Transfer Metric Learning with Semi-Supervised Extension

Yu Zhang, Hong Kong University of Science and Technology
Dit-Yan Yeung, Hong Kong University of Science and Technology

Distance metric learning plays a very crucial role in many data mining algorithms because the performance of an algorithm relies heavily on choosing a good metric. However, the labeled data available in many applications is scarce and hence the metrics learned are often unsatisfactory. In this paper, we consider a transfer learning setting in which some related source tasks with labeled data are available to help the learning of the target task. We first propose a convex formulation for multi-task metric learning by modeling the task relationships in the form of a task covariance matrix. Then we regard transfer learning as a special case of multi-task learning and adapt the formulation of multi-task metric learning to the transfer learning setting for our method, called transfer metric learning (TML). In TML, we learn the metric and the task covariances between the source tasks and the target task under a unified convex formulation. To solve the convex optimization problem, we use an alternating method in which each subproblem has an efficient solution. Moreover, in many applications, some unlabeled data is also available in the target task and so we propose a semi-supervised extension of TML called STML to further improve the generalization performance by exploiting the unlabeled data based on the manifold assumption. Experimental results on some commonly used transfer learning applications demonstrate the effectiveness of our method.

Categories and Subject Descriptors: I.2.6 [**Artificial Intelligence**]: Learning; H.2.8 [**Database Management**]: Database Applications—*Data mining*

General Terms: Algorithms

Additional Key Words and Phrases: Metric Learning, Transfer Learning, Multi-Task Learning, Semi-Supervised Learning

ACM Reference Format:
Zhang, Y., and Yeung, D.-Y. 2011. Transfer Metric Learning with Semi-Supervised Extension. ACM Trans. Intell. Syst. Technol. 9, 4, Article 0 (March 2011), 28 pages.
DOI = 10.1145/0000000.0000000 http://doi.acm.org/10.1145/0000000.0000000

1. INTRODUCTION

Many data mining algorithms, e.g., k-means clustering algorithm and k-nearest neighbor classifier, work by relying on a distance metric. In order to deliver satisfactory results, finding a good distance metric for the problem at hand often plays a very crucial role. As such, metric learning [Xing et al. 2002] has received much attention in the research community [Hastie and Tibshirani 1996; Xing et al. 2002; Chang and Yeung 2004; Weinberger et al. 2005; Chen et al. 2007; Davis et al. 2007; Yeung and Chang 2007; Davis and Dhillon 2008; Yeung et al. 2008; Hoi et al. 2008; Jin et al. 2009; Zhan et al. 2009; Hoi et al. 2010]. Many metric learning methods have been proposed. From the perspective of the underlying learning paradigm, these methods can be grouped

Author's address: Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong.
Email: {zhangyu,dyyeung}@cse.ust.hk

DOI 10.1145/0000000.0000000 http://doi.acm.org/10.1145/0000000.0000000

into three categories, namely, supervised metric learning, unsupervised metric learning, and semi-supervised metric learning. Supervised metric learning learns a metric for some supervised learning tasks, such as classification, so that data points from the same class are kept close while those from different classes are far apart [Hastie and Tibshirani 1996; Weinberger et al. 2005; Davis et al. 2007; Davis and Dhillon 2008; Jin et al. 2009; Zhan et al. 2009]. It has also been used for regression by exploiting the manifold structure contained in the labeled data [Xiao et al. 2009]. Unsupervised metric learning utilizes some information contained in the data to learn a metric for some unsupervised learning task, such as clustering [Chen et al. 2007]. Semi-supervised metric learning, which can be viewed as a combination of the supervised and unsupervised paradigms, utilizes both label information from the labeled data and geometric information from the unlabeled data to learn a good metric for classification or clustering. The need for semi-supervised metric learning arises from the fact that the labeled data available in a number of real-life applications is scarce because labeling data is very laborious and costly. With only limited labeled data, the metrics learned are often unsatisfactory. Semi-supervised metric learning tries to exploit additional information from the unlabeled data to alleviate this problem which is known as the labeled data deficiency problem here.

In this paper, we consider situations similar to those for semi-supervised metric learning in which there is deficiency in labeled data. While the amount of labeled data available in one learning task is limited, it is not uncommon that there exist other related learning tasks with labeled data available. Unlike semi-supervised learning [Chapelle et al. 2006] which exploits unlabeled data, multi-task learning [Caruana 1997; Hastie and Tibshirani 1996; Thrun 1996] and transfer learning [Pan and Yang 2010] seek to alleviate the labeled data deficiency problem by utilizing some related learning tasks to help improve the learning performance. In some sense, they mimic human learning activities in that people may learn faster when several related tasks are learned simultaneously, e.g., playing different games. In essence, people often apply the knowledge gained from some previous learning tasks to help learn a new task. Even though both multi-task learning and transfer learning utilize information from other related learning tasks, there exist some differences between them in both the problem setting and the objective. In transfer learning, the learning tasks are usually classified into two types: source task and target task. It is assumed that there is enough data in the source tasks but not in the target task. The objective of transfer learning is to utilize the information in the source tasks to help learn the target task with no need for improving the performance of the source tasks. On the other hand, there is no distinction between the tasks in multi-task learning and the objective is to improve the performance of all tasks simultaneously.

Even though there exist differences between multi-task learning and transfer learning, a central issue common to both is to accurately characterize the relationships between tasks. Given the training data for multiple tasks, there are two important aspects that distinguish between different methods for characterizing the task relationships. The first aspect is on *how* to obtain the relationships, either from the model assumption or automatically learned from data. Many multi-task and transfer learning methods make some prior assumptions. For example, the latent data representation is shared by different tasks [Caruana 1997; Argyriou et al. 2008] or the learning models in different tasks have similar model parameters [Evgeniou and Pontil 2004; Kienzle and Chellapilla 2006]. Obviously, learning the task relationships from data automatically is the more favorable option because the model assumption adopted may be incorrect and, worse still, it is not easy to verify the correctness of the assumption from data. The second aspect is on *what* task relationships can be represented by a method. Generally speaking there are three types of task relationship: positive task correlation,

negative task correlation, and task unrelatedness.[1] Positive task correlation is a useful task relationship to characterize because similar tasks are likely to have similar model parameters. For negative task correlation, knowing the model parameters of one task will reduce the search space for the model parameters of a negatively correlated task. As for task unrelatedness, identifying outlier tasks can prevent them from impairing the performance of other tasks since outlier tasks are unrelated to other tasks.

In this paper, we study metric learning under the transfer learning setting in which some source tasks are available in addition to the target task. Based on a method called regularized distance metric learning (RDML) [Jin et al. 2009], we propose an extension for transfer learning called *transfer metric learning* (TML). Different from conventional transfer learning methods, we first propose a convex formulation for multi-task metric learning by modeling the task relationships in the form of a task covariance matrix which can model positive, negative and zero task correlations. Then we regard transfer learning as a special case of multi-task learning in that the source tasks are equally important and independent, and adapt the formulation of multi-task metric learning to the transfer learning setting for the formulation of TML. In TML, we learn the metric and the task covariances between the source tasks and the target task under a unified convex formulation. As in multi-task metric learning, the task covariance matrix can also model positive, negative and zero task correlations. To solve the convex optimization problem, we use an alternating method in which each subproblem has an efficient solution. Moreover, in many applications, some unlabeled data is also available in the target task and so we propose a semi-supervised extension of TML called STML to further improve the generalization performance by exploiting the unlabeled data based on the manifold assumption [Belkin et al. 2006]. Experimental results on some commonly used transfer learning applications demonstrate the effectiveness of our method.

The remainder of this paper is organized as follows. We first briefly introduce some background for metric learning and the related work in Section 2. We then present our multi-task metric learning and TML algorithms in Sections 3 and 4, respectively. Section 5 proposes a semi-supervised extension of TML and Section 6 reports experimental results on some transfer learning applications. Finally, some concluding remarks are given in the last section.[2]

2. BACKGROUND AND RELATED WORK

Suppose we are given a labeled training set $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ with the ith data point $\mathbf{x}_i \in \mathbb{R}^d$ and its class label $y_i \in \{1, \ldots, C\}$. In RDML [Jin et al. 2009], the learning problem is formulated as follows:

$$\begin{aligned} \min_{\mathbf{\Sigma}} \quad & \frac{2}{n(n-1)} \sum_{j<k} g\Big(y_{j,k}\left[1 - \|\mathbf{x}_j - \mathbf{x}_k\|_{\mathbf{\Sigma}}^2\right]\Big) + \frac{\lambda}{2}\|\mathbf{\Sigma}\|_F^2 \\ \text{s.t.} \quad & \mathbf{\Sigma} \succeq \mathbf{0}, \end{aligned} \tag{1}$$

where λ is the regularization parameter which balances the empirical loss and the regularization term, $\|\cdot\|_F$ denotes the Frobenius norm of a matrix, $y_{j,k}$ is equal to 1 when y_j and y_k are identical and -1 otherwise, $\mathbf{\Sigma} \succeq \mathbf{0}$ means that $\mathbf{\Sigma}$ is a positive semidefinite (PSD) matrix, $\|\mathbf{x}_j - \mathbf{x}_k\|_{\mathbf{\Sigma}}^2 = (\mathbf{x}_j - \mathbf{x}_k)^T\mathbf{\Sigma}(\mathbf{x}_j - \mathbf{x}_k)$, $g(z) = \max(0, b - z)$ which is similar to the hinge loss used in the support vector machine (SVM). Here b is a constant, satisfying $0 \leq b \leq 1$, which denotes the classification margin. In [Jin et al. 2009], b is set to 0.

[1] Task unrelatedness corresponds to zero or close to zero task correlation.

[2] An abridged version of this paper [Zhang and Yeung 2010b] was presented in KDD2010.

In [Jin et al. 2009], an online method is used to learn the optimal $\mathbf{\Sigma}$ and some properties of RDML, such as the generalization error, are studied. Moreover, theoretical analysis shows that RDML is robust against the number of feature dimensions.

To the best of our knowledge, [Zha et al. 2009] is the only previous work on transfer metric learning. In [Zha et al. 2009], it is assumed that there exist labeled data points for the target task as well as some prior information from the source tasks in the form of a metric matrix learned from each source task. The authors extended information-theoretic metric learning (ITML) [Davis et al. 2007] to transfer metric learning by treating the metric matrices learned from the source tasks as prior information to regularize the learning of the target task. The optimization problem for transfer metric learning in [Zha et al. 2009], which is called L-DML, is formulated as follows:

$$\begin{aligned} \min_{\mathbf{M}} \quad & \sum_{k=1}^{K} \mu_k \mathrm{tr}(\mathbf{M}_k^{-1}\mathbf{M}) - \log|\mathbf{M}| + \eta_s \mathrm{tr}(\mathbf{SM}) - \eta_d \mathrm{tr}(\mathbf{DM}) + \gamma\|\boldsymbol{\mu}\|_2^2 \\ \text{s.t.} \quad & \mathbf{M} \succeq 0 \\ & \sum_{k=1}^{K} \mu_k = 1, \mu_k \geq 0, \end{aligned} \tag{2}$$

where $\mathrm{tr}(\cdot)$ denotes the trace of a square matrix and $\|\cdot\|_2$ denotes the 2-norm of a vector. Here $\mathbf{S} = \sum_{y_i=y_j}(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T$, $\mathbf{D} = \sum_{y_i \neq y_j}(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T$, and $\mathbf{M}_k$ $(k = 1, \ldots, K)$ is the available metric matrix for the kth source task. The first and second terms in the objective function of problem (2) are derived from the log-determinant regularization function as used in [Davis et al. 2007] and μ_k is the weight that reflects the utility of the metric of the kth source task. The third term is to keep the data points in the same class as close as possible and the fourth term is to keep the data points from different classes far apart. The last term is to penalize the complexity of $\boldsymbol{\mu}$. Here $\boldsymbol{\mu}$ plays an important role in this formulation since there may exist outlier tasks in real applications and by learning $\boldsymbol{\mu}$ L-DML can identity them. However, each element of the vector $\boldsymbol{\mu}$ is non-negative and so it cannot model the negative transfer situation [Rosenstein et al. 2005]. Moreover, the constraint $\sum_{k=1}^{K} \mu_k = 1$ is not very reasonable. Consider a special case in which there is only one source task. Then $\mu_1 = 1$ even if this source task is an outlier task. When there are multiple source tasks and all of them are outlier tasks, we should set all μ_i to zero but then the constraint $\sum_{k=1}^{K} \mu_k = 1$ cannot be satisfied. Furthermore, problem (2) is not convex, making it easy to get trapped in (bad) local minima during the optimization procedure.

There exist some methods for transfer dimensionality reduction [Wang et al. 2008; Pan et al. 2008; Pan et al. 2009], where dimensionality reduction can be viewed as a special case of metric learning in that the metric learned is not of full rank. However, transfer dimensionality reduction is different from transfer metric learning and these methods are not applicable here. For example, [Wang et al. 2008] used a transformation matrix of the dimensionality reduction in the source tasks for subspace clustering in the target task and so the target task is an unsupervised learning task. Also, [Pan et al. 2008; Pan et al. 2009] proposed dimensionality reduction methods for domain adaption in which the target task has no labeled data, and so it is different from the setting here where we utilize the metric matrices learned from the source tasks to help the learning of the target task from labeled data.

3. MULTI-TASK METRIC LEARNING

In this section, we propose a multi-task metric learning method which can learn the task relationships between all pairs of tasks.

Suppose we are given m learning tasks $\{T_i\}_{i=1}^m$. For the ith task T_i, the training set $\mathcal{D}_i$ consists of n_i data points represented in the form of $(\mathbf{x}_j^i, y_j^i)$, $j = 1, \ldots, n_i$, with $\mathbf{x}_j^i \in \mathbb{R}^d$ and its corresponding class label $y_j^i \in \{1, \ldots, C_i\}$. Here the superscript denotes the task index and the subscript denotes the instance index in each task.

The optimization problem for multi-task metric learning is formulated as follows[3]:

$$\begin{aligned}
\min_{\{\mathbf{\Sigma}_i\},\mathbf{\Omega}} & \sum_{i=1}^{m} \frac{2}{n_i(n_i-1)} \sum_{j<k} g\Big(y_{j,k}^i \left[1 - \|\mathbf{x}_j^i - \mathbf{x}_k^i\|_{\mathbf{\Sigma}_i}^2\right]\Big) + \frac{\lambda_1}{2} \sum_{i=1}^{m} \|\mathbf{\Sigma}_i\|_F^2 + \frac{\lambda_2}{2} \mathrm{tr}(\tilde{\mathbf{\Sigma}} \mathbf{\Omega}^{-1} \tilde{\mathbf{\Sigma}}^T) \\
\text{s.t. } & \mathbf{\Sigma}_i \succeq \mathbf{0} \ \forall i \\
& \tilde{\mathbf{\Sigma}} = (\mathrm{vec}(\mathbf{\Sigma}_1), \ldots, \mathrm{vec}(\mathbf{\Sigma}_m)) \\
& \mathbf{\Omega} \succeq 0 \\
& \mathrm{tr}(\mathbf{\Omega}) = 1, \qquad (3)
\end{aligned}$$

where $y_{j,k}^i$ is equal to 1 when $y_j^i = y_k^i$ and -1 otherwise, $\mathrm{vec}(\cdot)$ denotes the operator which converts a matrix into a vector in a columnwise manner, and λ_1 and λ_2 are the regularization parameters. $\mathbf{\Omega}$ is a task covariance matrix which describes the relationships between tasks and so it is a PSD matrix. The first term in the objective function of problem (3) measures the empirical loss for the training sets of the m tasks, the second term penalizes the complexity of each $\mathbf{\Sigma}_i$, and the last term measures the task relationships between all pairs of tasks based on each $\mathbf{\Sigma}_i$. The last constraint in (3) is to restrict the scale of $\mathbf{\Omega}$ to prevent it from reaching a degenerate solution.

From a probabilistic viewpoint, RDML can be seen as obtaining the maximum a posteriori (MAP) solution of a probabilistic model where the likelihood corresponds to the first term in the objective function of problem (1) and the prior on the metric is Gaussian prior corresponding to the second term. Similar to RDML, our multi-task metric learning is also a MAP solution of a probabilistic model where the likelihood is the same as that in RDML for each task and the prior on the metrics of all tasks is matrix-variate normal distribution [Gupta and Nagar 2000].

We will prove below that problem (3) is a convex optimization problem by proving that each term in the objective function is convex and each constraint is also convex.

THEOREM 3.1. *Problem (3) is convex with respect to* $\{\mathbf{\Sigma}_i\}$ *and* $\mathbf{\Omega}$.

Proof

It is easy to see that the first two terms in the objective function are convex with respect to (w.r.t.) all variables and the constraints in (3) are also convex. We have proved in our previous work [Zhang and Yeung 2010a] that the third term in the objective function is convex with respect to all variables but we repeat it here to make this paper self-contained. We rewrite the third term as

$$\mathrm{tr}(\tilde{\mathbf{\Sigma}} \mathbf{\Omega}^{-1} \tilde{\mathbf{\Sigma}}^T) = \sum_t \tilde{\mathbf{\Sigma}}(t,:) \mathbf{\Omega}^{-1} \tilde{\mathbf{\Sigma}}(t,:)^T,$$

where $\tilde{\mathbf{\Sigma}}(t,:)$ is the tth row of $\tilde{\mathbf{\Sigma}}$. Since $\tilde{\mathbf{\Sigma}}(t,:)\mathbf{\Omega}^{-1}\tilde{\mathbf{\Sigma}}(t,:)^T$ is a matrix fractional function as in Example 3.4 on page 76 of [Boyd and Vandenberghe 2004], it is convex w.r.t. $\tilde{\mathbf{\Sigma}}(t,:)$ and $\mathbf{\Omega}$ when $\mathbf{\Omega}$ is a PSD matrix (which is satisfied by the first constraint of (3)). Since $\tilde{\mathbf{\Sigma}}(t,:)$ is a row of $\tilde{\mathbf{\Sigma}}$, $\tilde{\mathbf{\Sigma}}(t,:)\mathbf{\Omega}^{-1}\tilde{\mathbf{\Sigma}}(t,:)^T$ is also convex w.r.t. $\{\mathbf{\Sigma}_i\}$ and $\mathbf{\Omega}$. Because the

[3]Since $\mathbf{\Sigma}_i$ is a symmetric matrix, we can also define each column of $\tilde{\mathbf{\Sigma}}$ as the vectorization of the upper (or lower) triangle of the corresponding $\mathbf{\Sigma}_i$. These two formulation can be proved to be equivalent to each other.

summation operation can preserve convexity according to the analysis on page 79 of [Boyd and Vandenberghe 2004], $\mathrm{tr}(\tilde{\mathbf{\Sigma}}\mathbf{\Omega}^{-1}\tilde{\mathbf{\Sigma}}^T) = \sum_t \tilde{\mathbf{\Sigma}}(t,:)\mathbf{\Omega}^{-1}\tilde{\mathbf{\Sigma}}(t,:)^T$ is convex w.r.t. $\{\mathbf{\Sigma}_i\}$ and $\mathbf{\Omega}$. So the objective function and the constraints in problem (3) are convex w.r.t. all variables and hence problem (3) is jointly convex. □

Even though problem (3) is convex with respect to $\{\mathbf{\Sigma}_i\}$ and $\mathbf{\Omega}$ jointly, it is not easy to optimize it with respect to all the variables simultaneously. Here we propose an alternating method to solve the problem more efficiently. Specifically, we first optimize the objective function with respect to $\mathbf{\Sigma}_i$ when $\mathbf{\Omega}$ and $\{\mathbf{\Sigma}\}_{-i} \stackrel{\text{def}}{=} \{\mathbf{\Sigma}_1, \ldots, \mathbf{\Sigma}_{i-1}, \mathbf{\Sigma}_{i+1}, \ldots, \mathbf{\Sigma}_m\}$ are fixed, and then optimize it with respect to $\mathbf{\Omega}$ when $\{\mathbf{\Sigma}_i\}$ are fixed. This procedure is repeated until convergence. Since the original optimization problem is convex, the solution found by this alternating procedure is guaranteed to be the globally optimal solution [Bertsekas 1999].

Because multi-task metric learning is not the focus of this paper, we leave the detailed optimization procedure to Appendix A.

4. TRANSFER METRIC LEARNING

Based on the multi-task metric learning problem formulated in the previous section, we propose a transfer metric learning formulation as a special case which can learn the task relationships between all source tasks and the target task.

Suppose we are given $m-1$ source tasks $\{T_i\}_{i=1}^{m-1}$ and one target task T_m, for $m > 1$. In the target task, the training set contains n_m labeled data points $\{(\mathbf{x}_j^m, y_j^m)\}_{j=1}^{n_m}$. In transfer learning, it is assumed that each source task has enough labeled data and can learn an accurate model with no need to seek help from the other source tasks. So the source tasks are considered to be independent since each source task does not need help from other source tasks. So, similar to the setting in [Zha et al. 2009], we assume that the metric matrix $\mathbf{\Sigma}_i$ for the ith source task has been learned independently. We hope to use the metric matrices learned to help the learning of the target task because the labeled data there is scarce.

4.1. Optimization Problem

Based on problem (3), we formulate the optimization problem for TML as follows:

$$\begin{aligned}
\min_{\mathbf{\Sigma}_m, \mathbf{\Omega}} \ & \frac{2}{n_m(n_m-1)} \sum_{j<k} g\Big(y_{j,k}^m \left[1 - \|\mathbf{x}_j^m - \mathbf{x}_k^m\|_{\mathbf{\Sigma}_m}^2\right]\Big) + \frac{\lambda_1}{2}\|\mathbf{\Sigma}_m\|_F^2 + \frac{\lambda_2}{2}\mathrm{tr}(\tilde{\mathbf{\Sigma}}\mathbf{\Omega}^{-1}\tilde{\mathbf{\Sigma}}^T) \\
\text{s.t. } & \mathbf{\Sigma}_m \succeq \mathbf{0} \\
& \tilde{\mathbf{\Sigma}} = (\mathrm{vec}(\mathbf{\Sigma}_1), \ldots, \mathrm{vec}(\mathbf{\Sigma}_{m-1}), \mathrm{vec}(\mathbf{\Sigma}_m)) \\
& \mathbf{\Omega} \succeq 0 \\
& \mathrm{tr}(\mathbf{\Omega}) = 1.
\end{aligned} \tag{4}$$

Since we assume that the source tasks are independent of each other and each source task is of equal importance, we can express $\mathbf{\Omega}$ as

$$\mathbf{\Omega} = \begin{pmatrix} \alpha\mathbf{I}_{m-1} & \boldsymbol{\omega}_m \\ \boldsymbol{\omega}_m^T & \omega \end{pmatrix},$$

where $\mathbf{I}_a$ denotes the $a \times a$ identity matrix, $\boldsymbol{\omega}_m$ denotes the task covariances between the target task and the source tasks (implying that the target task and the source tasks are not independent), and ω denotes the variance of the target task. According

to the last constraint in problem (4), we can get

$$\alpha = \frac{1-\omega}{m-1}.$$

From Theorem 1, it is easy to show that problem (4) is also jointly convex with respect to all variables. Moreover, from the block matrix inversion formula, we can get

$$\begin{aligned}\boldsymbol{\Omega}^{-1} &= \begin{pmatrix} \frac{1-\omega}{m-1}\mathbf{I}_{m-1} & \boldsymbol{\omega}_m \\ \boldsymbol{\omega}_m^T & \omega \end{pmatrix}^{-1} \\ &= \begin{pmatrix} \mathbf{I}_{m-1} & \mathbf{a} \\ \mathbf{0}_{m-1}^T & 1 \end{pmatrix} \begin{pmatrix} \frac{(m-1)\mathbf{I}_{m-1}}{1-\omega} & \mathbf{0}_{m-1} \\ \mathbf{0}_{m-1}^T & \frac{1}{c} \end{pmatrix} \begin{pmatrix} \mathbf{I}_{m-1} & \mathbf{0}_{m-1} \\ \mathbf{a}^T & 1 \end{pmatrix},\end{aligned}$$

where $\mathbf{0}_d$ denotes the $d \times 1$ zero vector, $\mathbf{a} = -\frac{(m-1)\boldsymbol{\omega}_m}{1-\omega}$ and $c = \omega - \frac{(m-1)\boldsymbol{\omega}_m^T\boldsymbol{\omega}_m}{1-\omega}$.

Let $\tilde{\boldsymbol{\Sigma}}_s = (\mathrm{vec}(\boldsymbol{\Sigma}_1), \ldots, \mathrm{vec}(\boldsymbol{\Sigma}_{m-1}))$, which is a constant matrix here, denote the parameter matrix of the source tasks. Then we can get

$$\begin{aligned}&\mathrm{tr}(\tilde{\boldsymbol{\Sigma}}\boldsymbol{\Omega}^{-1}\tilde{\boldsymbol{\Sigma}}^T) \\ &= \mathrm{tr}\Big(\Big(\tilde{\boldsymbol{\Sigma}}_s, \mathrm{vec}(\boldsymbol{\Sigma}_m)\Big) \boldsymbol{\Omega}^{-1} \begin{pmatrix} \tilde{\boldsymbol{\Sigma}}_s^T \\ \mathrm{vec}(\boldsymbol{\Sigma}_m)^T \end{pmatrix} \Big) \\ &= \mathrm{tr}\Big(\begin{pmatrix} \tilde{\boldsymbol{\Sigma}}_s^T \\ \mathrm{vec}(\boldsymbol{\Sigma}_m)^T - \frac{(m-1)}{1-\omega}\boldsymbol{\omega}_m^T\tilde{\boldsymbol{\Sigma}}_s^T \end{pmatrix}^T \begin{pmatrix} \frac{(m-1)\mathbf{I}_{m-1}}{1-\omega} & \mathbf{0}_{m-1} \\ \mathbf{0}_{m-1}^T & \frac{1}{c} \end{pmatrix} \begin{pmatrix} \tilde{\boldsymbol{\Sigma}}_s^T \\ \mathrm{vec}(\boldsymbol{\Sigma}_m)^T - \frac{(m-1)}{1-\omega}\boldsymbol{\omega}_m^T\tilde{\boldsymbol{\Sigma}}_s^T \end{pmatrix} \Big) \\ &= \frac{m-1}{1-\omega}\mathrm{tr}(\tilde{\boldsymbol{\Sigma}}_s^T\tilde{\boldsymbol{\Sigma}}_s) + \frac{1}{c}\|\mathrm{vec}(\boldsymbol{\Sigma}_m) - \frac{(m-1)}{1-\omega}\tilde{\boldsymbol{\Sigma}}_s\boldsymbol{\omega}_m\|_2^2 \\ &= \frac{(1-\omega)\|\boldsymbol{\Sigma}_m\|_F^2 - 2(m-1)\mathrm{vec}(\boldsymbol{\Sigma}_m)^T\tilde{\boldsymbol{\Sigma}}_s\boldsymbol{\omega}_m + (m-1)\omega\mathrm{tr}(\tilde{\boldsymbol{\Sigma}}_s^T\tilde{\boldsymbol{\Sigma}}_s)}{\omega(1-\omega) - (m-1)\boldsymbol{\omega}_m^T\boldsymbol{\omega}_m}. \end{aligned} \tag{5}$$

Moreover, according to the Schur complement [Boyd and Vandenberghe 2004], we have

$$\boldsymbol{\Omega} \succeq 0 \iff \omega \geq \frac{m-1}{1-\omega}\boldsymbol{\omega}_m^T\boldsymbol{\omega}_m \text{ and } \frac{(m-1)\mathbf{I}_{m-1}}{1-\omega} \succeq 0,$$

which is equivalent to

$$\boldsymbol{\Omega} \succeq 0 \iff \omega(1-\omega) \geq (m-1)\boldsymbol{\omega}_m^T\boldsymbol{\omega}_m.$$

Then problem (4) can be simplified to

$$\begin{aligned}\min_{\boldsymbol{\Sigma}_m, \boldsymbol{\omega}_m, \omega, \boldsymbol{\Omega}} & \frac{2}{n_m(n_m-1)} \sum_{j<k} g\Big(y_{j,k}^m \left[1 - \|\mathbf{x}_j^m - \mathbf{x}_k^m\|_{\boldsymbol{\Sigma}_m}^2\right]\Big) + \frac{\lambda_1}{2}\|\boldsymbol{\Sigma}_m\|_F^2 + \frac{\lambda_2}{2}\mathrm{tr}(\tilde{\boldsymbol{\Sigma}}\boldsymbol{\Omega}^{-1}\tilde{\boldsymbol{\Sigma}}^T) \\ \text{s.t. } & \boldsymbol{\Sigma}_m \succeq \mathbf{0} \\ & \boldsymbol{\Omega} = \begin{pmatrix} \frac{1-\omega}{m-1}\mathbf{I}_{m-1} & \boldsymbol{\omega}_m \\ \boldsymbol{\omega}_m^T & \omega \end{pmatrix} \\ & \tilde{\boldsymbol{\Sigma}} = (\tilde{\boldsymbol{\Sigma}}_s, \mathrm{vec}(\boldsymbol{\Sigma}_m)) \\ & \omega(1-\omega) \geq (m-1)\boldsymbol{\omega}_m^T\boldsymbol{\omega}_m, \end{aligned} \tag{6}$$

where the last term in the objective function can be simplified as in Eq. (5).

Compared with the L-DML method in [Zha et al. 2009], our method has some advantages. First, the formulation of TML is convex and so there is guarantee to find the globally optimal solution. Second, similar to multi-task metric learning proposed in

the previous section, TML can model positive, negative and zero task correlations in a unified formulation but L-DML cannot model negative task correlation. As an extreme case, we can deal with the situation in which all source tasks are outlier tasks, but L-DML cannot handle it due to the constraint $\sum_{k=1}^{K}\mu_k = 1$ in problem (2).

Moreover, compared with problem (4), there is no PSD constraint on $\mathbf{\Omega}$ in problem (6) making it simpler than problem (4). In the next section, we will discuss how to solve problem (6).

4.2. Optimization Procedure

As in multi-task metric learning, problem (6) is a convex problem and we still use an alternating method to solve it. Specifically, we first optimize the objective function with respect to $\mathbf{\Sigma}_m$ when $\boldsymbol{\omega}_m$ and ω are fixed, and then optimize it with respect to $\boldsymbol{\omega}_m$ and ω when $\mathbf{\Sigma}_m$ is fixed. This procedure is repeated until convergence. As before, the solution found by this alternating procedure is globally optimal [Bertsekas 1999]. In what follows, we will present the two subproblems separately.

Optimizing w.r.t. $\mathbf{\Sigma}_m$ when $\boldsymbol{\omega}_m$ and ω are fixed

Utilizing Eq. (5), the optimization problem with respect to $\mathbf{\Sigma}_m$ is formulated as

$$\begin{aligned}\min_{\mathbf{\Sigma}_m} \quad & \frac{2}{n_m(n_m-1)}\sum_{j<k} g\Big(y_{j,k}^m\left[1-\|\mathbf{x}_j^m-\mathbf{x}_k^m\|_{\mathbf{\Sigma}_m}^2\right]\Big)+\frac{\lambda_1'}{2}\|\mathbf{\Sigma}_m\|_F^2-\lambda_2'\mathrm{tr}(\mathbf{\Sigma}_m\mathbf{M})\\ \text{s.t.} \quad & \mathbf{\Sigma}_m \succeq \mathbf{0},\end{aligned} \tag{7}$$

where

$$\begin{aligned}\lambda_1' &= \lambda_1+\frac{\lambda_2(1-\omega)}{\omega(1-\omega)-(m-1)\boldsymbol{\omega}_m^T\boldsymbol{\omega}_m},\\ \lambda_2' &= \frac{\lambda_2(m-1)}{\omega(1-\omega)-(m-1)\boldsymbol{\omega}_m^T\boldsymbol{\omega}_m},\end{aligned}$$

$\mathbf{M}$ is a matrix such that $\mathrm{vec}(\mathbf{M}) = \tilde{\mathbf{\Sigma}}_s\boldsymbol{\omega}_m$. It is easy to show that $\mathbf{M}$ is a combination of $\mathbf{\Sigma}_i$ $(i=1,\ldots,m-1)$ as $\mathbf{M}=\sum_{i=1}^{m-1}\omega_{mi}\mathbf{\Sigma}_i$ where ω_{mi} is the ith element of $\boldsymbol{\omega}_m$.

Similar to [Jin et al. 2009], we can use an online learning method to solve problem (7) and the algorithm is depicted in Table I. This algorithm is similar to that in [Jin et al. 2009] except the initial step for $\mathbf{\Sigma}_m^{(0)}$. In [Jin et al. 2009], the initial value for $\mathbf{\Sigma}_m^{(0)}$ is a zero matrix but here it is $\frac{\lambda_2'}{\lambda_1'}\mathbf{M}$. Note that $\mathbf{M}$ is a combination of the metrics learned from the source tasks where each combination weight is the task covariance between a source task and the target task. This agrees with our intuition that a positively correlated source task will have a large weight on the initial value for $\mathbf{\Sigma}_m$, an outlier task has negligle contribution and a negatively correlated task even has opposite effect.

Optimizing w.r.t. $\boldsymbol{\omega}_m$ and ω when $\mathbf{\Sigma}_m$ is fixed

Utilizing Eq. (5), the optimization problem with respect to $\boldsymbol{\omega}_m$ and ω is formulated as

$$\begin{aligned}\min_{\boldsymbol{\omega}_m,\omega,\mathbf{\Omega}} \quad & \mathrm{tr}(\tilde{\mathbf{\Sigma}}\mathbf{\Omega}^{-1}\tilde{\mathbf{\Sigma}}^T)\\ \text{s.t.} \quad & \mathbf{\Omega}=\begin{pmatrix}\frac{1-\omega}{m-1}\mathbf{I}_{m-1} & \boldsymbol{\omega}_m\\ \boldsymbol{\omega}_m^T & \omega\end{pmatrix}\\ & \omega(1-\omega)\geq(m-1)\boldsymbol{\omega}_m^T\boldsymbol{\omega}_m.\end{aligned} \tag{8}$$

Table I. Online Learning Algorithm for Problem (7)

Input: labeled data $(\mathbf{x}_j^m, y_j^m)$ $(j = 1, \ldots, n_m)$, matrix $\mathbf{M}$, λ_1', λ_2', and predefined learning rate η

Initialize $\mathbf{\Sigma}_m^{(0)} = \frac{\lambda_2'}{\lambda_1'}\mathbf{M}$;
for $t = 1, \ldots, T_{max}$ **do**
  Receive a pair of training data points $\{(\mathbf{x}_j^m, y_j^m), (\mathbf{x}_k^m, y_k^m)\}$;
  Compute y: $y = 1$ if $y_j^m = y_k^m$, and $y = -1$ otherwise;
  if the training pair $(\mathbf{x}_j^m, \mathbf{x}_k^m), y$ is classified correctly, i.e., $y(1 - \|\mathbf{x}_j^m - \mathbf{x}_k^m\|^2_{\mathbf{\Sigma}_m^{(t-1)}}) > 0$ **then**
    $\mathbf{\Sigma}_m^{(t)} = \mathbf{\Sigma}_m^{(t-1)}$;
  else if $y == -1$
    $\mathbf{\Sigma}_m^{(t)} = \mathbf{\Sigma}_m^{(t-1)} + \eta(\mathbf{x}_j^m - \mathbf{x}_k^m)(\mathbf{x}_j^m - \mathbf{x}_k^m)^T$;
  else
    $\mathbf{\Sigma}_m^{(t)} = \pi_{S_+}\left(\mathbf{\Sigma}_m^{(t-1)} - \eta(\mathbf{x}_j^m - \mathbf{x}_k^m)(\mathbf{x}_j^m - \mathbf{x}_k^m)^T\right)$ where $\pi_{S_+}(\mathbf{A})$ projects matrix $\mathbf{A}$ into the positive semidefinite cone;
  end if
end for

Output: metric $\mathbf{\Sigma}_m^{(T_{max})}$

We impose a constraint as $\tilde{\mathbf{\Sigma}}\mathbf{\Omega}^{-1}\tilde{\mathbf{\Sigma}}^T \preceq \frac{1}{t}\mathbf{I}_{d^2}$ and the objective function becomes $\min \frac{1}{t}$. Using the Schur complement, we can get

$$\tilde{\mathbf{\Sigma}}\mathbf{\Omega}^{-1}\tilde{\mathbf{\Sigma}}^T \preceq \frac{1}{t}\mathbf{I}_{d^2} \Longleftrightarrow \begin{pmatrix} \mathbf{\Omega} & \tilde{\mathbf{\Sigma}}^T \\ \tilde{\mathbf{\Sigma}} & \frac{1}{t}\mathbf{I}_{d^2} \end{pmatrix} \succeq \mathbf{0}.$$

By using the Schur complement again, we get

$$\begin{pmatrix} \mathbf{\Omega} & \tilde{\mathbf{\Sigma}}^T \\ \tilde{\mathbf{\Sigma}} & \frac{1}{t}\mathbf{I}_{d^2} \end{pmatrix} \succeq \mathbf{0} \Longleftrightarrow \mathbf{\Omega} - t\tilde{\mathbf{\Sigma}}^T\tilde{\mathbf{\Sigma}} \succeq \mathbf{0}.$$

We write $\tilde{\mathbf{\Sigma}}^T\tilde{\mathbf{\Sigma}} = \begin{pmatrix} \mathbf{\Psi}_{11} & \mathbf{\Psi}_{12} \\ \mathbf{\Psi}_{12}^T & \Psi_{22} \end{pmatrix}$ where $\mathbf{\Psi}_{11} \in \mathbb{R}^{(m-1)\times(m-1)}$, $\mathbf{\Psi}_{12} \in \mathbb{R}^{(m-1)\times 1}$ and $\Psi_{22} \in \mathbb{R}$. Then $\mathbf{\Omega} - t\tilde{\mathbf{\Sigma}}^T\tilde{\mathbf{\Sigma}} \succeq \mathbf{0}$ is equivalent to

$$\frac{1-\omega}{m-1}\mathbf{I}_{m-1} - t\mathbf{\Psi}_{11} \succeq \mathbf{0}$$
$$\omega - t\Psi_{22} \geq (\boldsymbol{\omega}_m - t\mathbf{\Psi}_{12})^T \left(\frac{1-\omega}{m-1}\mathbf{I}_{m-1} - t\mathbf{\Psi}_{11}\right)^{-1} (\boldsymbol{\omega}_m - t\mathbf{\Psi}_{12}).$$

Let $\mathbf{U}$ and $\lambda_1, \ldots, \lambda_{m-1}$ denote the eigenvector matrix and eigenvalues of $\mathbf{\Psi}_{11}$ with $\lambda_1 \geq \ldots \geq \lambda_{m-1} \geq 0$. Then

$$\frac{1-\omega}{m-1}\mathbf{I}_{m-1} - t\mathbf{\Psi}_{11} \succeq \mathbf{0} \Longleftrightarrow \frac{1-\omega}{m-1} \geq \lambda_1 t$$

and

$$\left(\frac{1-\omega}{m-1}\mathbf{I}_{m-1} - t\mathbf{\Psi}_{11}\right)^{-1} = \mathbf{U}\,\mathrm{diag}(\frac{1-\omega}{m-1} - t\lambda_1, \ldots, \frac{1-\omega}{m-1} - t\lambda_{m-1})\,\mathbf{U}^T.$$

Combining the above results, problem (8) is formulated as

$$\begin{aligned}\min_{\boldsymbol{\omega}_m,\omega,\mathbf{f},t} \quad & -t \\ \text{s.t.} \quad & \frac{1-\omega}{m-1} \geq t\lambda_1 \\ & \mathbf{f} = \mathbf{U}^T(\boldsymbol{\omega}_m - t\boldsymbol{\Psi}_{12}) \\ & \sum_{j=1}^{m-1} \frac{f_j^2}{\frac{1-\omega}{m-1} - t\lambda_j} \leq \omega - t\Psi_{22} \\ & \omega(1-\omega) \geq (m-1)\boldsymbol{\omega}_m^T\boldsymbol{\omega}_m, \end{aligned} \tag{9}$$

where f_j is the jth element of $\mathbf{f}$. By introducing new variables h_j and r_j ($j = 1, \ldots, m-1$), (9) is reformulated as

$$\begin{aligned}\min_{\boldsymbol{\omega}_m,\omega,\mathbf{f},t,\{h_j\},\{r_j\}} \quad & -t \\ \text{s.t.} \quad & \frac{1-\omega}{m-1} \geq t\lambda_1 \\ & \mathbf{f} = \mathbf{U}^T(\boldsymbol{\omega}_m - t\boldsymbol{\Psi}_{12}) \\ & \sum_{j=1}^{m-1} h_j \leq \omega - t\Psi_{22} \\ & r_j = \frac{1-\omega}{m-1} - t\lambda_j \ \forall j \\ & \frac{f_j^2}{r_j} \leq h_j \ \forall j \\ & \omega(1-\omega) \geq (m-1)\boldsymbol{\omega}_m^T\boldsymbol{\omega}_m. \end{aligned} \tag{10}$$

Since

$$\frac{f_j^2}{r_j} \leq h_j \iff \left\| \begin{pmatrix} f_j \\ \frac{r_j - h_j}{2} \end{pmatrix} \right\|_2 \leq \frac{r_j + h_j}{2}$$

and

$$\omega(1-\omega) \geq (m-1)\boldsymbol{\omega}_m^T\boldsymbol{\omega}_m \iff \left\| \begin{pmatrix} \sqrt{m-1}\boldsymbol{\omega}_m \\ \frac{\omega-1}{2} \\ \omega \end{pmatrix} \right\|_2 \leq \frac{\omega+1}{2},$$

problem (10) is a second-order cone programming (SOCP) problem [Lobo et al. 1998] with $O(m)$ variables and $O(m)$ constraints. In many applications, m is very small and we can use a standard solver to solve problem (10) very efficiently.

We set the initial value of ω to $\frac{1}{m}$ and that of $\boldsymbol{\omega}_m$ to a zero vector which corresponds to the assumption that the target task is unrelated to the source tasks.

After learning the optimal values of $\boldsymbol{\Sigma}_m$, we can make prediction for a new data point. Given a test data point $\mathbf{x}_\star^m$ for the target task T_m, we first calculate the distances between $\mathbf{x}_\star^m$ and all training data points in T_m based on the learned metric $\boldsymbol{\Sigma}_m$ and then use the k-nearest neighbor classifier to classify $\mathbf{x}_\star^m$, where we choose $k = 1$ for simplicity in our experiments.

4.3. Regularization Parameters λ_1 and λ_2

Setting the regularization parameters λ_1 and λ_2 to suitable values plays a crucial role in our method in order to deliver good performance. In the transfer learning setting, however, the labeled data available in the target task is so scarce that model selection techniques such as cross-validation cannot be used to determine the values of λ_1 and λ_2. On the other hand, Bayesian regularization [Williams 1995; Cawley et al. 2006; Foo et al. 2009] is usually very effective under this setting by integrating out the regularization parameters. We propose below a Bayesian regularization scheme for our model.

Based on the probabilistic interpretation in Section 3, problem (4) defines a probabilistic model:

$$p(\mathbf{y}_m|\mathbf{X}_m,\mathbf{\Sigma}_m) \propto \exp\Big\{-\frac{2}{n_m(n_m-1)}\sum_{j<k} g\Big(y^m_{j,k}\left[1-\|\mathbf{x}^m_j-\mathbf{x}^m_k\|^2_{\mathbf{\Sigma}_m}\right]\Big)\Big\} \tag{11}$$

$$p(\mathbf{\Sigma}_m|\lambda_1,\lambda_2) \propto \exp\Big\{-\frac{\lambda_1}{2}\|\mathbf{\Sigma}_m\|^2_F-\frac{\lambda_2}{2}\mathbf{tr}(\tilde{\mathbf{\Sigma}}\mathbf{\Omega}^{-1}\tilde{\mathbf{\Sigma}}^T)\Big\}, \tag{12}$$

where $\mathbf{X}_m = (\mathbf{x}^m_1,\ldots,\mathbf{x}^m_{n_m})$ is the data matrix for the target task T_m and $\mathbf{y}_m = (y^m_1,\ldots,y^m_{n_m})^T$. Eq. (11) defines the likelihood given $\mathbf{\Sigma}_m$ and the prior on $\mathbf{\Sigma}_m$ is defined in Eq. (12). Similar to [Foo et al. 2009], we impose Gamma priors on λ_1 and λ_2:

$$\begin{aligned}\lambda_1 &\sim \mathcal{G}(\alpha,\beta)\\ \lambda_2 &\sim \mathcal{G}(\alpha,\beta),\end{aligned}$$

where $\mathcal{G}(\alpha,\beta)$ denotes the Gamma distribution with probabilistic density function

$$p(s|\alpha,\beta) = \frac{s^{\alpha-1}\beta^\alpha\exp\{-\beta s\}}{\Gamma(\alpha)}$$

with $\Gamma(\cdot)$ denoting the Gamma function. In general, the parameters of the Gamma priors for λ_1 and λ_2 can be assigned different values. Here, for simplicity, we use the same parameter values for both λ_1 and λ_2 as [Foo et al. 2009] did. By integrating out λ_1 and λ_2, we get

$$\begin{aligned}p(\mathbf{\Sigma}_m) &= \int p(\mathbf{\Sigma}_m|\lambda_1,\lambda_2)p(\lambda_1)p(\lambda_2)d\lambda_1 d\lambda_2\\
&= \int_0^\infty \Big(\frac{\lambda_1}{2\pi}\Big)^{\frac{d^2}{2}}\exp\Big\{-\frac{\lambda_1}{2}\|\mathbf{\Sigma}_m\|^2_F\Big\}\frac{\lambda_1^{\alpha-1}\beta^\alpha\exp\{-\beta\lambda_1\}}{\Gamma(\alpha)}d\lambda_1 \times\\
&\quad \int_0^\infty \Big(\frac{\lambda_2}{2\pi}\Big)^{\frac{dm}{2}}\exp\Big\{-\frac{\lambda_2}{2}\mathbf{tr}(\tilde{\mathbf{\Sigma}}\mathbf{\Omega}^{-1}\tilde{\mathbf{\Sigma}}^T)\Big\}\frac{\lambda_2^{\alpha-1}\beta^\alpha\exp\{-\beta\lambda_2\}}{\Gamma(\alpha)}d\lambda_2\\
&= \int_0^\infty \frac{\beta^\alpha}{(2\pi)^{\frac{d^2}{2}}\Gamma(\alpha)}\lambda_1^{\alpha+\frac{d^2}{2}-1}\exp\Big\{-(\beta+\frac{1}{2}\|\mathbf{\Sigma}_m\|^2_F)\lambda_1\Big\}d\lambda_1 \times\\
&\quad \int_0^\infty \frac{\beta^\alpha}{(2\pi)^{\frac{dm}{2}}\Gamma(\alpha)}\lambda_2^{\alpha+\frac{dm}{2}-1}\exp\Big\{-(\beta+\frac{1}{2}\mathbf{tr}(\tilde{\mathbf{\Sigma}}\mathbf{\Omega}^{-1}\tilde{\mathbf{\Sigma}}^T))\lambda_2\Big\}d\lambda_2\\
&= \frac{\beta^\alpha\Gamma(\alpha+\frac{d^2}{2})}{(2\pi)^{\frac{d^2}{2}}\Gamma(\alpha)(\beta+\frac{1}{2}\|\mathbf{\Sigma}_m\|^2_F)^{\alpha+\frac{d^2}{2}}}\frac{\beta^\alpha\Gamma(\alpha+\frac{dm}{2})}{(2\pi)^{\frac{dm}{2}}\Gamma(\alpha)(\beta+\frac{1}{2}\mathbf{tr}(\tilde{\mathbf{\Sigma}}\mathbf{\Omega}^{-1}\tilde{\mathbf{\Sigma}}^T))^{\alpha+\frac{dm}{2}}}\end{aligned}$$

and so

$$\ln p(\mathbf{\Sigma}_m) = -(\alpha+\frac{d^2}{2})\ln(\beta+\frac{1}{2}\|\mathbf{\Sigma}_m\|^2_F)-(\alpha+\frac{dm}{2})\ln(\beta+\frac{1}{2}\mathbf{tr}(\tilde{\mathbf{\Sigma}}\mathbf{\Omega}^{-1}\tilde{\mathbf{\Sigma}}^T))+\text{Const.}$$

The maximum a posterior (MAP) solution can be obtained by solving the following optimization problem

$$\begin{aligned}\min_{\mathbf{\Sigma}_m,\mathbf{\Omega}} & \frac{2}{n_m(n_m-1)}\sum_{j<k} g\Big(y_{j,k}^m\left[1-\|\mathbf{x}_j^m-\mathbf{x}_k^m\|^2_{\mathbf{\Sigma}_m}\right]\Big)+(\alpha+\frac{d^2}{2})\ln(\beta+\frac{1}{2}\|\mathbf{\Sigma}_m\|_F^2)\\ &+(\alpha+\frac{dm}{2})\ln(\beta+\frac{1}{2}\mathbf{tr}(\tilde{\mathbf{\Sigma}}\mathbf{\Omega}^{-1}\tilde{\mathbf{\Sigma}}^T))\\ \text{s.t. } & \mathbf{\Sigma}_m\succeq\mathbf{0}\\ & \tilde{\mathbf{\Sigma}}=(\mathbf{vec}(\mathbf{\Sigma}_1),\ldots,\mathbf{vec}(\mathbf{\Sigma}_{m-1}),\mathbf{vec}(\mathbf{\Sigma}_m))\\ & \mathbf{\Omega}\succeq 0\\ & \mathbf{tr}(\mathbf{\Omega})=1.\end{aligned}\tag{13}$$

Since problem (13) is non-convex due to the non-convexity of the logarithm function, we use a majorization-minimization (MM) algorithm [Lange et al. 2000] to solve it. An MM algorithm, which can be viewed as a generalization of the expectation-maximization (EM) algorithm [Dempster et al. 1977], is an iterative algorithm which derives an upper bound of the objective function as a surrogate function in each iteration based on the solution obtained in the previous iteration and then minimizes the surrogate function instead of the original objective function. The solution found by an MM algorithm can be proved to be the locally optimal solution. Since the logarithm function is a concave function, we have $\ln(a)\leq\ln(b)+\frac{a-b}{b}$ for any two scalars a and b due to the first-order property of a concave function. In the $(k+1)$th iteration of an MM algorithm, we have

$$\ln(\beta+\frac{1}{2}\|\mathbf{\Sigma}_m\|_F^2)\leq\ln(\beta+\frac{1}{2}\|\mathbf{\Sigma}_m^{(k)}\|_F^2)+\frac{\beta+\frac{1}{2}\|\mathbf{\Sigma}_m\|_F^2}{\beta+\frac{1}{2}\|\mathbf{\Sigma}_m^{(k)}\|_F^2}-1$$

and

$$\ln\left(\beta+\frac{1}{2}\mathbf{tr}(\tilde{\mathbf{\Sigma}}\mathbf{\Omega}^{-1}\tilde{\mathbf{\Sigma}}^T)\right)\leq\ln\left(\beta+\frac{1}{2}\mathbf{tr}\big(\tilde{\mathbf{\Sigma}}^{(k)}\mathbf{\Omega}^{-1}(\tilde{\mathbf{\Sigma}}^{(k)})^T\big)\right)+\frac{\beta+\frac{1}{2}\mathbf{tr}(\tilde{\mathbf{\Sigma}}\mathbf{\Omega}^{-1}\tilde{\mathbf{\Sigma}}^T)}{\beta+\frac{1}{2}\mathbf{tr}\big(\tilde{\mathbf{\Sigma}}^{(k)}\mathbf{\Omega}^{-1}(\tilde{\mathbf{\Sigma}}^{(k)})^T\big)}-1$$

where $\mathbf{\Sigma}_m^{(k)}$ (and also $\tilde{\mathbf{\Sigma}}^{(k)}$) denotes the solution obtained in the kth iteration. So in the $(k+1)$th iteration, the optimization problem can be formulated as

$$\begin{aligned}\min_{\mathbf{\Sigma}_m,\mathbf{\Omega}} & \frac{2}{n_m(n_m-1)}\sum_{j<k} g\Big(y_{j,k}^m\left[1-\|\mathbf{x}_j^m-\mathbf{x}_k^m\|^2_{\mathbf{\Sigma}_m}\right]\Big)+\frac{\tilde{\lambda}_1}{2}\|\mathbf{\Sigma}_m\|_F^2+\frac{\tilde{\lambda}_2}{2}\mathbf{tr}(\tilde{\mathbf{\Sigma}}\mathbf{\Omega}^{-1}\tilde{\mathbf{\Sigma}}^T)\\ \text{s.t. } & \mathbf{\Sigma}_m\succeq\mathbf{0}\\ & \tilde{\mathbf{\Sigma}}=(\mathbf{vec}(\mathbf{\Sigma}_1),\ldots,\mathbf{vec}(\mathbf{\Sigma}_{m-1}),\mathbf{vec}(\mathbf{\Sigma}_m))\\ & \mathbf{\Omega}\succeq 0\\ & \mathbf{tr}(\mathbf{\Omega})=1,\end{aligned}\tag{14}$$

where

$$\begin{aligned}\tilde{\lambda}_1 &= \frac{\alpha+\frac{d^2}{2}}{\beta+\frac{1}{2}\|\mathbf{\Sigma}_m^{(k)}\|_F^2}\\ \tilde{\lambda}_2 &= \frac{\alpha+\frac{dm}{2}}{\beta+\frac{1}{2}\mathbf{tr}\big(\tilde{\mathbf{\Sigma}}^{(k)}\mathbf{\Omega}^{-1}(\tilde{\mathbf{\Sigma}}^{(k)})^T\big)}.\end{aligned}$$

Problem (14) is almost identical to problem (4) with the exception of the regularization parameters. In problem (4), the regularization parameters λ_1 and λ_2 are set by the user but the regularization parameters $\tilde{\lambda}_1$ and $\tilde{\lambda}_2$ are automatically determined by the solution obtained in the previous iteration. Since problems (14) and (4) have the same formulation, we can still use the optimization procedure developed in the previous sections with the only modification in the values of the regularization parameters. Thus this model selection technique can be incorporated into the learning procedure easily. Moreover, the experiments in [Foo et al. 2009] show that the system performance is not sensitive to the values of α and β and hence their values are easy to set.

5. SEMI-SUPERVISED EXTENSION

In many applications, even though the labeled data available in the target task is scarce due to the costly labeling effort required, large quantities of unlabeled data are available at very low cost. Semi-supervised learning [Chapelle et al. 2006] is an increasingly popular learning paradigm that seeks to exploit the unlabeled data especially under situations in which the labeled data is scarce. When some conditions about the data are satisfied, it has been demonstrated by many researchers that the generalization performance of learning systems can be improved by learning under the semi-supervised setting. In this section, we investigate a semi-supervised extension of transfer metric learning that explicitly exploits unlabeled data in the target task.

The manifold assumption [Belkin et al. 2006] is widely adopted by many semi-supervised learning algorithms. Under this assumption, nearby points are more likely to have the same class label for classification problems and similar low-dimensional representation for dimensionality reduction problems. Similarly, for metric learning, we want nearby points to remain near to each other in the new space after learning the metric. This is the central idea on which our semi-supervised extension of transfer metric learning is based.

Given l_m labeled data points $\{(\mathbf{x}_j^m, y_j^m)\}_{j=1}^{l_m}$ and u_m unlabeled points $\{\mathbf{x}_j^m\}_{j=l_m+1}^{l_m+u_m}$ (i.e., a total of $n_m = l_m + u_m$ data points) for the target task T_m, we first construct a K-nearest neighbor graph $G = (V, E)$ in a way called local scaling [Zelnik-Manor and Perona 2004], with the vertex set $V = \{1, \ldots, n_m\}$ corresponding to the labeled and unlabeled data points and the edge set $E \subseteq V \times V$ representing the relationships between data points. Each edge is assigned a weight w_{ij} which reflects the similarity between points $\mathbf{x}_i^m$ and $\mathbf{x}_j^m$:

$$w_{ij} = \begin{cases} \exp\left(-\frac{\|\mathbf{x}_i^m - \mathbf{x}_j^m\|_2^2}{\sigma_i \sigma_j}\right) & \text{if } \mathbf{x}_i^m \in N_K(\mathbf{x}_j^m) \text{ or } \mathbf{x}_j^m \in N_K(\mathbf{x}_i^m) \\ 0 & \text{otherwise} \end{cases}$$

where $\|\cdot\|_2$ denotes the 2-norm of a vector, $N_K(\mathbf{x}_i^m)$ denotes the neighborhood set of K-nearest neighbors of $\mathbf{x}_i^m$, and σ_i (σ_j) is the distance between $\mathbf{x}_i^m$ ($\mathbf{x}_j^m$) and its Kth nearest neighbor.

Then we define a new regularization term based on the manifold assumption as follows:

$$
\begin{aligned}
&\sum_{i=1}^{n_m}\sum_{j=1}^{n_m} w_{ij}\|\mathbf{x}_i^m - \mathbf{x}_j^m\|_{\mathbf{\Sigma}_m}^2 \\
&= \sum_{i=1}^{n_m}\sum_{j=1}^{n_m} w_{ij}(\mathbf{x}_i^m - \mathbf{x}_j^m)^T\mathbf{\Sigma}_m(\mathbf{x}_i^m - \mathbf{x}_j^m) \\
&= \sum_{i=1}^{n_m}\sum_{j=1}^{n_m} w_{ij}\mathbf{tr}\big(\mathbf{\Sigma}_m(\mathbf{x}_i^m - \mathbf{x}_j^m)(\mathbf{x}_i^m - \mathbf{x}_j^m)^T\big) \\
&= \mathbf{tr}\Big(\mathbf{\Sigma}_m\sum_{i=1}^{n_m}\sum_{j=1}^{n_m} w_{ij}(\mathbf{x}_i^m - \mathbf{x}_j^m)(\mathbf{x}_i^m - \mathbf{x}_j^m)^T\Big) \\
&= \mathbf{tr}\Big(\mathbf{\Sigma}_m\sum_{i=1}^{n_m}\sum_{j=1}^{n_m} w_{ij}\big(\|\mathbf{x}_i^m\|_2^2 + \|\mathbf{x}_j^m\|_2^2 - 2(\mathbf{x}_j^m)^T\mathbf{x}_i^m\big)\Big) \\
&= 2\mathbf{tr}\Big(\mathbf{\Sigma}_m\big(\sum_{i=1}^{n_m} d_i\|\mathbf{x}_i^m\|_2^2 - \sum_{i=1}^{n_m}\sum_{j=1}^{n_m} w_{ij}(\mathbf{x}_j^m)^T\mathbf{x}_i^m\big)\Big) \\
&= 2\mathbf{tr}\Big(\mathbf{\Sigma}_m\mathbf{X}_m(\mathbf{D}-\mathbf{W})\mathbf{X}_m^T\Big) = 2\mathbf{tr}\Big(\mathbf{\Sigma}_m\mathbf{X}_m\mathbf{L}\mathbf{X}_m^T\Big),
\end{aligned}
\tag{15}
$$

where $d_i = \sum_{j=1}^{n_m} w_{ij}$, $\mathbf{D} = \mathbf{diag}(d_1, \ldots, d_{n_m})$ is a diagonal matrix with the ith diagonal element being d_i, $\mathbf{W}$ is the similarity matrix with the (i,j)th element being w_{ij}, $\mathbf{L} = \mathbf{D} - \mathbf{W}$ is the Laplacian matrix [Chung 1997] of $\mathbf{W}$, and $\mathbf{X}_m = (\mathbf{x}_1^m, \ldots, \mathbf{x}_{n_m}^m)$ is the data matrix for the target task T_m.

Adding the new regularization term in Eq. (15) into the objective function of problem (6), we get the optimization problem for semi-supervised transfer metric learning as follows:

$$
\begin{aligned}
\min_{\mathbf{\Sigma}_m, \boldsymbol{\omega}_m, \omega, \mathbf{\Omega}} \quad & \frac{2}{n_m(n_m-1)}\sum_{j<k} g\Big(y_{j,k}^m\left[1 - \|\mathbf{x}_j^m - \mathbf{x}_k^m\|_{\mathbf{\Sigma}_m}^2\right]\Big) + \frac{\lambda_1}{2}\|\mathbf{\Sigma}_m\|_F^2 \\
& + \frac{\lambda_2}{2}\mathbf{tr}(\tilde{\mathbf{\Sigma}}\mathbf{\Omega}^{-1}\tilde{\mathbf{\Sigma}}^T) + \lambda_3\mathbf{tr}\Big(\mathbf{\Sigma}_m\mathbf{X}_m\mathbf{L}\mathbf{X}_m^T\Big) \\
\text{s.t.} \quad & \mathbf{\Sigma}_m \succeq \mathbf{0} \\
& \mathbf{\Omega} = \begin{pmatrix} \frac{1-\omega}{m-1}\mathbf{I}_{m-1} & \boldsymbol{\omega}_m \\ \boldsymbol{\omega}_m^T & \omega \end{pmatrix} \\
& \tilde{\mathbf{\Sigma}} = (\tilde{\mathbf{\Sigma}}_s, \mathbf{vec}(\mathbf{\Sigma}_m)) \\
& \omega(1-\omega) \geq (m-1)\boldsymbol{\omega}_m^T\boldsymbol{\omega}_m,
\end{aligned}
\tag{16}
$$

where λ_3 is the regularization parameter for the newly added regularization term. Since the new regularization term is linear with respect to $\mathbf{\Sigma}_m$, problem (16) is still a convex optimization problem and hence we still use an alternating method to solve it.

When $\boldsymbol{\omega}_m$ and ω are fixed, the optimization problem with respect to $\boldsymbol{\Sigma}_m$ is formulated as

$$\begin{aligned}\min_{\boldsymbol{\Sigma}_m} \quad & \frac{2}{n_m(n_m-1)}\sum_{j<k} g\Big(y_{j,k}^m\left[1-\|\mathbf{x}_j^m-\mathbf{x}_k^m\|_{\boldsymbol{\Sigma}_m}^2\right]\Big)+\frac{\lambda_1'}{2}\|\boldsymbol{\Sigma}_m\|_F^2-\lambda_2'\mathbf{tr}(\boldsymbol{\Sigma}_m\mathbf{M})\\ & +\lambda_3\mathbf{tr}\Big(\boldsymbol{\Sigma}_m\mathbf{X}_m\mathbf{L}\mathbf{X}_m^T\Big)\\ \text{s.t.} \quad & \boldsymbol{\Sigma}_m\succeq\mathbf{0}. \end{aligned} \tag{17}$$

Similar to problem (7), we still use the online algorithm in Table I with the only modification being that the initial value for $\boldsymbol{\Sigma}_m$ is set to $\frac{1}{\lambda_1'}\Big(\lambda_2'\mathbf{M}-\lambda_3\mathbf{X}_m\mathbf{L}\mathbf{X}_m^T\Big)$. When $\boldsymbol{\Sigma}_m$ is fixed, the optimization problem with respect to $\boldsymbol{\omega}_m$ and ω is identical to problem (10) which is an SOCP problem. These two steps iterate until convergence to the optimal solution.

5.1. Regularization Parameters λ_1, λ_2 and λ_3

Similar to Sec. 4.3, we use Bayesian regularization to address the model selection problem for the regularization parameters λ_1, λ_2 and λ_3. The likelihood is also defined as in Eq. (11) and the change lies in the prior on $\boldsymbol{\Sigma}_m$:

$$p(\boldsymbol{\Sigma}_m|\lambda_1,\lambda_2,\lambda_3)\propto\exp\{-\frac{\lambda_1}{2}\|\boldsymbol{\Sigma}_m\|_F^2-\frac{\lambda_2}{2}\mathbf{tr}(\tilde{\boldsymbol{\Sigma}}\boldsymbol{\Omega}^{-1}\tilde{\boldsymbol{\Sigma}}^T)-\lambda_3\mathbf{tr}\Big(\boldsymbol{\Sigma}_m\mathbf{X}_m\mathbf{L}\mathbf{X}_m^T\Big)\}. \tag{18}$$

Similar to the supervised version, we impose Gamma priors on λ_1, λ_2 and λ_3:

$$\begin{aligned}\lambda_1 &\sim \mathcal{G}(\alpha,\beta)\\ \lambda_2 &\sim \mathcal{G}(\alpha,\beta)\\ \lambda_3 &\sim \mathcal{G}(\alpha,\beta).\end{aligned}$$

Then, by integrating out all the regularization parameters, we get

$$\begin{aligned}& p(\boldsymbol{\Sigma}_m)\\ =&\frac{\beta^\alpha\Gamma(\alpha+\frac{d^2}{2})}{(2\pi)^{\frac{d^2}{2}}\Gamma(\alpha)(\beta+\frac{1}{2}\|\boldsymbol{\Sigma}_m\|_F^2)^{\alpha+\frac{d^2}{2}}}\times\frac{\beta^\alpha\Gamma(\alpha+\frac{dm}{2})}{(2\pi)^{\frac{dm}{2}}\Gamma(\alpha)(\beta+\frac{1}{2}\mathbf{tr}(\tilde{\boldsymbol{\Sigma}}\boldsymbol{\Omega}^{-1}\tilde{\boldsymbol{\Sigma}}^T))^{\alpha+\frac{dm}{2}}}\times\\ &\frac{\beta^\alpha}{(\beta+\mathbf{tr}(\boldsymbol{\Sigma}_m\mathbf{M}))^\alpha}\times F(\lambda_1,\lambda_3),\end{aligned} \tag{19}$$

where $\mathbf{M}=\mathbf{X}_m\mathbf{L}\mathbf{X}_m^T$ and $F(\lambda_1,\lambda_3)=\int_0^\infty\int_0^\infty\exp\Big\{-\frac{\lambda_3^2}{2\lambda_1}\|\mathbf{M}\|_F^2\Big\}p(\lambda_1)p(\lambda_3)d\lambda_1 d\lambda_3$ is an irrelevant term with respect to $\boldsymbol{\Sigma}_m$. The detailed derivation for Eq. (19) is presented in Appendix B. The MAP solution can be obtained by solving the following problem

$$\begin{aligned}\min_{\boldsymbol{\Sigma}_m,\boldsymbol{\omega}_m,\omega,\boldsymbol{\Omega}} \quad & \frac{2}{n_m(n_m-1)}\sum_{j<k} g\Big(y_{j,k}^m\left[1-\|\mathbf{x}_j^m-\mathbf{x}_k^m\|_{\boldsymbol{\Sigma}_m}^2\right]\Big)+(\alpha+\frac{d^2}{2})\ln(\beta+\frac{1}{2}\|\boldsymbol{\Sigma}_m\|_F^2)\\ & +(\alpha+\frac{dm}{2})\ln(\beta+\frac{1}{2}\mathbf{tr}(\tilde{\boldsymbol{\Sigma}}\boldsymbol{\Omega}^{-1}\tilde{\boldsymbol{\Sigma}}^T))+\alpha\ln\Big(\beta+\mathbf{tr}\big(\boldsymbol{\Sigma}_m\mathbf{X}_m\mathbf{L}\mathbf{X}_m^T\big)\Big)\\ \text{s.t.} \quad & \boldsymbol{\Sigma}_m\succeq\mathbf{0}\\ & \boldsymbol{\Omega}=\begin{pmatrix}\frac{1-\omega}{m-1}\mathbf{I}_{m-1} & \boldsymbol{\omega}_m\\ \boldsymbol{\omega}_m^T & \omega\end{pmatrix}\\ & \tilde{\boldsymbol{\Sigma}}=(\tilde{\boldsymbol{\Sigma}}_s,\mathbf{vec}(\boldsymbol{\Sigma}_m))\\ & \omega(1-\omega)\geq(m-1)\boldsymbol{\omega}_m^T\boldsymbol{\omega}_m.\end{aligned} \tag{20}$$

We also use an MM algorithm to solve problem (20). In the $(k+1)$th iteration, the optimization problem is identical to problem (16) with the only difference lying in the definition of the regularization parameters:

$$\begin{aligned}\tilde{\lambda}_1 &= \frac{\alpha + \frac{d^2}{2}}{\beta + \frac{1}{2}\|\mathbf{\Sigma}_m^{(k)}\|_F^2} \\ \tilde{\lambda}_2 &= \frac{\alpha + \frac{dm}{2}}{\beta + \frac{1}{2}\mathbf{tr}\big(\tilde{\mathbf{\Sigma}}^{(k)}\mathbf{\Omega}^{-1}(\tilde{\mathbf{\Sigma}}^{(k)})^T\big)} \\ \tilde{\lambda}_3 &= \frac{\alpha}{\beta + \mathbf{tr}\big(\mathbf{\Sigma}_m^{(k)}\mathbf{X}_m\mathbf{L}\mathbf{X}_m^T\big)}.\end{aligned}$$

6. EXPERIMENTS

We study TML and STML empirically in this section.

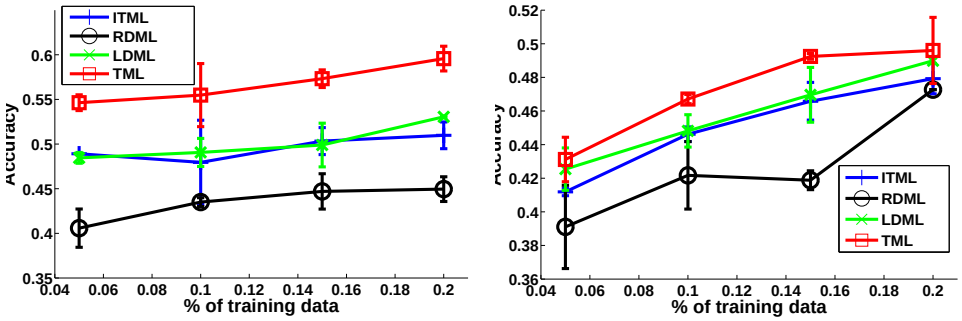
6.1. Experiments under Supervised Setting

We compare TML with two metric learning methods, ITML[4] [Davis et al. 2007] and RDML [Jin et al. 2009], and another metric learning method for transfer learning, L-DML [Zha et al. 2009]. We use the CVX solver [Grant and Boyd 2009][5] to solve problem (10). We set the learning rate η in Table I to 0.01. For ITML, RDML and L-DML, the best parameters reported in [Davis et al. 2007; Jin et al. 2009; Zha et al. 2009] are used. For classification, we use a 1-Nearest-Neighbor classifier. The distance metric matrices of the source tasks are learned using the RDML method.

6.1.1. Wine Quality Classification. The wine dataset[6] is about wine quality including red and white wine samples. The features include objective tests (e.g., PH values) and the output is based on sensory data. The labels are given by experts with grades between 0 (very bad) and 10 (very excellent). There are 1599 records for the red wine and 4898 for the white wine and so there are two tasks, one for red wine classification and the other for white wine classification. So the two tasks are both multi-class classification problems. Since both RDML and our proposed method in Table I formulate the multi-class classification problem as a binary classification problem by assigning a point pair a positive label if these two data points are from the same class and a negative label otherwise, there is no difference to process the task with multi-class classification problem or binary classification problem for our method. Each task is treated as the target task and the other task as the source task. To see the effect of varying the size of the training set, we vary the percentage of the training data used from 5% to 20%. Each configuration is repeated 10 times. The mean and standard deviation of the classification accuracy are reported in Fig. 1(a) and 1(b). From the results, we can see that the performance of L-DML is comparable to that of ITML and RDML, and TML is always the best one for both tasks.

Moreover, to demonstrate the ability of our method to detect outlier tasks, we add a synthetic task as a source task. The synthetic task is an XOR classification task and its 2-dimensional representation obtained by principal component analysis (PCA) is illustrated in Fig. 2. We use our method to perform transfer metric learning with the settings similar to those above. The mean task correlation matrix derived from the

[4]The implementation of ITML can be found in http://www.cs.utexas.edu/users/pjain/itml/.
[5]http://stanford.edu/∼boyd/cvx
[6]http://archive.ics.uci.edu/ml/datasets/Wine+Quality



(a) 1st task (b) 2nd task

Fig. 1. Overall performance on wine quality classification application. Subfigure (a) reports the performance when the first task is used as the target task and the second task is used as the source task, and subfigure (b) reports the performance when the roles of the two tasks are reversed.

task covariance matrix is

$$\mathbf{C} = \begin{pmatrix} 1.0000 & 0.0000 & 0.5623 \\ 0.0000 & 1.0000 & -0.0438 \\ 0.5623 & -0.0438 & 1.0000 \end{pmatrix}$$

when the red wine classification task is treated as the target task. When the white wine classification task is treated as the target task, the mean task correlation matrix is

$$\mathbf{C} = \begin{pmatrix} 1.0000 & 0.0000 & 0.6217 \\ 0.0000 & 1.0000 & -0.0531 \\ 0.6217 & -0.0531 & 1.0000 \end{pmatrix}.$$

From the results, we can see that the correlation between the target and synthetic source tasks is close to 0 (-0.0438 or -0.0531), showing that our method can make use of task relationships learned to detect outlier tasks. Moreover, the correlation between the white and red wine tasks is much higher indicating that the two tasks are similar to each other.

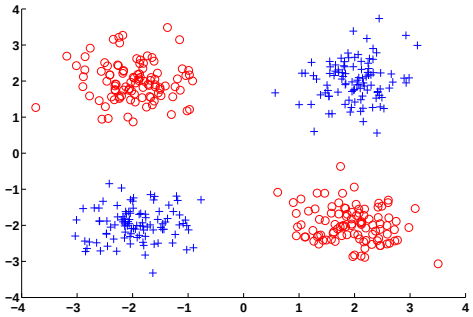
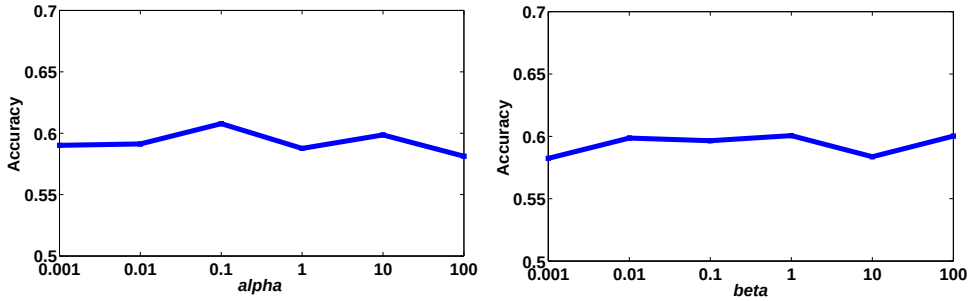
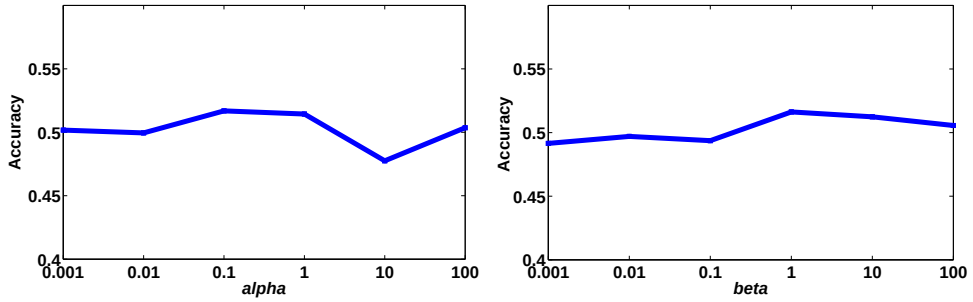


Fig. 2. 2-dimensional projection of XOR data by PCA, in which data points from the same class are shown with the same symbol (color).

To test the sensitivity of the performance of our method to the values of α and β used in the Gamma prior, we try different values of α and β in a setting with 20% of the data points used for training. The results are reported in Fig. 3(a) to Fig. 3(d). We can see that our method is not very sensitive to the choice of α and β.



(a) 1st task (b) 1st task



(c) 2nd task (d) 2nd task

Fig. 3. Effect of varying α and β on the accuracy of the wine quality classification application.

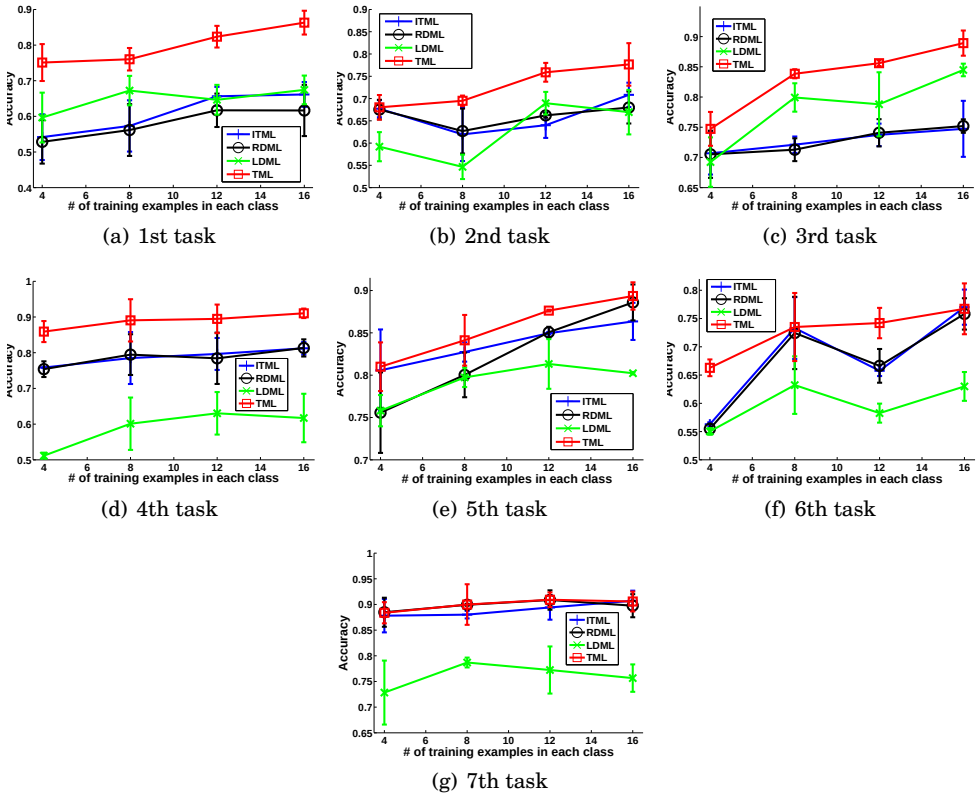
6.1.2. Handwritten Letter Classification. The handwritten letter classification applicaton[7] consists of seven tasks where each task is a binary classification problem. The corresponding letters for each task are: c/e, g/y, m/n, a/g, a/o, f/t and h/n. Each data point has 128 features corresponding to the pixel values of the handwritten letter images. For each task, there are about 1000 positive and 1000 negative data points. The experimental settings are the same as those for wine quality classification above. The results are plotted in Fig. 4(a) to 4(g). From the results, we find that the performance of L-MDL is worse than that of ITML and RDML on some tasks (4th, 6th and 7th tasks). This may be due to the fact that the objective function of L-MDL is non-convex and hence it is easy to get trapped in bad local minima. TML gives the best performance on almost every task.

6.1.3. USPS Digit Classification. The USPS digit dataset[7] contains 7291 examples each of 255 features. There are nine classification tasks, each corresponding to the classification of two digits. The experimental settings are the same as those for handwritten letter classification. The results are reported in Fig. 5(a) to 5(i). Similar to handwritten digit classification, L-MDL is worse than ITML and RDML on some tasks and TML is better than other methods on almost all tasks.

6.2. Experiments under Semi-Supervised Setting

In this section, we empirically study STML under the semi-supervised setting. We compare STML with TML to demonstrate how unlabeled data can improve the generalization performance. Similar to the experimental settings above, each task is treated as the target task and the other task as the source task. We select 80% of the data

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



(a) 1st task (b) 2nd task (c) 3rd task

(d) 4th task (e) 5th task (f) 6th task

(g) 7th task

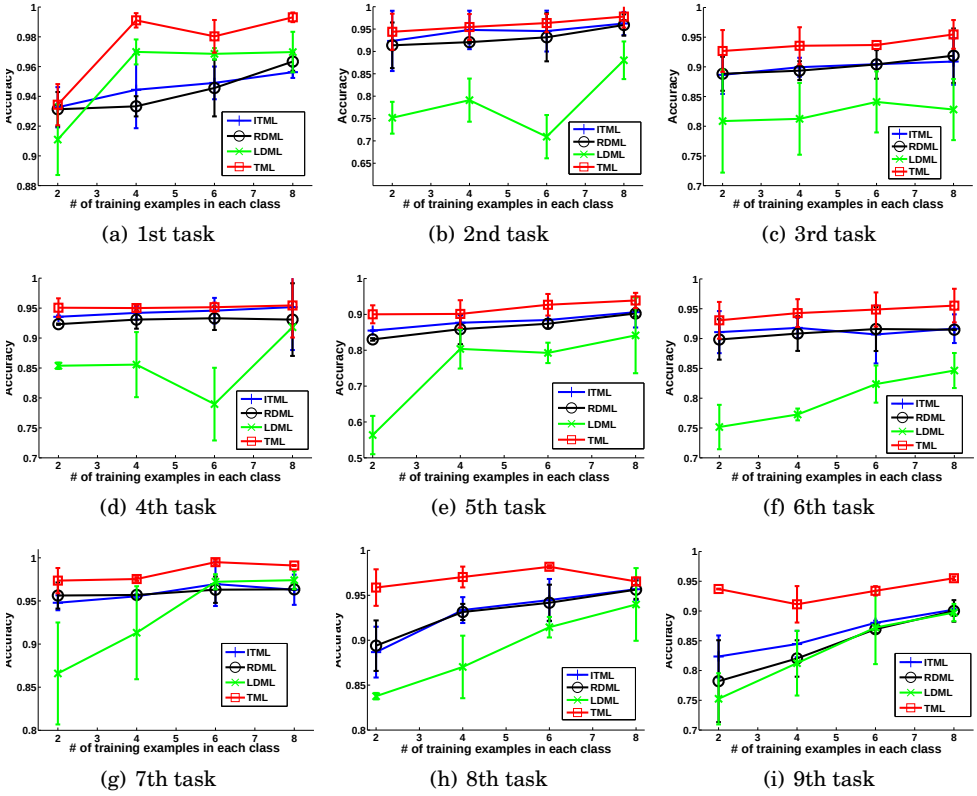
Fig. 4. Overall performance on handwritten letter classification application when one task is the target and the others are source tasks.

as training data which includes both labeled and unlabeled data and we vary the percentage of the labeled data used from 5% to 20% at intervals of 5% to see the effect of varying the size of the labeled data set. Each configuration is repeated 10 times and the results are in the form of the mean and standard deviation of the classification accuracy. The results on the unlabeled training data of the three data sets are recorded in Figs. 6, 7 and 8 and the results on the test data are recorded in Figs. 9, 10 and 11. From the results, we can see that the performance of STML is comparable to or even better than that of TML.

6.2.1. Sensitivity Analysis. We also perform sensitivity analysis with respect to α and β on the wine dataset. We use 20% of the data as labeled data and 60% of the data as unlabeled data in the training set. The results reported in Figs. 12 and 13 again show that our method is not very sensitive to the parameter values used.

7. CONCLUSION

In this paper, we have proposed a transfer metric learning method to alleviate the labeled data deficiency problem in the target learning task by exploiting useful information from some source tasks. The learning of the distance metrics from the source tasks and the relationships between the source tasks and the target task is formulated as a convex optimization problem which can be solved efficiently. We have also proposed an extension of TML to the semi-supervised setting by exploiting useful in-

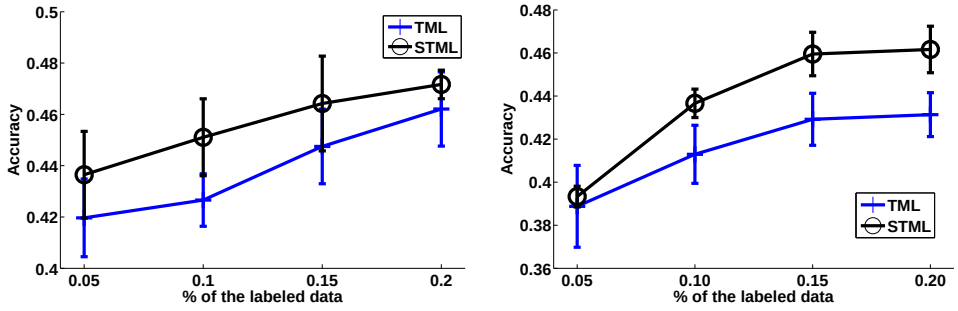


(a) 1st task (b) 2nd task (c) 3rd task

(d) 4th task (e) 5th task (f) 6th task

(g) 7th task (h) 8th task (i) 9th task

Fig. 5. Overall performance on USPS digit classification application.



(a) 1st task (b) 2nd task

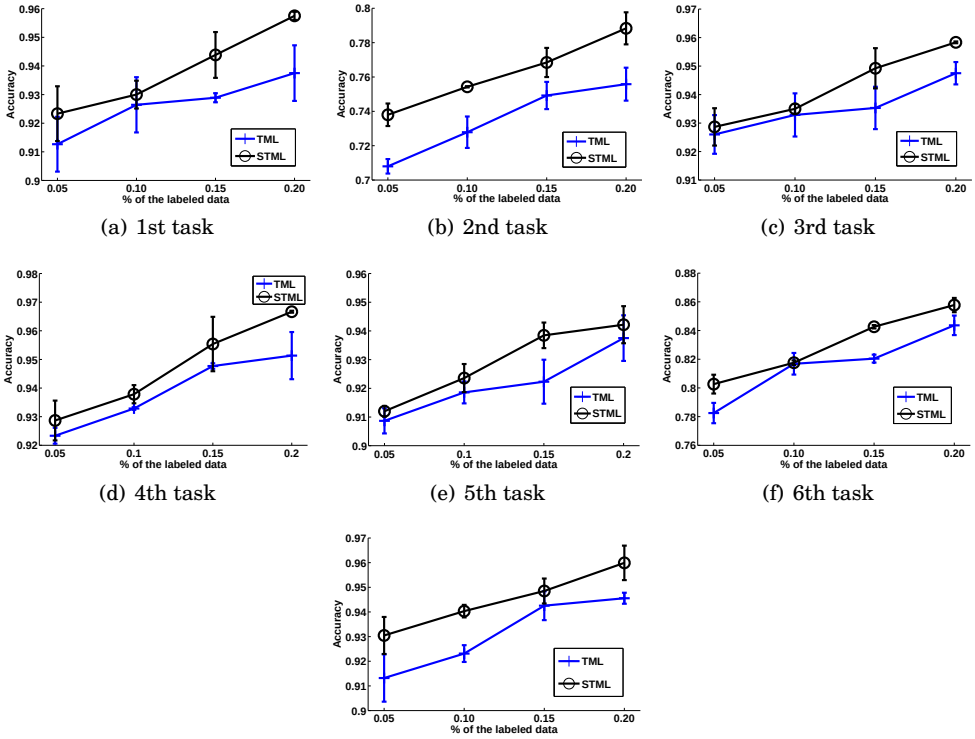
Fig. 6. Comparison of TML and STML on wine quality classification application (unlabeled training data).

formation contained in the unlabeled data. In our future research, we will apply our method to a wider range of supervised and semi-supervised learning applications.

Appendix: Optimization Procedure for Problem (3)

We present here the optimization procedure for solving problem (3). We use an alternating method with two subproblems to be presented separately below.

Optimizing w.r.t. Σ_i when Ω and $\{\Sigma\}_{-i}$ are fixed



(a) 1st task (b) 2nd task (c) 3rd task

(d) 4th task (e) 5th task (f) 6th task

(g) 7th task

Fig. 7. Comparison of TML and STML on handwritten letter classification application when one task is the target and the others are source tasks (unlabeled training data).

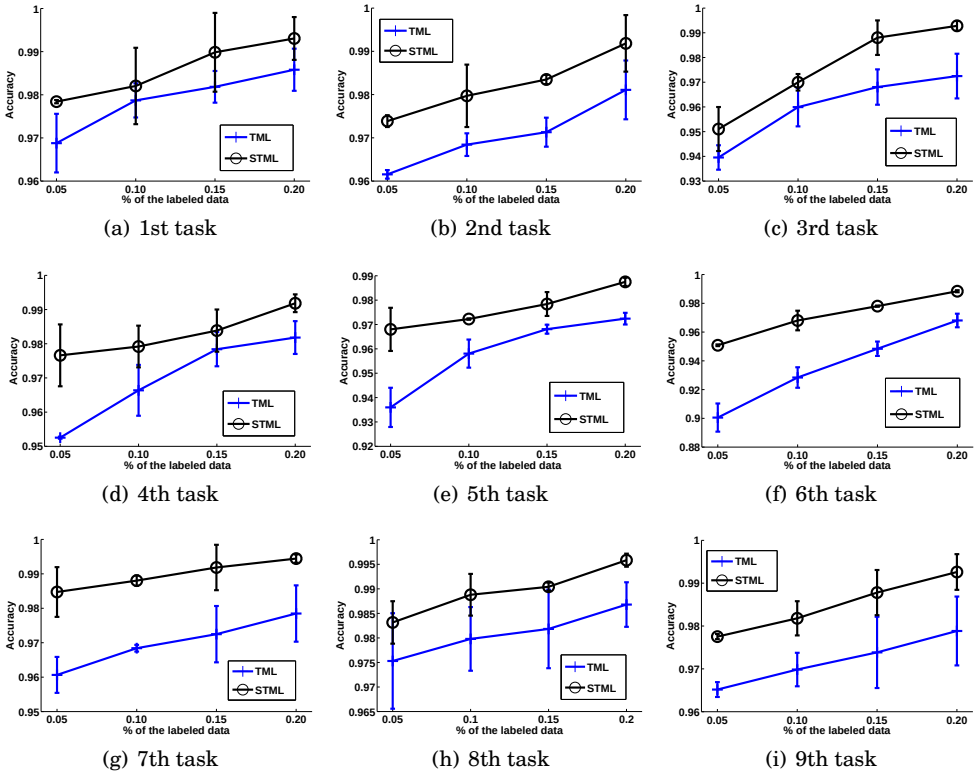
We first define $\tilde{\mathbf{\Sigma}}$ and $\mathbf{\Omega}^{-1}$ as

$$\tilde{\mathbf{\Sigma}} = \Big(\mathrm{vec}(\mathbf{\Sigma}_i), \tilde{\mathbf{\Sigma}}_{-i}\Big)$$
$$\mathbf{\Omega}^{-1} = \begin{pmatrix} \gamma_{ii} & \boldsymbol{\gamma}_i^T \\ \boldsymbol{\gamma}_i & \mathbf{\Gamma}_{-i} \end{pmatrix}.$$

Then the third term in the objective function of problem (3) can be rewritten as

$$\begin{aligned}
&\frac{\lambda_2}{2}\mathrm{tr}(\tilde{\mathbf{\Sigma}}\mathbf{\Omega}^{-1}\tilde{\mathbf{\Sigma}}^T) \\
=&\frac{\lambda_2}{2}\mathrm{tr}\Big((\mathrm{vec}(\mathbf{\Sigma}_i), \tilde{\mathbf{\Sigma}}_{-i}) \begin{pmatrix} \gamma_{ii} & \boldsymbol{\gamma}_i^T \\ \boldsymbol{\gamma}_i & \mathbf{\Gamma}_{-i} \end{pmatrix} \begin{pmatrix} \mathrm{vec}(\mathbf{\Sigma}_i)^T \\ \tilde{\mathbf{\Sigma}}_{-i}^T \end{pmatrix}\Big) \\
=&\frac{\lambda_2}{2}\Big(\gamma_{ii}\|\mathrm{vec}(\mathbf{\Sigma}_i)\|_2^2 + 2\boldsymbol{\gamma}_i^T\tilde{\mathbf{\Sigma}}_{-i}^T\mathrm{vec}(\mathbf{\Sigma}_i) + \mathrm{tr}(\tilde{\mathbf{\Sigma}}_{-i}\mathbf{\Gamma}_{-i}\tilde{\mathbf{\Sigma}}_{-i}^T)\Big) \\
=&\frac{\lambda_2}{2}\Big(\gamma_{ii}\|\mathbf{\Sigma}_i\|_F^2 + 2\mathrm{tr}(\mathbf{M}\mathbf{\Sigma}_i) + \mathrm{tr}(\tilde{\mathbf{\Sigma}}_{-i}\mathbf{\Gamma}_{-i}\tilde{\mathbf{\Sigma}}_{-i}^T)\Big),
\end{aligned}$$

where $\|\cdot\|_2$ denotes the 2-norm of a vector and $\mathbf{M}$ is a matrix such that $\mathrm{vec}(\mathbf{M}) = \tilde{\mathbf{\Sigma}}_{-i}\boldsymbol{\gamma}_i$. Note that the third term in the last equation above is independent of $\mathbf{\Sigma}_i$. It is easy to show that $\mathbf{M}$ is a symmetric matrix. The optimization problem with respect to $\mathbf{\Sigma}_i$

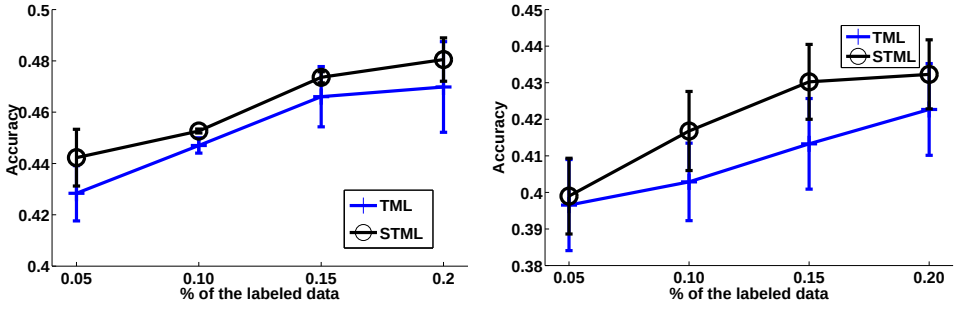


(a) 1st task (b) 2nd task (c) 3rd task

(d) 4th task (e) 5th task (f) 6th task

(g) 7th task (h) 8th task (i) 9th task

Fig. 8. Comparison of TML and STML on USPS digit classification application (unlabeled training data).



(a) 1st task (b) 2nd task

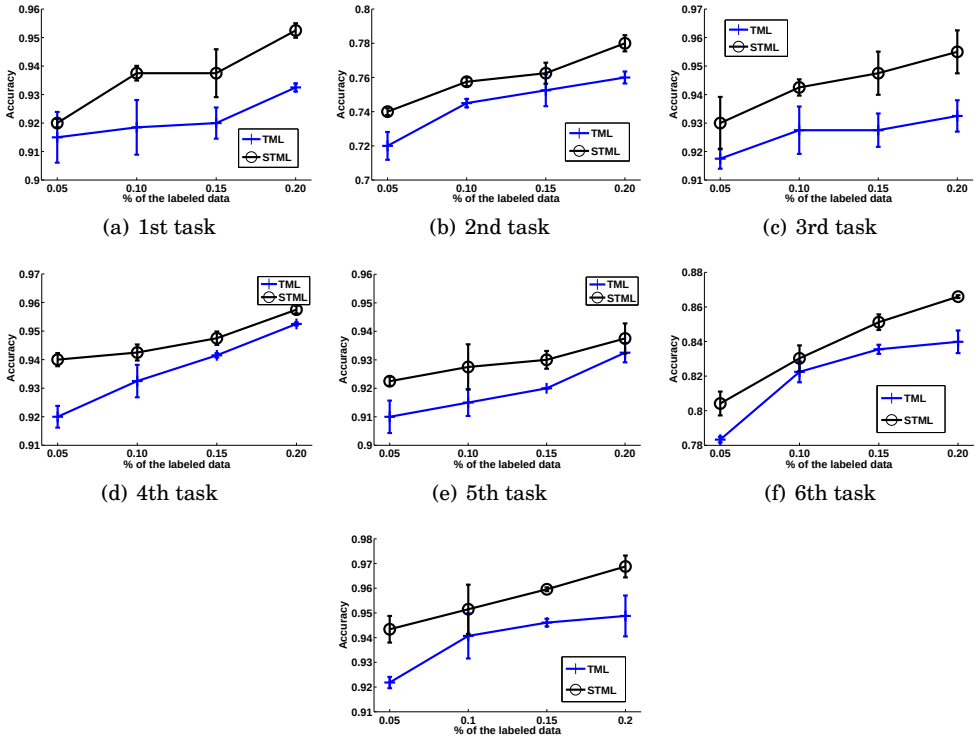
Fig. 9. Comparison of TML and STML on wine quality classification application (test data).

becomes

$$\min_{\mathbf{\Sigma}_i} \quad \frac{2}{n_i(n_i-1)} \sum_{j<k} g\Big(y^i_{j,k}\left[1-\|\mathbf{x}^i_j-\mathbf{x}^i_k\|^2_{\mathbf{\Sigma}_i}\right]\Big) + \frac{\lambda_1+\lambda_2\gamma_{ii}}{2}\|\mathbf{\Sigma}_i\|^2_F + \lambda_2 \mathbf{tr}(\mathbf{M}\mathbf{\Sigma}_i)$$

$$\text{s.t.} \quad \mathbf{\Sigma}_i \succeq \mathbf{0}. \tag{21}$$

It is easy to see that this problem is a convex semidefinite programming (SDP) problem since the objective function is convex with respect to $\mathbf{\Sigma}_i$ and the constraint is a PSD constraint on $\mathbf{\Sigma}_i$. Even though solving an SDP problem is computationally demanding



(a) 1st task (b) 2nd task (c) 3rd task

(d) 4th task (e) 5th task (f) 6th task

(g) 7th task

Fig. 10. Comparison of TML and STML on handwritten letter classification application when one task is the target and the others are source tasks (test data).

with poor scalability, we can adopt the technique in [Weinberger and Saul 2008] to use gradient projection to solve it. Moreover, we can also use the online algorithm in [Jin et al. 2009] to solve this problem.

Optimizing w.r.t. Ω when $\{\Sigma_i\}$ are fixed

When $\{\Sigma_i\}$ are fixed, the optimization problem for finding Ω becomes

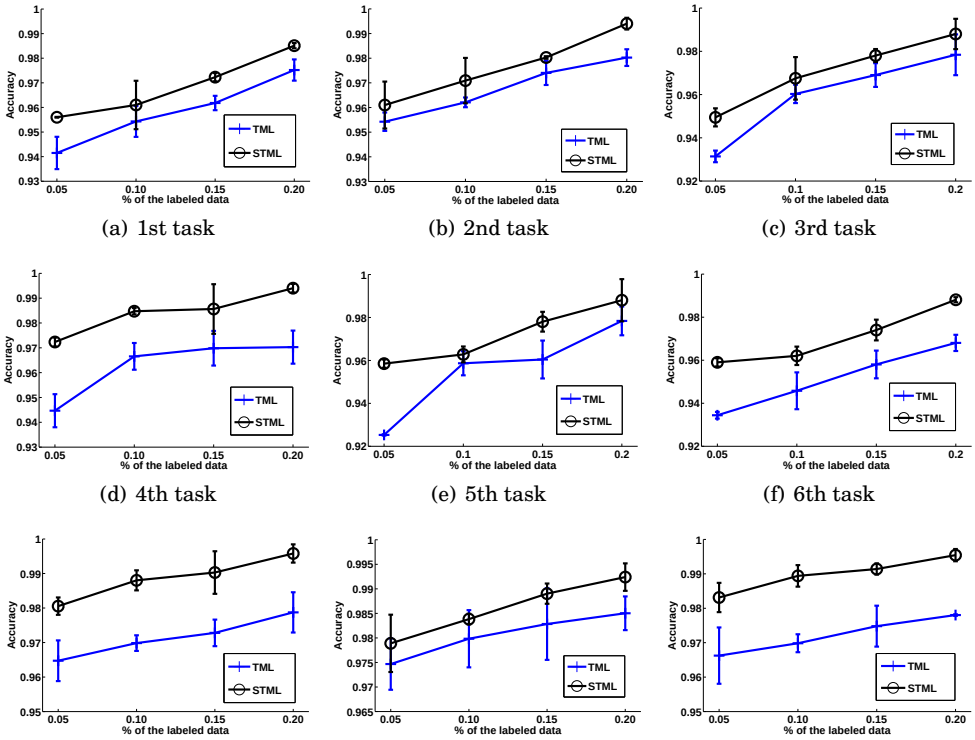
$$\begin{aligned} \min_{\mathbf{\Omega}} \quad & \mathrm{tr}(\mathbf{\Omega}^{-1}\tilde{\mathbf{\Sigma}}^T\tilde{\mathbf{\Sigma}}) \\ \text{s.t.} \quad & \mathbf{\Omega} \succeq 0 \\ & \mathrm{tr}(\mathbf{\Omega}) = 1. \end{aligned} \tag{22}$$

Then we have

$$\begin{aligned} \mathrm{tr}(\mathbf{\Omega}^{-1}\mathbf{A}) &= \mathrm{tr}(\mathbf{\Omega}^{-1}\mathbf{A})\mathrm{tr}(\mathbf{\Omega}) \\ &= \mathrm{tr}((\mathbf{\Omega}^{-\frac{1}{2}}\mathbf{A}^{\frac{1}{2}})(\mathbf{A}^{\frac{1}{2}}\mathbf{\Omega}^{-\frac{1}{2}}))\mathrm{tr}(\mathbf{\Omega}^{\frac{1}{2}}\mathbf{\Omega}^{\frac{1}{2}}) \\ &\geq (\mathrm{tr}(\mathbf{\Omega}^{-\frac{1}{2}}\mathbf{A}^{\frac{1}{2}}\mathbf{\Omega}^{\frac{1}{2}}))^2 = (\mathrm{tr}(\mathbf{A}^{\frac{1}{2}}))^2, \end{aligned}$$

where $\mathbf{A} = \tilde{\mathbf{\Sigma}}^T\tilde{\mathbf{\Sigma}}$. The first equality holds because of the last constraint in problem (22) and the last inequality holds because of the Cauchy-Schwarz inequality for the Frobenius norm. Moreover, $\mathrm{tr}(\mathbf{\Omega}^{-1}\mathbf{A})$ attains its minimum value $(\mathrm{tr}(\mathbf{A}^{\frac{1}{2}}))^2$ if and only if

$$\mathbf{\Omega}^{-\frac{1}{2}}\mathbf{A}^{\frac{1}{2}} = a\mathbf{\Omega}^{\frac{1}{2}}$$



(a) 1st task (b) 2nd task (c) 3rd task

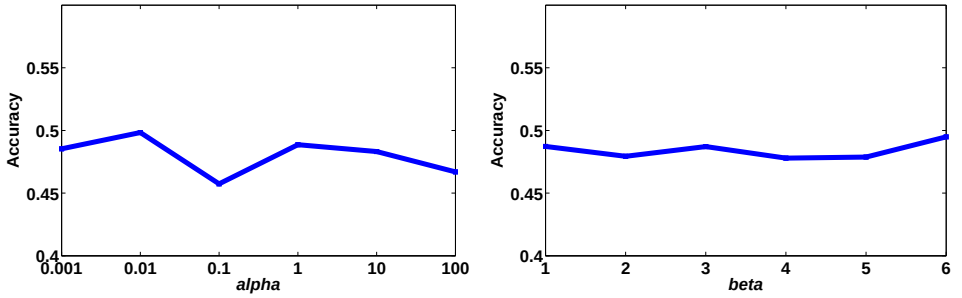
(d) 4th task (e) 5th task (f) 6th task

(g) 7th task (h) 8th task (i) 9th task

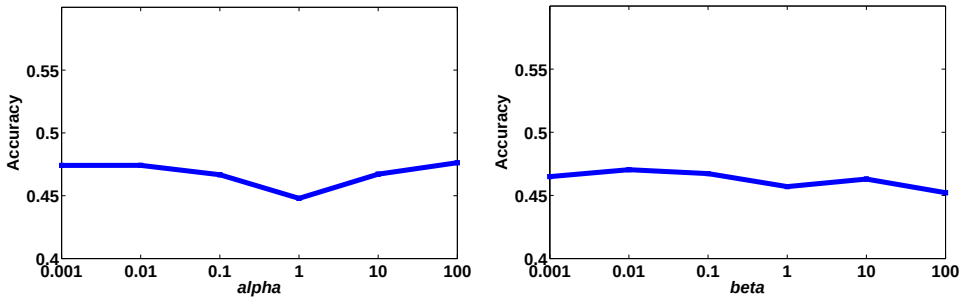
Fig. 11. Comparison of TML and STML on USPS digit classification application (test data).

for some constant a and $\mathrm{tr}(\mathbf{\Omega}) = 1$. So we can get the following analytical solution:

$$\mathbf{\Omega} = \frac{\left(\tilde{\mathbf{\Sigma}}^T \tilde{\mathbf{\Sigma}}\right)^{\frac{1}{2}}}{\mathrm{tr}\left(\left(\tilde{\mathbf{\Sigma}}^T \tilde{\mathbf{\Sigma}}\right)^{\frac{1}{2}}\right)}.$$



(a) 1st task (b) 1st task

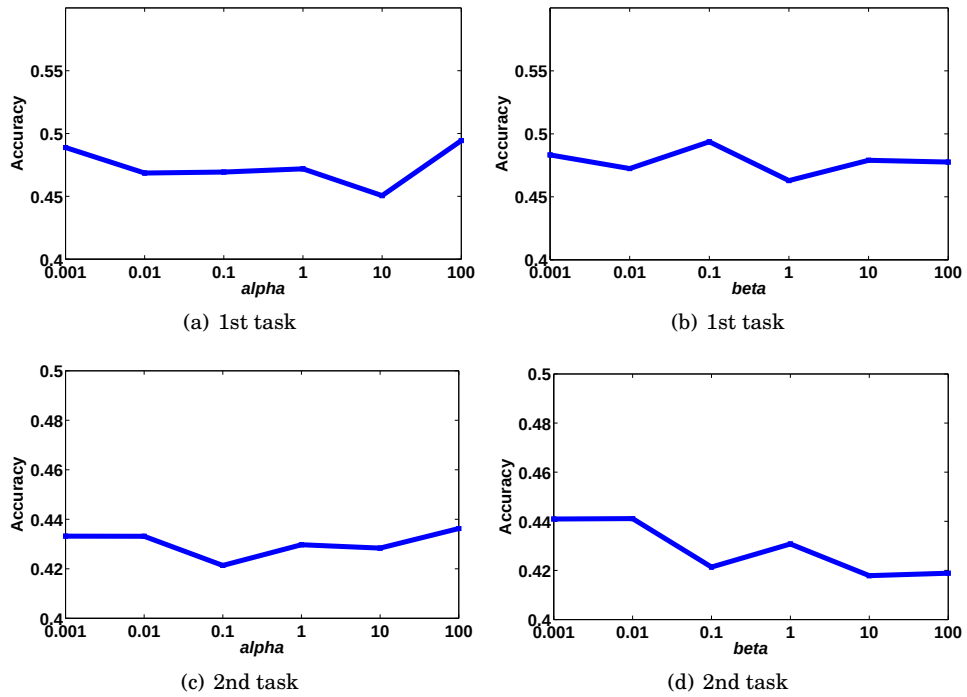


(c) 2nd task (d) 2nd task

Fig. 12. Effect of varying α and β in STML on the accuracy of the wine quality classification application (unlabeled training data).

Appendix: Detailed derivation for Equation (19)

$$
\begin{aligned}
& p(\mathbf{\Sigma}_m) \\
=& \int p(\mathbf{\Sigma}_m|\lambda_1,\lambda_2,\lambda_3)p(\lambda_1)p(\lambda_2)p(\lambda_3)d\lambda_1 d\lambda_2 d\lambda_3 \\
=& \int_0^\infty \int_0^\infty \Big(\frac{\lambda_1}{2\pi}\Big)^{\frac{d^2}{2}} \exp\Big\{-\frac{\lambda_1}{2}\|\mathbf{\Sigma}_m+\frac{\lambda_3}{\lambda_1}\mathbf{M}\|_F^2\Big\}p(\lambda_1)p(\lambda_3)d\lambda_1 d\lambda_3 \times \\
& \int_0^\infty \Big(\frac{\lambda_2}{2\pi}\Big)^{\frac{dm}{2}} \exp\Big\{-\frac{\lambda_2}{2}\mathbf{tr}(\tilde{\mathbf{\Sigma}}\mathbf{\Omega}^{-1}\tilde{\mathbf{\Sigma}}^T)\Big\}\frac{\lambda_2^{\alpha-1}\beta^\alpha \exp\{-\beta\lambda_2\}}{\Gamma(\alpha)}d\lambda_2 \\
=& \int_0^\infty \frac{\beta^\alpha}{(2\pi)^{\frac{d^2}{2}}\Gamma(\alpha)}\lambda_1^{\alpha+\frac{d^2}{2}-1}\exp\Big\{-(\beta+\frac{1}{2}\|\mathbf{\Sigma}_m\|_F^2)\lambda_1\Big\}d\lambda_1 \times \\
& \int_0^\infty \frac{\beta^\alpha}{(2\pi)^{\frac{dm}{2}}\Gamma(\alpha)}\lambda_2^{\alpha+\frac{dm}{2}-1}\exp\Big\{-(\beta+\frac{1}{2}\mathbf{tr}(\tilde{\mathbf{\Sigma}}\mathbf{\Omega}^{-1}\tilde{\mathbf{\Sigma}}^T))\lambda_2\Big\}d\lambda_2 \times \\
& \int_0^\infty \frac{\beta^\alpha}{\Gamma(\alpha)}\lambda_3^{\alpha-1}\exp\Big\{-(\beta+\mathbf{tr}(\mathbf{\Sigma}_m\mathbf{M}))\lambda_3\Big\}d\lambda_3 \times F(\lambda_1,\lambda_3) \\
=& \frac{\beta^\alpha\Gamma(\alpha+\frac{d^2}{2})}{(2\pi)^{\frac{d^2}{2}}\Gamma(\alpha)(\beta+\frac{1}{2}\|\mathbf{\Sigma}_m\|_F^2)^{\alpha+\frac{d^2}{2}}} \times \frac{\beta^\alpha\Gamma(\alpha+\frac{dm}{2})}{(2\pi)^{\frac{dm}{2}}\Gamma(\alpha)(\beta+\frac{1}{2}\mathbf{tr}(\tilde{\mathbf{\Sigma}}\mathbf{\Omega}^{-1}\tilde{\mathbf{\Sigma}}^T))^{\alpha+\frac{dm}{2}}} \times \\
& \frac{\beta^\alpha}{(\beta+\mathbf{tr}(\mathbf{\Sigma}_m\mathbf{M}))^\alpha} \times F(\lambda_1,\lambda_3)
\end{aligned}
$$



(a) 1st task (b) 1st task

(c) 2nd task (d) 2nd task

Fig. 13. Effect of varying α and β in STML on the accuracy of the wine quality classification application (test data).

ACKNOWLEDGMENTS

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

REFERENCES

ARGYRIOU, A., EVGENIOU, T., AND PONTIL, M. 2008. Convex multi-task feature learning. *Machine Learning 73,* 3, 243–272.

BELKIN, M., NIYOGI, P., AND SINDHWANI, V. 2006. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research* 7, 2399–2434.

BERTSEKAS, D. P. 1999. *Nonlinear Programming*. Athena Scientific.

BOYD, S. AND VANDENBERGHE, L. 2004. *Convex Optimization*. Cambridge University Press, New York, NY.

CARUANA, R. 1997. Multitask learning. *Machine Learning 28,* 1, 41–75.

CAWLEY, G. C., TALBOT, N. L. C., AND GIROLAMI, M. 2006. Sparse multinomial logistic regression via bayesian l1 regularisation. In *Advances in Neural Information Processing Systems 19*, B. Schölkopf, J. C. Platt, and T. Hoffman, Eds. Vancouver, British Columbia, Canada, 209–216.

CHANG, H. AND YEUNG, D.-Y. 2004. Locally linear metric adaptation for semi-supervised clustering. In *Proceedings of the Twenty-first International Conference on Machine Learning*. Banff, Alberta, Canada.

CHAPELLE, O., ZIEN, A., AND SCHÖLKOPF, B., Eds. 2006. *Semi-Supervised Learning*. MIT Press, Boston.

CHEN, J., ZHAO, Z., YE, J., AND LIU, H. 2007. Nonlinear adaptive distance metric learning for clustering. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. San Jose, California, USA, 123–132.

CHUNG, F. R. K. 1997. *Spectral Graph Theory*. American Mathematical Society, Rhode Island.

DAVIS, J. V. AND DHILLON, I. S. 2008. Structured metric learning for high dimensional problems. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Las Vegas, Nevada, USA, 195–203.

DAVIS, J. V., KULIS, B., JAIN, P., SRA, S., AND DHILLON, I. S. 2007. Information-theoretic metric learning. In *Proceedings of the Twenty-Fourth International Conference on Machine Learning*. Corvalis, Oregon, USA, 209–216.

DEMPSTER, A. P., LAIRD, N. M., AND RUBIN, D. B. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistic Society, B 39,* 1, 1–38.

EVGENIOU, T. AND PONTIL, M. 2004. Regularized multi-task learning. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Seattle, Washington, USA, 109–117.

FOO, C.-S., DO, C. B., AND NG, A. Y. 2009. A majorization-minimization algorithm for (multiple) hyperparameter learning. In *Proceedings of the 26th International Conference on Machine Learning*. Montreal, Quebec, Canada, 321–328.

GRANT, M. AND BOYD, S. 2009. CVX: Matlab software for disciplined convex programming (web page and software).

GUPTA, A. K. AND NAGAR, D. K. 2000. *Matrix Variate Distributions*. Chapman & Hall.

HASTIE, T. AND TIBSHIRANI, R. 1996. Discriminant adaptive nearest neighbor classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence 18,* 6, 607–616.

HOI, S. C. H., LIU, W., AND CHANG, S.-F. 2008. Semi-supervised distance metric learning for collaborative image retrieval. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Anchorage, Alaska, USA.

HOI, S. C. H., LIU, W., AND CHANG, S.-F. 2010. Semi-supervised distance metric learning for collaborative image retrieval and clustering. *ACM Transactions on Multimedia Computing, Communications, and Applications 6,* 3.

JIN, R., WANG, S., AND ZHOU, Y. 2009. Regularized distance metric learning: Theory and algorithm. In *Advances in Neural Information Processing Systems 22*, Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, Eds. Vancouver, British Columbia, Canada, 862–870.

KIENZLE, W. AND CHELLAPILLA, K. 2006. Personalized handwriting recognition via biased regularization. In *Proceedings of the Twenty-Third International Conference on Machine Learning*. 457–464.

LANGE, K., HUNTER, D. R., AND YANG, I. 2000. Optimization transfer using surrogate objective functions. *Journal of Computational and Graphical Statistics 9,* 1, 1–59.

LOBO, M. S., VANDENBERGHE, L., BOYD, S., AND LEBRET, H. 1998. Applications of second-order cone programming. *Linear Algebra and its Applications 284*, 193–228.

PAN, S. AND YANG, Q. 2010. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering 22,* 10, 1345–1359.

PAN, S. J., KWOK, J. T., AND YANG, Q. 2008. Transfer learning via dimensionality reduction. In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence*. Chicago, Illinois, USA, 677–682.

PAN, S. J., TSANG, I. W., KWOK, J. T., AND YANG, Q. 2009. Domain adaptation via transfer component analysis. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence*. Pasadena, California, USA, 1187–1192.

ROSENSTEIN, M. T., MARX, Z., AND KAELBLING, L. P. 2005. To transfer or not to transfer. In *NIPS-05 Workshop on Inductive Transfer: 10 Years Later*.

THRUN, S. 1996. Is learning the n-th thing any easier than learning the first? In *Advances in Neural Information Processing Systems 8*, D. S. Touretzky, M. Mozer, and M. E. Hasselmo, Eds. Denver, CO, 640–646.

WANG, Z., SONG, Y., AND ZHANG, C. 2008. Transferred dimensionality reduction. In *Proceedings of European Conference on Machine Learning and Knowledge Discovery in Databases*. Antwerp, Belgium, 550–565.

WEINBERGER, K. Q., BLITZER, J., AND SAUL, L. K. 2005. Distance metric learning for large margin nearest neighbor classification. In *Advances in Neural Information Processing Systems 18*, Y. Weiss, B. Schölkopf, and J. Platt, Eds. Vancouver, British Columbia, Canada, 1473–1480.

WEINBERGER, K. Q. AND SAUL, L. K. 2008. Fast solvers and efficient implementations for distance metric learning. In *Proceedings of the Twenty-Fifth International Conference on Machine Learning*. Helsinki, Finland, 1160–1167.

WILLIAMS, P. M. 1995. Bayesian regularization and pruning using a laplace prior. *Neural Computation 7,* 1, 117–143.

XIAO, B., YANG, X., XU, Y., AND ZHA, H. 2009. Learning distance metric for regression by semidefinite programming with application to human age estimation. In *Proceedings of the 17th ACM International Conference on Multimedia*. 451–460.

XING, E. P., NG, A. Y., JORDAN, M. I., AND RUSSELL, S. J. 2002. Distance metric learning with application to clustering with side-information. In *Advances in Neural Information Processing Systems 15*, S. Becker, S. Thrun, and K. Obermayer, Eds. Vancouver, British Columbia, Canada, 505–512.

YEUNG, D.-Y. AND CHANG, H. 2007. A kernel approach for semisupervised metric learning. *IEEE Transactions on Neural Networks 18,* 1, 141–149.

YEUNG, D.-Y., CHANG, H., AND DAI, G. 2008. A scalable kernel-based semi-supervised metric learning algorithm with out-of-sample generalization ability. *Neural Computation 20,* 11, 2839–2861.

ZELNIK-MANOR, L. AND PERONA, P. 2004. Self-tuning spectral clustering. In *Advances in Neural Information Processing Systems 17*. Vancouver, British Columbia, Canada, 1601–1608.

ZHA, Z.-J., MEI, T., WANG, M., WANG, Z., AND HUA, X.-S. 2009. Robust distance metric learning with auxiliary knowledge. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence*. Pasadena, California, USA, 1327–1332.

ZHAN, D.-C., LI, M., LI, Y.-F., AND ZHOU, Z.-H. 2009. Learning instance specific distances using metric propagation. In *Proceedings of the 26th International Conference on Machine Learning*. Montreal, Quebec, Canada, 1225–1232.

ZHANG, Y. AND YEUNG, D.-Y. 2010a. A convex formulation for learning task relationships in multi-task learning. In *Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence*. Catalina Island, California, 733–742.

ZHANG, Y. AND YEUNG, D.-Y. 2010b. Transfer metric learning by learning task relationships. In *Proceedings of the 16th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. Washington, DC, USA, 1199–1208.

Received October 2010; revised February 2011; accepted April 2011