Dual Transfer Learning

Mingsheng Long*† Jianmin Wang†‡ Guiguang Ding†‡ Wei Cheng§ Xiang Zhang¶
Wei Wang§

Abstract

Transfer learning aims to leverage the knowledge in the source domain to facilitate the learning tasks in the target domain. It has attracted extensive research interests recently due to its effectiveness in a wide range of applications. The general idea of the existing methods is to utilize the common latent structure shared across domains as the bridge for knowledge transfer. These methods usually model the common latent structure by using either the marginal distribution or the conditional distribution. However, without exploring the duality between these two distributions, these *single* bridge methods may not achieve optimal capability of knowledge transfer.

In this paper, we propose a novel approach, Dual Transfer Learning (DTL), which simultaneously learns the marginal and conditional distributions, and exploits the *duality* between them in a principled way. The key idea behind DTL is that learning one distribution can help to learn the other. This duality property leads to mutual reinforcement when adapting both distributions across domains to transfer knowledge. The proposed method is formulated as an optimization problem based on joint nonnegative matrix trifactorizations (NMTF). The two distributions are learned from the decomposed latent factors that exhibit the duality property. An efficient alternating minimization algorithm is developed to solve the optimization problem with convergence guarantee. Extensive experimental results demonstrate that DTL is more effective than alternative transfer learning methods.

Keywords Transfer Learning, Cross-Domain Classification, Dual Transfer Learning, Nonnegative Matrix Tri-Factorization.

*Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China. longmingsheng@gmail.com.

†School of Software, Tsinghua University, Beijing 100084, China. jimwang@tsinghua.edu.cn, dinggg@tsinghua.edu.cn.

‡Key Laboratory for Information System Security, Ministry of Education; Tsinghua National Laboratory for Information Science and Technology (TNLIST).

§Department of Computer Science, University of North Carolina at Chapel Hill, NC, USA. {weicheng,weiwang}@cs.unc.edu.

¶Dept of Electrical Engineering and Computer Science, Case Western Reserve University, OH, USA. xiang.zhang@case.edu.

1 Introduction

In many real-life applications, it may be impractical to obtain a large number of labeled examples. Lacking sufficient labeled examples poses a great challenge to traditional learning methods [2, 6, 17]. Transfer learning, which utilizes the labeled data in some source domain to help to train better models in the target domain, has recently attracted extensive research interests. It is widely applied to cross-domain classification, clustering, and information retrieval problems. The key idea is that, although the data distributions between the source domain and target domain are different, there is some common knowledge structure shared across domains. Such common structure can be utilized for the learning task in the target domain where the labeled examples are hard to collect [18].

Let $P_{src}(\mathbf{x}, y)$ and $P_{tar}(\mathbf{x}, y)$ be the data distributions in the source domain and the target domain respectively, where $\mathbf{x}$ is an example and y is its label. Transfer learning aims to fit distribution $P_{tar}(\mathbf{x}, y)$ with the labeled data drawn from $P_{src}(\mathbf{x}, y)$ and unlabeled data drawn from $P_{tar}(\mathbf{x}, y)$. The main challenge is to identify regions, in either the original or transformed latent space, where $P_{src}(\mathbf{x}, y)$ and $P_{tar}(\mathbf{x}, y)$ are similar. In this way, the shared common structure can be explored for across domain learning [17].

Many methods have been proposed to extract the common structure across the domains so that the distribution divergence between domains is reduced and the traditional learning algorithms can be applied. By definition, $P(\mathbf{x}, y) = P(\mathbf{x}) \cdot P(y|\mathbf{x})$, where $P(\mathbf{x})$ is the marginal distribution, and $P(y|\mathbf{x})$ is the conditional distribution that can be viewed as a classification model. A common assumption of the existing transfer learning methods is that, if the marginal distributions of examples are similar in some latent space, then the conditional distributions of the corresponding examples will also be similar [22]. More intuitively, if two data points are close in the latent space, then their class labels should also be similar [1]. Some methods have been proposed to learn the marginal distribution for knowledge transfer [6, 21, 17]. These methods try to find a latent feature space, where the marginal distributions are

drawn closer across domains. Other methods have been developed to learn the conditional distribution [24, 25]. They directly learn a latent space using the association between feature clusters and class labels so that the conditional distributions are drawn closer.

These methods use either the marginal distribution or the conditional distribution as the bridge for knowledge transfer. We refer to such approach as the *single* bridge transfer learning. The limitation of the existing single bridge approach is two-fold. (1) To transfer knowledge, these methods usually construct a latent space to represent the common structure shared across domains. However, not all latent factors in the space can be used to draw the two distributions closer. Some of the latent factors may represent the discrepancy between the distributions across domains. The "sharing all" latent factors assumption may cause the existing methods to underperform when the marginal distribution or the conditional distribution can only be drawn closer in a subspace of the latent space. In the worst case, when the distribution divergence between domains is so large that little knowledge can be shared, this strict assumption will result in negative transfer [4]. (2) These methods do not consider the *duality* between the marginal distribution and the conditional distribution. The duality between these two distributions is that learning one distribution can help to learn the other. It will lead to mutual reinforcement when learning the two distributions simultaneously. Without exploiting the duality, the existing methods may only transfer the common knowledge based on the marginal distribution and leave the common knowledge from the conditional distribution untransferred or only partially transferred, or vice versa. This may result in ineffective transfer.

In this paper, we propose a novel approach, Dual Transfer Learning (DTL), which simultaneously learns the marginal and conditional distributions, and exploits the duality between them to achieve effective knowledge transfer. The intuition behind DTL is as follows. If we can find a latent space where the marginal distributions across domains are close to each other, then a classification model can be learned in this space and shared across domains to draw the conditional distributions across domains closer. On the other hand, the learned conditional distributions can be used to refine the latent space so that the marginal distributions may become even closer. More specifically, the distribution of the observed data may be dominated by several latent factors. Some of the latent factors may cause the discrepancy between the distributions of the data in different domains. We refer to these factors as the *domain-specific latent factors*. On some other latent factors, the data distributions may be similar across domains. We refer to these latent factors as the *common latent factors*. We can uncover the common latent factors so that the marginal distributions across domains will be close in the common latent space [17]. We can learn a shared classification model in this latent space so that the conditional distributions across domains are close. On the other hand, if we have learned an accurate shared classification model across domains, we can use it to supervise the learning of the common latent factors so that the marginal distributions become even closer. The common latent factors and the common classification model are the means for adapting the marginal distributions and conditional distributions across domains respectively. By taking advantage of the duality, they reinforce the learning of the two distributions and improve each other to facilitate effective transfer learning.

The proposed DTL method enhances the transfer capability in a principled way. It finds the latent feature space where the marginal distributions across domains are close, and simultaneously learns a shared classification model in the latent space to make the conditional distributions across domains closer. The main contributions of this paper are: (1) distinguishing the common and domain-specific latent factors to avoid negative transfer; (2) exploiting the duality between the marginal distribution and conditional distribution to achieve effective transfer. The method is formulated as an optimization problem of joint nonnegative matrix tri-factorizations (NMTF), where the two distributions are learned with the help of the decomposed latent factors that exhibit the duality property. An efficient alternating minimization algorithm is developed to solve the optimization with convergence guarantee. Extensive experiments on benchmark data sets demonstrate the effectiveness of the proposed method.

The remainder of this paper is organized as follows. The related work is discussed in Section 2. In Section 3, we review NMTF and discuss its relevance to distribution learning, which motivates the formulation of our method. In Section 4, we describe the proposed DTL method, along with the learning algorithm, followed by the optimization derivation, and the proof of convergence. The experimental results are reported in Section 5. The conclusions and future work are discussed in Section 6.

2 Related Work

In this section, we review several existing work that are most related to our work, including transfer learning and semi-supervised learning.

2.1 Transfer Learning Transfer learning is widely applied in the applications where the training data

and the test data are obtained from different resources and with different distributions. Most transfer learning methods assume that there is some knowledge structure that defines the domain relatedness, and incorporate this structure in the learning process. The existing methods can be categorized into two types: instance transfer learning and feature representation transfer learning. Please refer to [18] for a comprehensive survey.

Instance transfer learning uses a re-weighting strategy for instances. The general idea is to increase the weights of instances in the source domain that are close to the instances in the target domain, and decreases the weights of instances in the source domain that are far away from the instances in the target domain [7, 11]. Feature representation transfer learning aims to discover a shared feature space in which the data distributions across domains are close to each other. The shared feature space can be constructed either in the original feature space [2, 16], or in the transformed subspace [6, 21, 25, 14, 5, 10, 15, 17]. In the original feature space, the correspondence among features are identified by modeling their correlations with pivot features that behave similarly across domains. In the transformed subspace, dimensionality reduction methods are applied to extract the underlying common structure.

The existing feature representation transfer learning methods focus on learning either the marginal distribution or the conditional distribution for knowledge transfer. For example, Co-Clustering based Classification (CoCC) [6] and Label Propagation [21] transfer the common feature clusters, which can be regarded as learning the marginal distribution. Collaborative Dual-PLSA [24] and Matrix Tri-Factorization based Classification (MTrick) [25] transfer the common association between feature clusters and example classes, which can be regarded as learning the conditional distribution. Another well-designed method for learning the marginal distribution is Joint Subspace Nonnegative Matrix Factorization [10]. It learns the common latent factors and domain-specific latent factors that span a shared subspace where the marginal distributions across domains are close. However, this method does not learn the conditional distribution thus can not be applied to cross-domain classification tasks. The proposed DTL method also adopts the block matrix factorization technique. The key difference between DTL and previous single bridge learning methods is that DTL simultaneously learns the marginal and conditional distributions. Exploiting the duality between these two distributions is a crucial step to enhance the transfer capability.

Two very recent methods, cross domain distribution adaptation via kernel mapping [22] and dual knowledge transfer [20], started exploring the idea of learning both distributions for knowledge transfer. In [22], both the marginal distribution and the conditional distribution are adapted. It uses kernel mapping to draw the marginal distributions across domains closer. A sample selection strategy is applied to ensure the closeness of the conditional distributions across domains. However, the two distributions are learned separately without exploiting the duality for mutual reinforcement. Moreover, it requires a few labeled data available in the target domain for inducing transfer learners for that domain. Therefore, it cannot be applied to *transductive* transfer learning, where no labeled data are available in the target domain. In [20], the proposed method discovers two types of latent factors from nonnegative matrix tri-factorization as two paths, through which knowledge can be transferred across domains. However, it does not exploit the duality between the marginal distribution and the conditional distribution. Our method integrates the duality in a unified subspace learning paradigm and requires fewer trade-off parameters. The effectiveness of DTL over the existing methods is demonstrated by extensive experimental evaluation.

2.2 Semi-Supervised Learning Semi-supervised learning also aims to learn from both labeled and unlabeled data [19, 23]. In semi-supervised learning, the labeled and unlabeled instances are sampled from the same domain and follow the dame data distribution. In transfer learning, the labeled instances and unlabeled instances are sampled from different domains with different distributions. Semi-supervised learning cannot be applied in this problem setting. For example, the nonnegative matrix tri-factorization used in [19] for semi-supervised clustering cannot be directly applied to solve transfer learning problems, since it assumes that all the latent factors are shared across domains. This assumption is not valid if data have different distributions across domains.

3 Nonnegative Matrix Tri-Factorization and Its Relationship to Distribution Learning

DTL is based on the nonnegative matrix tri-factorization (NMTF) model [8]. NMTF is very effective for mining text and image data. In this section, we explore the intuitions behind NMTF and discuss two different but closely related interpretations of the NMTF model. One is the traditional clustering interpretation that has been extensively studied. The other one is the transformation interpretation that motivates our DTL method.

In NMTF, a data matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$ is decomposed into a product of three nonnegative factors $\mathbf{U} \in \mathbb{R}^{m \times k}$,

$\mathbf{H} \in \mathbb{R}^{k\times c}$, and $\mathbf{V} \in \mathbb{R}^{n\times c}$, such that $\mathbf{X} \approx \mathbf{UHV}^{\mathrm{T}}$. This approximation can be achieved by the following matrix norm optimization:

$$\min_{\mathbf{U},\mathbf{H},\mathbf{V}\geq\mathbf{0}} \mathcal{L}_{\mathrm{NMTF}} = \left\|\mathbf{X} - \mathbf{UHV}^{\mathrm{T}}\right\| \tag{3.1}$$

where $\|\cdot\|$ is the Frobenius norm of matrix. $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$ is an $m \times n$ data matrix containing n examples. Each example $\mathbf{x}_i$ is an $m \times 1$ feature vector in the original feature space of m dimensions.

3.1 Clustering Interpretation of NMTF From a clustering perspective, the three nonnegative factors decomposed from NMTF can be interpreted in the following way [8]:

- $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_k]$ is an $m \times k$ cluster assignment matrix representing the feature clusters. If $\arg\max_j(\mathbf{U})_{ij} = j^*$, then the ith feature belongs to the j^*th feature cluster. Each $\mathbf{u}_i$ is a probability distribution over m features and is referred to as a *feature cluster*. There are k feature clusters in total.

- $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_c]$ is an $n \times c$ cluster assignment matrix representing example clusters. If $\arg\max_j(\mathbf{V})_{ij} = j^*$, then the ith example belongs to the j^*th example cluster. Each $\mathbf{v}_i$ is a probability distribution over n examples and is referred to as an *example cluster*. There are c example clusters in total. In the classification setting, each example cluster can be regarded as a class or category.

- $\mathbf{H} = [\mathbf{h}_1, \ldots, \mathbf{h}_c]$ is a $k \times c$ weight matrix representing the association between feature clusters and example clusters. $(\mathbf{H})_{ij}$ is the probability that the ith feature cluster is associated with the jth example cluster. Thus $\mathbf{H}$ is referred to as the *cluster association* matrix.

By clustering both sides of the data matrix simultaneously, NMTF makes use of the interrelatedness between features and examples, resulting in superior performance over other clustering methods. The clustering interpretation helps to define the terminologies for the transformation interpretation of NMTF.

3.2 Transformation Interpretation of NMTF We observe that, in addition to the widely used clustering interpretation, NMTF can also be interpreted from the viewpoint of linear transformation. This interpretation can be used in learning the marginal and conditional distributions.

Note that Equation (3.1) can be viewed as a combination of the following two alternating factorizations:

$$\min_{\mathbf{U},\mathbf{X}'\geq\mathbf{0}} \mathcal{L}_{\phi} = \|\mathbf{X} - \mathbf{UX}'\| \tag{3.2}$$

$$\min_{\mathbf{H},\mathbf{V}\geq\mathbf{0}} \mathcal{L}_{\theta} = \left\|\mathbf{X}' - \mathbf{HV}^{\mathrm{T}}\right\| \tag{3.3}$$

where $\mathbf{X}' = [\mathbf{x}'_1, \ldots, \mathbf{x}'_n]$ is a $k \times n$ dimension-reduced data matrix containing n examples. Each example $\mathbf{x}'$ is a $k \times 1$ feature vector in the latent feature space of k dimensions, spanned by the columns of $\mathbf{U}$.

Assume that data in the original feature space follow the marginal distribution $P(\mathbf{x})$ and the conditional distribution $P(y|\mathbf{x})$. The above two factorizations can be interpreted in terms of a linear transformation on the data distributions:

- Factorization in Equation (3.2) derives a linear transformation $\phi : \mathbb{R}^m \to \mathbb{R}^k$. $\mathbf{x}' = \phi(\mathbf{x})$ maps each example $\mathbf{x} \in \mathbb{R}^{m\times 1}$ in the original feature space to $\mathbf{x}' \in \mathbb{R}^{k\times 1}$ in the latent feature space spanned by the columns of $\mathbf{U}$. After mapping ϕ, the marginal distribution changes from $P(\mathbf{x})$ to $P(\mathbf{x}')$. We refer to ϕ as the *marginal mapping*. ϕ and $\mathbf{U}$ are related to learning the marginal distribution.

- Factorization in Equation (3.3) derives another linear transformation $\theta : \mathbb{R}^k \to \mathbb{R}^c$. $\mathbf{v}^{\mathrm{T}} = \theta(\mathbf{x}')$ maps each example $\mathbf{x}' \in \mathbb{R}^{k\times 1}$ in the latent feature space to $\mathbf{v}^{\mathrm{T}} \in \mathbb{R}^{c\times 1}$ in the example cluster/class space spanned by the columns of $\mathbf{H}$. After mapping θ, the conditional distribution changes from $P(y|\mathbf{x})$ to $P(y|\mathbf{x}')$. For any example $\mathbf{x}_i$ in the original feature space, its label can be predicted by $j^* = \arg\max_j \{P(y_i = j|\mathbf{x}'_i) = (\mathbf{V})_{ij}\}$. We refer to θ as the *conditional mapping*. θ and $\mathbf{H}$ are related to learning the conditional distribution.

From the discussion above, we observe that the latent factors $\mathbf{U}$ and $\mathbf{H}$ are closely related to the linear transformations ϕ and θ for learning the marginal and conditional distributions. However, NMTF in Equation (3.1) is designed for one domain. In transfer learning, there may be several source and target domains. Utilizing the relationship between NMTF and distribution learning, in the next section, we will show how to use NMTF to perform linear transformation on the marginal and conditional distributions. Such transformation will draw the data points closer across domains. The duality between these two distributions will also be exploited to facilitate knowledge transfer.

It is worth noting that the duality addressed by our work is a general property. NMTF is only one of many possible techniques to formulate distribution learning and realize dual transfer learning.

4 Dual Transfer Learning

In this section, we present the DTL method for cross-domain classification. DTL exploits the duality property that learning one distribution can help to learn the other for mutual reinforcement. DTL is formulated as an optimization problem of joint nonnegative matrix tri-factorizations.

4.1 Problem Definition We focus on *transductive transfer learning* where the source domains have abundant labeled examples while the target domains only have unlabeled examples. Let $\mathcal{D}$ be a domain, t be the total number of domains, s be the number of source domains, $\tau \in [1, t]$ be the domain index. Then $\{\mathcal{D}_\tau\}_{\tau=1}^{s}$ represent the source domains, and $\{\mathcal{D}_\tau\}_{\tau=s+1}^{t}$ represent the target domains. For each domain $\mathcal{D}_\tau$, let $\mathbf{X}_\tau = [\mathbf{x}_1^\tau, \ldots, \mathbf{x}_{n_\tau}^\tau] \in \mathbb{R}^{m \times n_\tau}$ be the feature-example co-occurrence matrix of n_τ examples, $\mathbf{Y}_\tau \in \mathbb{R}^{n_\tau \times c}$ be the corresponding labels if $\tau \in [1, s]$. Each example $\mathbf{x}_i^\tau$ is a feature vector in the shared feature space spanned by m different features, and can be calculated by *tf-idf* (term frequency-inverse document frequency) for text data. Each example $\mathbf{x}_i^\tau$ in source domain $\mathcal{D}_\tau$ is associated with one of the c labels: $y_{ij}^\tau = 1$ if $\mathbf{x}_i^\tau$ belongs to class j, $j \in [1, c]$, and $y_{ij}^\tau = 0$ otherwise. For clarity, frequently used notations and their descriptions are summarized in Table 1.

Given labeled examples $\{\mathbf{X}_\tau, \mathbf{Y}_\tau\}_{\tau=1}^{s}$ in the source domains and unlabeled examples $\{\mathbf{X}_\tau\}_{\tau=s+1}^{t}$ in the target domains, transfer learning aims at finding a function f that for any unlabeled example $\mathbf{x}$ in the target domain, predicts its correct label y, i.e., $y = f(\mathbf{x})$. A standard way to learn this function is to minimize the loss between the label predicted by the function and the true label. However, since the distribution divergence between the source and target domains is large in cross-domain learning, f may not generalize well in the target domain. The goal of transfer learning is to alleviate this difficulty by making data distributions between domains drawn closer in a latent space so that we can train the cross-domain classifier f as accurate as possible.

4.2 The Proposed Method Existing clustering based methods [6, 21, 24, 25] assume that the cluster structures hidden across domains can be extracted to learn the marginal distribution or the conditional distribution for knowledge transfer. The clustering of data in domain τ can be performed using NMTF

$$\min_{\mathbf{U}'_\tau, \mathbf{H}_\tau, \mathbf{V}_\tau \geq \mathbf{0}} \mathcal{L}_\tau = \left\| \mathbf{X}_\tau - \mathbf{U}'_\tau \mathbf{H}_\tau \mathbf{V}_\tau^{\mathrm{T}} \right\|^2 \tag{4.4}$$

Here $\mathbf{U}'_\tau \in \mathbb{R}^{m \times k}$, $\mathbf{V}_\tau \in \mathbb{R}^{n_\tau \times c}$ and $\mathbf{H}_\tau \in \mathbb{R}^{k \times c}$ are the k feature clusters, c example clusters/classes, and the association between them in domain $\mathcal{D}_\tau$.

Table 1: Notations and descriptions used in this paper.

Notation	Description
t, s	#total/source domains
τ	domain index $1 \leq \tau \leq t$
$\mathcal{D}_\tau$	domain τ
n_τ	#examples in $\mathcal{D}_\tau$
m	#features in the shared feature space
c	#classes in the shared label space
k, κ	#total/common feature clusters
ϕ, θ	marginal/conditional mapping
$\mathbf{X}_\tau$	$m \times n_\tau$ data matrix of $\mathcal{D}_\tau$
$\mathbf{Y}_\tau$	$n_\tau \times c$ label matrix of $\mathcal{D}_\tau$
$\mathbf{U}, \mathbf{U}_\tau$	common/domain-specific feature clusters of $\mathcal{D}_\tau$
$\mathbf{V}_\tau$	example clusters of $\mathcal{D}_\tau$
$\mathbf{H}$	association between feature/example clusters
$\mathbf{H}_\mu$, $\mathbf{H}_\upsilon$	$\mathbf{H}_\mu \equiv \mathbf{H}(1:\kappa,:)$, $\mathbf{H}_\upsilon \equiv \mathbf{H}(\kappa+1:k,:)$
$\mathbf{1}_m$	$m \times 1$ vector of ones
$\mathbf{\Lambda}_\tau$, $\mathbf{\Gamma}_\tau$	Lagrange multipliers for constraints in $\mathcal{D}_\tau$
$\circ, \frac{[\cdot]}{[\cdot]}, \sqrt{\cdot}$	element-wise product/division/root

As discussed in Section 3.2, $\mathbf{U}'_\tau \sim \phi'_\tau$ is related to learning the marginal distribution of $\mathcal{D}_\tau$, and $\mathbf{H}_\tau \sim \theta_\tau$ is related to learning the conditional distribution of $\mathcal{D}_\tau$. By extracting the common parts of $\mathbf{U}'_\tau$ or $\mathbf{H}_\tau$, data from different domains are mapped to a shared latent space in which the divergence of the marginal distributions or the conditional distributions across domains is reduced.

Next we will formulate the learning of the two distributions as an optimization problem of joint nonnegative matrix tri-factorizations, in which the duality property is realized naturally by using NMTF.

4.2.1 Marginal Mapping We derive the marginal mapping ϕ'_τ by learning feature clusters $\mathbf{U}'_\tau$. Motivated by [10], the feature clusters across domains can be reasonably partitioned into a common part and a domain-specific part. The common part is used to draw the marginal distributions across domains closer. So we partition $\mathbf{U}'_\tau$ into κ *common* feature clusters $\mathbf{U} \in \mathbb{R}^{m \times \kappa}$ and $k - \kappa$ *domain-specific* feature clusters $\mathbf{U}_\tau \in \mathbb{R}^{m \times (k-\kappa)}$. That is, $\mathbf{U}'_\tau = [\mathbf{U}, \mathbf{U}_\tau]$. This also leads to the partition of the marginal mapping $\phi'_\tau = [\phi, \phi_\tau]$, where only the common part ϕ can be shared across domains to draw the marginal distributions closer, while the domain-specific part ϕ_τ is used to respect domain-specific knowledge. By doing so, the marginal distribution is learned in an adaptive way with the sharing level controlled depending on the relatedness between domains. We extend Equation (4.4) so that it can exactly model the marginal distribution learning process:

$$\min_{\mathbf{U}, \mathbf{U}_\tau, \mathbf{H}_\tau, \mathbf{V}_\tau \geq \mathbf{0}} \mathcal{L}_\tau = \left\| \mathbf{X}_\tau - [\mathbf{U}, \mathbf{U}_\tau] \, \mathbf{H}_\tau \mathbf{V}_\tau^{\mathrm{T}} \right\|^2 \tag{4.5}$$

4.2.2 Conditional Mapping We derive the conditional mapping θ_τ by learning the cluster association matrix $\mathbf{H}_\tau$. Motivated by [25], the association between feature clusters and example clusters usually remain stable or unchanged across domains. The validity of this assumption is further strengthened since we have learned the feature clusters in an adaptive way in the marginal mapping, which in return can help learning the conditional distribution more effectively due to the duality property. Therefore, we let $\mathbf{H}_\tau = \mathbf{H}$ and $\theta_\tau = \theta$ for all domains, and share the entire conditional mapping θ across domains to draw the conditional distributions closer. We can further extend Equation (4.5) so that the learning of the two distributions are integrated in a unified subspace learning paradigm:

$$\min_{\mathbf{U},\mathbf{U}_\tau,\mathbf{H},\mathbf{V}_\tau \geq \mathbf{0}} \mathcal{L}_\tau = \left\| \mathbf{X}_\tau - [\mathbf{U}, \mathbf{U}_\tau]\, \mathbf{H}\mathbf{V}_\tau^{\mathrm{T}} \right\|^2 \tag{4.6}$$

The duality between the two distributions is realized in Equation (4.6) since $\mathbf{U}$ and $\mathbf{H}$ in the extended NMTF model also exhibit the duality property: a better clustering of features can help clustering the examples better, as well as the association between them; and vice versa. By learning the two distributions simultaneously for mutual reinforcement, this duality property enables optimal knowledge transfer across domains.

4.2.3 Optimization The common marginal mapping ϕ and the conditional mapping θ can serve as two means for learning the marginal distribution and the conditional distribution, respectively. The learning of ϕ and θ has been addressed in Equation (4.6). Here we share them to multiple domains simultaneously, arriving at the following objective function:

$$\mathcal{L} = \sum_{\tau=1}^{t} \left\| \mathbf{X}_\tau - [\mathbf{U}, \mathbf{U}_\tau]\, \mathbf{H}\mathbf{V}_\tau^{\mathrm{T}} \right\|^2 \tag{4.7}$$

Learning the objective in Equation (4.7) involves the following optimization problem, which is termed *joint nonnegative matrix tri-factorizations*:

$$\begin{aligned} &\min_{\mathbf{U},\mathbf{U}_\tau,\mathbf{H},\mathbf{V}_\tau \geq \mathbf{0}} \mathcal{L} \\ &s.t. \quad [\mathbf{U}, \mathbf{U}_\tau]^{\mathrm{T}} \mathbf{1}_m = \mathbf{1}_k, \mathbf{V}_\tau \mathbf{1}_c = \mathbf{1}_{n_\tau}, \forall \tau \in [1, t] \end{aligned} \tag{4.8}$$

The source domain label supervision is incorporated by $\{\mathbf{V}_\tau \equiv \mathbf{Y}_\tau\}_{\tau=1}^{s}$. The ℓ_1 normalization constraints on each column of $\mathbf{U}_\tau$ and each row of $\mathbf{V}_\tau$ are used to make the optimization well defined. The assigned label for any example $\mathbf{x}_i^\tau$ in domain τ can be determined by

$$f(\mathbf{x}_i^\tau) = \arg\max_j \left(\mathbf{V}_\tau\right)_{ij} \tag{4.9}$$

The DTL optimization is performed in an alternating minimization process until convergence. During each iteration, it extracts the common and domain-specific feature clusters to learn the marginal distribution; and simultaneously, it extracts the common association between feature clusters and example clusters/classes to learn the conditional distribution. After the iterative process, both distributions across domains are drawn closer by exploiting the duality property so that knowledge can be transferred effectively.

4.3 Learning Algorithm We present the solution to the DTL optimization problem in Equation (4.8) as the following theorem. The theoretical aspects of the optimization are presented in the next subsection.

THEOREM 4.1. *Updating* $\mathbf{U}_\tau$, $\mathbf{U}$, $\mathbf{V}_\tau$ *and* $\mathbf{H}$ *using Equations* (4.10)∼(4.13) *for each domain* τ *will monotonically decrease the objective function in Equation* (4.8) *until convergence.*

$$\mathbf{U}_\tau \leftarrow \mathbf{U}_\tau \circ \sqrt{\frac{[\mathbf{X}_\tau \mathbf{V}_\tau \mathbf{H}_\upsilon^{\mathrm{T}}]}{\left[[\mathbf{U}, \mathbf{U}_\tau]\, \mathbf{H}\mathbf{V}_\tau^{\mathrm{T}} \mathbf{V}_\tau \mathbf{H}_\upsilon^{\mathrm{T}}\right]}} \tag{4.10}$$

$$\mathbf{U} \leftarrow \mathbf{U} \circ \sqrt{\frac{\left[\sum_{\tau=1}^{t} \mathbf{X}_\tau \mathbf{V}_\tau \mathbf{H}_\mu^{\mathrm{T}}\right]}{\left[\sum_{\tau=1}^{t} [\mathbf{U}, \mathbf{U}_\tau]\, \mathbf{H}\mathbf{V}_\tau^{\mathrm{T}} \mathbf{V}_\tau \mathbf{H}_\mu^{\mathrm{T}}\right]}} \tag{4.11}$$

$$\mathbf{V}_\tau \leftarrow \mathbf{V}_\tau \circ \sqrt{\frac{[\mathbf{X}_\tau^{\mathrm{T}} [\mathbf{U}, \mathbf{U}_\tau]\, \mathbf{H}]}{\left[\mathbf{V}_\tau \mathbf{H}^{\mathrm{T}} [\mathbf{U}, \mathbf{U}_\tau]^{\mathrm{T}} [\mathbf{U}, \mathbf{U}_\tau]\, \mathbf{H}\right]}} \tag{4.12}$$

$$\mathbf{H} \leftarrow \mathbf{H} \circ \sqrt{\frac{\left[\sum_{\tau=1}^{t} [\mathbf{U}, \mathbf{U}_\tau]^{\mathrm{T}} \mathbf{X}_\tau \mathbf{V}_\tau\right]}{\left[\sum_{\tau=1}^{t} [\mathbf{U}, \mathbf{U}_\tau]^{\mathrm{T}} [\mathbf{U}, \mathbf{U}_\tau]\, \mathbf{H}\mathbf{V}_\tau^{\mathrm{T}} \mathbf{V}_\tau\right]}} \tag{4.13}$$

where $\mathbf{H}_\mu \equiv \mathbf{H}(1:\kappa, :)$, $\mathbf{H}_\upsilon \equiv \mathbf{H}(\kappa+1:k, :)$, *operator* $\circ$ *is element-wise product,* $\frac{[\cdot]}{[\cdot]}$ *is element-wise division,* $\sqrt{\cdot}$ *is element-wise root.*

Based on Theorem 4.1, we develop the learning algorithm for the DTL optimization and summarize it in Algorithm 1. To make the algorithm converge faster, we initialize the labels of target domain data by logistic regression trained on the source domain data. We also keep the labels of source domain data unchanged, i.e., $\{\mathbf{V}_\tau \equiv \mathbf{Y}_\tau\}_{\tau=1}^{s}$, instead of updating them during iterations. The time complexity of Algorithm 1 is $\mathcal{O}\left(\sum_{\tau=1}^{t} maxIter \cdot kmn_\tau\right)$ on t domains, which is approximately the sum of the NMTF running time on each domain.

Algorithm 1 DTL: Dual Transfer Learning

input: data sets $\{\mathbf{X}_\tau\}_{\tau=1}^t, \{\mathbf{Y}_\tau\}_{\tau=1}^s$, parameters k, κ representing the number of total/common feature clusters.
ouput: feature clusters $\{\mathbf{U}_\tau\}_{\tau=1}^t, \mathbf{U}$, cluster association $\mathbf{H}$, classification results in target domains $\{\mathbf{V}_\tau\}_{\tau=s+1}^t$.

1: normalize each data set into probability by $\mathbf{X}_\tau \leftarrow \mathbf{X}_\tau / \sum_{ij}(\mathbf{X}_\tau)_{ij}, \tau \in [1, t]$.
2: initialize $\{\mathbf{U}_\tau\}_{\tau=1}^t, \mathbf{U}, \mathbf{H}$ by random positives, $\{\mathbf{V}_\tau\}_{\tau=1}^s$ by $\{\mathbf{Y}\}_{\tau=1}^s$, $\{\mathbf{V}_\tau\}_{\tau=s+1}^t$ by logistic regression trained on $\{\mathbf{X}_\tau, \mathbf{Y}_\tau\}_{\tau=1}^s$.
3: **for** $iter \leftarrow 1$ to $maxIter$ **do**
4: **for** $\tau \leftarrow 1$ to t **do**
5: update $\mathbf{U}_\tau, \mathbf{U}, \mathbf{V}_\tau, \mathbf{H}$ by Equations (4.10)∼(4.13) with $\{\mathbf{V}_\tau \equiv \mathbf{Y}_\tau\}_{\tau=1}^s$ fixed.
6: for each update above, normalize each column of $[\mathbf{U}, \mathbf{U}_\tau]$ or each row of $\mathbf{V}_\tau$ by ℓ_1 norm.
7: compute objective $\mathcal{L}^{iter}$ by Equation (4.7).
8: **end for**
9: **end for**

4.4 Theoretical Analysis

4.4.1 Derivation We derive the solution to Equation (4.8) following the theory of constrained optimization [3]. Specifically, we will optimize one variable and derive its updating rule while fixing the rest variables. The procedure repeats until convergence.

We formulate the Lagrange function for the optimization with normalization constraints as follows

$$\begin{aligned} L =& \sum_{\tau=1}^t \left\| \mathbf{X}_\tau - [\mathbf{U}, \mathbf{U}_\tau] \mathbf{H} \mathbf{V}_\tau^{\mathrm{T}} \right\|^2 \\ &+ \sum_{\tau=1}^t \mathrm{tr}\left(\mathbf{\Gamma}_\tau ([\mathbf{U}, \mathbf{U}_\tau]^{\mathrm{T}} \mathbf{1}_m - \mathbf{1}_k)([\mathbf{U}, \mathbf{U}_\tau]^{\mathrm{T}} \mathbf{1}_m - \mathbf{1}_k)^{\mathrm{T}} \right) \\ &+ \sum_{\tau=1}^t \mathrm{tr}\left(\mathbf{\Lambda}_\tau \left(\mathbf{V}_\tau \mathbf{1}_c - \mathbf{1}_{n_\tau} \right) \left(\mathbf{V}_\tau \mathbf{1}_c - \mathbf{1}_{n_\tau} \right)^{\mathrm{T}} \right) \end{aligned}$$

where $\mathbf{\Gamma}_\tau \in \mathbb{R}^{k \times k}$, $\mathbf{\Lambda}_\tau \in \mathbb{R}^{n_\tau \times n_\tau}$ are the Lagrange multipliers for the normalization constraints.

Without loss of generality, we only show detailed derivation of the updating rule for $\mathbf{V}_\tau$. Using the Karush-Kuhn-Tucker (KKT) complementarity condition [3] for the constraint on $\mathbf{V}_\tau$ we have

$$\nabla_{\mathbf{V}_\tau} L \circ \mathbf{V}_\tau = \left\{ \begin{array}{l} -2\mathbf{X}_\tau^{\mathrm{T}} [\mathbf{U}, \mathbf{U}_\tau] \mathbf{H} \\ +2\mathbf{V}_\tau \mathbf{H}^{\mathrm{T}} [\mathbf{U}, \mathbf{U}_\tau]^{\mathrm{T}} [\mathbf{U}, \mathbf{U}_\tau] \mathbf{H} \\ +2\mathbf{\Lambda}_\tau \mathbf{V}_\tau \mathbf{1}_c \mathbf{1}_c^{\mathrm{T}} - 2\mathbf{\Lambda}_\tau \mathbf{1}_{n_\tau} \mathbf{1}_c^{\mathrm{T}} \end{array} \right\} \circ \mathbf{V}_\tau = \mathbf{0}$$

This leads to the following update formula

$$\mathbf{V}_\tau \leftarrow \mathbf{V}_\tau \circ \sqrt{\frac{[\mathbf{X}_\tau^{\mathrm{T}} [\mathbf{U}, \mathbf{U}_\tau] \mathbf{H} + \mathbf{\Lambda}_\tau \mathbf{1}_{n_\tau} \mathbf{1}_c^{\mathrm{T}}]}{\left[\mathbf{V}_\tau \mathbf{H}^{\mathrm{T}} [\mathbf{U}, \mathbf{U}_\tau]^{\mathrm{T}} [\mathbf{U}, \mathbf{U}_\tau] \mathbf{H} + \mathbf{\Lambda}_\tau \mathbf{V}_\tau \mathbf{1}_c \mathbf{1}_c^{\mathrm{T}} \right]}}$$

We need to compute the Lagrange multiplier $\mathbf{\Lambda}_\tau$ to obtain the updating rule for $\mathbf{V}_\tau$. We adopt an iterative normalization technique [25] to satisfy the constraint in the optimization independent of the computation of $\mathbf{\Lambda}_\tau$. Specifically, at each iteration, we normalize each row of $\mathbf{V}_\tau$ so that $\mathbf{V}_\tau \mathbf{1}_c = \mathbf{1}_{n_\tau}$. After that, we get two equal terms $\mathbf{\Lambda}_\tau \mathbf{1}_{n_\tau} \mathbf{1}_c^{\mathrm{T}} = \mathbf{\Lambda}_\tau \mathbf{V}_\tau \mathbf{1}_c \mathbf{1}_c^{\mathrm{T}}$ that depend only on $\mathbf{\Lambda}_\tau$ and can be omitted from the above updating formula without influencing convergence. This leads to the updating rule for $\mathbf{V}_\tau$ in Equation (4.12).

Following the similar derivations as shown above, we can obtain the updating rules for all the rest variables in the DTL optimization, as shown in Equations (4.10)∼(4.13).

4.4.2 Convergence We use the auxiliary function approach [13] to prove the convergence of Theorem 4.1 and Algorithm 1. We first introduce the definitions of auxiliary function as follows.

Definition 4.1. *[13] $A(\mathbf{Z}, \tilde{\mathbf{Z}})$ is an auxiliary function for $L(\mathbf{Z})$ if the conditions*

$$A(\mathbf{Z}, \tilde{\mathbf{Z}}) \geq L(\mathbf{Z}) \text{ and } A(\mathbf{Z}, \mathbf{Z}) = L(\mathbf{Z})$$

are satisfied for any given $\mathbf{Z}$, $\tilde{\mathbf{Z}}$.

Lemma 4.1. *[13] If A is an auxiliary function for L, then L is non-increasing under the update*

$$\mathbf{Z}^{(t+1)} = \arg\min_{\mathbf{Z}} A(\mathbf{Z}, \mathbf{Z}^{(t)})$$

Theorem 4.2. *Let $L(\mathbf{V}_\tau)$ denote the sum of all terms in L that contain $\mathbf{V}_\tau$. The following function*

$$\begin{aligned} A\left(\mathbf{V}_\tau, \tilde{\mathbf{V}}_\tau\right) =& \sum_{ij} \left(\tilde{\mathbf{V}}_\tau \mathbf{H}^{\mathrm{T}} [\mathbf{U}, \mathbf{U}_\tau]^{\mathrm{T}} [\mathbf{U}, \mathbf{U}_\tau] \mathbf{H} + \mathbf{\Lambda}_\tau \tilde{\mathbf{V}}_\tau \mathbf{1}_c \mathbf{1}_c^{\mathrm{T}} \right)_{ij} \frac{(\mathbf{V}_\tau)_{ij}^2}{(\tilde{\mathbf{V}}_\tau)_{ij}} \\ & -2 \sum_{ij} \left(\mathbf{X}_\tau^{\mathrm{T}} [\mathbf{U}, \mathbf{U}_\tau] \mathbf{H} + \mathbf{\Lambda}_\tau \mathbf{1}_{n_\tau} \mathbf{1}_c^{\mathrm{T}} \right)_{ij} (\tilde{\mathbf{V}}_\tau)_{ij} \left(1 + \log \frac{(\mathbf{V}_\tau)_{ij}}{(\tilde{\mathbf{V}}_\tau)_{ij}} \right) \end{aligned}$$

is an auxiliary function for $L(\mathbf{V}_\tau)$. Furthermore, it is a convex function in $\mathbf{V}_\tau$ and has a global minimum.

Theorem 4.2 can be proved similarly to [8] by validating $A(\mathbf{V}_\tau, \tilde{\mathbf{V}}_\tau) \geq L(\mathbf{V}_\tau)$, $A(\mathbf{V}_\tau, \mathbf{V}_\tau) = L(\mathbf{V}_\tau)$, and the Hessian matrix $\nabla\nabla_{\mathbf{V}_\tau} A(\mathbf{V}_\tau, \tilde{\mathbf{V}}_\tau) \succeq 0$. Due to limited space, we omit the details of the validation.

Based on Theorem 4.2, we can minimize $A(\mathbf{V}_\tau, \tilde{\mathbf{V}}_\tau)$ with respect to $\mathbf{V}_\tau$ with $\tilde{\mathbf{V}}_\tau$ fixed. Set $\nabla_{\mathbf{V}_\tau} A(\mathbf{V}_\tau, \tilde{\mathbf{V}}_\tau) = \mathbf{0}$ we get the following updating formula

$$\mathbf{V}_\tau \leftarrow \mathbf{V}_\tau \circ \sqrt{\frac{[\mathbf{X}_\tau^{\mathrm{T}}[\mathbf{U}, \mathbf{U}_\tau]\mathbf{H} + \mathbf{\Lambda}_\tau \mathbf{1}_{n_\tau} \mathbf{1}_c^{\mathrm{T}}]}{\left[\tilde{\mathbf{V}}_\tau \mathbf{H}^{\mathrm{T}}[\mathbf{U}, \mathbf{U}_\tau]^{\mathrm{T}}[\mathbf{U}, \mathbf{U}_\tau]\mathbf{H} + \mathbf{\Lambda}_\tau \tilde{\mathbf{V}}_\tau \mathbf{1}_c \mathbf{1}_c^{\mathrm{T}}\right]}}$$

which is consistent with the updating formula derived from the KKT condition aforementioned.

By Lemma 4.1 and Theorem 4.2, for each subsequent iteration of updating $\mathbf{V}_\tau$, we have $L(\mathbf{V}_\tau^0) = Z(\mathbf{V}_\tau^0, \mathbf{V}_\tau^0) \geq Z(\mathbf{V}_\tau^1, \mathbf{V}_\tau^0) \geq Z(\mathbf{V}_\tau^1, \mathbf{V}_\tau^1) = L(\mathbf{V}_\tau^1) \geq \ldots \geq L(\mathbf{V}_\tau^{maxIter})$. So $L(\mathbf{V}_\tau)$ is monotonically decreasing. This is also true for the other variables. Since objective $\mathcal{L}$ is lower bounded by 0, the correctness and convergence of Theorem 4.1 and Algorithm 1 are proved.

5 Experiments

In this section, we conduct experiments on two benchmark data sets to evaluate the proposed DTL algorithm and compare it with several state-of-the-art unsupervised, semi-supervised, and transfer learning methods.

5.1 Data Sets and Evaluation Criteria The cross-domain data sets for classification shown in Table 2 are generated from 20-NewsGroups[1] and Reuters-21578[2] utilizing their hierarchy structures following the work [6]. On one hand, data sets constructed this way are divergent across domains since they contain different subcategories. On the other hand, these data sets are also related since they are from the same top categories. Such data sets have also been widely used in previous transfer learning studies as benchmarks for performance evaluation [6, 9, 15, 21, 24, 25].

20-NewsGroups This data set contains 20,000 documents distributed evenly in 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 higher level, while others remain as separate categories. For example, the top category *sci* contains 4 subcategories *sci.crypt*, *sci.electronics*, *sci.med* and *sci.space* in the science field. Any two top categories can be selected to construct the cross-domain data set. The source domain contains some subcategories from the two top categories. The target domain contains the rest of the subcategories. The details of the constructed data sets are listed in Table 2, among which the first 6 data sets are generated from 20-NewsGroups. We preprocess the 20-NewsGroups data by removing stop words and words occurring in less than 15 documents. There are 15,981 features left after preprocess.

Table 2: Cross-domain data sets generated from 20-NewsGroups and Reuters-21578.

Data Set	Source Domain	Target Domain
comp vs rec	comp.{graphics, os} rec.{autos, motorcycles}	comp.sys.{ibm, mac} rec.sport.{baseball, hokey}
comp vs sci	comp.{graphics, os} sci.{crypt, med}	comp.sys.{ibm, mac} sci.{electronics, space}
comp vs talk	comp.{graphics, os} politics.{guns, mideast}	comp.sys.{ibm, mac} talk.{politics.misc, religion}
rec vs sci	rec.{autos, motorcycles} sci.{crypt, med}	rec.sport.{baseball, hokey} sci.{electronics, space}
rec vs talk	rec.{autos, motorcycles} politics.{guns, mideast}	rec.sport.{baseball, hokey} talk.{politics.misc, religion}
sci vs talk	sci.{crypt, med} politics.{guns, mideast}	sci.{electronics, space} talk.{politics.misc, religion}
orgs vs people	orgs.{...}, people.{...}	orgs.{...}, people.{...}
orgs vs place	orgs.{...}, place.{...}	orgs.{...}, place.{...}
people vs place	people.{...}, place.{...}	people.{...}, place.{...}

Reuters-21578 This data set contains 5 top categories and many subcategories. We directly use the preprocessed data in [9], which are generated from Reuters-21578 for testing cross-domain learning algorithms. The preprocessed data consists of three data sets *orgs vs people*, *orgs vs place* and *people vs place* as shown in Table 2.

We use the *Accuracy* of predicting unlabeled data in the target domains as the evaluation criteria. It has been widely used in the literature [6, 15, 21, 24, 25].

$$Accuracy = \frac{\left|\left\{\mathbf{x} : \mathbf{x} \in \{\mathcal{D}_\tau\}_{\tau=s+1}^t \wedge f(\mathbf{x}) = y(\mathbf{x})\right\}\right|}{\left|\left\{\mathbf{x} : \mathbf{x} \in \{\mathcal{D}_\tau\}_{\tau=s+1}^t\right\}\right|}$$

where $y(\mathbf{x})$ is the true label of example $\mathbf{x}$, and $f(\mathbf{x})$ is the label predicted by the classification algorithm.

5.2 Baseline Methods and Parameter Settings The cross-domain classification task is performed in a binary-class setting. One source domain and one target domain are used in each experiment following the same setting as in [6, 21, 25, 20]. Note that due to its general formulation, DTL can be easily applied to multi-class classification problems in multiple domains.

Several state-of-the-art methods are compared in the experiments: (1) Unsupervised method Nonnegative Matrix Factorization (NMF) [13]. It is directly applied to the target domain data. (2) Supervised method, including Support Vector Machine[3] (SVM) and Logistic Regression[4] (LR). They are trained on the source domain data and tested on the target domain data. (3) Semi-supervised method Transductive Support Vector Machine (TSVM) [12]. It works in a transductive setting using all data. (4) Transfer learning method, in-

[1] http://people.csail.mit.edu/jrennie/20Newsgroups

[2] http://www.daviddlewis.com/resources/testcollections/reuters21578

[3] http://www.csie.ntu.edu.tw/~cjlin/libsvm

[4] http://research.microsoft.com/en-us/um/people/minka/papers/logreg

Table 3: Average classification accuracy (%) on cross-domain data sets (10 repeated experiments).

Data Set	NMF	SVM	LR	TSVM	CoCC	MTrick	DKT	DTL
comp vs rec	95.79	83.50	87.22	90.20	95.80	94.18	95.20	**98.53**
comp vs sci	57.17	68.30	80.77	81.70	87.00	87.57	88.46	**95.25**
comp vs talk	53.73	89.70	95.30	90.30	98.00	93.20	94.35	**99.85**
rec vs sci	78.57	78.80	75.70	93.80	94.50	97.99	97.55	**98.68**
rec vs talk	54.49	76.70	93.95	96.00	96.50	98.71	98.35	**99.35**
sci vs talk	53.65	77.40	89.98	89.20	94.60	96.37	96.50	**98.32**
orgs vs people	63.29	74.25	74.88	73.80	79.79	80.80	80.95	**82.56**
orgs vs place	72.63	69.99	71.89	69.89	74.18	76.77	76.90	**78.46**
people vs place	60.49	59.05	58.06	58.43	66.94	69.02	69.32	**70.28**

cluding Co-Clustering based Classification (CoCC) [6], Matrix Tri-Factorization based Classification (MTrick) [25], and the recently proposed Dual Knowledge Transfer (DKT) [20]. They are trained on all data and are tested on the target domain data. All these methods are tested using their optimal parameter settings reported in the original papers.

Note that, though being closely related to our work, Collaborative Dual-PLSA (CD-PLSA) [24], Joint Subspace Nonnegative Matrix Factorization (JSNMF) [10], and Domain Distribution Adaptation via Kernel Mapping (KMap) [22] are not compared due to the following reasons. CD-PLSA is the probabilistic reformulation of MTrick [25] with similar performance. We only need to evaluate one of them. JSNMF is originally designed to handle the cross-domain retrieval task. It does not involve learning the conditional distribution for training classifiers, thus cannot be directly used to solve the cross-domain classification task. KMap requires a few labeled data available in the target domain for inducing transfer learners for that domain. It cannot be applied to transductive transfer learning, where no labeled data are available in the target domain.

The parameters of DTL, i.e., the number of total and common feature clusters k and κ, are tuned on data set *comp vs sci* by cross validation. Then the tuned parameters are applied to all other data sets. For simplicity, we keep $k = 20$ fixed, since the performance of DTL is very stable with respect to it. Therefore, in the comparison experiments, we set $t = 2$, $s = 1$, $k = 20$, $\kappa = 10$, $c = 2$, $maxIter = 50$.

5.3 Experimental Results Table 3 shows the comparison results, where the average classification accuracy of each method on each data set is computed over 10 repeated experiments. Figure 1 gives a more intuitive visualization of these results. From the results, we have several key observations.

5.3.1 Unsupervised Method NMF performs well on the data sets (e.g., *comp vs rec*) where data are already well separated and the cluster structures are consistent with the classes. However, when data are unseparated or inconsistent (e.g., *comp vs sci*), NMF performs poorly. This indicates that additional supervision is needed to improve the classification accuracy. In cross-domain classification problems, we can leverage knowledge from labeled data in the source domains.

5.3.2 Supervised Method SVM and LR trained on source domain data fail to discriminate target domain data on some data sets (e.g., *rec vs sci*). The reason is that they assume that the training data and test data follow identical probability distribution. When this assumption is violated, their performance may drop dramatically. In the worst case, these methods will even underperform unsupervised clustering methods such as NMF (e.g., on the *comp vs rec* data set). The results also indicate that SVM is more likely to have degraded performance in cross-domain classification than LR. The reason is that the decision boundary of SVM is determined by support vectors that are more likely to be influenced by divergent distributions across domains.

5.3.3 Semi-Supervised Method TSVM outperforms NMF, SVM and LR on many data sets. This verifies that the unlabeled data in target domains can indeed help the classifier fit the unseen data better, which is a major advantage of semi-supervised learning. However, TSVM performs worse when data across domains are significantly different and unseparated, (e.g., on the *comp vs sci* and *sci vs talk* data sets). This is because its identical distribution assumption is violated. Therefore, treating data from different domains as if they were drawn from a homogenous body typically leads to poor performance. The traditional no-transfer methods do not perform well for the cross-domain classification task.

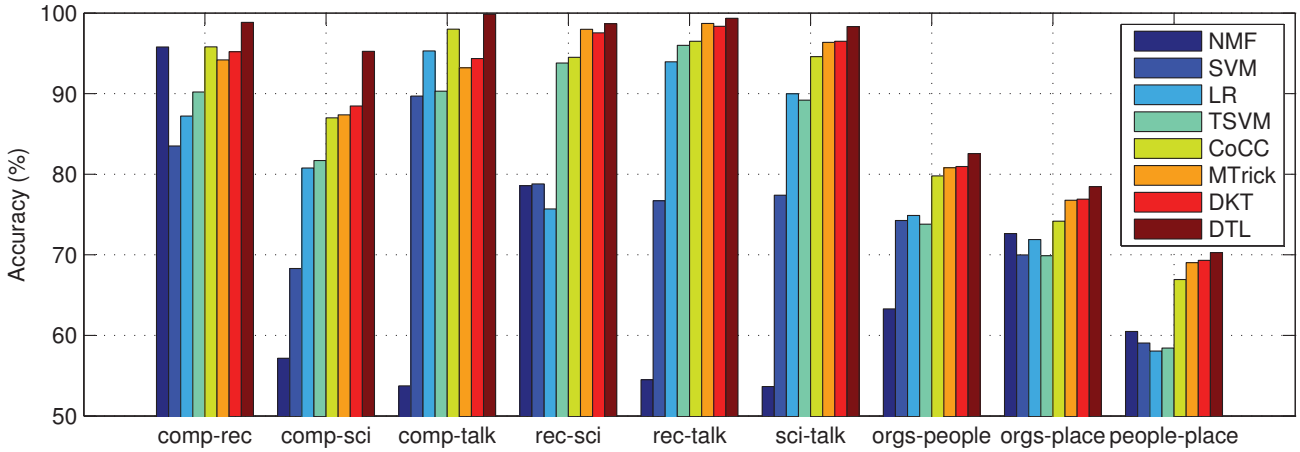

Figure 1: Experimental results of average classification accuracy (%) on cross-domain data sets (10 repeated experiments). The results demonstrate the superior performance of the proposed DTL method.

5.3.4 Transfer Learning Method CoCC, MTrick, and DKT generally perform better than the no-transfer methods. These transfer learning methods try to classify data that are not well separated by leveraging knowledge from the source domains (e.g., *sci vs talk*), while avoiding degraded performance when data are already well separated (e.g., *comp vs rec*). However, these methods have not reached the best performance for all data sets (e.g., *comp vs sci*). The reasons are two folds. (1) These methods are based on a strict assumption that are prone to negative transfer in the data set where the marginal distribution can only be partially drawn closer across domains (e.g., *comp vs rec*). (2) These methods, except DKT, adopt a single bridge transfer mechanism, without exploiting the duality between the marginal and conditional distributions for mutual reinforcement. For example, CoCC only transfers knowledge from the marginal distribution. MTrick only transfers knowledge from the conditional distribution. Thus they might not reach optimal transfer capability.

Note that, Dual Knowledge Transfer (DKT) is the first attempt to discover two possible paths and use them both for knowledge transfer. However, it does not address the duality property between the marginal distribution and the conditional distribution, which can reinforce the learning of each other. It does not solve dual transfer learning in a principled way using a unified subspace learning as our method does. The two paths for transferring knowledge in DKT are separated into a matrix factorization plus a regularization process, which require more trade-off parameters and have less power for mutual reinforcement. Therefore, the performance improvement of DKT over its baseline MTrick is not as significant as our method. These observations are also consistent with DKT's original work [20].

5.3.5 Dual Transfer Learning Method Our method DTL achieves the highest classification accuracy on all data sets by taking advantage of dual transfer learning mechanism. The reason is that DTL can enhance the transfer capability by exploiting the duality between the marginal distribution and conditional distribution and learning them simultaneously for mutual reinforcement. In particular, if there exists some knowledge that can be transferred, DTL will transfer the common knowledge from the marginal distribution and the conditional distribution across domains as much as possible. Otherwise, DTL will leave it as domain-specific knowledge to avoid negative transfer.

The results also show that DTL performs well even on those hard-to-classify data sets, such as *comp vs sci*. The *comp vs sci* data set is difficult due to the unbalance underlying the feature clusters and example clusters. The data distributions in *comp* category are similar across domains, while the data distributions in *sci* category are very different across domains. If we transfer all the feature clusters across domains, the domain-specific ones (mainly from the *sci* category) will result in negative transfer. On the other hand, if we transfer only the common feature clusters (mainly from the *comp* category), the stable conditional relation between the domain-specific feature clusters and the example classes is lost. This will lead to ineffective transfer. DTL tackles these two difficulties in a principled way and achieves superior performance.

5.4 Effectiveness of Dual Transfer Learning We compare DTL with CoCC, MTrick, and DKT to show

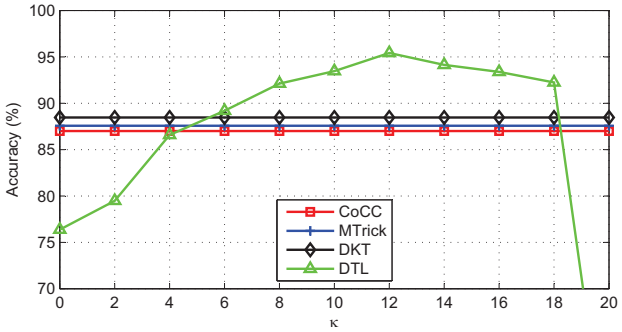

Figure 2: Accuracy of CoCC, MTrick and DTL with respect #common feature clusters κ ($k = 20$) on data set *comp vs sci*, which verifies the effectiveness of the duality property addressed by this work.

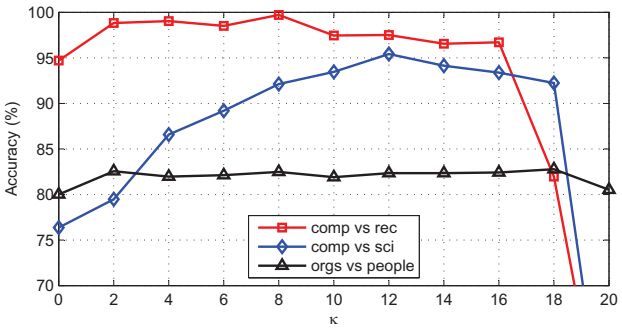

Figure 3: Accuracy of DTL with respect to #common feature clusters κ ($k = 20$) on data sets *comp vs rec*, *comp vs sci*, *orgs vs people*, which shows DTL is stable with respect to a wide range of parameter settings.

that the duality between the marginal distribution and conditional distribution indeed brings effectiveness to transfer learning. For a fair comparison, the number of total feature clusters is set to $k = 20$ for all four methods. For DTL in particular, the number of common feature clusters can vary in the interval $\kappa \in [0, k]$. Figure 2 shows the accuracy of the four methods on data set *comp vs sci* as a function of the parameter κ. From the figure, we see that when there exists some common knowledge which can be transferred across domains (i.e., $\kappa \in [8, 16]$), DTL is more effective than the other three methods. By exploiting the duality between the two distributions, DTL achieves optimal transferability.

5.5 Parameter Sensitivity We evalutate parameter sensitivity of DTL with respect to κ fixing $k = 20$. Figures 3 shows the accuracy of DTL on data sets *comp vs rec*, *comp vs sci* and *orgs vs people* with respect to parameter κ. It can be seen that when the parameter

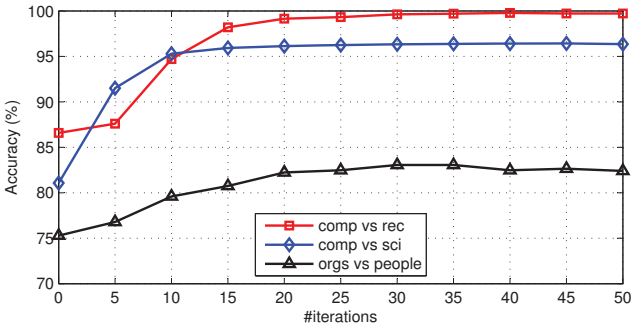

Figure 4: Accuracy of DTL with respect to #iterations on data sets *comp vs rec*, *comp vs sci*, *orgs vs people*, which shows that DTL converges with guarantee.

varies in a wide range $\kappa \in [8, 16]$, DTL performs quite stably and consistently outperforms the baseline methods by a large margin. When $\kappa \to 0$, the learning of the marginal distribution vanishes and DTL degenerates to single bridge transfer. When $\kappa \to 20$, all the feature clusters are used as the common ones while the domain-specific ones are not well fitted. Thus DTL suffers from negative transfer when κ/k is large. Nevertheless, by exploiting the duality between the marginal distribution and conditional distribution, the adaptiveness of the transfer process is greatly enhanced. Therefore, DTL is generally not sensitive to parameter κ, except the extreme cases when $\kappa \to 0$ or $\kappa \to k$.

5.6 Algorithm Convergence Since DTL employs an iterative algorithm, an important issue is its convergence property. We check the convergence of DTL empirically by testing it on several data sets: *comp vs rec*, *comp vs sci*, and *orgs vs people*. Figure 4 shows the accuracy with respect to the number of iterations. We see that the accuracy of DTL increases with more iterations and it usually converges after 50 iterations.

6 Conclusion

In this paper, we present a novel dual transfer learning (DTL) method based on the duality between the marginal and conditional distributions. We observe that learning one distribution can help to learn the other for mutual reinforcement. DTL is formulated as an optimization problem of joint nonnegative matrix trifactorizations. This integrated formulation can naturally explore the duality property. An efficient algorithm is developed to solve the optimization problem. We conduct extensive experiments on benchmark data sets to compare DTL with several state-of-the-art methods. Experimental results demonstrate that DTL outperforms alternative methods in all data sets.

7 Acknowledgments

The work is supported by the National HeGaoJi Key Project (No. 2010ZX01042-002-002-01), the National Basic Research Program of China (973 Program) (No. 2009CB320706), and the National Natural Science Foundation of China (No. 61050010, No. 61073005, No. 60972096). We would like to thank the anonymous reviewers and the shepherd for their insightful comments.

References

[1] M. Belkin, P. Niyogi, and V. Sindhwani, *Manifold regularization: A geometric framework for learning from labeled and unlabeled examples*, Journal of Machine Learning Research, 7 (2006), pp. 2399–2434.

[2] J. Blitzer, R. McDonald, and F. Pereira, *Domain adaptation with structural correspondence learning*, in Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, EMNLP, 2006.

[3] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.

[4] B. Cao, S. J. Pan, Y. Zhang, D.-Y. Yeung, and Q. Yang, *Adaptive transfer learning*, in Proceedings of the 24th AAAI Conference on Artificial Intelligence, AAAI, 2010.

[5] W. Dai, O. Jin, G.-R. Xue, Q. Yang, and Y. Yu, *Eigentransfer: A unified framework for transfer learning*, in Proceedings of the 26th International Conference on Machine Learning, ICML, 2009.

[6] W. Dai, G.-R. Xue, Q. Yang, and Y. Yu, *Co-clustering based classification for out-of-domain documents*, in Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD, 2007.

[7] W. Dai, Q. Yang, G.-R. Xue, and Y. Yu, *Boosting for transfer learning*, in Proceedings of the 24th International Conference on Machine Learning, ICML, 2007.

[8] C. Ding, T. Li, W. Peng, and H. Park, *Orthogonal nonnegative matrix tri-factorizations for clustering*, in Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD, 2006.

[9] J. Gao, W. Fan, J. Jiang, and J. Han, *Knowledge transfer via multiple model local structure mapping*, in Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD, 2008.

[10] S. K. Gupta, D. Phung, B. Adams, T. Tran, and S. Venkatesh, *Nonnegative shared subspace learning and its application to social media retrieval*, in Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD, 2010.

[11] J. Jiang and C. Zhai, *Instance weighting for domain adaptation in nlp*, in Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, ACL, 2007.

[12] T. Joachims, *Transductive inference for text classification using support vector machines*, in Proceedings of the 16th International Conference on Machine Learning, ICML, 1999.

[13] D. D. Lee and H. S. Seung, *Algorithms for non-negative matrix factorization*, in Proceedings of Neural Information Processing Systems 14, NIPS, 2000.

[14] X. Ling, W. Dai, G.-R. Xue, Q. Yang, and Y. Yu, *Spectral domain-transfer learning*, in Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD, 2008.

[15] M. Long, W. Cheng, X. Jin, J. Wang, and D. Shen, *Transfer learning via cluster correspondence inference*, in Proceedings of the 10th IEEE International Conference on Data Mining, ICDM, 2010.

[16] S. J. Pan, X. Ni, J.-T. Sun, Q. Yang, and Z. Chen, *Cross-domain sentiment classification via spectral feature alignment*, in Proceedings of the 19th International Conference on World Wide Web, WWW, 2010.

[17] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, *Domain adaptation via transfer component analysis*, IEEE Transactions on Neural Networks, 22 (2011), pp. 199–210.

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

[19] F. Wang, T. Li, and C. Zhang, *Semi-supervised clustering via matrix factorization*, in Proceedings of the 8th SIAM International Conference on Data Mining, SDM, 2008.

[20] H. Wang, H. Huang, F. Nie, and C. Ding, *Cross-language web page classification via dual knowledge transfer using nonnegative matrix tri-factorization*, in Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR, 2011.

[21] Z. Wang, Y. Song, and C. Zhang, *Knowledge transfer on hybrid graph*, in Proceedings of the 21st International Joint Conference on Artificial Intelligence, IJCAI, 2009.

[22] E. Zhong, W. Fan, J. Peng, K. Zhang, J. Ren, D. Turaga, and O. Verscheure, *Cross domain distribution adaptation via kernel mapping*, in Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD, 2009.

[23] X. Zhu, *Semi-supervised learning literature survey*, tech. report, 2008.

[24] F. Zhuang, P. Luo, Z. Shen, Q. He, Y. Xiong, Z. Shi, and H. Xiong, *Collaborative dual-plsa mining distinction and commonality across multiple domains for text classification*, in Proceedings of the 19th ACM International Conference on Information and Knowledge Management, CIKM, 2010.

[25] 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 Proceedings of the 10th SIAM International Conference on Data Mining, SDM, 2010.