Decomposition-Based Transfer Distance Metric Learning for Image Classification

Yong Luo, Tongliang Liu, Dacheng Tao, *Senior Member, IEEE*, and Chao Xu, *Member, IEEE*

***Abstract*— Distance metric learning (DML) is a critical factor for image analysis and pattern recognition. To learn a robust distance metric for a target task, we need abundant side information (i.e., the similarity/dissimilarity pairwise constraints over the labeled data), which is usually unavailable in practice due to the high labeling cost. This paper considers the transfer learning setting by exploiting the large quantity of side information from certain related, but different source tasks to help with target metric learning (with only a little side information). The state-of-the-art metric learning algorithms usually fail in this setting because the data distributions of the source task and target task are often quite different. We address this problem by assuming that the target distance metric lies in the space spanned by the eigenvectors of the source metrics (or other randomly generated bases). The target metric is represented as a combination of the base metrics, which are computed using the decomposed components of the source metrics (or simply a set of random bases); we call the proposed method, decomposition-based transfer DML (DTDML). In particular, DTDML learns a sparse combination of the base metrics to construct the target metric by forcing the target metric to be close to an integration of the source metrics. The main advantage of the proposed method compared with existing transfer metric learning approaches is that we directly learn the base metric coefficients instead of the target metric. To this end, far fewer variables need to be learned. We therefore obtain more reliable solutions given the limited side information and the optimization tends to be faster. Experiments on the popular handwritten image (digit, letter) classification and challenge natural image annotation tasks demonstrate the effectiveness of the proposed method.**

***Index Terms*— Distance metric learning, transfer learning, decomposition, base metric, image classification.**

I. Introduction

THE performance of computer vision, data mining and multimedia systems is heavily dependent on the distance metric between samples. For example, the simple k-nearest neighbor (kNN) classifier that uses a proper distance metric can be very competitive, and is sometimes superior to other well designed classifiers in many applications such as face recognition, image annotation, etc. In [1], the authors learn a distance metric for nearest neighbor classification so that the nearest neighbors tend to belong to the same class and the samples from different classes are separated by a large margin. The kNN classifier based on the learned metric was shown to be comparable to the state-of-the-art multiclass support vector machine (SVM) in several applications including face recognition and text categorization. A weighted nearest neighbor model was proposed in [2] for image annotation that learned a discriminative distance metric. This model was demonstrated empirically to significantly out-perform the state-of-the-art annotation methods on three challenge datasets. Actually, distance metric learning (DML) is also critical to many other popular algorithms, e.g., k-means clustering and kernel machines such as SVM.

It is therefore essential to learn a robust distance metric to reveal the data relationships. To achieve this goal, we need a large amount of side information [3] such as the constraints that indicate whether a pair of samples is similar or not. Real-world applications, e.g. image annotation [4]–[8], usually have few training samples in the instance space of the target learning task due to the high labeling cost. However, we can easily obtain a large number of labeled samples from the instance spaces of different, but related learning tasks, or from the same instance space with different distribution. Therefore, we can leverage the samples from the related tasks for the target task learning. This is known as transfer learning, and the related tasks are usually called source tasks. This article focuses on utilizing the large quantity of side information in the source tasks to discover a reliable distance metric for the target task.

A number of existing metric learning algorithms [1], [3], [9]–[12] can be utilized to learn a useful distance metric for the source task with adequate training data. The training criterion is usually to minimize the distance between two samples if they are from the same class, and otherwise maximize their distance. However, directly applying the learned source metric to the target task may not result in good performance because it may be biased to the sample distribution of the source task, while the data distributions between the source task and target task maybe quite different. More sophisticated methods should therefore be developed to tackle the metric learning problem in the transfer scenario.

This paper proposes a decomposition-based method for transfer distance metric learning (DTDML) by assuming that

Manuscript received October 12, 2013; revised March 21, 2014; accepted June 16, 2014. Date of publication June 23, 2014; date of current version July 22, 2014. This work was supported in part by the National Basic Research Program of China under Grant 2011CB302400, in part by the National Natural Science Foundation of China under Grant 61375026, Grant 61121002, and Grant JCYJ20120614152136201, and in part by the Australian Research Council Project under Grant DP-140102164 and Grant FT-130101457. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Nilanjan Ray.

Y. Luo and C. Xu are with the Key Laboratory of Machine Perception, Ministry of Education, School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China (e-mail: yluo180@gmail.com; xuchao@cis.pku.edu.cn).

T. Liu and D. Tao are with the Centre for Quantum Computation and Intelligent Systems, Faculty of Engineering and Information Technology, University of Technology, Sydney, Ultimo, NSW 2007, Australia (e-mail: tliang.liu@gmail.com; dacheng.tao@uts.edu.au).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2014.2332398

the target metric (distance metric of the target task) lies in the space spanned by the eigenvectors of the source metrics, or other randomly generated bases. The target metric is represented as a combination of "base metrics" that are derived from the decomposition of the source metrics, or simply computed using the random bases. In particular, DTDML learns a sparse combination of the "base metrics" to construct the target metric by forcing the target metric to be close to an integration of the source metrics. The optimization is performed by alternating between the calculation of the "base metric" coefficients and source metric integration weights, and both of the two sub-problems can be solved efficiently.

Recent research on transfer metric learning includes the following. In [13], the target metric is learned by minimizing the log-determinant divergence between the source metrics and target metric. Zhang and Yeung [14] proposed to learn the task relationships in transfer metric learning, and therefore, allow modeling of negative and zero transfer. These two methods are also optimized using the alternating strategy. However, in each iteration of their alternating procedures, they both rely on direct estimation of the target metric and have a large number of d^2 variables to be learned. Here, d is the feature dimensionality, which is usually very high for an image, while in DTDML, the number of variables is only md if we use the eigenvectors of m source metrics to construct the base metrics, and it is common for $m \ll d$. Therefore, we can obtain more reliable solutions, given the limited side information in the target task, and the optimization tends to be faster because we have far fewer variables to be estimated. We adopt Nesterov's optimal method [15] for optimization, so do not require costly semi-definite programming in the learning of the target metric, and have a rapid convergence rate. We performed extensive experiments on two popular handwritten image datasets and the challenge NUS-WIDE [16] web image dataset. The results confirmed the effectiveness and efficiency of DTDML.

The article is organized as follows. We summarize closely related works in Section II. Section III includes the description, formulation, and some theoretical analysis of the proposed DTDML. Extensive experiments are presented in Section IV and we conclude this paper in Section V.

II. Related Work

A. Distance Metric Learning

The goal of distance metric learning (DML) [17] is to learn an appropriate distance function for a given problem. DML is very important for many learning models, e.g., the kNN rule and SVMs. A popular categorization of the DML methods is: supervised DML [1], [3], [9], [10], [12], [18] and unsupervised DML [19], [20], according to the underlying learning paradigm. There are also some semi-supervised works that combine these two paradigms [21], [22]. Our research is built on supervised metric learning, so we only review some representative works in this category.

A classical algorithm for supervised DML was presented in [3], where the authors proposed a constrained convex optimization problem for the metric learning. Relevant component analysis (RCA) [23] utilizes the so-called chunklets to learn a metric by reducing the weights of irrelevant dimensions and amplifying the weights of the relevant dimensions. In [18], the relative comparison constraints that can be easily obtained using the query feedbacks were introduced for DML. The formulation is a quadratic programming problem, which was solved by adapting the standard SVM solver. Neighborhood component analysis (NCA) [9] learns a metric that directly maximizes the nearest neighbor (NN) classification performance. This is achieved by optimizing the leave-one-out classification error on the training set with stochastic neighborhood selection. Large margin nearest neighbor (LMNN) [1] is also based on NN classification, but using a large margin strategy. From the perspective of information theoretic, Davis et al. [10] proposed to learn a Mahanobis matrix that is close to a given prior distance metric in the sense of differential relative entropy, and simultaneously satisfies the distance constraints. In [12], an efficient online algorithm was presented for regularized DML, in which it was proved that the generalization error can be independent from the feature dimensionality if appropriate constraints are utilized.

B. Transfer Learning

Transfer learning [24] aims to utilize the knowledge obtained from source domains to help the target domain learning, because the training samples in the target domain are insufficient to train a robust model. Dozens of transfer learning algorithms have been proposed in the literature and can be roughly grouped into homogeneous and heterogeneous transfers. The former refers to samples in target and source domains that are drawn from the same instance space but different distributions [25]–[28], and the latter refers to samples in target and source domains that are drawn from different, but related instance spaces [29]–[31]. This research considers the homogeneous setting, and omits are view of the heterogeneous works.

According to [32], transfer learning can be grouped into instance transfer [26], [33], feature representation transfer [27], [34], parameter transfer [25] and relational knowledge transfer [35], based on "what to transfer". A kernel mean matching (KMM) method was presented in [26] to match the data distribution of the target domain using the source domain samples. TrAdaboost [33] extends AdaBoost to leverage the abundant source data for the target task learning by iteratively filtering out "bad" source data. Argyriou et al. [27] presented a sparse representation based learning algorithm that learns (or selects) some common features shared across related tasks by using a L_1-norm regularizer. In [34], an unsupervised approach called self-taught learning was proposed to learn features for transfer from unlabeled data. Evgeniou and Pontil [25] learned the parameters of the source and target task simultaneously by assuming the parameter for each task can be separated into two terms, one of which is shared between the source and target task. In [35], the relational knowledge represented with Markov logic networks (MLNs) was transferred from the source domain to the target domain by first constructing a predicate mapping, and then refining the mapped structure in the target domain. There are lots of other

works on homogeneous and heterogeneous transfer learning, and we refer to [32] for a more comprehensive survey.

Despite the proposal of many transfer learning algorithms, to the best of our knowledge, only a few works [13], [14] consider homogeneous distance metric transfer. Zha et al. [13] developed two algorithms for learning a distance metric from a small number of training samples by transferring the prior knowledge from auxiliary data and using a large number of unlabeled samples. Zhang and Yeung [14] proposed a convex formulation for transferring the metric by encoding task relationships in a task covariance matrix. This matrix models positive, negative and zero task correlations. Both algorithms perform well on some applications, but the proposed DTDML will outperform them due to the reasons discussed above in Section I. Before presenting the proposed DTDML, we first present certain notations that are used throughout this paper.

Notations: Let $\mathcal{D} = \{(x_i, x_j, y_{ij})\}_{i,j=1}^{N}$ denotes the training set for the target task, wherein $x_i, x_j \in \mathbb{R}^d$ are vectors of d dimension and $y_{ij} = \pm 1$ indicates x_i and x_j are similar/dissimilar to each other. The number of target training samples N is very small, so we are also given m relevant source training sets, $\mathcal{D}_p = \{(x_{pi}, x_{pj}, y_{pij})\}_{i,j=1}^{N_p}$, $p = 1, \ldots, m$, each contains a large amount of training data. In the homogeneous transfer setting, $x_{pi}, x_{pj} \in \mathbb{R}^d$ belong to the same feature space as x_i, x_j.

III. Decomposition Based Transfer Distance Metric Learning

Similar to [14], our method is also built on the regularized DML (RDML) [12] and we introduce it here. In DML, we intend to learn a distance function $dst(x_i, x_j|A)$ parameterized by a distance metric A so that the similarity/dissimilarity between a new instance pair x_i and x_j is reflected by comparing $dst(x_i, x_j|A)$ with a constant threshold c. In particular, the regularized distance metric learning (RDML) needs to learn a metric A by the use of the following optimization problem:

$$\begin{aligned} &\underset{A}{\operatorname{argmin}} \ \tfrac{2}{N(N-1)} \textstyle\sum_{i<j} g\left(y_{ij}[1 - \|x_i - x_j\|_A^2]\right) + \tfrac{\eta}{2}\|A\|_F^2, \\ &\text{s.t.} \quad A \succcurlyeq 0. \end{aligned} \tag{1}$$

where $\|x_i - x_j\|_A^2 = (x_i - x_j)^T A (x_i - x_j)$ is the distance between two samples x_i and x_j, and $g(z) = \max(0, b - z)$ is the hinge loss, where b is set to zero in [12]; The Frobenius norm of the metric A, i.e., $\|A\|_F$ is a regularizer that is used to control the model complexity, and η is a trade-off parameter. The constraint means that A is positive semi-definite.

An online method was presented in [12] to solve problem (1). However, when training data are limited, RDML performs poorly. Our decomposition based transfer distance metric learning (DTDML) method improves RDML by using training data from certain relevant source domains. As we know that, any metric A can be decomposed as $A = U\Lambda U^T = \sum_{i=1}^{d} \lambda_i u_i u_i^T$. This indicates that the optimal target metric can be represented as a linear combination of at most d target "base metrics" $B_i = u_i u_i^T$. However, the target base metrics are not available. We thus propose to approximate the target base metric by combining some base metrics derived from the source metrics. This approximation is reasonable since the source tasks are related to the target task. Actually, we can also approximate the target base metric using some randomly generated bases and the effectiveness will be demonstrated empirically in our experiments. The proposed target metric learning strategy is advantageous compared to the traditional transfer metric learning algorithm, since we have fewer variables to be learned and thus can obtain more reliable solutions.

The diagram of the proposed DTDML is shown in Fig. 1. Given m source domains with adequate labeled training data for each, we learn their corresponding metrics $A_p \in \mathbb{R}^{d\times d}$, $p = 1, \ldots, m$ independently. These source metrics are weighted and integrated as $A_S = \sum_{p=1}^{m} \alpha_p A_p$, which is used for the target metric estimation later. At the same time, we apply singular value decomposition (SVD) to the obtained source metrics $A_1, \ldots, A_m$ and obtain a set of source eigenvectors $U = [u_1, \ldots, u_n]$, with each $u_r \in \mathbb{R}^d, r = 1, \ldots, n$. Alternatively, U can be a set of randomly generated base vectors. We represent the target metric as $A = U\mathrm{diag}(\theta)U^T = \sum_{r=1}^{n=m\times d} \theta_r u_r u_r^T$, which is actually a combination of base metrics $B_r = u_r u_r^T$. Finally, we learn the source metric integration weights α and the base metric combination coefficients θ simultaneously by minimizing the divergence between A_S and A, as well as leveraging the limited labeled training samples in the target domain. The result metric is given by $A = U\mathrm{diag}(\theta^*)U^T$, where θ^* is the learned coefficients. The technical details are given below.

A. Problem Formulation

The general formulation of the proposed DTDML for learning the target metric matrix A is given by

$$\begin{aligned} \underset{\alpha,\theta}{\operatorname{argmin}} \ & \tfrac{2}{N(N-1)} \textstyle\sum_{i<j} V(x_i, x_j, y_{ij}) + \tfrac{\gamma_A}{2}\|A - A_S\|_F^2 \\ & + \frac{\gamma_B}{2}\|\alpha\|_2^2 + \gamma_C\|\theta\|_1, \\ \text{s.t.} \quad & \textstyle\sum_{p=1}^{m} \alpha_p = 1, \alpha_p \geq 0, p = 1, \ldots, m. \end{aligned} \tag{2}$$

where $A = \sum_{r=1}^{n} \theta_r u_r u_r^T$, and the integrated metric $A_S = \sum_{p=1}^{m} \alpha_p A_p$. The term $\|A - A_S\|_F^2$ is a measure of the difference between A and A_S, which are expected to be close. Both $\|\alpha\|_2^2$ and $\|\theta\|_1$ are used to control the model complexity. As depicted above, at most d optimal base metrics are needed to construct the optimal target metric. In practice, most base metric combination coefficients λ_i are small and approximate to zero. Therefore, many input base metrics of the proposed model are redundant or noisy. We thus constraint the base metric coefficients θ to be sparse in order to suppress noisy [36]; γ_A, γ_B and γ_C are positive trade-off parameters.

Following [12], we choose $V(x_i, x_j, y_{ij}) = g(y_{ij}[1 - \|x_i - x_j\|_A^2])$ and adopt the hinge loss [37] for g, i.e., $g(z) = \max(0, b - z)$. Here, b is set to be zero. Then we find the following optimization problem:

$$\begin{aligned} \underset{\alpha,\theta}{\operatorname{argmin}} \ & \tfrac{2}{N(N-1)} \textstyle\sum_{i<j} g\left(y_{ij}[1 - \|x_i - x_j\|_A^2]\right) \\ & + \frac{\gamma_A}{2}\|A - A_S\|_F^2 + \frac{\gamma_B}{2}\|\alpha\|_2^2 + \gamma_C\|\theta\|_1, \\ \text{s.t.} \quad & \sum_{p=1}^{m} \alpha_p = 1, \alpha_p \geq 0, p = 1, \ldots, m. \end{aligned} \tag{3}$$

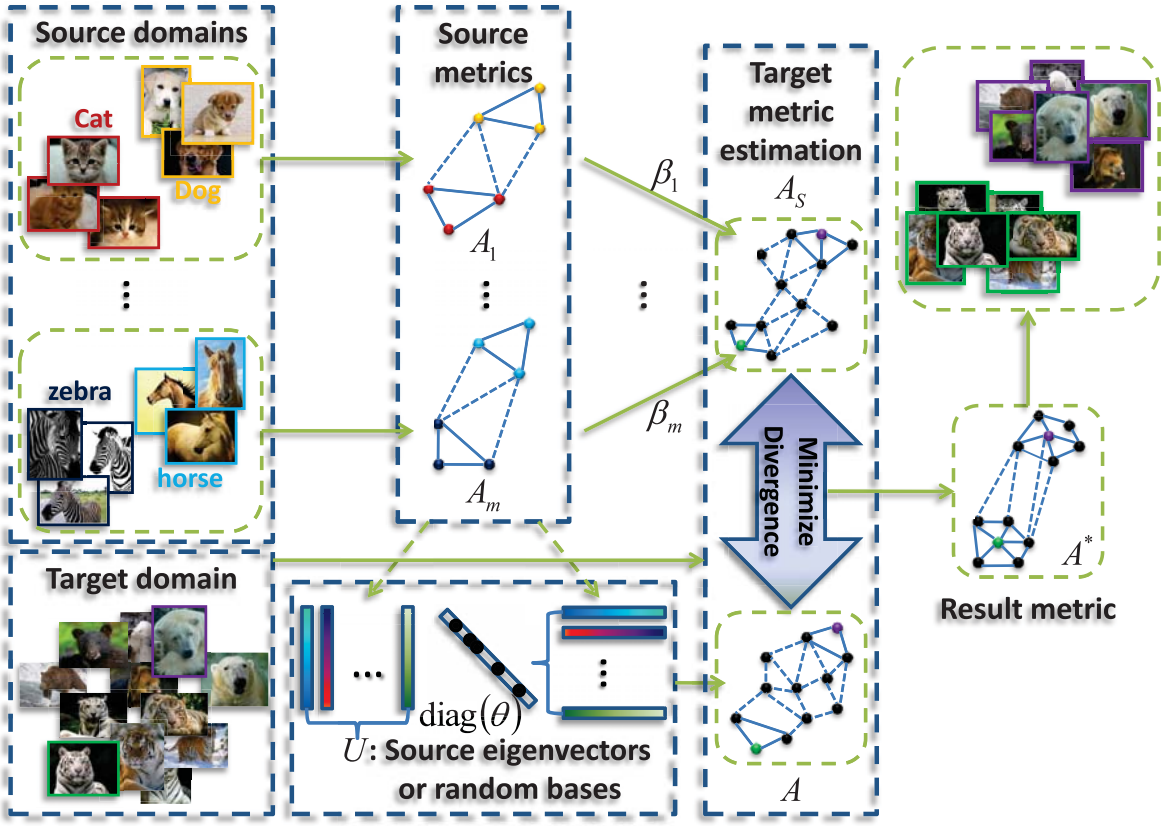

Fig. 1. Diagram of the proposed DTDML algorithm. Given source domains and a large number of labeled samples for each of them, we learn the corresponding source metrics independently. These metrics are combined as A_S for target metric estimation. At the same time, the source metrics are decomposed into a set of eigenvectors. Alternatively, we can randomly generate a set of bases. The target metric $A = U\text{diag}(\theta)U^T$ is actually a combination of certain "base metrics" derived from the source eigenvectors or random bases. By minimizing the divergence between A_S and A, we learn the combination coefficients and the source metric integration weights simultaneously, and finally, obtain the result metric.

For notation simplicity, we denote x_i, x_j and y_{ij} as x_k^1, x_k^2 and y_k respectively, where $k = 1, \ldots, N' = \frac{N(N-1)}{2}$. We also set $\delta_k = x_k^1 - x_k^2$ so that $\|x_k^1 - x_k^2\|_A^2 = \sum_{r=1}^n \theta_r \delta_k^T u_r u_r^T \delta_k = \theta^T h_k$ where $h_k = [h_k^1, \ldots, h_k^n]^T$ with each $h_k^r = \delta_k^T u_r u_r^T \delta_k$. Then, the problem (3) becomes

$$\begin{aligned} \underset{\alpha,\theta}{\text{argmin}} \ & \tfrac{1}{N'} \textstyle\sum_{k=1}^{N'} g\left(y_k(1-\theta^T h_k)\right) + \tfrac{\gamma_A}{2}\|A - A_S\|_F^2 \\ & + \frac{\gamma_B}{2}\|\alpha\|_2^2 + \gamma_C\|\theta\|_1, \\ \text{s.t.} \quad & \textstyle\sum_{p=1}^m \alpha_p = 1, \alpha_p \geq 0, p = 1, \ldots, m. \end{aligned} \tag{4}$$

The solution can be obtained by alternating between two sub-problems (which correspond to the minimization w.r.t. $\alpha = [\alpha_1, \ldots, \alpha_m]^T$ and $\theta = [\theta_1, \ldots, \theta_n]^T$ respectively) until convergence.

B. Optimization Procedure

For fixed α, the optimization problem with respect to θ is formulated as

$$\underset{\theta}{\text{argmin}}\, F(\theta) = \Phi(\theta) + \Omega(\theta), \tag{5}$$

where $\Phi(\theta) = \frac{1}{N'}\sum_{k=1}^{N'} g(y_k(1-\theta^T h_k)) + \gamma_C\|\theta\|_1$, and $\Omega(\theta) = \frac{\gamma_A}{2}\|A - A_S\|_F^2$. The loss function $\Phi(\theta)$ is non-differentiable. Hence, we firstly smooth the loss and then use Nesterov's optimal gradient method [15] to solve (5). According to [15], the smoothed version of the hinge loss $g(h_k, y_k, \theta) = \max\{0, -y_k(1-\theta^T h_k)\}$ can be given by

$$g_\sigma = \max_{v \in \mathcal{Q}} v_k\big(-y_k(1-\theta^T h_k)\big) - \frac{\sigma}{2}\|h_k\|_\infty v_k^2, \tag{6}$$

where $\mathcal{Q} = \{v : 0 \leq v_k \leq 1, v \in \mathbb{R}^{N'}\}$ and σ is the smooth parameter. A larger σ induces a more smooth approximation with larger approximation error. On the other hand, a small σ induces a slow convergence rate, and thus leads to high time complexity. Therefore, the parameter σ should neither be too large nor too small, and we empirically set it as 5 in our implementation. The $\|h_k\|_\infty$ used here is served as a normalization, so that the appropriate value for parameter σ does not change too much for different h_k. We refer to [38] for a comprehensive study of the smoothed hinge loss. By setting the objective function of (6) to become zero and then projecting v_k on $\mathcal{Q}$, we obtain the following solution,

$$v_k = \text{median}\left\{\frac{-y_k(1-\theta^T h_k)}{\sigma\|h_k\|_\infty}, 0, 1\right\}. \tag{7}$$

By substituting the solution (7) back into (6), we have the piece-wise approximation of g, i.e.,

$$g_\sigma = \begin{cases} 0, & y_k(1-\theta^T h_k) > 0; \\ -y_k(1-\theta^T h_k) - \frac{\sigma}{2}\|h_k\|_\infty, & y_k(1-\theta^T h_k) < -\sigma\|h_k\|_\infty; \\ \frac{(y_k(1-\theta^T h_k))^2}{2\sigma\|h_k\|_\infty}, & \text{otherwise.} \end{cases} \tag{8}$$

We adopt the Nesterov's method to solve the smoothed version of problem (5) since it can achieve the optimal convergence rate at $O(1/k^2)$, which indicates a low time complexity [15], [38]. To utilize Nesterov's method for optimization, we have to compute the gradient of the smoothed hinge loss to determine the descent direction, as well as the Lipschitz constant to determine the step size of each iteration. We summarize the results in the following theorem.

Theorem 1: The gradient of the smoothed hinge loss $g_\sigma(\theta)$ is

$$\frac{\partial g_\sigma(h_k, y_k, \theta)}{\partial \theta} = y_k h_k v_k. \tag{9}$$

The sum of the gradient over all the samples is

$$\frac{\partial g_\sigma(\theta)}{\partial \theta} = \sum_k y_k h_k v_k = H^\Phi Y v, \tag{10}$$

where $H^\Phi = [h_1, \ldots, h_{N'}]$ and $Y = \mathrm{diag}(y)$. The Lipschitz constant of $g_\sigma(\theta)$ is

$$L^g(\theta) = \frac{N'}{\sigma} \max_k \frac{\|h_k h_k^T\|_2}{\|h_k\|_\infty}. \tag{11}$$

We leave the proof in the Appendix.

Similarly, let $l(\theta) = \|\theta\|_1$, so we have the following piecewise approximation of l with the smooth parameter σ':

$$l'_\sigma = \begin{cases} -\theta_r - \frac{\sigma'}{2}, & \theta_r < -\sigma'; \\ \theta_r - \frac{\sigma'}{2}, & \theta_r > \sigma'; \\ \theta_r^2/(2\sigma'), & \text{otherwise.} \end{cases} \tag{12}$$

The gradient is given by $\partial(\sum_{r=1}^n l_{\sigma'}(\theta_r))/\partial\theta = v'$ with each $v'_r = \mathrm{median}\{\theta_r/\sigma', -1, 1\}$ and the Lipschitz constant $L^l(\theta) = 1/\sigma'$.

In addition, the gradient of $\Omega(\theta)$ is given by

$$\frac{\partial \Omega(\theta)}{\partial \theta} = H^\Omega \theta - h^\Omega, \tag{13}$$

where $H^\Omega_{st} = \gamma_A \mathrm{tr}((u_s u_s^T)(u_t u_t^T))$ and $h^\Omega_r = \gamma_A \mathrm{tr}(A_S^T (u_r u_r^T))$.

Therefore, the gradient of the smoothed $F(\theta)$, is

$$\frac{\partial F_\sigma(\theta)}{\partial \theta} = \frac{1}{N'} H^\Phi Y v + \gamma_C v' + H^\Omega \theta - h^\Omega, \tag{14}$$

and the Lipschitz constant is

$$L_\sigma = \frac{1}{\sigma} \max_k \frac{\|h_k h_k^T\|_2}{\|h_k\|_\infty} + \frac{\gamma_C}{\sigma'} + \|H^\Omega\|_2. \tag{15}$$

Finally, based on the obtained gradient and Lipschitz constant, we apply Nesterov's method to minimize the smoothed primal $F_\sigma(\theta)$. In the t'th iteration round, two auxiliary optimizations are constructed and their solutions are used to build the solution of problem (5). We use θ^t, y^t and z^t to represent the solutions of DTDML w.r.t. θ and its two auxiliary optimizations at the t'th iteration round, respectively. The Lipschitz constant of $F_\sigma(\theta)$ is L_σ and the two auxiliary optimizations are,

$$\min_y \langle \nabla F_\sigma(\theta^t), y - \theta^t \rangle + \frac{L_\sigma}{2} \|y - \theta^t\|_2^2,$$

$$\min_z \sum_{i=0}^t \frac{i+1}{2} [F_\sigma(\theta^i) + \langle \nabla F_\sigma(\theta^i), z - \theta^i \rangle] + \frac{L_\sigma}{2} \|z - \hat{\theta}\|_2^2.$$

where $\hat{\theta}$ is a guessed solution of θ. By directly setting the gradients of the two objective functions in the auxiliary optimizations as zeros, we can obtain y^t and z^t, respectively,

$$y^t = \theta^t - \frac{1}{L_\sigma} \nabla F_\sigma(\theta^t), \tag{16}$$

$$z^t = \hat{\theta} - \frac{1}{L_\sigma} \sum_{i=0}^t \frac{i+1}{2} \nabla F_\sigma(\theta^i). \tag{17}$$

The solution after the t'th iteration round is the weighted sum of y^t and z^t, i.e.,

$$\theta^{t+1} = \frac{2}{t+3} z^t + \frac{t+1}{t+3} y^t. \tag{18}$$

The stop criterion is $|F_\sigma(\theta^{t+1}) - F_\sigma(\theta^t)| < \epsilon$. The initialization θ^0 and guessed solution $\hat{\theta}$ are set as the zero vectors.

For fixed θ, the optimization problem with respect to α can be formulated as

$$\begin{aligned} &\operatorname*{argmin}_\alpha \frac{\gamma_A}{2} \|A - \sum_{p=1}^m \alpha_p A_p\|_F^2 + \frac{\gamma_B}{2} \|\alpha\|_2^2, \\ &\text{s.t.} \quad \sum_{p=1}^m \alpha_p = 1, \alpha_p \geq 0, p = 1, \ldots, m. \end{aligned} \tag{19}$$

This is a standard quadratic programming problem and can be rewritten in compact form as

$$\begin{aligned} &\operatorname*{argmin}_\alpha \frac{1}{2} \alpha^T H \alpha - \alpha^T h + \frac{\gamma_B}{2} \|\alpha\|_2^2, \\ &\text{s.t.} \quad \sum_{p=1}^m \alpha_p = 1, \alpha_p \geq 0, p = 1, \ldots, m. \end{aligned} \tag{20}$$

where the constant term has been omitted, $h = [h_1, \ldots, h_m]$ with each $h_p = \gamma_A \mathrm{tr}(A^T A_p)$, and H is a symmetric positive semi-definite matrix with the entry $H_{st} = \gamma_A \mathrm{tr}(A_s^T A_t)$. This is a constrained quadratic optimization problem and can be solved efficiently using the coordinate descent algorithm. In each iteration, we select two elements α_i and α_j to update, and leave the others to be fixed. To satisfy the constraint $\sum_{p=1}^m \alpha_p = 1$, we have $\alpha_i^* + \alpha_j^* = \alpha_i + \alpha_j$, where α_i^* and α_j^* are the solutions of the current iteration. In addition, by using the Lagrangian of (20), we obtain the following updating rule:

$$\begin{cases} \alpha_i^* = \dfrac{\gamma_B(\alpha_i + \alpha_j) + (h_i - h_j) + \varepsilon_{ij}}{(H_{ii} - H_{ij} - H_{ji} + H_{jj}) + 2\gamma_B}, \\ \alpha_j^* = \alpha_i + \alpha_j - \alpha_i^*, \end{cases} \tag{21}$$

where $\varepsilon_{ij} = (H_{ii} - H_{ji} - H_{ij} + H_{jj})\alpha_i - \sum_k (H_{ik} - H_{jk})\alpha_k$. The obtained α_i^* or α_j^* may violate the constraint $\alpha_p \geq 0$, so we set

$$\begin{cases} \alpha_i^* = 0, \alpha_j^* = \alpha_i + \alpha_j, & \text{if } \gamma_B(\alpha_i + \alpha_j) + (h_i - h_j) + \varepsilon_{ij} \leq 0, \\ \alpha_j^* = 0, \alpha_i^* = \alpha_i + \alpha_j, & \text{if } \gamma_B(\alpha_i + \alpha_j) + (h_j - h_i) + \varepsilon_{ji} \leq 0. \end{cases}$$

C. Automatic Determination of the Regularization Parameters γ_B and γ_C

In the proposed model (4), we have three parameters γ_A, γ_B and γ_C to determine. Determination of all these parameters is nontrivial [46] due to the limited number of labeled data

Algorithm 1 The Optimization Procedure of the Proposed DTDML Algorithm With Automatic Determination of the Regularization Parameters γ_B and γ_C. Both ρ_C and ρ_B Are Empirically Set to One

Initialize: $\alpha^{(0)}$, $\theta^{(0)}$, $\gamma_B^{(0)}$ and $\gamma_C^{(0)}$. Set $t \leftarrow 0$, construct $A^{(0)} = \sum_{r=1}^{n} \theta_r^{(0)} u_r u_r^T$ and $A_S^{(0)} = \sum_{p=1}^{m} \alpha_p^{(0)} A_p$.

1: **Iterate**

2: Optimize

$$\theta^{(t+1)} \leftarrow \underset{\theta}{\operatorname{argmin}} \frac{1}{N'} \sum_{k=1}^{N'} g\left(y_k(1-\theta^T h_k)\right) + \frac{\gamma_A}{2}\|A - A_S^{(t)}\|_F^2 + \gamma_C^{(t)}\|\theta\|_1$$

and update $A^{(t+1)} = \sum_{r=1}^{n} \theta_r^{(t+1)} u_r u_r^T$;

3: Optimize

$$\alpha^{(t+1)} \leftarrow \underset{\alpha}{\operatorname{argmin}} \frac{\gamma_A}{2}\|A^{(t+1)} - A_S\|_F^2 + \frac{\gamma_B^{(t)}}{2}\|\alpha\|_2^2$$

and update $A_S^{(t+1)} = \sum_{p=1}^{m} \alpha_p^{(t+1)} A_p$;

4: Compute

$$\gamma_C^{(t+1)} = |\rho_C|\Big[\frac{1}{N'}\sum_{k=1}^{N'} g\left(y_k(1-(\theta^{(t+1)})^T h_k)\right) + \frac{\gamma_A}{2}\|A^{(t+1)} - A_S^{(t)}\|_F^2\Big] / \|\theta^{(t+1)}\|_1;$$

5: Compute

$$\gamma_B^{(t+1)} = |\rho_B|\big[\gamma_A\|A^{(t+1)} - A_S^{(t+1)}\|_F^2\big] / \|\alpha^{(t+1)}\|_2^2;$$

6: $t \leftarrow t+1$.

7: **Until convergence**

available in the target task. Therefore, we present an automatic determination algorithm for the regularization parameters γ_B and γ_C. This algorithm is inappropriate for the determination of γ_A because the corresponding regularization term $\|A - A_S\|_F^2$ is a coupling of α and θ.

The algorithm is based on the L-curve, which graphically displays the trade-off between approximation error and solution size as the regularization parameter varies [39], [40]. The proper regularization parameter value is associated with the corner of the curve, where both solution and approximation error have small norms. Following [40], we choose a tangency-based method [39] to find the L-corner since it has a convergence guarantee and the computation is fast. The procedure is shown in Algorithm 1, where ρ_C and ρ_B are slopes of the straight line that are tangent to the L-curves, and are set to be one, empirically, in this paper.

The stopping criterion for terminating the algorithm can be the difference of the objective value $\frac{1}{N'}\sum_{k=1}^{N'} g(y_k(1-\theta^T h_k)) + \frac{\gamma_A}{2}\|A - A_S\|_F^2 + \frac{\gamma_B}{2}\|\alpha\|_2^2 + \gamma_C\|\theta\|_1$ between two consecutive steps. Alternatively, we can stop the iterations when the variation of α and θ are both smaller than a predefined threshold. Our implementation is based on the difference of the objective value, i.e., if the value $|O_k - O_{k-1}|/|O_k - O_0|$ is smaller than a predefined threshold, then the iteration stops, where O_k is the objective value of the k'th iteration step.

D. Theoretical Analysis

The generalization error bound of the proposed DTDML algorithm is now provided. We derive the generalization bound using the uniform stability [41].

1) Uniform Stability:

Definition 1 (Uniform Stability [41]): An algorithm has uniform stability β with respect to the loss function l if the following holds

$$\forall s \in \mathcal{Z}^m, \quad \forall i \in \{1, \ldots, m\}, \|l(h_s, \cdot) - l(h_{s^i}, \cdot)\|_\infty \leq \beta, \quad (22)$$

where $\mathcal{Z}$ is the sample space, h_s is the hypothesis function returned by the algorithm learning with the set of samples s, and $s^i = \{z_1, \ldots, z_{i-1}, z_{i'}, z_{i+1}, \ldots, z_m\}$ denotes a set of samples with the i'th element z_i replaced by $z_{i'}$.

To obtain the uniform stability, we use the Bregman divergence [42]. Bregman divergence is defined for any convex and differentiable function $F : \mathcal{H} \to \mathbb{R}$ as follows (here $\mathcal{H}$ denotes the Hilbert space):

$$\forall f, g \in \mathcal{H}, B_F(f \parallel g) = F(f) - F(g) - \operatorname{tr}(\langle f - g, \nabla F(g)\rangle). \quad (23)$$

For non-differential loss function, we use the generalized Bregman divergence. The sub-gradient of F at h (see [43]) is defined as

$$\partial F(h) = \{g \in \mathcal{H} \mid \forall h' \in \mathcal{H}, F(h') - F(h) \geq \operatorname{tr}(\langle h' - h, g\rangle)\}. \quad (24)$$

Let $\delta F(h)$ be an arbitrary element of $\partial F(h)$. The generalized Bregman divergence to F is then defined as

$$\forall h', h \in \mathcal{H}, B_F(h' \parallel h) = F(h') - F(h) - \operatorname{tr}(\langle h' - h, \delta F(h)\rangle). \quad (25)$$

According to the definition of sub-gradient, we have $B_F(h' \parallel h) \geq 0$ and $B_{P+Q} = B_P + B_Q$ for any convex functions P and Q. That is, the generalized Bregman divergence is non-negative and additive.

In addition, to derive the uniform stability, we need the following lemma cited from [12] (Proposition 2 therein):

Lemma 1: For any two distance metrics A and A', the following inequality holds for any sample z_i and z_j

$$|V(A, z_i, z_j) - V(A', z_i, z_j)| \leq 4LR^2\|A - A'\|_F. \quad (26)$$

Then we present the uniform stability for our model.

Theorem 2: Let β be the uniform stability of the developed algorithm for problem (2) and assume $\|x\|_2 \leq R$ for any sample x. Then,

$$\beta \leq \frac{64L^2R^4}{\gamma_A N}. \quad (27)$$

where L is the Lipschitz constant of the function g.

The detailed proof of Theorem 2 can be found in the Appendix. We then derive the generalization bound via the uniform stability.

2) Generalization Error Bound: Let $\mathcal{N}$ denote the sample set and $V(A, z_i, z_j) = g(y_{ij}[1 - \|x_i - x_j\|_A^2])$. The empirical risk and expected risk can be defined as $R_{\mathcal{N}}(A) = \frac{2}{N(N-1)} \sum_{i<j} V(A, z_i, z_j)$ and $R(A) = E_{(z_i,z_j)}[V(A, z_i, z_j)]$, respectively. A probabilistic bound on the defect $R(A) - R_{\mathcal{N}}(A)$ is called the generalization bound.

The bound can be derived by utilizing the obtained uniform stability and the following McDiarmid inequality [44].

Theorem 3 (McDiarmid Inequality [44]): Let $z_1, \ldots, z_N \in \mathcal{Z}$ be a set of N independent random variables and assume that there exist $c_1, \ldots, c_N$ such that $f : \{z_i\}_{i=1}^N \mapsto \mathbb{R}$ satisfying

$$\sup_{z_1,\ldots,z_N,z_{i'}} |f(z_1, \ldots, z_N) - f(z_1, \ldots, z_{i-1}, z_{i'}, z_{i+1}, z_N)| \leq c_i, \tag{28}$$

for all $i \in [1, N]$ and any point $z_{i'} \in \mathcal{Z}$. Let $f(\mathcal{N}) = f(z_1, \ldots, z_N)$. Then, for all $\epsilon > 0$, the following holds:

$$Pr(f(\mathcal{N}) - E[f(\mathcal{N})] \geq \epsilon) \leq \exp\left(\frac{-2\epsilon^2}{\sum_{i=1}^N c_i^2}\right). \tag{29}$$

Then we present the generalization bound for our model.

Theorem 4: Let $\mathcal{N}$ be a set of N randomly selected samples and $A_{\mathcal{N}}$ be the distance metric learned by solving (2). With probability at least $1 - \delta$, we have

$$R(A_{\mathcal{N}}) - R_{\mathcal{N}}(A_{\mathcal{N}}) \leq \frac{128L^2R^4}{\gamma_A N} + M\sqrt{\frac{\ln(1/\delta)}{2N}}, \tag{30}$$

where

$$M = \frac{128L^2R^4 + 4\gamma_A g_{A_S} + 16\sqrt{2\gamma_A}LR^2\sqrt{g_{A_S} + \gamma_C\|\theta_S\|_1}}{\gamma_A}.$$

Here, θ_S is a solution of $A = A_S$ and $g_{A_S} = \sup_{z_i,z_j} V(A_S, z_i, z_j)$ is the largest loss when the distance metric is A_S.

To prove Theorem 4, we need an additional lemma:

Lemma 2: The following two inequalities hold:

1)

$$\|A_{\mathcal{N}} - A_S\|_F \leq \sqrt{\left(2(g_{A_S} + \gamma_C(\|\theta_S\|_1 - \|\theta_{\mathcal{N}}\|_1))\right)/\gamma_A}$$

and

2)

$$\|A_{\mathcal{N}'} - A_S\|_F \leq \sqrt{\left(2(g_{A_S} + \gamma_C(\|\theta_S\|_1 - \|\theta_{\mathcal{N}'}\|_1))\right)/\gamma_A} \leq \sqrt{\left(2(g_{A_S} + \gamma_C\|\theta_S\|_1)\right)/\gamma_A}.$$

The detailed proof of Theorem 4 for the bound can be found in the Appendix.

Remark 1: In the upper bound of generalization error, A_S and θ_S are learned from the source data. They are the information that was transferred from the source data to the target data.

IV. Experimental Evaluation

This section outlines the validation of the effectiveness of the proposed DTDML empirically on two popular handwritten image datasets, and a challenging natural image dataset. The first two datasets are obtained from [14]. Specifically, we compare the following methods:

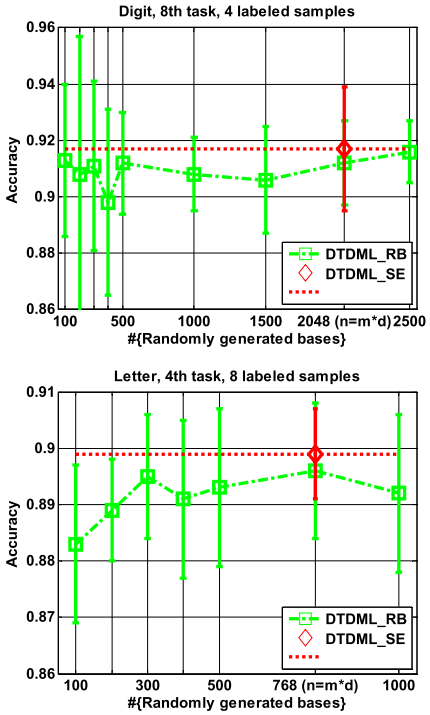

Fig. 2. A comparison of DTDML using source eigenvectors (DTDML_SE) and random bases (DTDML_RB). A different number of random bases is investigated. (Top row: Handwritten digit; Bottom row: Handwritten letter.)

- **RDML [12]:** an online algorithm that has been demonstrated empirically to be effective and quite efficient in learning a distance metric, and can handle high dimensional data. This algorithm serves as a baseline here since it learns only from the target task and leverages nothing from the source tasks.
- **RDML_AGG:** a simple aggregation strategy, which is to learn the target metric by directly applying RDML on the training set that consists of data from both the source and target tasks.
- **LDML [13]:** a transfer distance metric learning algorithm that is based on [10], and is formulated as:

$$\begin{aligned} \underset{A}{\text{argmin}} \sum_{p=1}^m \beta_p \left(\text{tr}(A_p^{-1}A)\right) - \log\det A + \gamma_S \text{tr}(SA) \\ -\gamma_D \text{tr}(DA) + \gamma_B\|\beta\|_2^2, \\ \text{s.t. } A \succcurlyeq 0, \sum_{p=1}^m \beta_p = 1, \beta_p \geq 0, p = 1, 2, \ldots, m, \end{aligned} \tag{31}$$

  where S and D are matrices of the similar and dissimilar constraints. The above formulation contains a semi-definite programming (SDP) problem, and in our re-implementation it is solved using the SDPT3 solver. According to [13], the parameters can be set empirically as $\gamma_D = \frac{1}{4}\gamma_S$ and $\gamma_B = \frac{1}{8}\gamma_S$. Therefore, only γ_S needs to be tuned.
- **TML [14]:** a recently proposed transfer metric learning algorithm. Similar to [12], an online algorithm is developed to learn the target metric. The task relationship is learned for transfer by solving a second-order cone programming (SOCP) problem using the CVX solver.

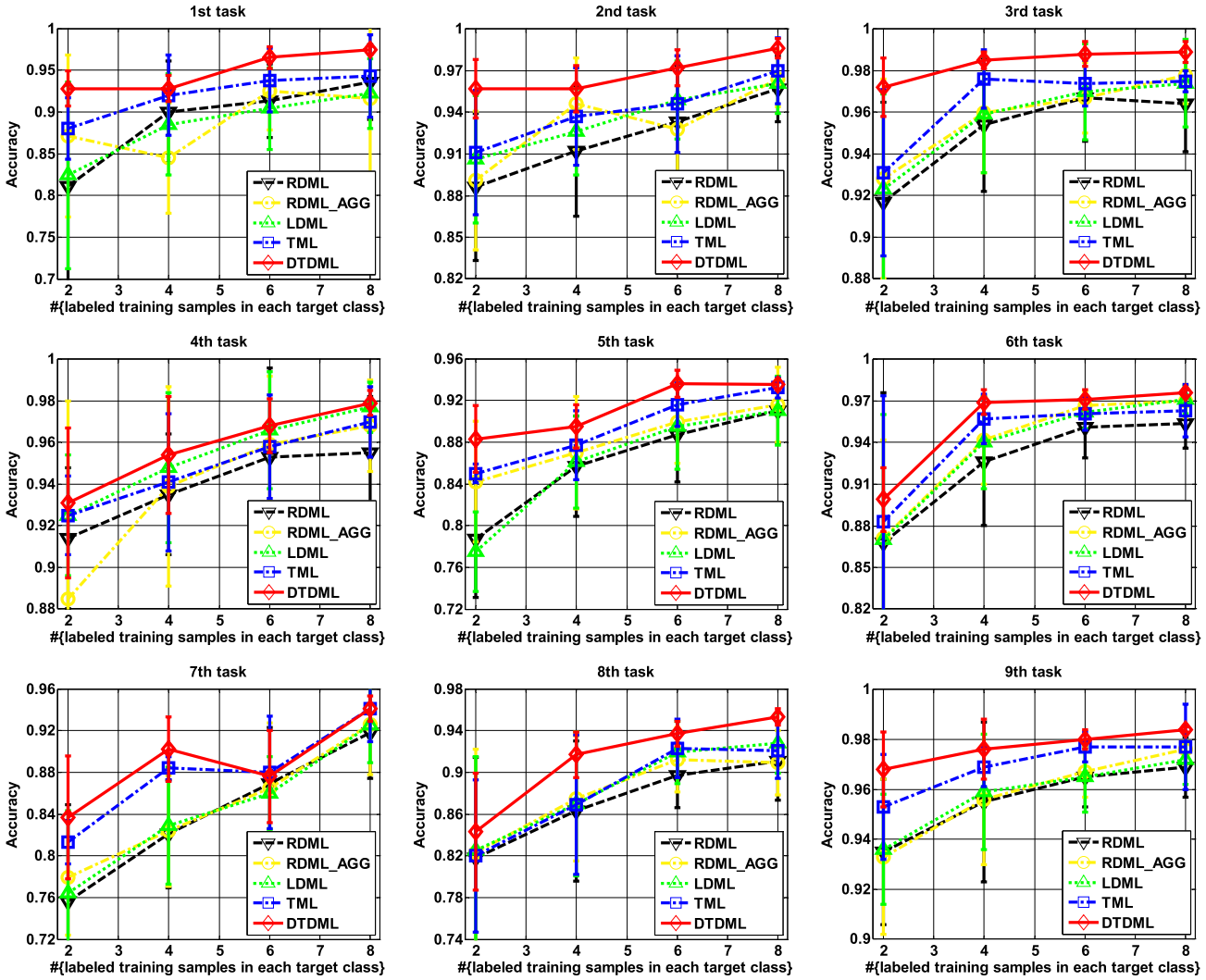

Fig. 3. Classification performance vs. the number of labeled training samples on the USPS digit dataset.

TABLE I
AVERAGE PERFORMANCE OVER ALL TASKS ON USPS DIGIT CLASSIFICATION

Methods	2	4	6	8
RDML [12]	0.855±0.071	0.903±0.046	0.926±0.035	0.942±0.029
RDML_AGG [12]	0.869±0.067	0.907±0.045	0.932±0.032	0.947±0.033
LDML [13]	0.861±0.061	0.909±0.042	0.932±0.029	0.949±0.024
TML [14]	0.885±0.040	0.926±0.030	0.941±0.025	0.954±0.022
DTDML	**0.913±0.031**	**0.943±0.018**	**0.955±0.014**	**0.968±0.006**

TABLE II
AVERAGE PERFORMANCE OVER ALL TASKS ON HANDWRITTEN LETTER CLASSIFICATION

Methods	4	8	12	16
RDML [12]	0.782±0.049	0.818±0.036	0.823±0.033	0.854±0.029
RDML_AGG [12]	0.775±0.060	0.818±0.041	0.842±0.036	0.858±0.029
LDML [13]	0.792±0.049	0.816±0.037	0.831±0.027	0.851±0.025
TML [14]	0.803±0.031	0.826±0.038	0.846±0.030	0.867±0.022
DTDML	**0.835±0.042**	**0.845±0.021**	**0.857±0.019**	**0.877±0.020**

In addition, the parameters are automatically determined by adopting a Bayesian regularization scheme for the model.

- **DTDML:** the proposed decomposition based transfer distance metric learning. The parameters γ_B and γ_C are determined automatically and we only need to optimize γ_A.

We train the source metrics using the RDML method and all the available data in the source tasks. We split the data into equal training and test sets for the target task. The number of labeled samples that are chosen from the training set is gradually increased to see the performance variation w.r.t. the size of the labeled set. We evaluate the learned target metric by applying the 1-nearest-neighbor classifier on the test set. Ten random choices of the labeled samples are used in our experiments. Both the mean and standard deviation of the accuracies are reported.

A. Handwritten Image Classification

One of the handwritten image datasets we use is the well-known USPS digit dataset,[1] which contains 7,291 samples. Each sample is an image of size 16×16 in raw pixels, and the feature dimension $d = 256$. We consider nine classification tasks, i.e., 0/6, 0/8, 1/4, 2/7, 3/5, 4/7, 4/9,

[1] http://www.csie.ntu.edu.tw/∼cjlin/libsvmtools/datasets/multiclass.html#usps

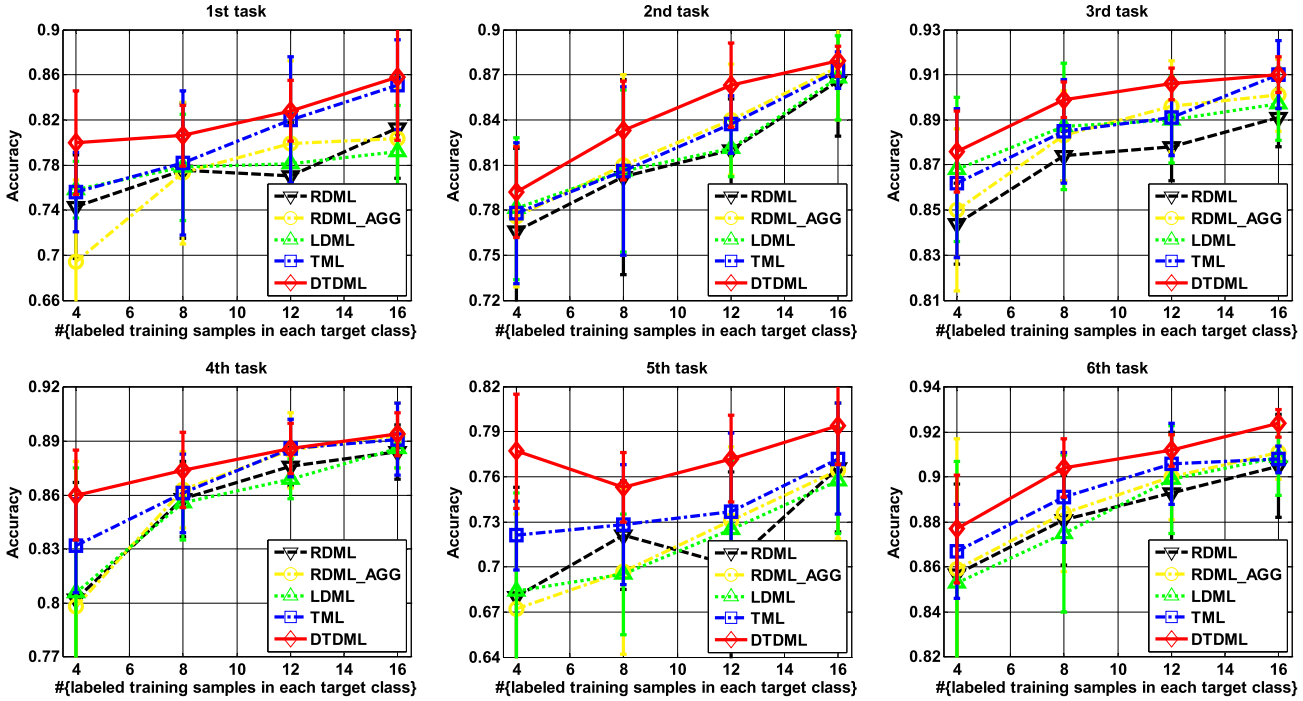

Fig. 4. Classification performance vs. the number of labeled training samples on the handwritten letter dataset.

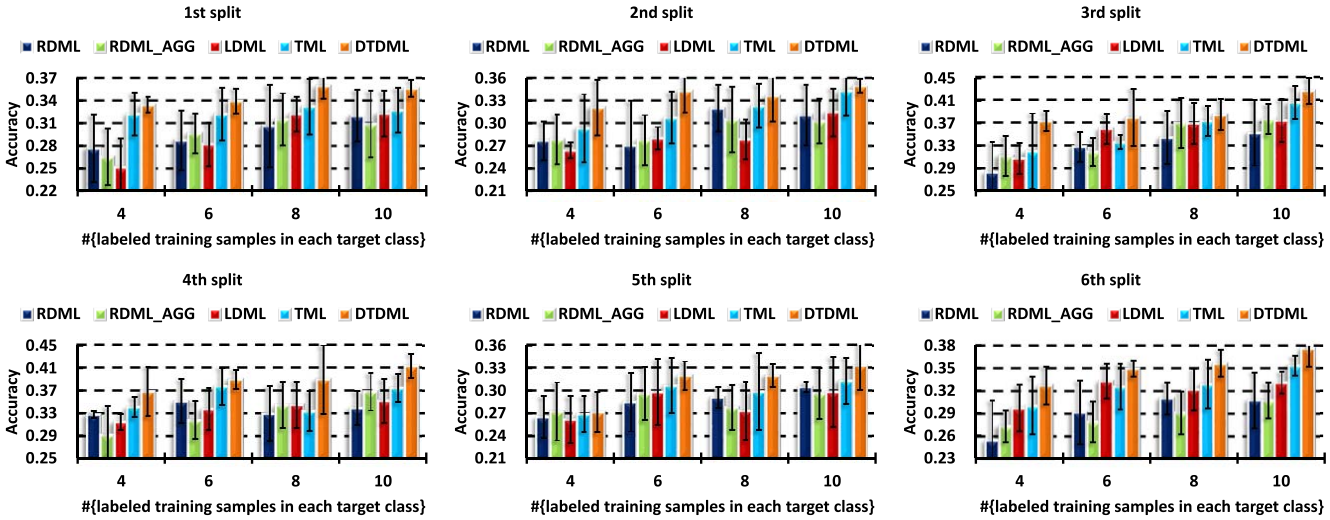

Fig. 5. Annotation performance vs. the number of labeled training samples on the NUS-WIDE dataset.

5/8, and 6/8, each corresponding to a classification of two digits. One of the nine tasks is treated as the target task and the others are the source tasks (each task is treated as the target task in turn).

The other is a handwritten letter dataset,[2] which is a little different from the dataset presented in [14] since it cannot be downloaded immediately, according to the web link provided. The letter dataset used in this paper consists of 52, 152 samples and the feature dimension is 128. Six binary classification problems, i.e., c/e, m/n, a/g, a/o, f/t, and h/n, are considered. For each task, we randomly select at most 1 000 positive and 1 000 negative samples from the dataset. The experimental settings are the same as for those of the digit classification.

1) A Self-Comparison of DTDML Using Source Eigenvectors and Random Bases: As depicted in this paper, the "base metrics" $B_r = u_r u_r^T$ that are utilized to construct the target metric $A = \sum_{r=1}^{n} \theta_r B_r$ can be derived from either the source eigenvectors, or other randomly generated bases. Therefore, we first investigate the performance of these two strategies, which are denoted as DTDML_SE and DTDML_RB, respectively. For DTDML_SE, the $u_r, r = 1, \ldots, n$ in problem (2) are eigenvectors of the source metrics, so the number of base metrics is fixed as $n = m \times d$. For DTDML_RB, each u_r is an eigenvector of some random matrix, and thus we can

[2]http://ai.stanford.edu/∼btaskar/ocr/

generate arbitrary number of base metrics. We randomly select one task from each of the two handwritten datasets, and report the results in Fig. 2.

The results demonstrate that: 1) Even when the number of random bases n_b is very small, e.g., 100, we can still obtain satisfactory accuracy; 2) The accuracy of DTDML_RB tends to be higher when n_b is increased, but usually cannot outperform DTDML_SE. To this end, we adopt DTDML_SE in the following experiments. Another reason for choosing DTDML_SE is to avoid tuning the additional parameter n_b.

2) A Comparison With the Other Algorithms: The classification accuracies of different methods, under different settings on the digit dataset are shown in Fig. 3. We observe from the results that: 1) when the number of labeled training samples increases, the performance of all the compared methods tends to be better (higher mean accuracy and smaller variance); 2) the transfer metric learning algorithms (LDML, TML, DTDML) that utilize the source task information for target task learning are usually superior to RDML, which only learns on the target task. RDML is comparable to LDML on some tasks (e.g., the 5'th, 7'th, and 9'th task), which may be due to the finding of a bad local minima in LDML; 3) Overall, the performance of RDML_AGG, which directly utilizes both the source and target training data without transfer, is better than RDML but worse than the transfer methods. This indicates that the distributions of the source and target datasets are different but related, and the transfer is necessary; 4) TML is better than LDML and RDML in most cases, while the proposed DTDML consistently outperforms all of them. In addition, we present the average performance over all settings in Table I. The results indicate a significant 3.2% improvement compared with TML when using two labeled training samples. The level of improvement drops when more labeled samples are available. This is because DTDML has far fewer variables to learn than TML. The significance of this advantage gradually decreases since variable estimation can be steadily improved with an increase of labeled training samples. This indicates that the proposed algorithm is more suitable for the transfer scenario, since the labeled sample size of the target task is usually very small.

We report the performance on the letter dataset in Fig. 4. Similar to the digit classification, LDML is comparable to RDML and RDML_AGG sometimes, and DTDML is superior to other methods significantly on almost all tasks. The average performance is presented in Table II and we observe a significant 3.8% improvement compared against TML when using four labeled training samples.

B. Web Image Annotation

This section provides details of the experiments conducted on a natural image dataset NUS-WIDE [16] to further verify the effectiveness of the proposed algorithm. This dataset contains 269, 648 images and the features used in our experiments are 500-D bag of visual words based on SIFT [45] descriptors. To perform a meaningful transfer, we select 12 animal concepts: bear, bird, cat, cow, dog, elk, fish, fox, horse, tiger, whale, and zebra. For each concept, 100 samples were randomly selected from the dataset.

In this set of experiments, the source task requires annotation of six randomly selected concepts, and the target task requires annotation of all others. Both are multi-class problems, but there is no difference in training compared to the binary case since the sample pairs are used and only the pair labels are needed. A pair of samples is labeled as positive if they are from the same class, and negative otherwise.

We perform six random splits of the concept set, and show the result of each split in Fig. 5. Similar conclusions can be obtained as in the handwritten image classification. DTDML always performs the best for all splits and in particular, we obtain an 8.1% improvement on the average performance over all splits compared with TML when using four labeled samples.

V. Conclusion and Discussion

Existing transfer metric learning approaches usually learn entries of the target metric directly, so the amount of variables is large, especially for the high dimensional image features. To resolve this problem, we have presented a decomposition based method called DTDML that assumes the target metric can be represented as a combination of "base metrics". DTDML has far less variables because we only have to learn the combination coefficients of the "base metrics", so better solutions can be obtained. In addition, we adopt Nesterov's optimal method to learn the coefficients and the optimization is quite efficient.

From the experimental validation on the popular handwritten image datasets and a challenging natural image dataset, we conclude that: 1) both source eigenvectors and random bases can be used to construct the target metric and the former performs a little better; 2) In the transfer scenario, using "base metrics" to induce the target metric is more effective than learning the target metric variables directly, even when the "base metrics" are randomly generated.

Appendix A
Proof of Theorem 1

Proof: According to (7) and (8), we can calculate the gradient of g_σ for the k'th sample as

$$\frac{\partial g_\sigma}{\partial \theta} = \begin{cases} 0, & v_k = 0; \\ y_k h_k, & v_k = 1; \\ \frac{y_k h_k (-y_k (1-\theta^T h_k))}{\sigma \|h_k\|_\infty}, & v_k = \frac{-y_k (1-\theta^T h_k)}{\sigma \|h_k\|_\infty}. \end{cases} \tag{32}$$

This leads to (9). Given function $g(x)$, for any x^1 and x^2, the Lipschitz constant L satisfies

$$\|\nabla g(x^1) - \nabla g(x^2)\|_2 \leq L \|x^1 - x^2\|_2. \tag{33}$$

Hence the Lipschitz constant of g_σ can be calculated from

$$\max \frac{\|\frac{\partial g_\sigma}{\partial \theta^1} - \frac{\partial g_\sigma}{\partial \theta^2}\|_2}{\|\theta^1 - \theta^2\|_2} \leq L^g. \tag{34}$$

According to (32), we have

$$\frac{\partial g_\sigma}{\partial \theta^1} - \frac{\partial g_\sigma}{\partial \theta^2} = \begin{cases} 0, & y_k(1-\theta^T h_k) < -\sigma \|h_k\|_\infty \text{ or} \\ & y_k(1-\theta^T h_k) > 0; \\ \frac{h_k h_k^T (\theta^1 - \theta^2)}{\sigma \|h_k\|_\infty}, & \text{otherwise.} \end{cases} \tag{35}$$

Therefore,

$$\max_k \frac{\|h_k h_k^T(\theta^1-\theta^2)\|_2}{\sigma\|h_k\|_\infty\|\theta^1-\theta^2\|_2} \le \max_k \frac{\|h_k h_k^T\|_2\|(\theta^1-\theta^2)\|_2}{\sigma\|h_k\|_\infty\|\theta^1-\theta^2\|_2} = \frac{\|h_k h_k^T\|_2}{\sigma\|h_k\|_\infty} = L^g(h_k, y_k, \theta). \quad (36)$$

To this end, the Lipschitz constant of $L^g(\theta)$ is calculated as

$$\sum_k L^g(h_k, y_k, \theta) \le N' \max_k L^g(h_k, y_k, \theta) = \frac{N'}{\sigma}\max_k \frac{\|h_k h_k^T\|_2}{\|h_k\|_\infty} = L^g(\theta). \quad (37)$$

This completes the proof. ■

Appendix B
Proof of Theorem 2

Proof: Let's denote $F_{\mathcal{N}}(\theta) = P_{\mathcal{N}}(\theta) + Q(\theta)$, where $P_{\mathcal{N}}(\theta) = \frac{2}{N(N-1)}\sum_{i<j} V(A, z_i, z_j)$ and $Q(\theta) = \frac{\gamma_A}{2}\|A - A_S\|_F^2 + \gamma_C\|\theta\|_1$. It is obvious that both $P_{\mathcal{N}}(\theta)$ and $Q(\theta)$ are convex. We assume $\theta_{\mathcal{N}}$ and $\theta_{\mathcal{N}'}$ to be the minimizers of $F_{\mathcal{N}}(\theta)$ and $F_{\mathcal{N}'}(\theta)$, respectively, where $\mathcal{N}'$ is the collection of examples that replaces $z_i \in \mathcal{N}$ with another example $z_{i'}$.

Because the generalized Bregman divergence is non-negative and additive, we have

$$B_{F_{\mathcal{N}}}(\theta_{\mathcal{N}'} \parallel \theta_{\mathcal{N}}) + B_{F_{\mathcal{N}'}}(\theta_{\mathcal{N}} \parallel \theta_{\mathcal{N}'}) \ge B_Q(\theta_{\mathcal{N}'} \parallel \theta_{\mathcal{N}}) + B_Q(\theta_{\mathcal{N}} \parallel \theta_{\mathcal{N}'}). \quad (38)$$

Besides, $\partial Q(\theta_{\mathcal{N}})/\partial\theta = \frac{\gamma_A}{2}\partial(\|A - A_S\|_F^2)/\partial\theta + \gamma_C\delta f(\theta)$, where $\delta f(\theta)$ is the subgradient of $\|\theta\|_1$, so we can obtain

$$B_Q(\theta_{\mathcal{N}'} \parallel \theta_{\mathcal{N}}) + B_Q(\theta_{\mathcal{N}} \parallel \theta_{\mathcal{N}'}) = \gamma_A\|A_{\mathcal{N}'} - A_{\mathcal{N}}\|_F^2 + \gamma_C\Delta, \quad (39)$$

where $\Delta = \|\theta_{\mathcal{N}}\|_1 - \langle\theta_{\mathcal{N}}, \mathrm{sgn}(\theta_{\mathcal{N}'})\rangle + \|\theta_{\mathcal{N}'}\|_1 - \langle\theta_{\mathcal{N}'}, \mathrm{sgn}(\theta_{\mathcal{N}})\rangle \ge 0$, $\mathrm{sgn}(\theta) = [\mathrm{sgn}(\theta_1), \ldots, \mathrm{sgn}(\theta_n)]^T$ and $\mathrm{sgn}(x) = 1$ if $x > 0$ and -1 otherwise.

According to (38) and (39), we have

$$\begin{aligned}
&\gamma_A\|A_{\mathcal{N}'} - A_{\mathcal{N}}\|_F^2 \le B_{F_{\mathcal{N}}}(\theta_{\mathcal{N}'} \parallel \theta_{\mathcal{N}}) + B_{F'_{\mathcal{N}}}(\theta_{\mathcal{N}} \parallel \theta_{\mathcal{N}'}) \\
&= F_{\mathcal{N}}(\theta_{\mathcal{N}'}) - F_{\mathcal{N}}(\theta_{\mathcal{N}}) - \langle\theta_{\mathcal{N}'} - \theta_{\mathcal{N}}, \partial F_{\mathcal{N}}(\theta_{\mathcal{N}})\rangle \\
&\quad + F_{\mathcal{N}'}(\theta_{\mathcal{N}}) - F_{\mathcal{N}'}(\theta_{\mathcal{N}'}) - \langle\theta_{\mathcal{N}} - \theta_{\mathcal{N}'}, \partial F_{\mathcal{N}'}(\theta_{\mathcal{N}'})\rangle \\
&= F_{\mathcal{N}}(\theta_{\mathcal{N}'}) - F_{\mathcal{N}}(\theta_{\mathcal{N}}) + F_{\mathcal{N}'}(\theta_{\mathcal{N}}) - F_{\mathcal{N}'}(\theta_{\mathcal{N}'}) \\
&= P_{\mathcal{N}}(\theta_{\mathcal{N}'}) - P_{\mathcal{N}}(\theta_{\mathcal{N}}) + P_{\mathcal{N}'}(\theta_{\mathcal{N}}) - P_{\mathcal{N}'}(\theta_{\mathcal{N}'}) \\
&= \frac{2}{N(N-1)}\left(\sum_{\mathcal{N}} V(A_{\mathcal{N}'}, z_i, z_j) - \sum_{\mathcal{N}} V(A_{\mathcal{N}}, z_i, z_j) + \sum_{\mathcal{N}'} V(A_{\mathcal{N}}, z_{i'}, z_j) - \sum_{\mathcal{N}'} V(A_{\mathcal{N}'}, z_{i'}, z_j)\right) \\
&\le \frac{2}{N(N-1)}\left(\sum_{\mathcal{N}} |V(A_{\mathcal{N}'}, z_i, z_j) - V(A_{\mathcal{N}}, z_i, z_j)| + \sum_{\mathcal{N}'} |V(A_{\mathcal{N}}, z_{i'}, z_j) - V(A_{\mathcal{N}'}, z_{i'}, z_j)|\right) \\
&\le \frac{16LR^2}{N}\|A_{\mathcal{N}'} - A_{\mathcal{N}}\|_F. \quad (40)
\end{aligned}$$

The second equality holds because $\theta_{\mathcal{N}}$ and $\theta_{\mathcal{N}'}$ are minimizers of $F_{\mathcal{N}}(\theta)$ and $F_{\mathcal{N}'}(\theta)$ respectively, which implies that $\partial F_{\mathcal{N}}(\theta_{\mathcal{N}}) = \partial F_{\mathcal{N}'}(\theta_{\mathcal{N}'}) = 0$. The last inequality holds because of Lemma 1. By comparing the left and right side of (40), we obtain

$$\|A_{\mathcal{N}} - A_{\mathcal{N}'}\|_F \le \frac{16LR^2}{\gamma_A N}. \quad (41)$$

By further utilizing Lemma 1, i.e., $|V(A_{\mathcal{N}}, z_i, z_j) - V(A_{\mathcal{N}'}, z_i, z_j)| \le 4LR^2\|A_{\mathcal{N}} - A_{\mathcal{N}'}\|_F$, we have

$$|V(A_{\mathcal{N}}, z_i, z_j) - V(A_{\mathcal{N}'}, z_i, z_j)| \le \frac{64L^2R^4}{\gamma_A N}. \quad (42)$$

This completes the proof. ■

Appendix C
Proof of Lemma 2

Proof: Because θ_S is a solution of $A = A_S$, so we have

$$\begin{aligned}
&\frac{2}{N(N-1)}\sum_{i<j} V(A_{\mathcal{N}}, z_i, z_j) + \frac{\gamma_A}{2}\|A_{\mathcal{N}} - A_S\|_F^2 + \frac{\gamma_B}{2}\|\alpha\|_2^2 + \gamma_C\|\theta_{\mathcal{N}}\|_1 \\
&\le \frac{2}{N(N-1)}\sum_{i<j} V(A_S, z_i, z_j) + \frac{\gamma_B}{2}\|\alpha\|_2^2 + \gamma_C\|\theta_S\|_1, \quad (43)
\end{aligned}$$

This leads to

$$\begin{aligned}
\frac{\gamma_A}{2}\|A_{\mathcal{N}} - A_S\|_F^2 &\le \frac{2}{N(N-1)}\sum_{i<j} V(A_S, z_i, z_j) + \gamma_C(\|\theta_S\|_1 - \|\theta_{\mathcal{N}}\|_1). \quad (44)
\end{aligned}$$

since $\frac{2}{N(N-1)}\sum_{i<j} V(A_{\mathcal{N}}, z_i, z_j) \ge 0$. Therefore, we have $\|A_{\mathcal{N}} - A_S\|_F \le \sqrt{\left(2(g_{A_S} + \gamma_C(\|\theta_S\|_1 - \|\theta_{\mathcal{N}}\|_1))\right)/\gamma_A}$. The same procedure can be applied to bounding $\|A_{\mathcal{N}'} - A_S\|_F$. ■

Appendix D
Proof of Theorem 4

Proof: Let $\Phi(A_{\mathcal{N}}) = R(A_{\mathcal{N}}) - R_{\mathcal{N}}(A_{\mathcal{N}})$. It follows from [41] that $\Phi(A_{\mathcal{N}}) \le 2\beta$. Besides,

$$\begin{aligned}
&|\Phi(A_{\mathcal{N}}) - \Phi(A_{\mathcal{N}'})| \\
&= |R(A_{\mathcal{N}}) - R_{\mathcal{N}}(A_{\mathcal{N}}) - R(A_{\mathcal{N}'}) + R_{\mathcal{N}'}(A_{\mathcal{N}'})| \\
&\le |R(A_{\mathcal{N}}) - R(A_{\mathcal{N}'})| + |R_{\mathcal{N}}(A_{\mathcal{N}}) - R_{\mathcal{N}'}(A_{\mathcal{N}'})| \\
&\le \beta + \frac{2}{N(N-1)}\left(\frac{N(N-1)}{2} - (N-1)\right)\beta \\
&\quad + \left|\frac{2}{N(N-1)}\left(\sum_{j\neq i} V(A_{\mathcal{N}}, z_i, z_j) - \sum_{j\neq i'} V(A_{\mathcal{N}'}, z_{i'}, z_j)\right)\right| \\
&\le 2\beta + \frac{2}{N(N-1)}\sum_{j\neq i} |V(A_{\mathcal{N}}, z_i, z_j) - V(A_S, z_i, z_j)| \\
&\quad + \frac{2}{N(N-1)}\sum_{j\neq i'} |V(A_{\mathcal{N}'}, z_{i'}, z_j) - V(A_S, z_{i'}, z_j)| + \frac{4g_{A_S}}{N} \\
&\le 2\beta + \frac{4g_{A_S}}{N} + \frac{8LR^2(\|A_{\mathcal{N}} - A_S\|_F + \|A_{\mathcal{N}'} - A_S\|_F)}{N} \\
&\le 2\beta + \frac{4g_{A_S}}{N} + \Big[\left(8\sqrt{2}LR^2(\sqrt{g_{A_S} + \gamma_C(\|\theta_S\|_1 - \|\theta_{\mathcal{N}}\|_1)}\right. \\
&\quad \left. + \sqrt{g_{A_S} + \gamma_C\|\theta_S\|_1}\right)/(\sqrt{\gamma_A}N)\Big] \triangleq M'. \quad (45)
\end{aligned}$$

where $g_{A_S} = \sup_{z_i, z_j} V(A_S, z_i, z_j)$ is the largest loss when the distance metric is A_S. The last inequality holds because of Lemma 2.

Given $\delta > 0$, using the McDiarmid inequality, with probability at least $1 - \delta$, we have

$$\Phi(A_{\mathcal{N}}) \leq E[\Phi(A_{\mathcal{N}})] + NM' \sqrt{\frac{\ln(1/\delta)}{2N}}. \tag{46}$$

In addition, we can conclude that $E[\Phi(A_{\mathcal{N}})] \leq 2\beta$:

$$\begin{aligned} &|E\Phi(A_{\mathcal{N}})| \\ &\quad = |E\left(R(A_{\mathcal{N}}) - R(A_{\mathcal{N}'}) + R(A_{\mathcal{N}'}) - R_{\mathcal{N}}(A_{\mathcal{N}})\right)| \\ &\quad \leq |E(R(A_{\mathcal{N}}) - R(A_{\mathcal{N}'}))| + |E(R(A_{\mathcal{N}'}) - R_{\mathcal{N}}(A_{\mathcal{N}}))| \\ &\quad = 2|E(V(A_{\mathcal{N}}, z_i, z_j) - V(A_{\mathcal{N}'}, z_i, z_j))| \leq 2\beta. \end{aligned} \tag{47}$$

This completes the proof. ■

ACKNOWLEDGMENT

The authors would like to thank the handling associate editor Prof. Dr. Nilanjan Ray and all the three anonymous reviewers for their constructive comments.

REFERENCES

[1] K. Q. Weinberger, J. Blitzer, and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," in *Proc. Adv. Neural Inform. Process. Syst.*, 2005, pp. 1473–1480.

[2] M. Guillaumin, T. Mensink, J. Verbeek, and C. Schmid, "Tagprop: Discriminative metric learning in nearest neighbor models for image auto-annotation," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep./Oct. 2009, pp. 309–316.

[3] E. P. Xing, M. I. Jordan, S. Russell, and A. Y. Ng, "Distance metric learning with application to clustering with side-information," in *Proc. Adv. Neural Inform. Process. Syst.*, 2002, pp. 505–512.

[4] Y. Aytar and A. Zisserman, "Tabula rasa: Model transfer for object category detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Nov. 2011, pp. 2252–2259.

[5] Y. Luo, D. Tao, B. Geng, C. Xu, and S. J. Maybank, "Manifold regularized multitask learning for semi-supervised multilabel image classification," *IEEE Trans. Image Process.*, vol. 22, no. 2, pp. 523–536, Feb. 2013.

[6] N. Sawant, J. Z. Wang, and L. Jia, "Enhancing training collections for image annotation: An instance-weighted mixture modeling approach," *IEEE Trans. Image Process.*, vol. 22, no. 9, pp. 3562–3577, Sep. 2013.

[7] Y. Luo, D. Tao, C. Xu, C. Xu, H. Liu, and Y. Wen, "Multiview vector-valued manifold regularization for multilabel image classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 5, pp. 709–722, May 2013.

[8] J. Gui, D. Tao, Z. Sun, Y. Luo, X. You, and Y. Tang, "Group sparse multiview patch alignment framework with view consistency for image classification," *IEEE Trans. Image Process.*, vol. 23, no. 7, pp. 3126–3137, Jul. 2014.

[9] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov, "Neighbourhood components analysis," in *Proc. Adv. Neural Inform. Process. Syst.*, Dec. 2004, pp. 513–520.

[10] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon, "Information-theoretic metric learning," in *Proc. 24th Int. Conf. Mach. Learn. (ICML)*, 2007, pp. 209–216.

[11] K. Q. Weinberger and L. K. Saul, "Fast solvers and efficient implementations for distance metric learning," in *Proc. 25th Int. Conf. Mach. Learn. (ICML)*, 2008, pp. 1160–1167.

[12] R. Jin, S. Wang, and Y. Zhou, "Regularized distance metric learning: Theory and algorithm," in *Proc. Adv. Neural Inform. Process. Syst.*, 2009, pp. 862–870.

[13] Z.-J. Zha, T. Mei, M. Wang, Z. Wang, and X.-S. Hua, "Robust distance metric learning with auxiliary knowledge," in *Proc. Int. Joint Conf. Artif. Intell.*, 2009, pp. 1327–1332.

[14] Y. Zhang and D.-Y. Yeung, "Transfer metric learning with semi-supervised extension," *ACM Trans. Intell. Syst. Technol.*, vol. 3, no. 3, pp. 54:1–54:28, May 2012.

[15] Y. Nesterov, "Smooth minimization of non-smooth functions," *Math. Program.*, vol. 103, no. 1, pp. 127–152, May 2005.

[16] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y. Zheng, "NUS-WIDE: A real-world web image database from national university of singapore," in *Proc. ACM Int. Conf. Image Video Retr.*, 2009, pp. 48:1–48:9.

[17] L. Yang and R. Jin, "Distance metric learning: A comprehensive survey," Dept. Comput. Sci. Eng., Michigan State Univ., Tech. Rep., Lansing, MI, USA, May 2006.

[18] M. Schultz and T. Joachims, "Learning a distance metric from relative comparisons," in *Proc. Adv. Neural Inform. Process. Syst.*, 2003, pp. 41–48.

[19] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, Dec. 2000.

[20] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," in *Proc. Adv. Neural Inform. Process. Syst.*, 2001, pp. 585–591.

[21] S. C. H. Hoi, W. Liu, M. R. Lyu, and W.-Y. Ma, "Learning distance metrics with contextual constraints for image retrieval," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2006, pp. 2072–2078.

[22] M. S. Baghshah and S. B. Shouraki, "Semi-supervised metric learning using pairwise constraints," in *Proc. Int. Joint Conf. Artif. Intell.*, 2009, pp. 1217–1222.

[23] N. Shental, T. Hertz, D. Weinshall, and M. Pavel, "Adjustment learning and relevant component analysis," in *Proc. 7th Eur. Conf. Comput. Vis.*, May 2002, pp. 776–790.

[24] J. Quionero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence, *Dataset Shift in Machine Learning*. Cambridge, MA, USA: MIT Press, 2009.

[25] T. Evgeniou and M. Pontil, "Regularized multi-task learning," in *Proc. ACM Int. Conf. Knowl. Discovery Data Mining*, 2004, pp. 109–117.

[26] J. Huang, A. Gretton, K. M. Borgwardt, B. Schölkopf, and A. J. Smola, "Correcting sample selection bias by unlabeled data," in *Proc. Adv. Neural Inform. Process. Syst.*, 2007, pp. 601–608.

[27] A. Argyriou, T. Evgeniou, and M. Pontil, "Convex multi-task feature learning," *Mach. Learn.*, vol. 73, no. 3, pp. 243–272, Dec. 2008.

[28] B. Geng, D. Tao, and C. Xu, "DAML: Domain adaptation metric learning," *IEEE Trans. Image Process.*, vol. 20, no. 10, pp. 2980–2989, Oct. 2011.

[29] J. Blitzer, R. McDonald, and F. Pereira, "Domain adaptation with structural correspondence learning," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2006, pp. 120–128.

[30] A. Argyriou, T. Evgeniou, and M. Pontil, "Multi-task feature learning," in *Proc. Adv. Neural Inform. Process. Syst.*, 2007, pp. 41–48.

[31] R. Gupta and L.-A. Ratinov, "Text categorization with knowledge transfer from heterogeneous data sources," in *Proc. 23rd Nat. Conf. Artif. Intell.*, 2008, pp. 842–847.

[32] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

[33] W. Dai, Q. Yang, G.-R. Xue, and Y. Yu, "Boosting for transfer learning," in *Proc. Int. Conf. Mach. Learn.*, Jun. 2007, pp. 193–200.

[34] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng, "Self-taught learning: Transfer learning from unlabeled data," in *Proc. 24th Int. Conf. Mach. Learn.*, 2007, pp. 759–766.

[35] L. Mihalkova, T. Huynh, and R. J. Mooney, "Mapping and revising Markov logic networks for transfer learning," in *Proc. 22nd Nat. Conf. Artif. Intell.*, 2007, pp. 608–614.

[36] H. Van Nguyen, V. Patel, N. M. Nasrabad, and R. Chellappa, "Design of non-linear Kernel dictionaries for object recognition," *IEEE Trans. Image Process.*, vol. 22, no. 12, pp. 5123–5135, Dec. 2013.

[37] A. Rodriguez, V. N. Boddeti, B. Vijaya Kumar, and A. Mahalanobis, "Maximum margin correlation filter: A new approach for localization and classification," *IEEE Trans. Image Process.*, vol. 22, no. 2, pp. 631–643, Feb. 2013.

[38] T. Zhou, D. Tao, and X. Wu, "NESVM: A fast gradient method for support vector machines," in *Proc. Int. Conf. Data Mining*, 2010, pp. 679–688.

[39] S. Oraintara, W. C. Karl, D. A. Castanon, and T. Q. Nguyen, "A method for choosing the regularization parameter in generalized Tikhonov regularized linear inverse problems," in *Proc. Int. Conf. Image Process.*, 2000, pp. 93–96.

[40] A. Mirzal, "A converged algorithm for Tikhonov regularized nonnegative matrix factorization with automatic regularization parameters determination," 2012.

[41] O. Bousquet and A. Elisseeff, "Stability and generalization," *J. Mach. Learn. Res.*, vol. 2, pp. 499–526, Mar. 2002.

[42] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of Machine Learning*. Cambridge, MA, USA: MIT Press, 2012.
[43] G. A. Watson, "Characterization of the subdifferential of some matrix norms," *Linear Algebra Appl.*, vol. 170, pp. 33–45, Jun. 1992.
[44] C. McDiarmid, "On the method of bounded differences," in *Proc. Surv. Combinat.*, 1989, pp. 148–188.
[45] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.
[46] J. Gui, Z. Sun, J. Cheng, S. Ji, and X. Wu, "How to estimate the regularization parameter for spectral regression discriminant analysis and its kernel version?" *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 2, pp. 211–223, 2014.

Yong Luo received the B.E. degree in computer science from Northwestern Polytechnical University, Xi'an, China, in 2009. He is currently pursuing the Ph.D. degree with the School of Electronics Engineering and Computer Science, Peking University, Beijing, China. He was a Visiting Student with the School of Computing, Nanyang Technological University, Singapore, and the Faculty of Engineering and Information Technology, University of Technology, Sydney, Ultimo, NSW, Australia. His research interests are primarily on machine learning with applications on image classification and annotation. He has authored several scientific articles at top venues, including the IEEE TRANSACTIONS ON IMAGE PROCESSING, the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, and the Association for the Advancement of Artificial Intelligence.

Tongliang Liu received the B.E. degree from the University of Science and Technology of China, Hefei, China, in 2012. He is currently pursuing the Ph.D. degree in computer science from the University of Technology, Sydney, Ultimo, NSW, Australia. His research interests include machine learning, computer vision, and optimization.

Dacheng Tao (M'07–SM'12) is a Professor of Computer Science with the Centre for Quantum Computation and Intelligent Systems and the Faculty of Engineering and Information Technology at the University of Technology, Sydney, Ultimo, NSW, Australia. He mainly applies statistics and mathematics for data analysis problems in data mining, computer vision, machine learning, multimedia, and video surveillance. He has authored and co-authored more than 100 scientific articles at top venues, including the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, the IEEE TRANSACTIONS ON IMAGE PROCESSING, the Neural Information Processing Systems Conference, the International Conference on Machine Learning, the International Conference on Artificial Intelligence and Statistics, the IEEE International Conference on Data Mining (ICDM), the IEEE Conference on Computer Vision and Pattern Recognition, the International Conference on Computer Vision, the European Conference on Computer Vision, the *ACM Transactions on Knowledge Discovery from Data*, the ACM Multimedia Conference, and the ACM Conference on Knowledge Discovery and Data Mining, and was a recipient of the Best Theory/Algorithm Paper Runner Up Award at the IEEE ICDM'07 and the Best Student Paper Award at the IEEE ICDM'13.

Chao Xu (M'02) received the B.E. degree from Tsinghua University, Beijing, China, in 1988, the M.S. degree from the University of Science and Technology of China, Hefei, China, in 1991, and the Ph.D. degree from the Institute of Electronics, Chinese Academy of Sciences, Beijing, in 1997. From 1991 to 1994, he was an Assistant Researcher with the University of Science and Technology of China. In 1997, he joined the Key Laboratory of Machine Perception (Ministry of Education), Peking University, Beijing, where he has been a Professor since 2005. His research interests are in image and video processing, and multimedia technology. He has authored and co-authored more than 100 publications, and holds six patents in these fields.