Discriminative Transfer Learning on Manifold

Zheng Fang* Zhongfei (Mark) Zhang †

Abstract

Collective matrix factorization has achieved a remarkable success in document classification in the literature of transfer learning. However, the learned latent factors still suffer from the divergence between different domains and thus are usually not discriminative for an appropriate assignment of category labels. Based on these observations, we impose a discriminative regression model over the latent factors to enhance the capability of label prediction. Moreover, we propose to minimize the Maximum Mean Discrepancy in the latent manifold subspace, as opposed to typically in the original data space, to bridge the gap between different domains. Specifically, we formulate these objectives into a joint optimization framework with two matrix tri-factorizations for the source and target domains simultaneously. An iterative algorithm DTLM is developed and the theoretical analysis of its convergence is discussed. Empirical study on benchmark datasets validates that DTLM improves the classification accuracy consistently compared with the state-of-the-art transfer learning methods.

1 Introduction

In real-world applications, we are often encountered with the situation where there is lack of labeled data for training in one domain while there are abundant labeled data in another domain. To deal with this situation, transfer learning has been proposed and is shown very effective for leveraging labeled data in the source domain to build an accurate classifier in the target domain. Many existing transfer learning methods explore common latent factors shared by both domains to reduce the distribution divergence and bridge the gap between different domains [3, 13, 17, 5]. Many of the transfer learning algorithms which are based on the collective matrix tri-factorization achieve a remarkable succuss in the recent literature [19, 21, 14, 12].

This paper focuses on the literature of collective matrix factorization based transfer learning. Though there is a significant success, the learned latent factors still suffer from the divergence between different domains and thus are usually not discriminative for an appropriate assignment of category labels. Specifically, there are several issues that the existing literature on transfer learning either fail to address appropriately or ignore completely.

First, in the literature, the learned latent factors serve two roles simultaneously. They represent the cluster structures as one role during the matrix factorization, and the category structures as another role through the supervised guidance of given labels during the classification. The cluster structures are determined by the original data whereas the category structures are determined by the concept summarization, typically supervised by the given labels. Since all the existing collective matrix factorization based transfer learning methods make the matrix factorization and the classification as two separate stages, a semantic gap exists between the two roles for the same latent factors, which is completely ignored in the literature. For examples, in image document classification, images of red balloons and red apples might be first mapped into the same latent factors based on the original color data through matrix factorization and then would have to be classified into different classes through the supervised learning with the given labels.

Second, since the matrix factorization and the classification are done separately, if the learned latent factors from the matrix factorization stage are wrong, it may be difficult to "correct" them back during the classification stage even with given labels, as these latent factors would be unable to be appropriately assigned correct category labels in the low dimensional manifold space. Figure (1) illustrates this issue, where the resulting latent factors obtained from the Graph coregularized Collective Matrix tri-Factorization (GCMF) [15] algorithm used to indicate the categories are shown in the 2D latent space,, together with the decision boundary of $argmax(\cdot)$ for categories. Clearly it is "too late" to assign some of the "circles" to a correct category label when they are already on the other side of the decision boundary. This issue is similar to the trivial solution and scale transfer problems [9] caused from the collective matrix factorization.

Third, in transfer learning, the distributions of the latent factors in source domain and target domain

*Dept. of Information Science and Electronic Engineering, Zhejiang University, fangzheng354@zju.edu.cn.

†Dept. of Information Science and Electronic Engineering, Zhejiang University, zhongfei@zju.edu.cn.

are largely divergent, making the latent factors in the target domain difficult to appropriately predict the correct category labels though the learning in the source domain.

To address these issues, we propose a domain transfer learning method which incorporates the discriminative regression model to bridge the gap between the two roles of the learned latent factors and minimizes the distribution divergence of the latent factors directly, as opposed to typically in the original data space, between the source and the target domains using Maximum Mean Discrepancy (MMD). Our objective is to minimize the regression empirical loss and the MMD measurement with respect to the latent factors which parameterize the embedded low dimensional manifold space in different domains simultaneously. Furthermore, we apply the graph Laplacian regularization to preserve the geometric structure in both source and target domains. Based on all these consideratoins, we develop a unified framework leading to an iterative algorithm called Discriminative Transfer Learning on Manifold (DTLM).

The remainder of this paper is organized as follows. In Section 2, we discuss the related work. Section 3 defines the symbol notations and presents the formulation of the proposed framework. The multiplicative iterative optimization solution is derived in Section 4. In Section 5, we provide a theoretical analysis of the DTLM convergence. The extensive experiments on benchmark datasets are reported in Section 6. Finally, Section 7 concludes the paper.

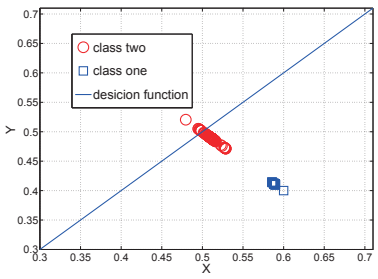


Figure 1: The latent factors learned from algorithm GCMF and the boundary of decision function $argmax(\cdot)$ to assign category labels

2 Related Work

In this section, we review several existing transfer learning methods that are related to our work. The existing methods of transfer learning can be summarized into four cases [16], transferring the knowledge of the instances [6], transferring the knowledge of the feature representations [17], transferring the knowledge of the parameters [8], and transferring the relational knowledge .

The collective matrix tri-factorization based methods [19, 21, 14, 12] can be categorized into the relation based transfering. Most of them share the associations between the word clusters and the document clusters across different domains. Moreover, Li et al. [11, 12] propose to share the information of the word clusters for the task of sentiment classification. However, Zhuang et al. [21] demonstrate that this assumption does not meet practical issues and propose a matrix tri-factorization based classification framework (MTrick) for cross domain transfer learning.

Recently, the most closely related literature to our algorithm is the efforts in [15] and [12]. Though Long et al. [15] propose GCMF to preserve the geometric structures of the datasets [2] in learning latent factors, the algorithm fails to incorporate the cross-domain supervision information for label predication. In [12], Li et al. introduce a linear prediction model over the latent factors. Nonetheless, the algorithm restricts the feature clusters to be the same across the different domains and fails to preserve the local geometric structures. Moreover, the collective matrix tri-factorizations in the source and target domains are two separate stages. Consequently, the two domains do not share the associations between the feature and instance clusters. To overcome both the weakness of these methods, we integrate the discriminative regression model in an unified latent factors learning framework. In order to eliminate the domain divergence, we minimize the MMD between the latent factor distributions in different domains whilst preserving the local geometric structures of data.

3 Notations and Problem Specification

In this section, we first introduce the basic concepts and mathematical notations used in this paper, and then formulate the framework.

3.1 Basic Concepts and Mathematical Notations We consider a source domain $\mathcal{D}_s$ and a target domain $\mathcal{D}_t$. The domain indices are $\mathcal{I} = \{s, t\}$. $\mathcal{D}_s$ and $\mathcal{D}_t$ share the same feature space and label space. There are m features and c classes. Let $\mathbf{X}_\pi = [\mathbf{x}^\pi_{\cdot 1}, \cdots, \mathbf{x}^\pi_{\cdot n_\pi}] \in \mathbb{R}^{m \times n_\pi}$, $\pi \in \mathcal{I}$, represent the feature-instance matrix of domain $\mathcal{D}_\pi$, where $\mathbf{x}^\pi_{\cdot i}$ is the ith instance in domain $\mathcal{D}_\pi$. Labels of the examples in the source domain $\mathcal{D}_\pi$ are given as $\mathbf{Y}_\pi \in \mathbb{R}^{c * n_\pi}$, where the element $y^\pi_{ij} = 1$ if $\mathbf{x}^\pi_{\cdot j}$ belongs to class i, and $y^\pi_{ij} = 0$ otherwise.

3.2 Unified Framework of Collective Matrix Factorization and Discriminative Regression Model We propose a domain transfer learning frame-

work based on the collective matrix tri-factorization which has been proven very effective in [15, 21, 19].

$$\min_{\mathbf{U}_\pi,\mathbf{H},\mathbf{V}_\pi \geq 0} \sum_{\pi \in \mathcal{I}} \|\mathbf{X}_\pi - \mathbf{U}_\pi \mathbf{H} \mathbf{V}_\pi\|^2$$

Conceptually, using the existing terminologies, $\mathbf{U}_\pi = [(\mathbf{u}^\pi_{1\cdot})^T, \cdots, (\mathbf{u}^\pi_{m\cdot})^T]^T \in \mathcal{R}^{m \times k_m}$ denotes the word cluster structures, where k_m is the number of the feature clusters. $\mathbf{V}_\pi = [\mathbf{v}^\pi_{\cdot 1}, \cdots, \mathbf{v}^\pi_{\cdot n_\pi}] \in \mathcal{R}^{k_n \times n_\pi}$ denotes the document cluster structures, where k_n is the number of the data instance clusters in domain $\mathcal{D}_\pi$. $\mathbf{H} \in \mathcal{R}^{k_m \times k_n}$ denotes the association between the word clusters and the document clusters which is shown to remain stable across different domains [21].

With the intuitive goal of discovering the intrinsic discriminative structures and looking for the clusters which are most linearly separable, we introduce a linear regression function for the classification on the latent factors $\mathbf{V}$ with the loss function $\|\mathbf{Y} - \mathbf{A}\mathbf{V}\|^2$, where matrix $\mathbf{A} \in \mathbb{R}^{c * k_n}$ is the regression coefficient matrix. Here we chose the least squares loss for optimization simplification. Considering that there are labeled data in the source or target domain for training, we also introduce the matrix $\mathbf{P}_\pi$ to indicate which data are used as the supervised information in the corresponding domain. $\mathbf{P}_\pi \in \mathcal{R}^{n_\pi \times n_\pi}$ is a diagonal matrix, where its element $P^\pi_{ii} = 1$ denotes the ith data instance in the corresponding domain used in the supervised training, and $P^\pi_{ii} = 0$ otherwise. The objective function of the unified framework is as follows, which combines the task of cross domain data co-clustering and the task of classification simultaneously.

$$\begin{aligned} &\min_{\mathbf{V}_\pi,\mathbf{U}_\pi,\mathbf{H},\mathbf{A}} \sum_{\pi \in \mathcal{I}} (\|\mathbf{X}_\pi - \mathbf{U}_\pi \mathbf{H} \mathbf{V}_\pi\|^2 \\ &+\beta \|\mathbf{Y}_\pi \mathbf{P}_\pi - \mathbf{A}\mathbf{V}_\pi \mathbf{P}_\pi\|^2) + \alpha \|\mathbf{A}\|^2 \end{aligned} \tag{3.1}$$

where β, α, λ are the trade-off regularization parameters. $\alpha\|\mathbf{A}\|^2$ is introduced to avoid the overfitting of the regression classification.

3.3 Maximum Mean Discrepancy

To transfer cross domain knowledge, we need to bridge the gap between $\mathcal{D}_s$ and $\mathcal{D}_t$. To this end, we employ a criterion based on Maximum Mean Discrepancy (MMD) [17, 1]. The empirical estimate of the distance between domains $\mathcal{D}_s$ and $\mathcal{D}_t$ defined by MMD is as follows.

$$Dist(\mathcal{D}_s, \mathcal{D}_t) = \| \frac{1}{|\mathcal{D}_s|} \sum_{\mathbf{x}_i \in \mathcal{D}_s} \phi(\mathbf{x}_i) - \frac{1}{|\mathcal{D}_t|} \sum_{\mathbf{x}_j \in \mathcal{D}_t} \phi(\mathbf{x}_j) \|^2 \tag{3.2}$$

where $|\cdot|$ denotes the size of a dataset in the corresponding domain. In our case, the function $\phi(\cdot)$ maps the original data, $\mathbf{x}_i \in \mathcal{D}_s$, $\mathbf{x}_j \in \mathcal{D}_t$, from different domains to the corresponding low dimensional manifold representations, $\mathbf{v}_i$, $\mathbf{v}_j$. That is $\phi(\mathbf{x}^\pi_{\cdot i}) = \mathbf{v}^\pi_{\cdot i}$, $i = 1, \cdots, |\mathcal{D}_\pi|$. The distance in Eq.(3.2) in our case is

$$Dist^v(\mathcal{D}_s, \mathcal{D}_t) = \| \frac{1}{n_s} \sum_{i=1}^{n_s} \mathbf{v}^s_{\cdot i} - \frac{1}{n_t} \sum_{j=1}^{n_t} \mathbf{v}^t_{\cdot j} \|^2 \tag{3.3}$$

Similarly, the distance based on MMD criterion for different domains in the feature space is

$$Dist^u(\mathcal{D}_s, \mathcal{D}_t) = \| \frac{1}{n_s} \sum_{i=1}^{n_s} \mathbf{u}^s_{i\cdot} - \frac{1}{n_t} \sum_{j=1}^{n_t} \mathbf{u}^t_{j\cdot} \|^2 \tag{3.4}$$

Bridging the gap between different domains now becomes minimizing the distances defined in Eqs.(3.3)(3.4) in the latent factor space, as opposed to typically in the original data space.

3.4 Data Manifold Geometric Regularization

From a manifold geometric perspective, the data points may be sampled from a distribution supported by a low-dimensional manifold embedded in a high dimensional space. Studies on spectral graph theory [4] and manifold learning theory have demonstrated that the local geometric structures can be effectively modeled through a nearest neighbor graph on a scatter of data points. Consider a data instance graph G^v_π with n_π vertices where each vertex corresponds to a data instance in domain $\mathcal{D}_\pi$. Define the edge weight matrix $\mathbf{W}^v_\pi$ as follows:

$$(W^v_\pi)_{ij} = \begin{cases} \cos(\mathbf{x}^\pi_{\cdot i}, \mathbf{x}^\pi_{\cdot j}) & if, \mathbf{x}^\pi_{\cdot i} \in N_p(\mathbf{x}^\pi_{\cdot j}) \quad or, \mathbf{x}^\pi_{\cdot j} \in N_p(\mathbf{x}^\pi_{\cdot i}) \\ 0 & otherwise \end{cases} \tag{3.5}$$

where $N_p(\mathbf{x}_{\cdot i})$ denotes the set of p nearest neighbors of $\mathbf{x}_{\cdot i}$. The data instance graph regularizer $\mathcal{R}^v_\pi$ used to measure the smoothness of the mapping function along the geodesics in the intrinsic geometry of the dataset is as follows .

$$\begin{aligned} \mathcal{R}^v_\pi &= \frac{1}{2} \sum_{ij} \|\mathbf{v}^\pi_{\cdot i} - \mathbf{v}^\pi_{\cdot j}\|^2 (\mathbf{W}^v_\pi)_{ij} \\ &= \sum_i tr(\mathbf{v}^\pi_{\cdot i} (\mathbf{v}^\pi_{\cdot i})^T) \mathbf{D}^v_{ii} - \sum_{ij} tr(\mathbf{v}^\pi_{\cdot i} (\mathbf{v}^\pi_{\cdot j})^T) \mathbf{W}^v_{ij} \\ &= tr(\mathbf{V}_\pi (\mathbf{D}^v_\pi - \mathbf{W}^v_\pi) \mathbf{V}^T_\pi) \end{aligned} \tag{3.6}$$

where $\mathbf{D}^v_\pi = diag(\sum_i (\mathbf{W}^v_\pi)_{ij})$. By minimizing $\mathcal{R}^v_\pi$ we get the low dimensional representations for the instances on the manifold, which preserve the intrinsic geometry of the data distribution.

Similarly, we also construct a feature graph G^u_π with m vertices where each vertex corresponds to a feature in domain $\mathcal{D}_\pi$. The edge weight matrix $\mathbf{W}^u_\pi$ of it is as follows:

$$(W^u_\pi)_{ij} = \begin{cases} \cos(\mathbf{x}^\pi_{i\cdot}, \mathbf{x}^\pi_{j\cdot}) & if, \mathbf{x}^\pi_{i\cdot} \in N_p(\mathbf{x}^\pi_{j\cdot}) \quad or, \mathbf{x}^\pi_{j\cdot} \in N_p(\mathbf{x}^\pi_{i\cdot}) \\ 0 & otherwise \end{cases} \tag{3.7}$$

Preserving the feature geometric structure in domain $\mathcal{D}_\pi$ requires minimizing the feature graph regularizer

$$\begin{aligned}\mathcal{R}_\pi^u &= \frac{1}{2}\sum_{ij}\|\mathbf{u}_{i\cdot}^\pi - \mathbf{u}_{j\cdot}^\pi\|^2(\mathbf{W}_\pi^u)_{ij} \\ &= \sum_i tr((\mathbf{u}_{i\cdot}^\pi)^T(\mathbf{u}_{i\cdot}^\pi))\mathbf{D}_{ii}^u - \sum_{ij} tr((\mathbf{u}_{i\cdot}^\pi)^T(\mathbf{u}_{j\cdot}^\pi))\mathbf{W}_{ij}^u \\ &= tr(\mathbf{U}_\pi^T(\mathbf{D}_\pi^u - \mathbf{W}_\pi^u)\mathbf{U}_\pi) \end{aligned} \tag{3.8}$$

where $\mathbf{D}_\pi^u = diag(\sum_i(\mathbf{W}_\pi^u)_{ij})$

3.5 Discriminative Transfer Learning on Manifold

Finally, we combine the optimization problems Eqs.(3.1-3.8) into a joint optimization objective to minimize. This allows us to reach the optimization problem of DTLM as defined in Equation (3.9).

$$\begin{aligned}\min_{\mathbf{V}_s,\mathbf{V}_t,\mathbf{U}_s,\mathbf{U}_t,\mathbf{H},\mathbf{A}} &\sum_{\pi\in\mathcal{I}}(\|\mathbf{X}_\pi - \mathbf{U}_\pi\mathbf{H}\mathbf{V}_\pi\|^2 \\ &+ \beta\|\mathbf{Y}_\pi\mathbf{P}_\pi - \mathbf{A}\mathbf{V}_\pi\mathbf{P}_\pi\|^2) + \alpha\|\mathbf{A}\|^2 \\ &+ \sum_{\pi\in\mathcal{I}}\lambda(\mathcal{R}_\pi^u + \mathcal{R}_\pi^v) + \|\frac{1}{m_s}\mathbf{1}_{m_s}^T\mathbf{U}_s - \frac{1}{m_t}\mathbf{1}_{m_t}^T\mathbf{U}_t\|^2 \\ &+ \|\frac{1}{n_s}\mathbf{V}_s\mathbf{1}_{n_s} - \frac{1}{n_t}\mathbf{V}_t\mathbf{1}_{n_t}\|^2 \\ s.t.\quad &\mathbf{V}_s,\mathbf{V}_t,\mathbf{U}_s,\mathbf{U}_t,\mathbf{H} \geq 0\end{aligned} \tag{3.9}$$

4 Solution to the Optimization Problem

Due to the space limitation and for simplicity, we consider computing the variables in domain $\mathcal{D}_\pi$ and introduce subscript $\bar{\pi}$ for the variables in the counterpart domain of π.

4.1 Computation of $\mathbf{V}_\pi$

Optimizing Eq.(3.9) with respect to $\mathbf{V}_\pi$ is equivalent to optimizing

$$\begin{aligned}\min_{\mathbf{V}_\pi} &\|\mathbf{X}_\pi - \mathbf{U}_\pi\mathbf{H}\mathbf{V}_\pi\|^2 + \beta\|\mathbf{Y}_\pi\mathbf{P}_\pi - \mathbf{A}\mathbf{V}_\pi\mathbf{P}_\pi\|^2 + \\ &\lambda tr(\mathbf{V}_\pi\mathbf{L}_\pi^v\mathbf{V}_\pi^T) + \|\frac{1}{n_\pi}\mathbf{V}_\pi\mathbf{1}_{n_\pi} - \frac{1}{n_{\bar{\pi}}}\mathbf{V}_{\bar{\pi}}\mathbf{1}_{n_{\bar{\pi}}}\|^2 \\ s.t.\quad &\mathbf{V}_\pi \geq 0, \qquad where \quad \mathbf{L}_\pi^v = \mathbf{D}_\pi^u - \mathbf{W}_\pi^u\end{aligned} \tag{4.10}$$

For the constraint $\mathbf{V}_\pi \geq 0$, we present an iterative multiplicative updating solution. We introduce the Lagrangian multiplier $\mathbf{\Phi} \in \mathbb{R}^{c\times n_\pi}$. Thus, the Lagrangian function is

$$\begin{aligned}L(\mathbf{V}_\pi) = &\|\mathbf{X}_\pi - \mathbf{U}_\pi\mathbf{H}\mathbf{V}_\pi\|^2 + \beta\|\mathbf{Y}_\pi\mathbf{P}_\pi - \mathbf{A}\mathbf{V}_\pi\mathbf{P}_\pi\|^2 + \\ &\lambda tr(\mathbf{V}_\pi\mathbf{L}_\pi^v\mathbf{V}_\pi^T) + \|\frac{1}{n_\pi}\mathbf{V}_\pi\mathbf{1}_{n_\pi} - \frac{1}{n_{\bar{\pi}}}\mathbf{V}_{\bar{\pi}}\mathbf{1}_{n_{\bar{\pi}}}\|^2 + tr(\mathbf{\Phi}\mathbf{V}_\pi^T)\end{aligned}$$

Setting $\frac{\partial L(\mathbf{V}_\pi)}{\partial \mathbf{V}_\pi} = 0$, we obtain

$$\begin{aligned}\mathbf{\Phi} = &2(\mathbf{U}_\pi\mathbf{H})^T\mathbf{X}_\pi - 2(\mathbf{U}_\pi\mathbf{H})^T\mathbf{U}_\pi\mathbf{H}\mathbf{V}_\pi + 2\beta\mathbf{B}_\pi - 2\beta\mathbf{E}_\pi \\ &+2\lambda\mathbf{V}_\pi\mathbf{L}_\pi^v + 2\frac{\mathbf{V}_{\bar{\pi}}\mathbf{1}_{\bar{\pi}}\mathbf{1}_\pi^T}{n_\pi n_{\bar{\pi}}} - 2\frac{\mathbf{V}_\pi\mathbf{1}_\pi\mathbf{1}_\pi^T}{n_\pi^2}\end{aligned} \tag{4.11}$$

where $\mathbf{B}_\pi = \mathbf{A}^T\mathbf{Y}_\pi\mathbf{P}_\pi\mathbf{P}_\pi^T$ and $\mathbf{E}_\pi = \mathbf{A}^T\mathbf{A}\mathbf{V}_\pi\mathbf{P}_\pi\mathbf{P}_\pi^T$. Using the Karush-Kuhn-Tucker condition $\mathbf{\Phi}_{ij}(\mathbf{V}_\pi)_{ij} = 0$, we get

$$\begin{aligned}&[(\mathbf{U}_\pi\mathbf{H})^T\mathbf{X}_\pi - (\mathbf{U}_\pi\mathbf{H})^T\mathbf{U}_\pi\mathbf{H}\mathbf{V}_\pi + \beta\mathbf{B}_\pi - \beta\mathbf{E}_\pi \\ &+\lambda\mathbf{V}_\pi\mathbf{L}_\pi^v + \frac{\mathbf{V}_{\bar{\pi}}\mathbf{1}_{\bar{\pi}}\mathbf{1}_\pi^T}{n_\pi n_{\bar{\pi}}} - \frac{\mathbf{V}_\pi\mathbf{1}_\pi\mathbf{1}_\pi^T}{n_\pi^2}]_{ij}(\mathbf{V}_\pi)_{ij} = 0\end{aligned} \tag{4.12}$$

By introducing $\mathbf{B}_\pi = \mathbf{B}_\pi^+ - \mathbf{B}_\pi^-$, where $\mathbf{B}_\pi^+ = (|(\mathbf{B}_\pi)_{ij}| + (\mathbf{B}_\pi)_{ij})/2$ and $\mathbf{B}_\pi^- = (|(\mathbf{B}_\pi)_{ij}| - (\mathbf{B}_\pi)_{ij})/2$, $\mathbf{E}_\pi = \mathbf{E}_\pi^+ - \mathbf{E}_\pi^-$, where $\mathbf{E}_\pi^+ = \mathbf{R}^+\mathbf{V}_\pi\mathbf{P}_\pi\mathbf{P}_\pi^T$, $\mathbf{E}_\pi^- = \mathbf{R}^-\mathbf{V}_\pi\mathbf{P}_\pi\mathbf{P}_\pi^T$, $\mathbf{R} = \mathbf{A}^T\mathbf{A}$, $\mathbf{R}^+ = (|(\mathbf{R})_{ij}| + (\mathbf{R})_{ij})/2$ and $\mathbf{R}^- = (|(\mathbf{R})_{ij}| - (\mathbf{R})_{ij})/2$, we obtain

$$\begin{aligned}&[(\mathbf{U}_\pi\mathbf{H})^T\mathbf{X}_\pi - (\mathbf{U}_\pi\mathbf{H})^T\mathbf{U}_\pi\mathbf{H}\mathbf{V}_\pi + \beta(\mathbf{B}_\pi^+ - \mathbf{B}_\pi^-) \\ &-\beta(\mathbf{E}_\pi^+ - \mathbf{E}_\pi^-) + \lambda\mathbf{V}_\pi(\mathbf{D}_\pi^v - \mathbf{W}_\pi^v) \\ &+\frac{\mathbf{V}_{\bar{\pi}}\mathbf{1}_{\bar{\pi}}\mathbf{1}_\pi^T}{n_\pi n_{\bar{\pi}}} - \frac{\mathbf{V}_\pi\mathbf{1}_\pi\mathbf{1}_\pi^T}{n_\pi^2}]_{ij}(\mathbf{V}_\pi)_{ij} = 0\end{aligned} \tag{4.13}$$

Eq.(4.13) leads to the following updating formula

$$\mathbf{V}_\pi = \mathbf{V}_\pi \odot \sqrt{\frac{\beta(\mathbf{B}_\pi^+ + \mathbf{E}_\pi^-) + \lambda\mathbf{V}_\pi\mathbf{W}_\pi^v + (\mathbf{U}_\pi\mathbf{H})^T\mathbf{X}_\pi + \frac{\mathbf{V}_{\bar{\pi}}\mathbf{1}_{n_{\bar{\pi}}}\mathbf{1}_{n_\pi}^T}{n_\pi n_{\bar{\pi}}}}{\beta(\mathbf{B}_\pi^- + \mathbf{E}_\pi^+) + \lambda\mathbf{V}_\pi\mathbf{D}_\pi^v + (\mathbf{U}_\pi\mathbf{H})^T\mathbf{U}_\pi\mathbf{H}\mathbf{V}_\pi + \frac{\mathbf{V}_\pi\mathbf{1}_{n_\pi}\mathbf{1}_{n_\pi}^T}{n_\pi^2}}} \tag{4.14}$$

4.2 Computation of $\mathbf{U}_\pi$, $\mathbf{H}$

Computation of $\mathbf{U}_\pi$, $\mathbf{H}$ is very similar to the computation of $\mathbf{V}_\pi$. Due to the limited space, we omit the derivation and present the updating formulas directly.

For $\mathbf{U}_\pi$ in domain π, the updating rule is as follows.

$$\mathbf{U}_\pi = \mathbf{U}_\pi \odot \sqrt{\frac{\mathbf{X}_\pi(\mathbf{H}\mathbf{V}_\pi)^T + \lambda\mathbf{W}_\pi^u\mathbf{U}_\pi + \frac{\mathbf{1}_{m_\pi}\mathbf{1}_{m_{\bar{\pi}}}^T\mathbf{U}_{\bar{\pi}}}{m_\pi m_{\bar{\pi}}}}{\mathbf{U}_\pi\mathbf{H}\mathbf{V}_\pi(\mathbf{H}\mathbf{V}_\pi)^T + \lambda\mathbf{D}_\pi^u\mathbf{U}_\pi + \frac{\mathbf{1}_{m_\pi}\mathbf{1}_{m_\pi}^T\mathbf{U}_\pi}{m_\pi^2}}} \tag{4.15}$$

The updating formula of $\mathbf{H}$ is as follows.

$$\mathbf{H} = \mathbf{H} \odot \sqrt{\frac{\sum_{\pi\in\mathcal{I}}\mathbf{U}_\pi^T\mathbf{X}_\pi\mathbf{V}_\pi^T}{\sum_{\pi\in\mathcal{I}}\mathbf{U}_\pi^T\mathbf{U}_\pi\mathbf{H}\mathbf{V}_\pi\mathbf{V}_\pi^T}} \tag{4.16}$$

4.3 Computation of $\mathbf{A}$

Fixing $\mathbf{U}_\pi$, $\mathbf{V}_\pi(\pi \in \mathcal{I})$, and $\mathbf{H}$, the problem in Eq.(3.9) reduces to the ridge regression problem as follows, which has a closed form solution.

$$\min_{\mathbf{A}} \sum_{\pi\in\mathcal{I}} \beta\|\mathbf{Y}_\pi\mathbf{P}_\pi - \mathbf{A}\mathbf{V}_\pi\mathbf{P}_\pi\|^2 + \alpha\|\mathbf{A}\|^2 \tag{4.17}$$

Let $J(\mathbf{A})$ denote the objective function. Taking the first order derivative of $J(\mathbf{A})$ with respect to $\mathbf{A}$ and requiring

it to be zero, we have

$$\frac{\partial J(\mathbf{A})}{\partial \mathbf{A}} = 2\beta\Big(\sum_{\pi\in\mathcal{I}} \mathbf{Y}_\pi \mathbf{P}_\pi (\mathbf{V}_\pi \mathbf{P}_\pi)^T + \mathbf{A}\sum_{\pi\in\mathcal{I}} \mathbf{V}_\pi \mathbf{P}_\pi (\mathbf{V}_\pi \mathbf{P}_\pi)^T\Big) + 2\alpha\mathbf{A} = 0 \tag{4.18}$$

which leads to the following updating formula

$$\mathbf{A} = \Big(\sum_{\pi\in\mathcal{I}} \mathbf{Y}_\pi \mathbf{P}_\pi (\mathbf{V}_\pi \mathbf{P}_\pi)^T\Big)\Big(\sum_{\pi\in\mathcal{I}} \mathbf{V}_\pi \mathbf{P}_\pi (\mathbf{V}_\pi \mathbf{P}_\pi)^T + \gamma\mathbf{I}\Big)^{-1} \tag{4.19}$$

where $\gamma = \frac{\alpha}{\beta}$. In summary, we present the iterative multiplicative updating algorithm of DTLM in Algorithm 1. To make the optimization well-defined, we normalize each row of $\mathbf{U}_\pi$ and each column of $\mathbf{V}_\pi$ after every iteration by l_1 norm as done in [21, 15].

Algorithm 1: The Discriminative Transfer Learning on Manifold (DTLM) Algorithm

Input: data matrices $\{\mathbf{X}\}_{\pi\in\mathcal{I}}$, label information matrix $\{\mathbf{Y}\}_{\pi\in\mathcal{I}}$, parameters α, β, λ, and p.
Output: classification results $\widetilde{\mathbf{Y}}_t$ on unlabeled data in the target domain.

Construct graphs G_π^v and G_π^u using Eq.(3.5) and Eq.(3.7). Initialize $\{\mathbf{U}\}_{\pi\in\mathcal{I}}$, $\mathbf{V}_s$, $\mathbf{H}$ following [15], and initialize $\mathbf{V}_t$ by a random positive matrix;
while $iter \le maxIter$ **do**
  Update $\{\mathbf{U}\}_{\pi\in\mathcal{I}}$ using Eq.(4.15).
  Update $\{\mathbf{V}\}_{\pi\in\mathcal{I}}$ using Eq.(4.14).
  Update $\mathbf{H}$ using Eq.(4.16).
  Update $\mathbf{A}$ using Eq.(4.19).
  Normalize each row of $\{\mathbf{U}_\pi\}_{\pi\in\mathcal{I}}$ and each column of $\{\mathbf{V}_\pi\}_{\pi\in\mathcal{I}}$ by l_1 norm.
  $iter := iter + 1$;
Predict labels for the unlabeled data in target domain using $\widetilde{\mathbf{Y}}_t = \mathbf{A}\mathbf{V}_t$;

5 Convergence Analysis

In this section, we investigate the convergence of Algorithm 1. We use the auxiliary function approach [18] to prove the convergence of the algorithm. Here we first introduce the definition of auxiliary function [18].

DEFINITION 5.1. *[18] $Z(h, h')$ is an auxiliary function for $J(h)$ if the conditions*

$$Z(h, h') \ge J(h), \qquad Z(h, h) = J(h)$$

are satisfied.

LEMMA 5.1. *[18] If Z is an auxiliary function for J, then J is non-increasing under the updating rule*

$$h^{(t+1)} = \arg\min_h Z(h, h^{(t)})$$

LEMMA 5.2. *[7] For any nonnegative matrices $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{B} \in \mathbb{R}^{k\times k}$, $\mathbf{S} \in \mathbb{R}^{n\times k}$, $\mathbf{S}' \in \mathbb{R}^{n\times k}$, and $\mathbf{A}$, $\mathbf{B}$ are symmetric, the following inequality holds*

$$\sum_{i=1}^{n}\sum_{j=1}^{k} \frac{(\mathbf{AS}'\mathbf{B})_{ij}\mathbf{S}_{ij}^2}{\mathbf{S}'_{ij}} \ge tr(\mathbf{S}^T\mathbf{ASB})$$

LEMMA 5.3. *Denote the sum of all the terms in objective function (3.9) that contain $\mathbf{V}_\pi$ as*

$$\begin{aligned} J(\mathbf{V}_\pi) = {} & tr(\mathbf{V}_\pi^T(\mathbf{U}_\pi\mathbf{H})^T\mathbf{U}_\pi\mathbf{H}\mathbf{V}_\pi - 2\mathbf{X}_\pi^T\mathbf{U}_\pi\mathbf{H}\mathbf{V}_\pi) \\ & + \beta tr(2\mathbf{V}_\pi^T\mathbf{B}^- - 2\mathbf{V}_\pi^T\mathbf{B}^+) + \beta tr(\mathbf{K}\mathbf{V}_\pi^T\mathbf{R}^+\mathbf{V}_\pi) \\ & - \beta tr(\mathbf{K}_\pi\mathbf{V}_\pi^T\mathbf{R}^-\mathbf{V}_\pi) + \lambda tr(\mathbf{V}_\pi\mathbf{D}_\pi^v\mathbf{V}_\pi^T - \mathbf{V}_\pi\mathbf{W}_\pi^v\mathbf{V}_\pi^T) \\ & + \frac{1}{n_\pi^2}\mathbf{1}_\pi^T\mathbf{V}_\pi^T\mathbf{V}_\pi\mathbf{1}_\pi - \frac{2}{n_\pi n_{\bar\pi}}\mathbf{1}_\pi^T\mathbf{V}_\pi^T\mathbf{V}_{\bar\pi}\mathbf{1}_{\bar\pi} \end{aligned} \tag{5.20}$$

where $\mathbf{K}_\pi = \mathbf{P}_\pi\mathbf{P}_\pi^T$ and $\mathbf{R} = \mathbf{A}^T\mathbf{A}$. $\mathbf{R}^+ = (|\mathbf{R}| + \mathbf{R})/2$, $\mathbf{R}^- = (|\mathbf{R}| - \mathbf{R})/2$.

The following function

$$\begin{aligned} Z(\mathbf{V}_\pi, \mathbf{V}'_\pi) = {} & \sum_{ij} \frac{((\mathbf{U}_\pi\mathbf{H})^T\mathbf{U}_\pi\mathbf{H}\mathbf{V}'_\pi)_{ij}(\mathbf{V}_\pi)_{ij}^2}{(\mathbf{V}'_\pi)_{ij}} \\ & - 2\sum_{ij}((\mathbf{U}_\pi\mathbf{H})^T\mathbf{X}_\pi)_{ij}(\mathbf{V}'_\pi)_{ij}\Big(1 + \log\frac{(\mathbf{V}_\pi)_{ij}}{(\mathbf{V}'_\pi)_{ij}}\Big) \\ & - 2\beta\Big(\sum_{ij}\mathbf{B}_{ij}^+(\mathbf{V}'_\pi)_{ij}\Big(1 + \log\frac{(\mathbf{V}_\pi)_{ij}}{(\mathbf{V}'_\pi)_{ij}}\Big) - \sum_{ij}\mathbf{B}_{ij}^-\frac{(\mathbf{V}_\pi)_{ij}^2 + (\mathbf{V}'_\pi)_{ij}^2}{2(\mathbf{V}'_\pi)_{ij}}\Big) \\ & + \beta\sum_{ij}\frac{(\mathbf{R}^+\mathbf{V}'_\pi\mathbf{K}_\pi)_{ij}(\mathbf{V}_\pi)_{ij}^2}{(\mathbf{V}'_\pi)_{ij}} \\ & - \beta\sum_{ijyz}(\mathbf{K}_\pi)_{jy}\mathbf{R}_{zi}^-(\mathbf{V}'_\pi)_{ij}(\mathbf{V}'_\pi)_{zy}\Big(1 + \log\frac{(\mathbf{V}_\pi)_{ij}(\mathbf{V}_\pi)_{zy}}{(\mathbf{V}'_\pi)_{ij}(\mathbf{V}'_\pi)_{zy}}\Big) \\ & + \lambda\sum_{ij}\frac{(\mathbf{V}'_\pi\mathbf{D}_\pi^v)_{ij}(\mathbf{V}_\pi)_{ij}^2}{(\mathbf{V}'_\pi)_{ij}} \\ & - \lambda\sum_{ijz}\mathbf{W}_{\pi jz}^{\mathbf{v}}(\mathbf{V}'_\pi)_{ij}(\mathbf{V}'_\pi)_{iz}\Big(1 + \log\frac{(\mathbf{V}_\pi)_{ij}(\mathbf{V}_\pi)_{iz}}{(\mathbf{V}'_\pi)_{ij}(\mathbf{V}'_\pi)_{iz}}\Big) \\ & + \frac{1}{n_\pi^2}\sum_{ij}\frac{(\mathbf{V}'_\pi\mathbf{1}_\pi\mathbf{1}_\pi^T)_{ij}(\mathbf{V}_\pi)_{ij}^2}{(\mathbf{V}'_\pi)_{ij}} \\ & - \frac{2}{n_\pi n_{\bar\pi}}\sum_{ij}(\mathbf{V}_{\bar\pi}\mathbf{1}_\pi\mathbf{1}_{\bar\pi}^T)_{ij}(\mathbf{V}'_\pi)_{ij}\Big(1 + \log\frac{(\mathbf{V}_\pi)_{ij}}{(\mathbf{V}'_\pi)_{ij}}\Big) \end{aligned}$$

is an auxiliary function for $J(\mathbf{V}_\pi)$. Furthermore, it is a convex function with respect to $\mathbf{V}_\pi$ and has a global minimum with $\mathbf{V}_\pi$ in the representation of Eq.(4.14).

THEOREM 5.1. *Updating $\mathbf{V}_\pi$ using Eq.(4.14) monotonically decreases the value of the objective in Eq.(3.9). Hence, Algorithm 1 converges.*

The detailed proofs of Lemma 5.3 and Theorem 5.1 are given in the supplementary file. Moreover, the

convergence analysis of the updating rules of $\mathbf{U}_\pi$ and $\mathbf{H}$ is similar to that of $\mathbf{V}_\pi$ by Lemma 5.1 and Lemma 5.3 and we omit the details here. The convergence of the updating rules of $\mathbf{A}$ is obvious from the optimization objective of Eq.(4.17). Consequently, the convergence of Algorithm 1 is achieved.

6 Complexity Analysis

Here we analyze the computation complexity briefly regrading with the space limitation. We count the arithmetic multiplication operations for each iteration. For updating the $\mathbf{V}_\pi$ of both domain in 4.14, the computational complexity is $O(3k_n(n_s^2 + n_t^2) + k_n k_m m(n_s + n_t) + k_n^2 k_m^2 m(n_s + n_t) + 2k_n n_s n_t + k_n(n_s + n_t))$. For updating the $\mathbf{U}_\pi$ of both domain in formula 4.15, the computational complexity is $O(8m^2 k_m + m(n_s + n_t)k_n k_m + m(n_s + n_t)k_n^2 k_m^2 + 2mk_m)$. For updating the $\mathbf{H}$ in 4.16, the computational complexity is $O(k_m k_n + k_m k_n m(n_s + n_t) + k_m^2 k_n^2 m(n_s + n_t))$. For updating the $\mathbf{A}$ in 4.19, the computational complexity is $O(k_n^3 + ck_n(n_s + n_t) + k_n^2(n_s + n_t))$. The total computational complexity of the DTLM algorithm is $O(k_n^2 k_m^2 m(n_s + n_t)p)$, where p is the iteration number.

7 Experiment

In this section, we demonstrate the promise of DTLM by conducting experiments on datasets generated from two benchmark data collections and compare the performance of DTLM with those of several state-of-the-art semi-supervised, and transfer learning methods.

7.1 Dataset We use the 20-Newsgroups corpus to conduct experiments on document classification. This corpus consists of approximately 20,000 news articles harvested from 20 different newsgroups. Each newsgroup corresponds to a different topic. Some of the newsgroups are closely related and can be grouped into one category at a top level, while others remain as separate categories. There are four top level categories used for class label, i.e. *comp*, *rec*, *sci*, and *talk*. Each of them has subcategories. For an example, under *sci* category there are four subcategories *sci.crypt*, *sci.electronics*, *sci.med*, and *sci.space*. We split each top category into two different groups as listed in Table 1. To construct a domain dataset, we randomly select two out of the four top categories, A and B, as positive class and negative class, respectively. The subcategory groups of A and B are $A1$, $A2$ and $B1$, $B2$. We merge $A1$ and $B1$ as the source domain data and merge $A2$ and $B2$ as the target domain data. This ensures that the two domains' data are related, but at the same time the domains are different because they are drawn from different subcategories. Such a preprocessing is a common practice for data preparation in transfer learning [20]. Consequently, we generate six domain datasets for binary classification in the transfer learning setting as in [15], i.e., *comp* vs *rec*, *comp* vs *sci*, *comp* vs *talk*, *rec* vs *sci*, *rec* vs *talk*, and *sci* vs *talk*.

To further validate our algorithm, we also perform experiments on the dataset Reuters-21578, which has a hierarchical structure and contains five top level categories. We evaluate DTLM on three classification tasks with the data collection constructed by Gao et al. [8], which contains three cross-domain datasets *orgs* vs *people*, *orgs* vs *place*, and *people* vs *place*.

7.2 Evaluation Metric In this paper, we employ the metric accuracy for comparing different algorithms by considering the binary classifications. Assume that Y is the function which maps from document d to its true class label $y = Y(d)$, and F the function which maps from document d to its prediction label $\widetilde{y} = F(d)$ by a classifier. The accuracy is defined as: $Accuracy = \frac{|\{d|d \in \mathcal{D}_t \wedge F(d) = Y(d)\}|}{|\mathcal{D}_t|}$.

7.3 Comparison Methods To verify the effectiveness of DTLM, we compare it with the state-of-the-art transfer learning methods Matrix Tri-factorization based Classification (MTrick) [21], Dual Knowledge Transfer (DKT) [19], and Graph co-regularized Collective Matrix tri-Factorization (GCMF) [15]. Support Vector Machine (SVM) and Semi-supervised learning method Transductive Support Vector Machine (TSVM) are also introduced in the comparison experiments.

7.4 Implementation Details TSVM and SVM are implemented by SVM^{light} [10] with the corresponding default parameters. For Mtrick, DKT, and GCMF, the parameters and initializations of these algorithms follow the settings of the experiments in the literature respectively.

In DTLM, the number of the data instance clusters in the source and target domains k_n is set as 2 to meet the number of the classes. The weight coefficients for the regression items, β and α, are both set as default value 10. We abbreviate the number of the feature clusters k_m as k with varying values 2, 4, 8, 16, 32, 64, 100. Similarly, we evaluate the trade-off regularization parameters λ in the values of $\{0.001, 0.005, 0.01, 0.05, 0.1, 1, 10, 50, 100, 250, 500, 1000\}$ for the parameter sensitivity analysis. In the comparison experiments with other methods, we use the parameter settings $\lambda = 0.1$, $k = 100$ for 20-Newsgroups datasets and $\lambda = 1$, $k = 100$ for Reuters-21578 datasets. $\mathbf{U}_s$, $\mathbf{U}_t$, and $\mathbf{H}$ are initialized as the random positive matrices. $\mathbf{V}_s$ is initialized by $\mathbf{Y}_s$ and $\mathbf{V}_t$ is is initialized

Table 1: Top categories and their groups. Each top category is partitioned into two groups 1 and 2.

Categories	Subcategories
comp	(1): comp.graphics, comp.os.ms-windows.misc (2): comp.sys.ibm.pc.hardware, comp.sys.mac.hardware
rec	(1): rec.autos, rec.motorcycles (2): rec.sport.baseball, rec.sport.hokey
sci	(1): sci.crypt, sci.electronics (2): sci.med, sci.space
talk	(1): talk.politics.guns, talk.politics.mideast (2): talk.politics.misc, talk.religion.misc

as the predicted results of Logistic Regression, which is trained based on the source domain data. We set the iteration number $maxIter$ as 100 for 20-Newsgroups and 210 for Reuters-21578.

7.5 Experimental Results and Discussion We perform all the six methods ten times for each case and the performance results are averaged over the ten times reported in Table 2. Since most of the comparison methods are unsupervised in the target domain, we use the target domain unsupervised version of DTLM for a fair comparison and set $\mathbf{P}_t = \mathbf{0}$.

From Table 2, we see that all the transfer learning methods perform better than non-transfer learning methods. Even the semi-supervised learning method TSVM cannot deliver a good performance as well as the transfer learning methods. This validates the fact that the transfer learning methods exploit the shared information between different domains and enhance the classification capability. Moreover, we see that DTLM performs the best of all the transfer learning methods. Though the transfer learning methods MTrick and DKT work better than the non-transfer learning methods, they fail to explore the geometric structures underlying the data manifold and cannot reach the best performance. This is consistent with the discussion in the literature [15]. For GCMF, though it adopts the geometric regularization to obtain an enhancement in data clustering, it still fails to address the divergence between the cluster structures and the categories of the labels. Superior to the other transfer learning methods, DTLM not only takes into account the intrinsic character of the data structures, but also incorporates the power of the discriminative regression model to correctly predict the category labels. Furthermore, the imposed MMD regularization constraint minimizes the gap between the latent factor distributions in different domains. GCMF is a special case of DTLM when parameters $\beta, \alpha = 0$ and the MMD regularization is degenerated. The improved capacity in transfer learning of DTLM is validated as seen in Table 2.

7.6 Parameter Effect In the following, we examine the impact of the parameters on the performance of DTLM. We show the performance of DTLM under different settings of λ, k on the six datasets from 20-Newsgroups in Fig (2a, 2b) and on the three datasets from Reuters-21578 in Fig (4a, 4b, 4c).

Fig (2a) shows the average classification accuracy of DTLM on 20-Newsgroups datasets under varying values of λ with fixed $k = 100$. We find that DTLM performs stably very well when λ spans over a wide range, i.e., $[0.1, 1000]$. Fig (2b) shows the average classification accuracy of DTLM under varying values of k, the number of feature clusters, with fixed $\lambda = 0.1$. We see that DTLM also performs stably well when k takes a value in a wide range, i.e., $[2, 64]$.

For Reuters-21578 datasets, DTLM's performance varies when λ is tuned in a range of $[0.1, 1000]$, in particular for *people* vs *place* dataset, which is seen from Fig (4a) with fixed $k = 100$. This is a common phenomenon in the graph geometric regularization literature, called the trivial solution and scale transfer problems, which is discussed in [9]. The phenomenon exists in GCMF, too. Without the MMD regularization and the discriminative prediction, the classification accuracy of GCMF stays at an even lower score over a wide range of λ value, i.e., $[0.1, 1000]$. To investigate the impact of k under different fixed λ values, we report the experiment results under different k values with λ set as 1 and 100, respectively, in Fig (4b) and Fig (4c). From these figures, it is easy to see that DTLM still stably achieves a good performance over a wide range of k, i.e., [2, 100], with a wide range of $\lambda = 1, 100$.

Figure (5) shows the DTLM's performance on semi-supervised classification in the target domain. From the figure, we see that the classification accuracy does not improve much with the increasing percentage of the labeled data in the target domain. This implies that the benefit of a portion of the data labeled in the target domain is relatively small and the complementary shared knowledge from the source domain is more significant instead in transfer learning, which further verifies the rationale of DTLM.

7.7 Convergence The method that we use to find the optimal objective value in Equation (3.9) is an multiplicative updating algorithm, which is an iterative process that converges to a local optimum. In this subsection we investigate the convergence of DTCM empirically. Fig (3a) and Fig (3b) show the average classification accuracy with respect to the number of iterations on datasets from 20-Newsgroups and Reuters-21578, respectively. Clearly, the average classification accuracy of DTLM increases stably with more iterations

Table 2: Performance comparison on different domain datasets with the measurement of average classification accuracy (10 repeated times). Due to space limitation, all the standard deviations of the comparing methods are omitted.

DataSet	*SVM*	*TSVM*	*DTK*	*MTrick*	*GCMF*	*DTLM*
comp vs rec	0.6879	0.7042	0.8641	0.8812	0.9275	$\mathbf{0.9727 \pm 0.0087}$
comp vs sci	0.6981	0.7278	0.9031	0.9113	0.9322	$\mathbf{0.9613 \pm 0.0254}$
comp vs talk	0.7023	0.7174	0.9106	0.9028	0.9399	$\mathbf{0.9545 \pm 0.0039}$
rec vs sci	0.6618	0.6944	0.8723	0.8872	0.9168	$\mathbf{0.9398 \pm 0.0059}$
rec vs talk	0.6714	0.6989	0.8401	0.8946	0.8964	$\mathbf{0.9646 \pm 0.0056}$
sci vs talk	0.6538	0.6754	0.8890	0.8862	0.9071	$\mathbf{0.9398 \pm 0.0180}$
orgs vs people	0.6643	0.6625	0.8042	0.7931	0.8228	$\mathbf{0.8836 \pm 0.0261}$
orgs vs place	0.6128	0.6419	0.7611	0.7784	0.7966	$\mathbf{0.8338 \pm 0.0118}$
people vs place	0.5911	0.5882	0.6910	0.6832	0.7002	$\mathbf{0.8246 \pm 0.0275}$
Average	0.6604	0.6790	0.8373	0.8464	0.8711	$\mathbf{0.9194 \pm 0.0148}$

and then converges after 50 iterations on 20-Newsgroups and 150 iterations on Reuters-21578, which verifies Theorem 5.1.

8 Conclusion

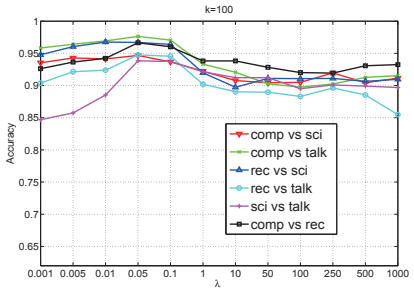
We argue that in the existing literature of collective matrix factorization based transfer learning, the learned latent factors still suffer from the divergence between different domains and thus are usually not discriminative for an appropriate assignment of category labels, resulting in a series of issues that are either not addressed well or ignored completely. To address these issues, we have developed a novel transfer learning framework as well as an iterative algorithm based on the framework called DTLM. Specifically, we apply a cross-domain matrix tri-factorization simultaneously incorporating a discriminative regression model and minimizing the MMD distance between the latent factor distributions in different domains. Meanwhile, we exploit the geometric graph structure to preserve the manifold geometric structures in both domains. Theoretical analysis and extensive empirical evaluations demonstrate that DTLM achieves a better performance consistently than all the comparing state-of-the-art methods in the literature.

9 Acknowledgment

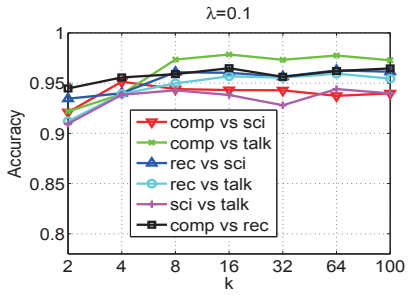
This work is supported in part by National Basic Research Program of China (2012CB316400), ZJU–Alibaba Financial Joint Lab, and Zhejiang Provincial Engineering Center on Media Data Cloud Processing and Analysis. ZZ is also supported in part by US NSF (IIS-0812114, CCF-1017828).

References

[1] K. Borgwardt, A. Gretton, M. Rasch, H. Kriegel, B. Schölkopf, and A. Smola. Integrating structured

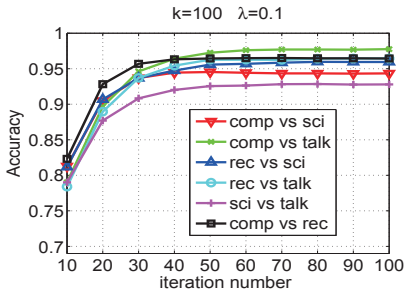


(a) Classification accuracy with respect to different values of λ with $k = 100$.

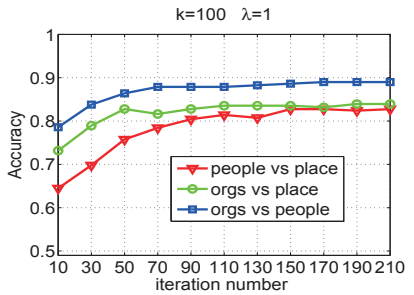


(b) Classification accuracy with respect to different numbers of feature clusters k with $\lambda = 0.1$.

Figure 2: Parameter sensitivity of DTLM on the cross-domain datasets generated from 20-Newsgroups.

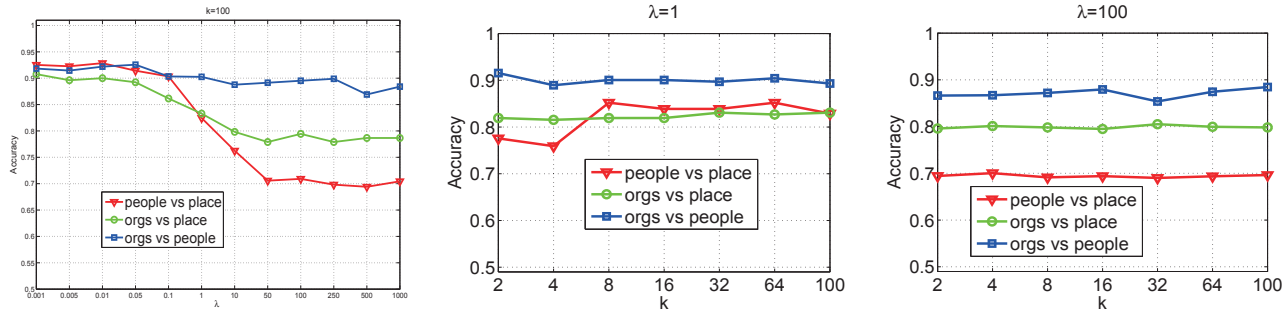


(a) Classification accuracy with respect to different numbers of iterations on datasets generated from 20-Newsgroups.



(b) Classification accuracy with respect to different numbers of iterations on datasets generated from Reuters-21578.

Figure 3: Convergence studies on DTLM

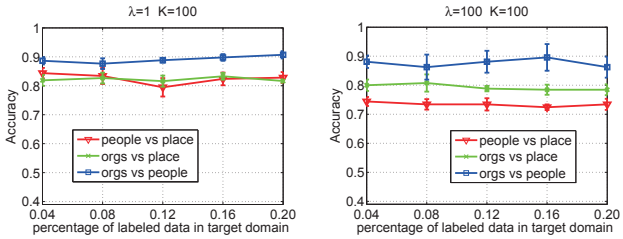


(a) Classification accuracy with respect to different values of λ with $k = 100$.

(b) Classification accuracy with respect to different numbers of feature clusters k with $\lambda = 1$.

(c) Classification accuracy with respect to different numbers of feature clusters k with $\lambda = 100$.

Figure 4: Parameter sensitivity of DTLM on the cross-domain datasets generated from Reuters-21578.



(a) (b)

Figure 5: Classification accuracy of DTLM with different percentages of the labeled data in the target domain on datasets generated from Reuters-21578.

biological data by kernel maximum mean discrepancy. *Bioinformatics*, 22(14):e49–e57, 2006.

[2] D. Cai, X. He, X. Wang, H. Bao, and J. Han. Locality preserving nonnegative matrix factorization. In *Proc. IJCAI*, pages 1010–1015, 2009.

[3] B. Chen, W. Lam, I. W. Tsang, and T.-L. Wong. Extracting discriminative concepts for domain adaptation in text mining. In *KDD*, pages 179–188, 2009.

[4] F. Chung. *Spectral graph theory*, volume 92. Amer Mathematical Society, 1997.

[5] W. Dai, G.-R. Xue, Q. Yang, and Y. Yu. Co-clustering based classification for out-of-domain documents. In *KDD*, pages 210–219, 2007.

[6] W. Dai, Q. Yang, G.-R. Xue, and Y. Yu. Boosting for transfer learning. In *ICML*, pages 193–200, 2007.

[7] C. H. Q. Ding, T. Li, and M. I. Jordan. Convex and semi-nonnegative matrix factorizations. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(1):45–55, 2010.

[8] J. Gao, W. Fan, J. Jiang, and J. Han. Knowledge transfer via multiple model local structure mapping. In *KDD*, pages 283–291, 2008.

[9] Q. Gu, C. H. Q. Ding, and J. Han. On trivial solution and scale transfer problems in graph regularized nmf. In *IJCAI*, pages 1288–1293, 2011.

[10] T. Joachims. Transductive inference for text classification using support vector machines. In *ICML*, pages 200–209, 1999.

[11] T. Li, V. Sindhwani, C. H. Q. Ding, and Y. Zhang. Knowledge transformation for cross-domain sentiment classification. In *SIGIR*, pages 716–717, 2009.

[12] T. Li, V. Sindhwani, C. H. Q. Ding, and Y. Zhang. Bridging domains with words: Opinion analysis with matrix tri-factorizations. In *SDM*, pages 293–302, 2010.

[13] X. Ling, W. Dai, G.-R. Xue, Q. Yang, and Y. Yu. Spectral domain-transfer learning. In *KDD*, pages 488–496, 2008.

[14] M. Long, J. Wang, G. Ding, W. Cheng, X. Zhang, and W. Wang. Dual transfer learning. In *SDM*, pages 540–551, 2012.

[15] M. Long, J. Wang, G. Ding, D. Shen, and Q. Yang. Transfer learning with graph co-regularization. In *AAAI*, 2012.

[16] S. Pan and Q. Yang. A survey on transfer learning. *Knowledge and Data Engineering, IEEE Transactions on*, 22(10):1345–1359, 2010.

[17] S. J. Pan, J. T. Kwok, Q. Yang, and Q. Yang. Transfer learning via dimensionality reduction. In *AAAI*, pages 677–682, 2008.

[18] D. Seung and L. Lee. Algorithms for non-negative matrix factorization. *Advances in neural information processing systems*, 13:556–562, 2001.

[19] H. Wang, H. Huang, F. Nie, and C. H. Q. Ding. Cross-language web page classification via dual knowledge transfer using nonnegative matrix tri-factorization. In *SIGIR*, pages 933–942, 2011.

[20] F. Zhuang, P. Luo, Z. Shen, Q. He, Y. Xiong, Z. Shi, and H. Xiong. Mining distinction and commonality across multiple domains using generative model for text classification. *IEEE Trans. Knowl. Data Eng.*, 24(11):2025–2039, 2012.

[21] F. Zhuang, P. Luo, H. Xiong, Q. He, Y. Xiong, and Z. Shi. Exploiting associations between word clusters and document classes for cross-domain text categorization. In *SDM*, pages 13–24, 2010.