 41th variables as categorical features and predict their values with remaining features. Because all the selected variables are about the information of income, these tasks are more liable to be correlated with each other. We use randomly

[4] http://www-i6.informatik.rwth-aachen.de/~keysers/usps.html.

[5] http://kdd.ics.uci.edu/databases/tic/tic.html.

selected $5, 10, 15\,‰$ of the training samples to learn a metric, and the test samples are given by the data set. The results on both test samples and training samples are shown in Fig. 7.

With these experiment results on test samples, we see that on most dimensionalities, our proposed mtLMCA model performs the best across all the data sets whatever the percentage of training samples are used. This clearly demonstrates the superiority of our proposed multi-task framework. While on training samples, the performance of our method is similar to the performance of single-task method. This agrees with the motivation of multi-task learning that is to improve the generalization performance. Besides, our method is especially suitable to learn a low-rank metric.

4 Conclusion

In this paper, we proposed a new framework capable of extending metric learning to the multi-task scenario. Based on the assumption that the discriminative information across all the tasks can be retained in a low-dimensional common subspace, our proposed framework can be easily solved via the standard gradient descend method. In particular, we applied our framework on a popular metric learning method called LMCA and developed a new model called multi-task LMCA (mtLMCA). In addition to learning an appropriate metric, this model optimized directly on a low-rank transformation matrix and demonstrated very good performance compared to many competitive methods. We conducted extensive experiments on one synthetic and four real multi-task data sets. Experiments results showed that our proposed mtLMCA model can consistently outperform the other three comparison algorithms.

Acknowledgments This work was supported by National Basic Research Program of China (973 Program) grant 2012CB316301, National Natural Science Foundation of China (NSFC) under grants 61075052 and 60825301, and Tsinghua National Laboratory for Information Science and Technology (TNList) Cross-discipline Foundation.

References

1. Xing EP, Ng AY, Jordan MI, Russell S (2003) Distance metric learning, with application to clustering with side-information. In: Advances in neural information processing systems, vol 15, pp 505–512
2. Goldberger J, Roweis S, Hinton G, Salakhutdinov R (2004) Neighbourhood components analysis. In: Advances in neural information processing systems, vol 17, pp 513–520
3. Weinberger KQ, Saul LK (2009) Distance metric learning for large margin nearest neighbor classification. J Mach Learn Res 10:207–244
4. Huang K, Ying Y, Campbell C (2011) Generalized sparse metric learning with relative comparisons. Knowl Inf Syst 28(1):25–45
5. Torresani L, Lee K (2007) Large margin component analysis. In: Advances in neural information processing, pp 505–512
6. Davis JV, Kulis B, Jain P, Sra S, Dhillon IS (2007) Information-theoretic metric learning. In: Proceedings of the 24th international conference on machine learning, pp 209–216
7. Caruana R (1997) Multitask learning. Mach Learn 28(1):41–75
8. Evgeniou T, Pontil M (2004) Regularized multi-task learning. In: Proceedings of the tenth ACM SIGKDD international conference on knowledge discovery and data mining, pp 109–117
9. Argyriou A, Evgeniou T (2008) Convex multi-task feature learning. Mach Learn 73(3):243–272
10. Micchelli CA, Ponti M (2004) Kernels for multi-task learning. In: Advances in neural information processing, pp 921–928
11. Zhang Y, Yeung DY, Xu Q (2010) Probabilistic multi-task feature selection. In: Advances in neural information processing systems, pp 2559–2567
12. Cole R, Fanty M (1990) Spoken letter recognition. In: Proceedings of the workshop on speech and natural language, pp 385–390
13. Parameswaran S, Weinberger K (2010) Large margin multi-task metric learning. In: Advances in neural information processing systems, vol 23, pp 1867–1875
14. Webb AR (2002) Statistical pattern recognition. 2nd edn. Wiley, Chichester
15. Yang P, Huang K, Liu CL (2011) Multi-task low-rank metric learning based on common subspace. In : Proceedings of International Conference on Neural information processing, vol 7063, pp 151–159
16. Rosales R, Fung G (2006) Learning sparse metrics via linear programming. In: Proceedings of the 12th ACM SIGKDD international conference on knowledge discovery and data mining, pp 367–373