Transfer Learning with Graph Co-Regularization

Mingsheng Long, Jianmin Wang, Guiguang Ding, Dou Shen, and Qiang Yang, *Fellow, IEEE*

Abstract—Transfer learning is established as an effective technology to leverage rich labeled data from some source domain to build an accurate classifier for the target domain. The basic assumption is that the input domains may share certain knowledge structure, which can be encoded into common latent factors and extracted by preserving important property of original data, e.g., statistical property and geometric structure. In this paper, we show that different properties of input data can be complementary to each other and exploring them simultaneously can make the learning model robust to the domain difference. We propose a general framework, referred to as *Graph Co-Regularized Transfer Learning* (GTL), where various matrix factorization models can be incorporated. Specifically, GTL aims to extract common latent factors for knowledge transfer by preserving the statistical property across domains, and simultaneously, refine the latent factors to alleviate negative transfer by preserving the geometric structure in each domain. Based on the framework, we propose two novel methods using NMF and NMTF, respectively. Extensive experiments verify that GTL can significantly outperform state-of-the-art learning methods on several public text and image datasets.

Index Terms—Transfer learning, negative transfer, graph regularization, matrix factorization, text mining, image classification

1 Introduction

The exponential growth of big data from a variety of domains has created a compelling demand for innovative methods to analyze and manage them. Unfortunately, for some newly-emerged target domains, labeled data are usually very sparse, making standard supervised learning algorithms infeasible. Moreover, collecting sufficient labeled data from scratch is very expensive. One may expect to leverage the abundant labeled data readily available in some related source domains for training an accurate classifier in the target domains. However, standard supervised learning algorithms cannot reuse the labeling knowledge across domains effectively, since they fundamentally require the training and test data to be sampled from the same distribution. Recently, the literature has witnessed an increasing interest in developing *transfer learning* [2] algorithms for cross-domain knowledge transfer problems. Transfer learning has proven to be promising in real-world applications, e.g., text categorization [3], sentiment analysis [4], image classification [5], video summarization [6], and collaborative filtering [7], etc.

One major computational problem of transfer learning is how to explore the shared knowledge structure underlying input domains as the bridge to propagate supervision information from the source domains to the target domains. Recent works focus on encoding the knowledge structure into common latent factors and extracting them by preserving specific property of the original data: 1) preserving the *statistical property*, i.e., maximizing embedded variance or minimizing reconstruction error [3], [8]–[12]; and 2) preserving the *geometric structure*, i.e., encoding similar examples with similar representations [13]–[17]. Specifically, the statistical property here refers to the *descriptive statistics* of input data, e.g., sample variance, or global variability [18]. The geometric structure refers to the embedded *manifold*, which supports the intrinsic distribution of input data and locally looks like a flat low-dimensional Euclidean space [19].

The main limitation of most prior transfer learning methods is that they do not simultaneously preserve both the statistical property and geometric structure. In reality, preserving these complementary properties together is important to make learning models robust to the domain difference. In some difficult scenarios, the intrinsic domain structure cannot be effectively explored with a single property of data. In this case, the prior methods may suffer from *ineffective transfer*, i.e., underfitting the target data. In other difficult scenarios, the domain difference can be so large that it may be difficult to extract common factors as the bridge for knowledge transfer. In this case, the prior methods may suffer from *negative transfer*, i.e., overfitting the target data. These issues motivate us to design a framework to explore both the statistical property and geometric structure for robust transfer learning.

- *M. Long is with the School of Software, Tsinghua University, Beijing 100084, China, and also with the Department of Computer Science, Tsinghua University, Beijing 100084, China. E-mail: longmingsheng@gmail.com.*
- *J. Wang and G. Ding are with the School of Software, Tsinghua University, Beijing 100084, China. E-mail: {jimwang, dinggg}@tsinghua.edu.cn.*
- *D. Shen is with Baidu, Beijing 100085, China. E-mail: doushen@live.com.*
- *Q. Yang is with Noah's Ark Lab, Huawei, and Hong Kong University of Science and Technology, Hong Kong. E-mail: qyang@cse.ust.hk.*

Manuscript received 24 Oct. 2012; revised 11 Mar. 2013; accepted 11 June 2013. Date of publication 18 June 2013; date of current version 9 July 2014.
Recommended for acceptance by G. Karypis.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier 10.1109/TKDE.2013.97

For the *ineffective transfer* problem, inspired by Zhu *et al.* [18], the statistical and geometric properties may focus on different aspects of the original data and are complementary to each other in reality. The justifications are as follows. On one hand, each data point may be associated with some latent factors. For example, a text document can be regarded as a combination of several hidden semantics. Extracting these latent factors involves preserving the statistical property of the original data [20]. On the other hand, from the geometric perspective, the data points may be sampled from a distribution supported by a low-dimensional manifold embedded in a high-dimensional space [19]. Preserving this geometric structure involves encoding similar examples or features with similar embeddings. By preserving the statistical and geometric properties together, we can improve the smoothness of latent factors and enhance the effectiveness of transfer learning.

For the *negative transfer* problem, we put forward the following justifications. When the source and target domains are different enough, it is impossible to extract some latent factors common to both domains. If we forcefully extract some "common" latent factors, then they may result in the inconsistency between the target domain cluster structure and the source domain discriminating structure. To alleviate this problem, we propose to preserve the geometric structure in each domain. Thus, the geometric structure of target domains will be respected even if it is contradicted with the common factors transferred from source domains.

In this paper, we propose a general framework, referred to as Graph Co-Regularized Transfer Learning (GTL), to achieve more effective and robust transfer learning. Specifically, GTL aims to extract some common latent factors for knowledge transfer by preserving the statistical property across domains, and simultaneously, refine the latent factors to alleviate negative transfer by preserving the geometric structure in each domain. The key assumptions of GTL are as follows: 1) by preserving the statistical and geometric properties simultaneously, we can improve the smoothness of latent factors and enhance effective transfer learning; 2) by preserving the geometric structure in each domain, the domain-specific geometric structure can be respected to alleviate negative transfer. The main contributions of this paper are summarized as follows.

- To cope with the considerable change in data distributions from different domains, GTL aims to simultaneously preserve the statistical property and geometric structure in a unified framework. The learning goal of GTL is to enhance effective transfer and alleviate negative transfer. To the best of our knowledge, GTL is the first transfer learning framework which has explored the criteria simultaneously to achieve the desirable learning goal.
- Many existing matrix factorization models, e.g., NMF [21] and Semi-NMF [22], can be readily incorporated into the GTL framework to tackle transfer learning problems. The implemented methods can be optimized by multiplicative update rules.
- Under the GTL framework, we further propose two new methods based on NMF [21] and Nonnegative Matrix Tri-Factorization (NMTF) [23], respectively. Both methods perform effectively for cross-domain text and image classification tasks.
- Comprehensive experiments on text (Reuters-21578 and 20-Newsgroups) and image (PIE, USPS, MNIST, MSRC, and VOC2007) datasets verify the effectiveness of GTL in real-life applications.

The rest of the paper is organized as follows. The related works are reviewed in Section 2. In Section 3, we present the general framework, implement two learning algorithms using NMF and NMTF, and analyze their computational complexity. In Section 4, we formally analyze the convergence property of the new algorithms. The experimental evaluations are discussed in Section 5. Finally, we conclude the paper in Section 6.

2 Related Work

Transfer learning is established to be effective in the applications where the training data and test data are obtained from different resources and with different distributions. Based on the literature survey [2], most transfer learning methods can be roughly organized into two categories: *instance re-weighting* [24], [25] and *feature extraction*. Our approach belongs to the feature extraction category, which can be roughly reorganized into three subcategories: correspondence learning, distribution matching, and preservation of data property.

Correspondence Learning: Well-known methods include Structural Correspondence Learning (SCL) [26] and Spectral Feature Alignment (SFA) [4]. These methods first identify the correspondence among non-pivot features by modeling their correlations with pivot features that appear frequently in both domains, and explore the correspondence for subspace learning.

Distribution Matching: Well-known methods are Maximum Mean Discrepancy Embedding (MMDE) [27], Transfer Component Analysis (TCA) [15], and Transfer Subspace Learning (TSL) [28]. These methods aim to extract a shared feature subspace in which the difference in distributions across domains can be reduced by minimizing predefined distance measures.

Property Preservation: A majority of the feature extraction based transfer learning methods belong to this subcategory. These methods assume that there exists common knowledge structure underlying multiple domains, which can be encoded into common latent factors and explored as the bridge for knowledge transfer across domains. The latent factors can be extracted by preserving important properties of input data, e.g., *statistical property* [3], [8]–[12], [29], and *geometric structure* [13]–[17].

Preservation of Statistical Property: Preserving the statistical property can be reduced to maximizing the empirical likelihood, or minimizing the reconstruction error, of the model parameters to fit input data statistics. *Collective Matrix Factorization* (CMF) proposed by Singh *et al.* [30] and its tri-factorization variants have been extensively studied for transfer learning recently [3], [5], [9]–[12]. CMF simultaneously factorizes multiple matrices with correspondences between rows and columns while enforcing a set of common latent factors that match rows and columns

across different matrices. In this process, the common latent factors are shared as the bridge for knowledge transfer. Wang *et al.* [9] proposed Label Propagation (LP) to share the feature clusters as the bridge for knowledge transfer. Zhuang *et al.* [3], [10] developed Matrix Tri-Factorization based Classification (MTrick) to share the associations between feature clusters and example classes for knowledge transfer. In essence, all CMF-based methods are underpinned by maximizing the empirical likelihood across multiple domains [30]. They only explore the statistical property of data, and may suffer from underfitting when data structure is complicated. Different from these methods, our GTL simultaneously explores both the statistical property and the geometric structure to discover more connections between domains, and builds up a better bridge.

Preservation of Geometric Property: Preserving the geometric structure can be executed by exploring the *local invariance assumption* [19], i.e., similar examples or features should have similar embeddings. Recently, several methods have been proposed to explore the geometric structure for transfer learning [13]–[17]. These methods aim to maximize the consistency between the in-domain geometric structure and out-of-domain discriminating structure using spectral learning. Contrary to the CMF-based methods, these methods focus only on the geometric structure and do not explore the statistical property. Different from these methods, our GTL explores the geometric structure underlying both example and feature spaces, and simultaneously, it explores the statistical property across domains. Thus, we can take advantage of both sets of methods to establish robust transfer learning.

3 Graph Co-Regularized Transfer Learning Framework

In this section, we first define the problem setting and learning goal for transfer learning. Then we present our Graph Co-Regularized Transfer Learning (GTL) framework. Based on the framework, we propose two novel methods using NMF and NMTF, respectively. Finally, we will analyze the computational complexity.

3.1 Problem Definition

We focus on *transductive* transfer learning: rich labeled data are available in source domains and only unlabeled data are available in target domains. We study *multi-class* classification on one source and one target domains, which can be extended to *multiple* domains.

Denote $\mathcal{D}_\pi$ the πth domain, where $\pi \in \Pi$ is the domain index. To distinguish different types of domains, we partition Π into source domain indices Π_s and target domain indices Π_t, i.e., $\Pi = \Pi_s \cup \Pi_t$, $\Pi_s \cap \Pi_t = \emptyset$. The domains share identical feature space $\mathcal{X}$ and label space $\mathcal{Y}$ with $|\mathcal{X}| = m$ features and $|\mathcal{Y}| = c$ labels, respectively. Denote $\mathbf{X}_\pi = [\mathbf{x}^\pi_{*1}, \ldots, \mathbf{x}^\pi_{*n_\pi}] \in \mathbb{R}^{m \times n_\pi}$ the feature-example matrix of domain $\mathcal{D}_\pi$, where $\mathbf{x}^\pi_{*i}$ is the ith example in domain $\mathcal{D}_\pi$. Denote $\mathbf{Y}_\pi \in \mathbb{R}^{n_\pi \times c}$ the label matrix of source domain $\mathcal{D}_\pi$, where $y^\pi_{ij} = 1$ if $\mathbf{x}^\pi_{*i}$ is assigned to class j, and $y^\pi_{ij} = 0$ otherwise. Frequently used notations are summarized in Table 1.

TABLE 1
Frequently Used Notations and Their Descriptions

Notation	Description	Notation	Description
$\mathcal{D}_\pi$	domain π, $\pi \in \Pi$	$\mathbf{X}_\pi$	$m \times n_\pi$ data matrix of $\mathcal{D}_\pi$
n_π	#examples in $\mathcal{D}_\pi$	$\mathbf{Y}_\pi$	$n_\pi \times c$ label matrix of $\mathcal{D}_\pi$
m, c	#features, #classes	$\mathbf{U}_\pi$	c feature clusters in $\mathcal{D}_\pi$
p	#nearest neighbors	$\mathbf{V}_\pi$	c example classes in $\mathcal{D}_\pi$
λ	feature graph reg.	$\mathbf{H}_\pi$	associations between $\mathbf{U}_\pi$ & $\mathbf{V}_\pi$
γ	example graph reg.	$\mathbf{L}^u_\pi$	feature graph Laplacian in $\mathcal{D}_\pi$
σ	orthogonality reg.	$\mathbf{L}^v_\pi$	example graph Laplacian in $\mathcal{D}_\pi$

Problem 1 (Learning Goal). *Given domains $\{\mathcal{D}_\pi\}_{\pi\in\Pi}$, learn a multi-class classifier $f{:}\mathcal{X} \mapsto \mathcal{Y}$ with low error rate on target domains $\{\mathcal{D}_\pi\}_{\pi\in\Pi_t}$, by simultaneously 1) preserving the statistical property across domains to facilitate knowledge transfer, and 2) preserving the geometric structure in each domain to alleviate negative transfer.*

In this paper, we propose a general framework, referred to as Graph Co-Regularized Transfer Learning (GTL), to achieve the learning goal. In GTL , we adopt a *regularized matrix factorization* technique. We assume that the input domains can share some common latent factors. We extract the common factors by *collective matrix factorization*, which can preserve the statistical property of the input data across domains. Simultaneously, we refine the common factors by *graph co-regularization*, which can preserve the geometric property of the input data in each domain. In this way, the learning model is made more robust to the domain difference. Our proposed justifications are two-folds: 1) if the statistical and geometric properties are consistent across domains, they can reinforce learning each other to enhance knowledge transfer; 2) otherwise, the in-domain geometric property will dominate learning task within each domain to alleviate negative transfer.

3.2 General Framework

The general GTL framework integrates two learning objectives into a unified optimization problem: collective matrix factorization and graph co-regularization.

3.2.1 Collective Matrix Factorization

First, we extract the latent factors by collective matrix factorization [30], through which data distributions between domains can be drawn close. We aim to preserve the statistical property of data across domains.

Collective Matrix Factorization: The latent factors in each domain $\mathcal{D}_\pi$ can be extracted by nonnegative matrix factorization (NMF) model [21], [22]. In NMF, a feature-example matrix $\mathbf{X}_\pi$ is decomposed into two low-rank nonnegative matrices $\mathbf{U}_\pi$ and $\mathbf{V}_\pi$, such that the reconstruction error of matrix $\mathbf{X}_\pi$ is minimized and the statistical property of input data is preserved. NMF amounts to the following optimization problem:

$$\min_{\mathbf{U}_\pi, \mathbf{V}_\pi \geq \mathbf{0}} \mathcal{L}\left(\mathbf{X}_\pi, h\left(\mathbf{U}_\pi \mathbf{V}_\pi^{\mathrm{T}}\right)\right), \tag{1}$$

where h is the prediction link and $\mathcal{L}$ is the loss function. $\mathbf{U}_\pi = [\mathbf{u}^\pi_{*1}, \ldots, \mathbf{u}^\pi_{*c}] \in \mathbb{R}^{m \times c}$ is the feature cluster matrix, with each $\mathbf{u}^\pi_{*i}$ representing a *feature cluster*; and $\mathbf{V}_\pi = [\mathbf{v}^\pi_{*1}, \ldots, \mathbf{v}^\pi_{*c}] \in \mathbb{R}^{n_\pi \times c}$ is the example class matrix, with each $\mathbf{v}^\pi_{*i}$ representing an *example class*. Intuitively, $\mathbf{U}_\pi$

and $\mathbf{V}_\pi$ are the co-clustering results for $\mathbf{X}_\pi$ on features and examples, respectively. According to Ding *et al.* [20], NMF models are equivalent to maximizing the empirical likelihood of the input data.

Given multiple domains with intrinsic correlations, we may improve classification accuracy by exploiting supervision information from labeled source domains and classifying unlabeled target domains, by sharing the *common factors* underlying these domains. This is the nature of transfer learning. Singh *et al.* [30] extend basic MF to simultaneously factorize multiple relevant matrices, leading to *collective matrix factorization* (CMF)

$$\min_{\mathbf{U}_\pi \in \mathcal{C}_u, \mathbf{V}_\pi \in \mathcal{C}_v} \sum\nolimits_{\pi \in \Pi} \mathcal{L}\left(\mathbf{X}_\pi, h\left(\mathbf{U}_\pi \mathbf{V}_\pi^{\mathrm{T}}\right)\right), \tag{2}$$

where $\mathcal{C}_u$ and $\mathcal{C}_v$ are proper constraints (e.g., nonnegativity, orthogonality) on the factor matrices $\mathbf{U}_\pi$ and $\mathbf{V}_\pi$, respectively. The key idea of CMF is to share the common factors across multiple matrices. In the literature, the feature clusters $\{\mathbf{U}_\pi\}_{\pi \in \Pi}$ are usually shared across multiple domains to facilitate transfer learning [8], [9], [31], i.e., $\mathcal{C}_u \triangleq \{\mathbf{U}_\pi \equiv \mathbf{U}{:}\forall \pi \in \Pi\}$. In this way, we can extract the common factors for knowledge transfer by preserving the statistical property of data.

Collective Matrix Tri-Factorization: Similarly, the latent factors can also be extracted by nonnegative matrix tri-factorization (NMTF) model [23]. In NMTF, the feature-example matrix $\mathbf{X}_\pi$ is decomposed into three low-rank nonnegative matrices $\mathbf{U}_\pi$, $\mathbf{H}_\pi$, and $\mathbf{V}_\pi$

$$\min_{\mathbf{U}_\pi, \mathbf{H}_\pi, \mathbf{V}_\pi \geq \mathbf{0}} \mathcal{L}\left(\mathbf{X}_\pi, h\left(\mathbf{U}_\pi \mathbf{H}_\pi \mathbf{V}_\pi^{\mathrm{T}}\right)\right). \tag{3}$$

$\mathbf{H}_\pi \in \mathbb{R}^{c \times c}$ is the *association* between feature clusters $\mathbf{U}_\pi$ and example classes $\mathbf{V}_\pi$ and can give a condensed view of $\mathbf{X}_\pi$. Similar to CMF, we extend basic NMTF to simultaneously factorize multiple relevant matrices, which leads to *collective matrix tri-factorization* (CMTF)

$$\min_{\mathbf{U}_\pi \in \mathcal{C}_u, \mathbf{H}_\pi \in \mathcal{C}_h, \mathbf{V}_\pi \in \mathcal{C}_v} \sum\nolimits_{\pi \in \Pi} \mathcal{L}\left(\mathbf{X}_\pi, h\left(\mathbf{U}_\pi \mathbf{H}_\pi \mathbf{V}_\pi^{\mathrm{T}}\right)\right). \tag{4}$$

Through CMTF, the feature clusters are usually shared across domains to facilitate transfer learning [9], [29], i.e., $\mathcal{C}_u \triangleq \{\mathbf{U}_\pi \equiv \mathbf{U}{:}\pi \in \Pi\}$. Similarly, the associations can also be shared across domains to facilitate stable transfer learning [3], [10], i.e., $\mathcal{C}_h \triangleq \{\mathbf{H}_\pi \equiv \mathbf{H}{:}\pi \in \Pi\}$. We study both CMF and CMTF for transfer learning.

3.2.2 Graph Co-Regularization

Secondly, we refine the latent factors by graph co-regularization, through which the data distribution in each domain can be respected. We aim to preserve the intrinsic geometric property of data in each domain.

Example Graph Regularization: From the geometric perspective, the data points may be sampled from a distribution supported by a low-dimensional manifold embedded in a high-dimensional space [19], [32]. Preserving this geometric structure can make the learning model carefully respect the domain-specific data distribution and substantially alleviate the negative transfer issue. By the *local invariance assumption* [33], if two examples $\mathbf{x}_{*i}^\pi$, $\mathbf{x}_{*j}^\pi$ are close in the intrinsic geometry of the data distribution underlying domain $\mathcal{D}_\pi$, then their embeddings $\mathbf{v}_{i*}^\pi$ and $\mathbf{v}_{j*}^\pi$ should also be close. The geometric structure can be effectively encoded by a p-nearest neighbor graph on the involved scatter of data points [19]. Consider an *example graph* G_π^v with n_π vertices each representing a data point in domain $\mathcal{D}_\pi$, and define the affinity matrix of G_π^v as

$$\left(\mathbf{W}_\pi^v\right)_{ij} = \begin{cases} \mathrm{sim}\left(\mathbf{x}_{*i}^\pi, \mathbf{x}_{*j}^\pi\right), \text{ if } \mathbf{x}_{*i}^\pi \in \mathcal{N}_p\left(\mathbf{x}_{*j}^\pi\right) \vee \mathbf{x}_{*j}^\pi \in \mathcal{N}_p\left(\mathbf{x}_{*i}^\pi\right) \\ 0, \text{ otherwise,} \end{cases} \tag{5}$$

where $\mathrm{sim}(\cdot,\cdot)$ is a proper similarity function, $\mathcal{N}_p\left(\mathbf{x}_{*i}^\pi\right)$ is the set of p-nearest neighbors of the example $\mathbf{x}_{*i}^\pi$.

Recall that the low-dimensional embedding of example $\mathbf{x}_{*i}^\pi$ extracted by CMF is $\mathbf{v}_{i*}^\pi = [v_{i1}^\pi, \ldots, v_{ic}^\pi]$. We use loss function ℓ to measure the closeness between each pair of embeddings $\mathbf{v}_{i*}^\pi$ and $\mathbf{v}_{j*}^\pi$, i.e., $\ell\left(\mathbf{v}_{i*}^\pi, \mathbf{v}_{j*}^\pi\right)$. According to Cai *et al.* [19], preserving the geometric structure in domain $\mathcal{D}_\pi$ with respect to graph G_π^v is achieved by the following *example graph regularization*

$$\mathcal{R}\left(\mathbf{V}_\pi\right) = \frac{1}{2} \sum\nolimits_{i,j=1}^{n_\pi} \ell\left(\mathbf{v}_{i*}^\pi, \mathbf{v}_{j*}^\pi\right) \left(\mathbf{W}_\pi^v\right)_{ij}. \tag{6}$$

Feature Graph Regularization: Considering *duality* between features and examples, the features are also sampled from a distribution supported by another low-dimensional manifold [34]. By the *local invariance assumption* [33], if two features $\mathbf{x}_{i*}^\pi$, $\mathbf{x}_{j*}^\pi$ are close in the intrinsic geometry of the data distribution underlying domain $\mathcal{D}_\pi$, then their embeddings $\mathbf{u}_{i*}^\pi$ and $\mathbf{u}_{j*}^\pi$ should also be close. As the example graph, consider a *feature graph* G_π^u with m vertices each representing a feature in domain $\mathcal{D}_\pi$, and define the affinity matrix of G_π^u as

$$\left(\mathbf{W}_\pi^u\right)_{ij} = \begin{cases} \mathrm{sim}\left(\mathbf{x}_{i*}^\pi, \mathbf{x}_{j*}^\pi\right), \text{ if } \mathbf{x}_{i*}^\pi \in \mathcal{N}_p\left(\mathbf{x}_{j*}^\pi\right) \vee \mathbf{x}_{j*}^\pi \in \mathcal{N}_p\left(\mathbf{x}_{i*}^\pi\right) \\ 0, \text{ otherwise,} \end{cases} \tag{7}$$

where $\mathrm{sim}(\cdot,\cdot)$ is a proper similarity function, $\mathcal{N}_p\left(\mathbf{x}_{i*}^\pi\right)$ is the set of p-nearest neighbors of the feature $\mathbf{x}_{i*}^\pi$.

The low-dimensional embedding of feature $\mathbf{x}_{i*}^\pi$ extracted by CMF is $\mathbf{u}_{i*}^\pi = [u_{i1}^\pi, \ldots, u_{ic}^\pi]$. Similar to the example graph regularization, preserving the geometric structure in domain $\mathcal{D}_\pi$ with respect to graph G_π^u is achieved by the following *feature graph regularization*

$$\mathcal{R}\left(\mathbf{U}_\pi\right) = \frac{1}{2} \sum\nolimits_{i,j=1}^{m} \ell\left(\mathbf{u}_{i*}^\pi, \mathbf{u}_{j*}^\pi\right) \left(\mathbf{W}_\pi^u\right)_{ij}. \tag{8}$$

We refer to the graph regularization terms in Equations (6) and (8) as *graph co-regularization*, since they are to preserve the geometric structure on examples and features simultaneously. We will use them to refine the latent factors for alleviating negative transfer.

3.2.3 Optimization Framework

To further boost the performance for cross-domain classification, the two learning objectives should be considered together. The reasons are: 1) with collective matrix factorization, the common latent factors are extracted through which knowledge can be transferred between domains; 2) with graph co-regularization, the geometric structure in each domain is preserved so that negative transfer

can be substantially alleviated. Furthermore, collective matrix factorization and graph co-regularization can be performed simultaneously to enjoy the intrinsic mutual reinforcement learning: 1) collective matrix factorization can extract a subspace with statistically-sufficient embeddings for all the examples and features, and 2) graph co-regularization can enrich the subspace with discriminating geometric structure for better classification performance. Therefore, we should integrate these two learning objectives seamlessly into unified GTL optimization framework

$$\min_{\mathbf{U}_\pi \in \mathcal{C}_u, \mathbf{V}_\pi \in \mathcal{C}_v} \sum_{\pi \in \Pi} \left[\mathcal{L}\left(\mathbf{X}_\pi, h\left(\mathbf{U}_\pi \mathbf{V}_\pi^{\mathrm{T}}\right)\right) + \lambda \mathcal{R}\left(\mathbf{U}_\pi\right) + \gamma \mathcal{R}\left(\mathbf{V}_\pi\right) \right]$$
$$\text{s.t.} \quad \mathcal{C}_u \triangleq \{\mathbf{U}_\pi \equiv \mathbf{U} : \pi \in \Pi\}, \mathcal{C}_v \triangleq \{\mathbf{V}_\pi \equiv \mathbf{Y}_\pi : \pi \in \Pi_s\}, \tag{9}$$

where λ is the feature graph regularization parameter, and γ is the example graph regularization parameter. Since true labels are available in the source domains, we incorporate them in the optimization framework by $\mathcal{C}_v \triangleq \{\mathbf{V}_\pi \equiv \mathbf{Y}_\pi : \forall \pi \in \Pi_s\}$. To facilitate transfer learning, we follow [8], [30], [31] and share the *feature clusters* across domains, i.e., $\mathcal{C}_u \triangleq \{\mathbf{U}_\pi \equiv \mathbf{U} : \forall \pi \in \Pi\}$, through which supervision information can be propagated from the source domains to the target domains.

Similarly, the GTL framework can also be formulated by using collective matrix tri-factorization (CMTF)

$$\min_{\mathbf{H}_\pi \in \mathcal{C}_h, \mathbf{V}_\pi \in \mathcal{C}_v} \sum_{\pi \in \Pi} \left[\mathcal{L}\left(\mathbf{X}_\pi, h\left(\mathbf{U}_\pi \mathbf{H}_\pi \mathbf{V}_\pi^{\mathrm{T}}\right)\right) + \lambda \mathcal{R}\left(\mathbf{U}_\pi\right) + \gamma \mathcal{R}\left(\mathbf{V}_\pi\right) \right]$$
$$\text{s.t.} \quad \mathcal{C}_h \triangleq \{\mathbf{H}_\pi \equiv \mathbf{H} : \pi \in \Pi\}, \mathcal{C}_v \triangleq \{\mathbf{V}_\pi \equiv \mathbf{Y}_\pi : \pi \in \Pi_s\}. \tag{10}$$

To facilitate transfer learning, we share the *associations* across domains as [3], [10], i.e., $\mathcal{C}_h \triangleq \{\mathbf{H}_\pi \equiv \mathbf{H} : \forall \pi \in \Pi\}$. With the optimization results, the label for example $\mathbf{x}_{*i}^\pi$ in target domain $\mathcal{D}_\pi$ can be predicted by

$$f\left(\mathbf{x}_{*i}^\pi\right) = \arg\max_j \left(\mathbf{V}_\pi\right)_{ij}. \tag{11}$$

The GTL framework is formulated to be general, in which we can choose various prediction link h, loss functions $\mathcal{L}$ and ℓ, similarity function sim, constraints $\mathcal{C}_u$ and $\mathcal{C}_v$. The widely adopted options are as follows:

- h can be either the identity function or the logistic function, i.e., $h(\mathbf{X}) = \mathbf{X}$ or $h\left(X_{ij}\right) = \frac{1}{1+e^{-X_{ij}}}$.
- $\mathcal{L}$ can be either the sum of squares loss or the matrix divergence [19], i.e., $\mathcal{L}(\mathbf{X}, \hat{\mathbf{X}}) = \|\mathbf{X} - \hat{\mathbf{X}}\|_F^2$ or $\mathcal{L}\left(\mathbf{X}, \hat{\mathbf{X}}\right) = \sum_{ij}\left(X_{ij} \log \frac{X_{ij}}{\hat{X}_{ij}} - X_{ij} + \hat{X}_{ij}\right)$.
- ℓ can be either the Euclidian distance or the generalized KL-divergence [19], i.e., $\ell(\mathbf{x}, \hat{\mathbf{x}}) = \|\mathbf{x} - \hat{\mathbf{x}}\|_2^2$ or $\ell\left(\mathbf{x}, \hat{\mathbf{x}}\right) = \sum_i \left(x_i \log \frac{x_i}{\hat{x}_i} - x_i + \hat{x}_i\right)$.
- sim can be either the cosine similarity or the heat kernel weighting [19], i.e., $\mathrm{sim}(\mathbf{x}, \hat{\mathbf{x}}) = \cos(\mathbf{x}, \hat{\mathbf{x}})$ or $\mathrm{sim}\left(\mathbf{x}, \hat{\mathbf{x}}\right) = \exp\left(-\frac{\|\mathbf{x}-\hat{\mathbf{x}}\|_2^2}{\beta^2}\right)$, where β is the bandwidth parameter of the heat kernel.
- $\mathcal{C}_u$ and $\mathcal{C}_v$ can incorporate nonnegative constraint (NMF), orthogonal constraint (SVD), probabilistic constraint (PLSA) [20], or sparse constraint.

Based on specific applications, we can choose the most appropriate configuration to achieve optimal performances.

3.3 Learning Algorithms

We extend standard algorithms NMF [21] and NMTF [23] under the GTL framework using proper settings. We choose linear models, i.e., $h(\mathbf{X}) = \mathbf{X}$, $\mathcal{L}(\mathbf{X}, \hat{\mathbf{X}}) = \|\mathbf{X} - \hat{\mathbf{X}}\|_F^2$, $\ell(\mathbf{x}, \hat{\mathbf{x}}) = \|\mathbf{x} - \hat{\mathbf{x}}\|_2^2$, and $\mathrm{sim}(\mathbf{x}, \hat{\mathbf{x}}) = \cos(\mathbf{x}, \hat{\mathbf{x}})$. We further impose *approximate* orthogonal constraints for $\mathcal{C}_v$, i.e., $\mathcal{C}_v \supset \left\{\left\|\mathbf{V}_\pi^{\mathrm{T}}\mathbf{V}_\pi - \mathbf{I}\right\|_F^2 \le \varepsilon : \forall \pi \in \Pi\right\}$, where ε is a small nonnegative constant. GTL will require the orthogonality constraints to combat the *trivial solution* problem.

3.3.1 GTL Using NMF

Using nonnegative matrix factorization (NMF) [21] as base model, the framework in (9) is reduced to GTL_2

$$\mathcal{O}_2 = \sum\nolimits_{\pi \in \Pi} \left\|\mathbf{X}_\pi - \mathbf{U}\mathbf{V}_\pi^{\mathrm{T}}\right\|_F^2 + \frac{\sigma}{2} \sum\nolimits_{\pi \in \Pi} \left\|\mathbf{V}_\pi^{\mathrm{T}}\mathbf{V}_\pi - \mathbf{I}\right\|_F^2 + \lambda \sum\nolimits_{\pi \in \Pi} \mathrm{tr}\left(\mathbf{U}^{\mathrm{T}}\mathbf{L}_\pi^u\mathbf{U}\right) + \gamma \sum\nolimits_{\pi \in \Pi} \mathrm{tr}\left(\mathbf{V}_\pi^{\mathrm{T}}\mathbf{L}_\pi^v\mathbf{V}_\pi\right), \tag{12}$$

where σ is the shrinkage regularization parameter; $\mathbf{L}_\pi^u$ and $\mathbf{L}_\pi^v$ are the *graph Laplacian* matrices which are computed as $\mathbf{L}_\pi^u = \mathbf{D}_\pi^u - \mathbf{W}_\pi^u$, $\mathbf{L}_\pi^v = \mathbf{D}_\pi^v - \mathbf{W}_\pi^v$; and $\mathbf{D}_\pi^u$ and $\mathbf{D}_\pi^v$ are diagonal degree matrices with each item $\left(\mathbf{D}_\pi^u\right)_{ii} = \sum_{j=1}^m \left(\mathbf{W}_\pi^u\right)_{ij}$, $\left(\mathbf{D}_\pi^v\right)_{ii} = \sum_{j=1}^{n_\pi} \left(\mathbf{W}_\pi^v\right)_{ij}$. Based on the shrinkage method, we can approximately satisfy the orthogonality constraints for $\mathbf{V}_\pi, \forall \pi \in \Pi$ by preventing the second term from getting too large. Due to the Lagrange multiplier method, given any ε, there is a proper σ such that $\left\|\mathbf{V}_\pi^{\mathrm{T}}\mathbf{V}_\pi - \mathbf{I}\right\|_F^2 \le \varepsilon$ is satisfied.

Remark on the Trivial Solution Problem: It is very important to note that, existing graph regularized NMF methods [19], [32], [34] may suffer from the *trivial solution* problem [35]: when $\gamma \to \infty$, the fourth term dominate the objective, with Equation (12) reduced to

$$\mathcal{O}_2' = \sum_{\pi \in \Pi} \mathrm{tr}\left(\mathbf{V}_\pi^{\mathrm{T}}\mathbf{L}_\pi^v\mathbf{V}_\pi\right) = \sum_{\pi \in \Pi} \sum_{k=1}^{c} \mathbf{v}_{*k}^{\pi\mathrm{T}}\mathbf{L}_\pi^v\mathbf{v}_{*k}^\pi. \tag{13}$$

Equation (13) is decomposed into $c|\Pi|$ independent subproblems: $\mathcal{O}_2'' = \mathbf{v}_{*k}^{\pi\mathrm{T}}\mathbf{L}_\pi^v\mathbf{v}_{*k}^\pi$. Each subproblem gets the same solution up to a scale, i.e., $\mathbf{v}_{*1}^\pi \propto \ldots \propto \mathbf{v}_{*c}^\pi$. Hence the class assignments by $\mathbf{V}_\pi$ tend to assign all examples to one class, which is misspecified. Gu *et al.* [35] imposed a normalized-cut style constraint on $\mathbf{V}_\pi$ and then solved a constrained optimization problem using the Lagrange multiplier method. However, this method suffers from unstable convergence performance. By satisfying the orthogonality constraints with a shrinkage methodology, GTL can fully address the trivial solution problem and can obtain stable convergence performance. Since typically $\lambda \in [0, 1] \ll \infty$, we need not impose the orthogonal constraints for $\mathcal{C}_u$.

The optimization problem in Equation (12) can be solved by an alternating optimization procedure, as stated in the following theorem. The detailed theoretical analysis of the theorem is presented in Section 4.

Algorithm 1: GTL_2: GTL Using NMF

Input: Data $\{\mathbf{X}_\pi\}_{\pi\in\Pi}$, $\{\mathbf{Y}_\pi\}_{\pi\in\Pi_s}$; parameters p, λ, γ, σ, T.
Output: Factors $\mathbf{U}$, $\{\mathbf{V}_\pi\}_{\pi\in\Pi}$, predictions $\{\hat{\mathbf{Y}}_\pi\}_{\pi\in\Pi_t}$.

begin
  Construct $\mathbf{W}_\pi^v$ and $\mathbf{W}_\pi^u$ by Equations (5) and (7).
  Initialize $\mathbf{U}$ randomly; $\mathbf{V}_\pi \leftarrow \mathbf{Y}_\pi$, $\forall\pi \in \Pi_s$; and $\mathbf{V}_\pi \leftarrow \text{LR}\left(\bigcup_{\pi'\in\Pi_s}\{\mathbf{X}_{\pi'}, \mathbf{Y}_{\pi'}\}\right)$, $\forall\pi \in \Pi_t$.
  for $t \leftarrow 1$ **to** T **do**
    Update $\mathbf{U}$ by Equation (14).
    foreach $\pi \in \Pi_t$ **do**
      Update $\mathbf{V}_\pi$ by Equation (15).
    Compute objective $\mathcal{O}_2^{(t)}$ by Equation (12).
  Predict target labels by Equation (11) $\hat{y}(\mathbf{x}_{*i}^\pi) = f\left(\mathbf{x}_{*i}^\pi\right)$.

Theorem 1. *Updating* $\mathbf{U}$, $\{\mathbf{V}_\pi\}_{\pi\in\Pi}$ *sequentially by Equations* (14)∼(15) *will monotonically decrease the objective function in Equation* (12) *until convergence.*

$$\mathbf{U} \leftarrow \mathbf{U} \odot \frac{\left[\sum_{\pi\in\Pi}\left(\mathbf{X}_\pi\mathbf{V}_\pi + \lambda\mathbf{W}_\pi^u\mathbf{U}\right)\right]}{\left[\sum_{\pi\in\Pi}\left(\mathbf{U}\mathbf{V}_\pi^{\mathrm{T}}\mathbf{V}_\pi + \lambda\mathbf{D}_\pi^u\mathbf{U}\right)\right]} \tag{14}$$

$$\mathbf{V}_\pi \leftarrow \mathbf{V}_\pi \odot \frac{\left[\mathbf{X}_\pi^{\mathrm{T}}\mathbf{U} + \gamma\mathbf{W}_\pi^v\mathbf{V}_\pi + \sigma\mathbf{V}_\pi\right]}{\left[\mathbf{V}_\pi\mathbf{U}^{\mathrm{T}}\mathbf{U} + \gamma\mathbf{D}_\pi^v\mathbf{V}_\pi + \sigma\mathbf{V}_\pi\mathbf{V}_\pi^{\mathrm{T}}\mathbf{V}_\pi\right]}, \tag{15}$$

where $\odot$ *and* $\frac{[\cdot]}{[\cdot]}$ *denote element-wise product and division respectively.*

The complete learning procedure is summarized in Algorithm 1. The source domains are labeled, so we keep $\{\mathbf{V}_\pi \equiv \mathbf{Y}_\pi : \pi \in \Pi_s\}$ throughout iteration. Since the optimization involves iterative update rules, it is risky that the procedure might get stuck in poor local optima. Therefore, we initialize the classes $\{\mathbf{V}_\pi\}_{\pi\in\Pi_t}$ of the target data by applying the Logistic Regression (LR) classifier trained on the labeled source data $\{\mathbf{X}_\pi, \mathbf{Y}_\pi\}_{\pi\in\Pi_s}$.

3.3.2 GTL Using NMTF

Using nonnegative matrix tri-factorization (NMTF) [23] as base model, framework (10) is reduced to GTL_3

$$\begin{aligned}\mathcal{O}_3 = &\sum\nolimits_{\pi\in\Pi}\left\|\mathbf{X}_\pi - \mathbf{U}_\pi\mathbf{H}\mathbf{V}_\pi^{\mathrm{T}}\right\|_F^2 + \frac{\sigma}{2}\sum\nolimits_{\pi\in\Pi}\left\|\mathbf{V}_\pi^{\mathrm{T}}\mathbf{V}_\pi - \mathbf{I}\right\|_F^2 \\ &+\lambda\sum\nolimits_{\pi\in\Pi}\mathrm{tr}\left(\mathbf{U}_\pi^{\mathrm{T}}\mathbf{L}_\pi^u\mathbf{U}_\pi\right) + \gamma\sum\nolimits_{\pi\in\Pi}\mathrm{tr}\left(\mathbf{V}_\pi^{\mathrm{T}}\mathbf{L}_\pi^v\mathbf{V}_\pi\right).\end{aligned} \tag{16}$$

The problem in Equation (16) can also be solved by an alternating optimization procedure, as stated in the following theorem. The theoretical analysis is similar to the NMF version and thus is omitted. The complete learning procedure is summarized in Algorithm 2.

Algorithm 2: GTL_3: GTL Using NMTF

Input: Data $\{\mathbf{X}_\pi\}_{\pi\in\Pi}$, $\{\mathbf{Y}_\pi\}_{\pi\in\Pi_s}$; parameters p, λ, γ, σ, T.
Output: Factors $\{\mathbf{U}_\pi, \mathbf{V}_\pi\}_{\pi\in\Pi}$, $\mathbf{H}$, predictions $\{\hat{\mathbf{Y}}_\pi\}_{\pi\in\Pi_t}$.

begin
  Construct $\mathbf{W}_\pi^v$ and $\mathbf{W}_\pi^u$ by Equations (5) and (7).
  Initialize $\{\mathbf{U}_\pi\}_{\pi\in\Pi}$, $\mathbf{H}$ randomly; $\mathbf{V}_\pi \leftarrow \mathbf{Y}_\pi$, $\forall\pi \in \Pi_s$; and $\mathbf{V}_\pi \leftarrow \text{LR}\left(\bigcup_{\pi'\in\Pi_s}\{\mathbf{X}_{\pi'}, \mathbf{Y}_{\pi'}\}\right)$, $\forall\pi \in \Pi_t$.
  for $t \leftarrow 1$ **to** T **do**
    foreach $\pi \in \Pi$ **do**
      Update $\mathbf{U}_\pi$ by Equation (17).
      if $\pi \in \Pi_t$ **then**
        Update $\mathbf{V}_\pi$ by Equation (18).
    Update $\mathbf{H}$ by Equation (19).
    Compute objective $\mathcal{O}_3^{(t)}$ by Equation (16).
  Predict target labels by Equation (11) $\hat{y}(\mathbf{x}_{*i}^\pi) = f\left(\mathbf{x}_{*i}^\pi\right)$.

Theorem 2. *Updating* $\{\mathbf{U}_\pi\}_{\pi\in\Pi}$, $\{\mathbf{V}_\pi\}_{\pi\in\Pi_t}$, $\mathbf{H}$ *sequentially by Equations* (17)∼(19) *will monotonically decrease the objective function in Equation* (16) *until convergence.*

$$\mathbf{U}_\pi \leftarrow \mathbf{U}_\pi \odot \frac{\left[\mathbf{X}_\pi\mathbf{V}_\pi\mathbf{H}^{\mathrm{T}} + \lambda\mathbf{W}_\pi^u\mathbf{U}_\pi\right]}{\left[\mathbf{U}_\pi\mathbf{H}\mathbf{V}_\pi^{\mathrm{T}}\mathbf{V}_\pi\mathbf{H}^{\mathrm{T}} + \lambda\mathbf{D}_\pi^u\mathbf{U}_\pi\right]} \tag{17}$$

$$\mathbf{V}_\pi \leftarrow \mathbf{V}_\pi \odot \frac{\left[\mathbf{X}_\pi^{\mathrm{T}}\mathbf{U}_\pi\mathbf{H} + \gamma\mathbf{W}_\pi^v\mathbf{V}_\pi + \sigma\mathbf{V}_\pi\right]}{\left[\mathbf{V}_\pi\mathbf{H}^{\mathrm{T}}\mathbf{U}_\pi^{\mathrm{T}}\mathbf{U}_\pi\mathbf{H} + \gamma\mathbf{D}_\pi^v\mathbf{V}_\pi + \sigma\mathbf{V}_\pi\mathbf{V}_\pi^{\mathrm{T}}\mathbf{V}_\pi\right]} \tag{18}$$

$$\mathbf{H} \leftarrow \mathbf{H} \odot \frac{\left[\sum_{\pi\in\Pi}\mathbf{U}_\pi^{\mathrm{T}}\mathbf{X}_\pi\mathbf{V}_\pi\right]}{\left[\sum_{\pi\in\Pi}\mathbf{U}_\pi^{\mathrm{T}}\mathbf{U}_\pi\mathbf{H}\mathbf{V}_\pi^{\mathrm{T}}\mathbf{V}_\pi\right]}, \tag{19}$$

where $\odot$ *and* $\frac{[\cdot]}{[\cdot]}$ *denote element-wise product and division respectively.*

Remark on GTL_2 vs. GTL_3: Firstly, GTL_2 is at the end of "over-transfer". It shares the feature clusters $\mathbf{U} \in \mathbb{R}^{m\times c}$, which are a large amount of parameters and can encode sufficient knowledge structure to enable transfer. However, it may suffer from *negative transfer*, i.e., the excessive transferred knowledge may be highly inconsistent with the target domain cluster structure. In this scenario, graph co-regularization can preserve the target geometric structure to combat negative transfer.

On contrary, GTL_3 is at the end of "under-transfer". It shares the associations $\mathbf{H} \in \mathbb{R}^{c\times c}$, which are a small amount of parameters and can perform robustly to substantial domain difference. However, it may suffer from *ineffective transfer*, i.e., no sufficient knowledge can be transferred across domains to improve the target tasks. In this scenario, graph co-regularization can facilitate reinforcement learning between different properties of input data to enhance effective transfer.

3.4 Computational Complexity

The computational cost consists of three parts as follows: $O\left(\sum_{\pi\in\Pi} Tc\left(mn_\pi + m^2 + n_\pi^2\right)\right)$ for computing multiplicative update rules; $O\left(\sum_{\pi\in\Pi}\left(mn_\pi^2 + m^2n_\pi\right)\right)$ for building nearest neighbor graphs; $O\left(\sum_{\pi\in\Pi} mn_\pi\right)$ for others. In all, the overall computational complexity is $O\left(\sum_{\pi\in\Pi} Tc\left(mn_\pi + m^2 + n_\pi^2\right) + m^2n_\pi + mn_\pi^2\right)$, which can be greatly reduced if the input data are sparse.

4 Theoretical Analysis

4.1 Optimization Derivation

We derive solutions to the GTL optimization problem in Equation (12) using the constrained optimization theory. Specifically, we will optimize one variable and compute its update rule while fixing the rest of variables. The procedure repeats until convergence.

Let Φ and Ψ_π be the Lagrange multipliers for the nonnegative constraints $\mathbf{U} \geq \mathbf{0}$ and $\mathbf{V}_\pi \geq \mathbf{0}$, $\forall\pi \in \Pi$ respectively. The Lagrange function is formulated as

$$L = \mathcal{O}_2 + \mathrm{tr}\left(\Phi\mathbf{U}^{\mathrm{T}}\right) + \sum\nolimits_{\pi\in\Pi}\mathrm{tr}\left(\Psi_\pi\mathbf{V}_\pi^{\mathrm{T}}\right).$$

The partial derivatives of L w.r.t. $\mathbf{U}$ and $\mathbf{V}_\pi$ are

$$\begin{aligned}\frac{\partial L}{\partial \mathbf{U}} &= -2\sum\nolimits_{\pi\in\Pi}\mathbf{X}_\pi\mathbf{V}_\pi + 2\sum\nolimits_{\pi\in\Pi}\mathbf{U}\mathbf{V}_\pi^{\mathrm{T}}\mathbf{V}_\pi \\ &\quad +2\sum\nolimits_{\pi\in\Pi}\lambda\mathbf{D}_\pi^u\mathbf{U} - 2\sum\nolimits_{\pi\in\Pi}\lambda\mathbf{W}_\pi^u\mathbf{U} + \Phi \\ \frac{\partial L}{\partial \mathbf{V}_\pi} &= -2\mathbf{X}_\pi^{\mathrm{T}}\mathbf{U} + 2\mathbf{V}_\pi\mathbf{U}^{\mathrm{T}}\mathbf{U} + 2\gamma\mathbf{D}_\pi^v\mathbf{V}_\pi \\ &\quad -2\gamma\mathbf{W}_\pi^v\mathbf{V}_\pi + 2\sigma\mathbf{V}_\pi\mathbf{V}_\pi^{\mathrm{T}}\mathbf{V}_\pi - 2\sigma\mathbf{V}_\pi + \Psi_\pi.\end{aligned}$$

Using the Karush-Kuhn-Tucker (KKT) complementarity conditions $\Phi\odot\mathbf{U}=\mathbf{0}$, $\Psi_\pi\odot\mathbf{V}_\pi=\mathbf{0}$, we can obtain

$$\begin{aligned}\frac{\partial L}{\partial \mathbf{U}}\odot\mathbf{U} &= \Big(\sum\nolimits_{\pi\in\Pi}\mathbf{U}\mathbf{V}_\pi^{\mathrm{T}}\mathbf{V}_\pi + \sum\nolimits_{\pi\in\Pi}\lambda\mathbf{D}_\pi^u\mathbf{U}\Big)\odot\mathbf{U} \\ &\quad -\Big(\sum\nolimits_{\pi\in\Pi}\mathbf{X}_\pi\mathbf{V}_\pi + \sum\nolimits_{\pi\in\Pi}\lambda\mathbf{W}_\pi^u\mathbf{U}\Big)\odot\mathbf{U} = \mathbf{0} \\ \frac{\partial L}{\partial \mathbf{V}_\pi}\odot\mathbf{V}_\pi &= \Big(\mathbf{V}_\pi\mathbf{U}^{\mathrm{T}}\mathbf{U} + \gamma\mathbf{D}_\pi^v\mathbf{V}_\pi + \sigma\mathbf{V}_\pi\mathbf{V}_\pi^{\mathrm{T}}\mathbf{V}_\pi\Big)\odot\mathbf{V}_\pi \\ &\quad -\Big(\mathbf{X}_\pi^{\mathrm{T}}\mathbf{U} + \gamma\mathbf{W}_\pi^v\mathbf{V}_\pi + \sigma\mathbf{V}_\pi\Big)\odot\mathbf{V}_\pi = \mathbf{0}.\end{aligned}$$

These lead to the update rules in Equations (14)∼(15).

4.2 Convergence Analysis

We use the auxiliary function approach [19], [21] to analyze the convergence aspect of Theorem 1. For clarity, we will only prove that the objective function $\mathcal{O}_2$ in Equation (12) is non-increasing under the update rule for $\mathbf{V}_\pi$ in Equation (15). The convergence property of the other update rules can be proved similarly. First, we introduce the definition of auxiliary function.

Definition 1. *[21] $A(z,\tilde{z})$ is an auxiliary function for $F(z)$ if the conditions*

$$A(z,\tilde{z}) \geq F(z) \text{ and } A(z,z) = F(z)$$

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

Lemma 1. *[21] If A is an auxiliary function for F, then F is non-increasing under the update*

$$z^{(t+1)} = \arg\min_z A(z, z^{(t)})$$

Proof.

$$F(z^{(t+1)}) \leq A(z^{(t+1)}, z^{(t)}) \leq A(z^{(t)}, z^{(t)}) = F(z^{(t)}). \qquad \square$$

In the sequel, we will show that Equation (15) is exactly the update rule in Lemma 1 with a proper auxiliary function. For any element v_{ij} in $\mathbf{V}_\pi$, we use F_{ij} to denote the part of $\mathcal{O}_2$ which is only relevant to v_{ij}. The corresponding first-order and second-order derivatives of F_{ij} with respect to v_{ij} are computed as

$$\begin{aligned}F'_{ij} &= \left(\frac{\partial\mathcal{O}_2}{\partial\mathbf{V}_\pi}\right)_{ij} \\ F''_{ij} &= 2\Big(\mathbf{U}^{\mathrm{T}}\mathbf{U}\Big)_{jj} + 2\gamma\big(\mathbf{D}_\pi^v - \mathbf{W}_\pi^v\big)_{ii} \\ &\quad +2\sigma\left(\Big(\mathbf{V}_\pi^{\mathrm{T}}\mathbf{V}_\pi\Big)_{jj} + v_{jj}^2 - 1\right).\end{aligned}$$

Lemma 2. *Function*

$$\begin{aligned}A\left(v, v_{ij}^{(t)}\right) &= F_{ij}\left(v_{ij}^{(t)}\right) + F'_{ij}\left(v_{ij}^{(t)}\right)\left(v - v_{ij}^{(t)}\right) \\ &+ \frac{\left(\mathbf{V}_\pi\mathbf{U}^{\mathrm{T}}\mathbf{U} + \gamma\mathbf{D}_\pi^v\mathbf{V}_\pi + \sigma\mathbf{V}_\pi\mathbf{V}_\pi^{\mathrm{T}}\mathbf{V}_\pi\right)_{ij}}{v_{ij}^{(t)}}\Big(v - v_{ij}^{(t)}\Big)^2\end{aligned}$$

is a proper auxiliary function for $F_{ij}(v)$.

Proof. It is straight-forward that $A(v,v) = F_{ij}(v)$, and thus we only need verify that $A\left(v, v_{ij}^{(t)}\right) \geq F_{ij}(v)$. To achieve this, we expand $F_{ij}(v)$ using Taylor series

$$\begin{aligned}F_{ij}(v) &= F_{ij}\left(v_{ij}^{(t)}\right) + F'_{ij}\left(v_{ij}^{(t)}\right)\left(v - v_{ij}^{(t)}\right) + \left(v - v_{ij}^{(t)}\right)^2 \cdot \\ &\Big(\big(\mathbf{U}^{\mathrm{T}}\mathbf{U}\big)_{jj} + \gamma\big(\mathbf{D}_\pi^v - \mathbf{W}_\pi^v\big)_{ii} + \sigma\left(\Big(\mathbf{V}_\pi^{\mathrm{T}}\mathbf{V}_\pi\Big)_{jj} + v_{jj}^2 - 1\right)\Big).\end{aligned}$$

Due to orthogonality, $1 \approx \left(\mathbf{V}_\pi^{\mathrm{T}}\mathbf{V}_\pi\right)_{jj} = \sum_i v_{ij}^2 \gg v_{jj}^2$. By algebra manipulations, we have three inequalities:

$$\begin{aligned}\Big(\mathbf{V}_\pi\mathbf{U}^{\mathrm{T}}\mathbf{U}\Big)_{ij} &= \sum\nolimits_l v_{il}^{(t)}\Big(\mathbf{U}^{\mathrm{T}}\mathbf{U}\Big)_{lj} \geq v_{ij}^{(t)}\Big(\mathbf{U}^{\mathrm{T}}\mathbf{U}\Big)_{jj} \\ \big(\mathbf{D}_\pi^v\mathbf{V}_\pi\big)_{ij} &= \sum\nolimits_l \big(\mathbf{D}_\pi^v\big)_{il}v_{lj}^{(t)} \geq v_{ij}^{(t)}\big(\mathbf{D}_\pi^v - \mathbf{W}_\pi^v\big)_{ii} \\ \Big(\mathbf{V}_\pi\mathbf{V}_\pi^{\mathrm{T}}\mathbf{V}_\pi\Big)_{ij} &= \sum\nolimits_l v_{il}^{(t)}\Big(\mathbf{V}_\pi^{\mathrm{T}}\mathbf{V}_\pi\Big)_{lj} \geq v_{ij}^{(t)}\Big(\mathbf{V}_\pi^{\mathrm{T}}\mathbf{V}_\pi\Big)_{jj} \\ &\geq v_{ij}^{(t)}\left(\Big(\mathbf{V}_\pi^{\mathrm{T}}\mathbf{V}_\pi\Big)_{jj} + v_{jj}^2 - 1\right).\end{aligned}$$

By comparing the above three inequalities collectively, we have $A\left(v, v_{ij}^{(t)}\right) \geq F_{ij}(v)$, and Lemma 2 holds. $\square$

Proof. of Theorem 1. Based on Lemmas 1 and 2, the update rule for $\mathbf{V}_\pi$ can be obtained by minimizing $A\left(v_{ij}^{(t+1)}, v_{ij}^{(t)}\right)$. Setting $\frac{\partial A\left(v_{ij}^{(t+1)}, v_{ij}^{(t)}\right)}{\partial v_{ij}^{(t+1)}} = 0$, we obtain

$$\begin{aligned}v_{ij}^{(t+1)} &= v_{ij}^{(t)} - \frac{v_{ij}^{(t)}F'_{ij}\left(v_{ij}^{(t)}\right)}{2\left(\mathbf{V}_\pi\mathbf{U}^{\mathrm{T}}\mathbf{U} + \gamma\mathbf{D}_\pi^v\mathbf{V}_\pi + \sigma\mathbf{V}_\pi\mathbf{V}_\pi^{\mathrm{T}}\mathbf{V}_\pi\right)_{ij}} \\ &= v_{ij}^{(t)}\frac{\left(\mathbf{X}_\pi^{\mathrm{T}}\mathbf{U} + \gamma\mathbf{W}_\pi^v\mathbf{V}_\pi + \sigma\mathbf{V}_\pi\right)_{ij}}{\left(\mathbf{V}_\pi\mathbf{U}^{\mathrm{T}}\mathbf{U} + \gamma\mathbf{D}_\pi^v\mathbf{V}_\pi + \sigma\mathbf{V}_\pi\mathbf{V}_\pi^{\mathrm{T}}\mathbf{V}_\pi\right)_{ij}}\end{aligned}$$

This update rule is consistent with Equation (15). For each iteration of updating $\mathbf{V}_\pi$, we have $\mathcal{O}_2\left(\mathbf{V}_\pi^{(0)}\right) = A\left(\mathbf{V}_\pi^{(0)}, \mathbf{V}_\pi^{(0)}\right) \geq A\left(\mathbf{V}_\pi^{(1)}, \mathbf{V}_\pi^{(0)}\right) \geq A\left(\mathbf{V}_\pi^{(1)}, \mathbf{V}_\pi^{(1)}\right) = \mathcal{O}_2\left(\mathbf{V}_\pi^{(1)}\right) \geq \ldots \geq \mathcal{O}_2\left(\mathbf{V}_\pi^{(T)}\right)$. Therefore, $\mathcal{O}_2(\mathbf{V}_\pi)$ is monotonically decreasing during iterations. Since the objective function in Equation (12) is lower bounded by 0, the convergence aspect of Theorem 1 is proved. $\square$

5 Experiments

In this section, we conduct extensive experiments on two real-world applications (i.e., text classification and image recognition) to evaluate the GTL approaches. In the sequel, we will use GTL to refer to both GTL_2 and GTL_3 for the ease of description.

5.1 Data Preparation

5.1.1 Text Datasets

The 222 cross-domain text datasets are generated from 20-Newsgroups and Reuters-21578, which are two benchmark text corpora widely used for evaluating transfer learning algorithms [3], [8], [13], [15], [36].

20-Newsgroups[1] has approximately 20,000 documents distributed evenly in 20 different subcategories. The corpus

1. http://people.csail.mit.edu/jrennie/20newsgroups

TABLE 2
Top Categories and Subcategories in 20-Newsgroups

Top Category	Subcategory	#Examples	#Features
comp	comp.graphics	970	25804
	comp.os.ms-windows.misc	963	
	comp.sys.ibm.pc.hardware	979	
	comp.sys.mac.hardware	958	
rec	rec.autos	987	
	rec.motorcycles	993	
	rec.sport.baseball	991	
	rec.sport.hokey	997	
sci	sci.crypt	989	
	sci.electronics	984	
	sci.med	987	
	sci.space	985	
talk	talk.politics.guns	909	
	talk.politics.mideast	940	
	talk.politics.misc	774	
	talk.religion.misc	627	

contains four top categories *comp*, *rec*, *sci* and *talk*. Each top category has four subcategories, which are listed in Table 2. In the experiments, we can construct 6 dataset groups for binary classification by randomly selecting two top categories (one for positive and the other one for negative) from the four top categories. The 6 dataset groups are *comp vs rec*, *comp vs sci*, *comp vs talk*, *rec vs sci*, *rec vs talk*, and *sci vs talk*. Similar to the approach in [8], we set up one dataset (including source domain and target domain) for cross-domain classification as follows. For each pair of top categories P and Q (e.g., P for positive and Q for negative), their four sub-categories are denoted by P_1, P_2, P_3, P_4 and Q_1, Q_2, Q_3, Q_4, respectively. We randomly select (without replacement) two subcategories from P (e.g., P_1 and P_2) and two subcategories from Q (e.g., Q_1 and Q_2) to form a source domain, then the remaining subcategories in P and Q (i.e., P_3, P_4 and Q_3, Q_4) are selected to form a target domain. This dataset construction strategy ensures that the domains of labeled and unlabeled data are related, since they are under the same top categories. Besides, the domains are also ensured to be different, since they are drawn from different subcategories. In this way, for each dataset group P *vs* Q, we can generate $C_4^2 \cdot C_4^2 = 36$ datasets. Clearly, for each example in the generated dataset group, its class label is either P or Q. In total, we can generate 6 dataset groups consisting of $6 \cdot 36 = 216$ datasets. For fair comparison, the 216 datasets are constructed using a preprocessed version of 20-Newsgroups [3], which contains 25,804 features and 15,033 documents, with each document weighted by *term frequency-inverse document frequency* (TF-IDF).

Reuters-21578[2] has three top categories *orgs*, *people*, and *place*. Using the same strategy, we can construct 3 cross-domain text datasets *orgs vs people*, *orgs vs place* and *people vs place*. We switch the source/target pair to get another 3 datasets *people vs orgs*, *place vs orgs* and *place vs people*. For fair comparison, we use the preprocessed version of Reuters-21578 studied in [37].

5.1.2 Image Datasets

USPS, MNIST, PIE, MSRC, and VOC2007 (see Fig. 1 and Table 3) are five handwritten digits/face/photo datasets broadly adopted in compute vision literature.

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

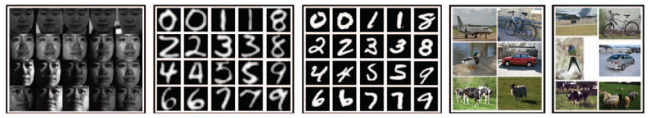


Fig. 1. Examples of PIE, USPS, MNIST, MSRC, and VOC.

TABLE 3
Statistics of the Six Benchmark Image Datasets

Dataset	Type	#Examples	#Features	#Classes
USPS	Digit	1,800	256	10
MNIST	Digit	2,000	256	10
PIE1	Face	2,856	1,024	68
PIE2	Face	3,329	1,024	68
MSRC	Photo	1,269	240	6
VOC2007	Photo	1,530	240	6

USPS[3] dataset composes of 7,291 training images and 2,007 test images of size 16×16.

MNIST[4] dataset has a training set of 60,000 examples and a test set of 10,000 examples of size 28×28.

From Fig. 1, we see that USPS and MNIST follow different distributions. They share 10 semantic classes, with each corresponding to one digit. We construct one dataset *USPS vs MNIST* by randomly sampling 1,800 images in USPS to form the source domain, and sampling 2,000 images in MNIST to form the target domain. Then we switch the source/target pair to get another dataset *MNIST vs USPS*. We uniformly rescale all images to size 16×16, and represent each image by a 256-dimensional vector encoding the gray-scale values of all pixels. In this way, the source and target domain are ensured to share the same feature space.

PIE[5], standing for "Pose, Illumination, Expression", is a benchmark face database. It has 68 individuals with 41,368 face images sized 32×32. The images were captured by 13 synchronized cameras and 21 flashes, under varying poses, illuminations, and expressions.

In our experiments, we simply adopt the preprocessed versions of PIE[6], i.e., **PIE1** [19] and **PIE2** [38], which are generated by randomly sampling the face images from the near-frontal poses (C27) under different lighting and illumination conditions. We construct one dataset *PIE1 vs PIE2* by selecting all 2,856 images in PIE1 to form the source domain, and all 3,329 images in PIE2 to form the target domain. We switch source/target pair to get another dataset *PIE2 vs PIE1*. Thus the source and target domains are guaranteed to follow different distributions in the same feature space, due to variations in lighting and illumination.

MSRC[7] dataset is provided by the computer vision group at Microsoft Research Cambridge, which contains 4,323 images labeled by 18 classes.

VOC2007[8] dataset (the training/validation subset) contains 5,011 images annotated with 20 concepts.

3. http://www-i6.informatik.rwth-aachen.de/~keysers/usps.html
4. http://yann.lecun.com/exdb/mnist
5. http://vasc.ri.cmu.edu/idb/html/face
6. http://www.cad.zju.edu.cn/home/dengcai/Data/FaceData.html
7. http://research.microsoft.com/en-us/projects/objectclassrecognition
8. http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2007

TABLE 4
Classification Accuracy (%) on 6 Cross-Domain Dataset Groups (216 Datasets) Generated from 20-Newsgroups

Dataset Group	Standard Learning			Transfer Learning					Our Methods		
	LR	SVM	LapSVM	SFA	TCA	LP	CMF	MTrick	GCMF	GTL_3	GTL_2
comp vs rec	88.37	87.51	81.93	89.73	95.12	95.21	95.73	95.96	97.72	**98.19**	**98.05±0.00**
comp vs sci	77.87	75.38	68.96	78.07	77.32	85.75	87.73	86.90	88.35	86.98	**91.43±0.00**
comp vs talk	96.31	95.44	95.40	95.85	97.20	96.89	97.11	97.77	98.25	**98.48**	**98.35±0.00**
rec vs sci	75.28	73.82	74.21	79.25	82.31	85.45	86.40	87.22	93.02	95.18	**95.95±0.03**
rec vs talk	82.28	83.27	87.44	86.98	86.58	94.16	94.89	94.33	97.70	**98.28**	**98.07±0.00**
sci vs talk	76.99	76.85	80.22	79.27	79.30	86.37	88.37	90.21	**96.17**	**96.32**	95.64±0.01
Average	82.85	82.05	81.36	84.86	86.31	90.64	91.70	92.06	95.20	95.57	**96.25±0.01**

From Fig. 1 we see that MSRC and VOC follow different distributions, since MSRC is from standard images for scientific studies, while VOC2007 is from digital photos in Flickr[9]. They share the following 6 semantic classes: "aeroplane", "bicycle", "bird", "car", "cow", "sheep". We construct one dataset *MSRC vs VOC* by selecting all 1,269 images in MSRC to form the source domain, and all 1,530 images in VOC2007 to form the target domain. We switch source/target pair to get another dataset *VOC vs MSRC*. We then uniformly rescale all images to be 256 pixels in length, and extract 128-dimensional dense SIFT (DSIFT) [39] features with grid size of 5 pixels. A 240-dimensional codebook is created, where K-means clustering is used to obtain the codewords. Thus the source and target domains are ensured to share the same feature space.

5.2 Experimental Setup

5.2.1 Baseline Methods

We compare GTL with nine state-of-the-art methods for cross-domain text and image classification tasks.

- Logistic Regression (LR)
- Support Vector Machine (SVM)
- Laplacian Support Vector Machine (LapSVM) [40]
- Spectral Feature Alignment (SFA) [4]
- Transfer Component Analysis (TCA) [15]
- Collective Matrix Factorization (CMF) [30]
- Label Propagation (LP) [9]
- Matrix Tri-Factorization Clustering (MTrick) [10]
- Our Graph Co-Regularized Collective Matrix Tri-Factorization (GCMF) in the preliminary work [1]

Specifically, CMF and GTL_2 adopt matrix factorization; LP, MTrick, GCMF, and GTL_3 adopt matrix tri-factorization. Our GCMF and GTL approaches distinguish from the other methods by imposing graph co-regularization on collective matrix (tri-) factorizations.

5.2.2 Implementation Details

Following [2], [10], [15], LR and SVM are trained on the labeled source data, and then tested on the unlabeled target data; SFA and TCA are run on all data as a dimensionality reduction procedure, then an LR classifier is trained on the source data to classify the target data; LapSVM, CMF, LP, MTrick, GCMF, and GTL are applied in a transductive way, i.e., trained on all data and then tested on unlabeled target data.

9. http://www.flickr.com

Under the experimental setup, it is impossible to automatically tune the optimal parameters for the target classifier using cross validation, since there are no labeled data in target domains. Thus, we evaluate the nine baseline methods on our datasets by empirically searching the parameter space for the optimal parameter settings, and report the best results. For LR[10] and SVM[11], we set the trade-off parameter C by searching $C \in \{0.1, 0.5, 1, 5, 10, 50, 100\}$. For LapSVM[12], we set the regularization parameters γ_A and γ_I by searching $\gamma_A, \gamma_I \in \{0.01, 0.05, 0.1, 0.5, 1, 5, 10\}$. For SFA, TCA, LP, MTrick, and GCMF, we set the optimal subspace dimension k by searching $k \in \{4, 8, 16, 32, 64, 128, 256\}$.

The GTL approaches has four model parameters: #nearest neighbors p, regularization parameters λ, γ, and σ. In the coming sections, we provide empirical analysis on parameter sensitivity, which validates that GTL can achieve stable performance under a wide range of parameter values. When comparing with the baseline methods, we fix $p = 10$, $\gamma = 10$, and $\sigma = 100$, and use these settings: 1) $\lambda = 0$ for text datasets, and 2) $\lambda = 0.1$ for image datasets. We set #iterations as $T = 100$ and run GTL 10 repeated times to remove any randomness caused by initialization.

We use the classification *Accuracy* on the test data (unlabeled target data) as the evaluation metric, since it is widely adopted in the literature [4], [13], [15], [36]

$$Accuracy = \frac{\left|\mathbf{x}:\mathbf{x} \in \bigcup_{\pi \in \Pi_t} \mathcal{D}_\pi \wedge f(\mathbf{x}) = y(\mathbf{x})\right|}{\left|\mathbf{x}:\mathbf{x} \in \bigcup_{\pi \in \Pi_t} \mathcal{D}_\pi\right|}, \quad (20)$$

where $y(\mathbf{x})$ is the groundtruth label of $\mathbf{x}$ while $f(\mathbf{x})$ is the label predicted by the classification algorithm.

5.3 Experimental Results

In this section, we compare GTL approaches with the baseline methods in terms of classification accuracy.

5.3.1 Cross-Domain Text Classification

As 20-Newsgroups and Reuters-21578 are different in hierarchical structure, we report the results separately.

20-Newsgroups: The average classification accuracy of GTL and the nine baseline methods on the 6 cross-domain dataset groups (216 datasets) are illustrated in Table 4. All the detailed results of the 6 dataset groups are listed through Figs. 2(a)∼2(f). Each of these six figures contains

10. http://www.csie.ntu.edu.tw/~cjlin/liblinear
11. http://www.csie.ntu.edu.tw/~cjlin/libsvm
12. http://vikas.sindhwani.org/manifoldregularization.html

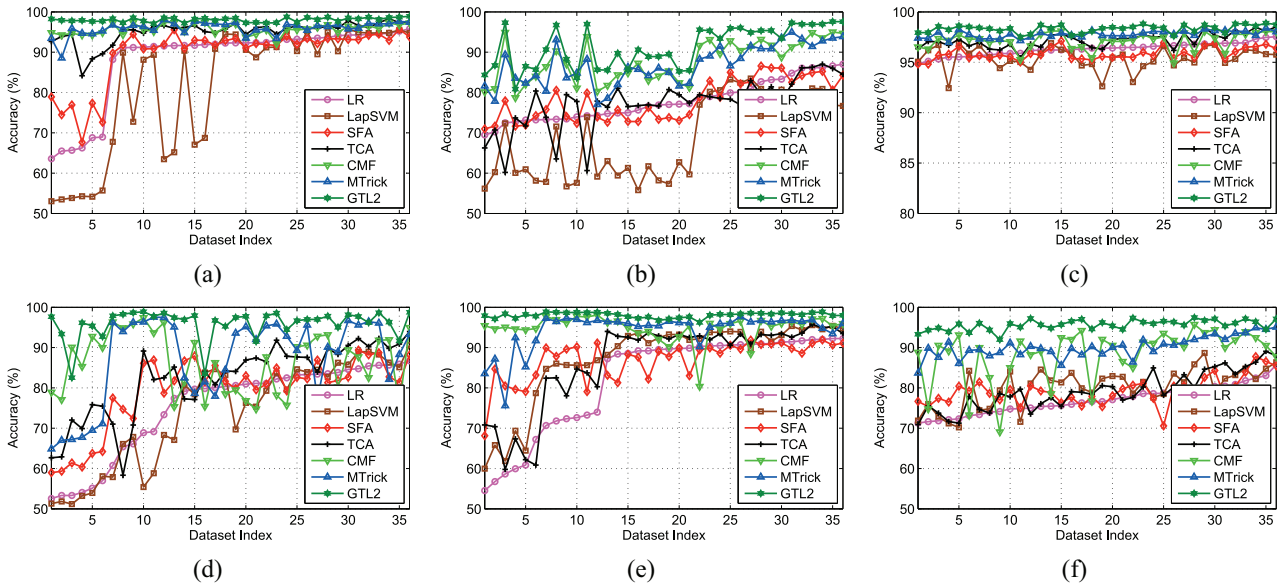


Fig. 2. Classification accuracy of LR, LapSVM, SFA, TCA, CMF, MTrick, and GTL_2 on *20-Newsgroups* datasets: (a) *comp vs rec.* (b) *comp vs sci.* (c) *comp vs talk.* (d) *rec vs sci.* (e) *rec vs talk.* (f) *sci vs talk.*

the results on the 36 datasets in the corresponding group. The 36 datasets are sorted by an increasing order of the classification accuracy obtained by Logistic Regression (LR). Therefore, the x-axis in each figure essentially indicates the degree of difficulty in cross-domain knowledge transfer. From these figures, we can make the following observations.

GTL can achieve much better performance than the first eight baseline methods (excluding GCMF) with statistical significance, and GTL_2 achieves slightly better performance than GTL_3. The average classification accuracy of GTL_2 on the 216 datasets is **96.25%**. The performance improvement is **4.19%** compared to the best baseline method MTrick, which means a significant error reduction of **52.78%**. Furthermore, from the results averaged by 10 repeated runs in Table 4, we see that the deviation is less than 0.01%, which validates that GTL performs stably to its random initialization.

Secondly, we observe that all the transfer learning methods can achieve better classification accuracy than the standard learning methods. A major limitation of existing standard learning methods is that they treat the data from different domains as if they were drawn from a homogenous distribution. In reality, the identical-distribution assumption does not hold in the cross-domain learning problems, and thus results in their unsatisfactory performance. It is important to notice that, the state-of-the-art semi-supervised learning method LapSVM cannot perform better than LR and SVM. Although LapSVM can explore the target data in a transductive way, it does not extract common factors to bridge domains and may overfit the target data when the domain difference is significantly large.

Thirdly, we notice that GTL has significantly outperformed SFA, TCA, LP, CMF, and MTrick, which are state-of-the-art transfer learning methods based on feature transformation. A major limitation of prior feature transformation based transfer learning methods is that they may be prone to overfitting the target data, due to their incapability to negotiate between the cross-domain statistical property and the in-domain geometric structure. In real-world applications, different properties from different domains are likely to be inconsistent with each other, resulting in a higher risk of negative transfer. Currently, GTL has alleviated these limitations and can achieve much better results.

Lastly, GTL often performs much more robustly than all the baseline methods under different degrees of difficulty in cross-domain knowledge transfer. This can be observed from Figs. 2(a)∼2(f), where each baseline method has many weak datasets while GTL performs effectively and stably on almost all datasets.

Reuters-21578: The classification accuracy of GTL and the baseline methods on the 6 datasets generated from Reuters-21578 are illustrated in Fig. 3(a). We observe that GTL has outperformed, or achieved comparable performance than the baseline methods.

We notice that, Reuters-21578 is more challenging than 20-Newsgroups, since each of its top categories consists of many subcategories, i.e., clusters or subclasses. Therefore, it is more difficult to propagate the supervision information between domains for knowledge transfer. This reason can explain the unsatisfactory performance obtained by the baseline methods.

GTL tackles the above limitation by simultaneously 1) extracting some common factors to propagate supervision information across domains, and 2) refining the latent factors in each domain to respect domain-specific geometric structure. Therefore, GTL can perform better on difficult datasets with many subclasses.

5.3.2 Cross-Domain Image Classification

The average classification accuracy of GTL_2, GTL_3 and the six baseline methods on the six image datasets is illustrated in Fig. 3(b). In this experiment, SFA is not compared since it cannot handle non-sparse image data, LapSVM is

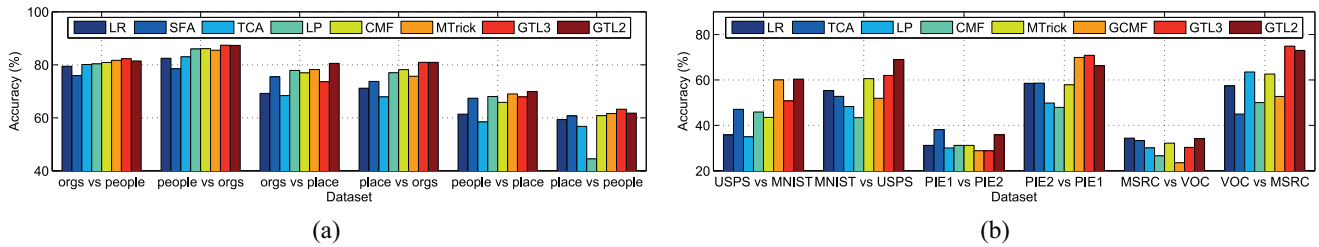


Fig. 3. Classification accuracy of LR, SFA, TCA, LP, CMF, MTrick, GCMF, and GTL on text and image datasets: (a) Reuters-21578. (b) Image datasets.

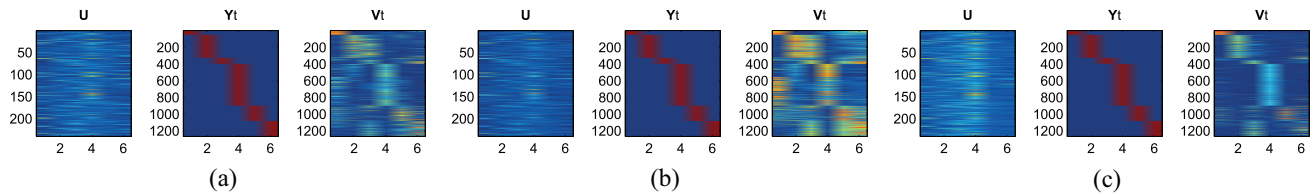


Fig. 4. Latent factor **U** and target predictions $\mathbf{V}_t$ output by GTL_2 on cross-domain image dataset *VOC vs MSRC*: (a) $\gamma = 0$. (b) $\sigma = 0$. (c) Optimal parameters.

not compared since its original implementation cannot deal with multi-class problems.

We observe that GTL_2 and GTL_3 significantly outperform all baseline methods on most of the datasets, which verifies that graph co-regularization is generally very effective for cross-domain knowledge transfer, and is transparent to base models (NMF and NMTF). However, GTL_3 does achieve considerably worse performance than GTL_2, which indicates that NMF may be a more suitable base model than NMTF for GTL. In this sense, GTL works better if sufficient knowledge can be transferred across domains (see Section 3.3.2).

It is also noteworthy that GTL has achieved much better performance than GCMF. The reason is that GCMF may suffer from the *trivial solution* problem. By imposing the orthogonality constraints, GTL can fully address this problem and thus perform more robustly.

We notice that, the transfer learning methods TCA, LP, CMF, and MTrick have underperformed standard learning methods LR on some datasets, e.g., *MNIST vs USPS* and *PIE2 vs PIE1*. This is an example of the *negative transfer* problem. In these datasets, the domain differences are so large that it is very difficult to extract some "common" factors to build up good connections for knowledge transfer. In other words, the common factor assumption made by the baseline methods is violated and thus negative transfer occurs.

Noteworthily, GTL performs robustly and does not suffer from the negative transfer problem. The main reason is that the domain-specific geometric structure is respected no matter whether some "common" latent factors can be discovered between different domains. In a word, if the statistical and geometric properties are inconsistent across domains, they can negotiate with each other and let in-domain geometric property dominate the target tasks to alleviate negative transfer.

5.4 Effectiveness Verification

We verify the effectiveness of GTL by inspecting the impacts of graph co-regularization and orthogonality.

5.4.1 Graph Co-Regularization

First, we remove the graph co-regularization terms by setting $\lambda = \gamma = 0$ as in Fig. 4a, where warmer colors indicate larger values. Notice that, each column of $\mathbf{V}_t$ corresponds to one example class, and each row of $\mathbf{V}_t$ is the class assignments to an example. Comparing $\mathbf{V}_t$ with $\mathbf{Y}_t$, the groundtruth target labels, we observe that many target examples are assigned to the wrong classes. The reason is that the geometric structure, i.e., similar examples should have similar labels, is not carefully preserved for each domain. In this case, the cross-domain statistical property and the in-domain geometric structure may be inconsistent, which leads to the violation of the target domain data distribution. By using graph co-regularization, GTL can respect the domain-specific geometric structure and thus output better class/cluster structure, as shown in Fig. 4(c).

5.4.2 Orthogonality Constraints

Secondly, we remove the orthogonality regularization term by setting $\sigma = 0$ as in Fig. 4(b). Comparing $\mathbf{V}_t$ with $\mathbf{Y}_t$, we observe that many target examples are assigned to multiple (and possibly wrong) classes. This is an example of the *trivial solution* problem. By imposing the shrinkage-style regularization for the orthogonality, this problem is fixed as in Fig. 4(c).

5.5 Parameter Sensitivity

We conduct empirical parameter sensitivity analysis, which validates that GTL can achieve optimal performance under wide range of parameter values. Due to space limitation, we randomly select one dataset from 20-Newsgroups,

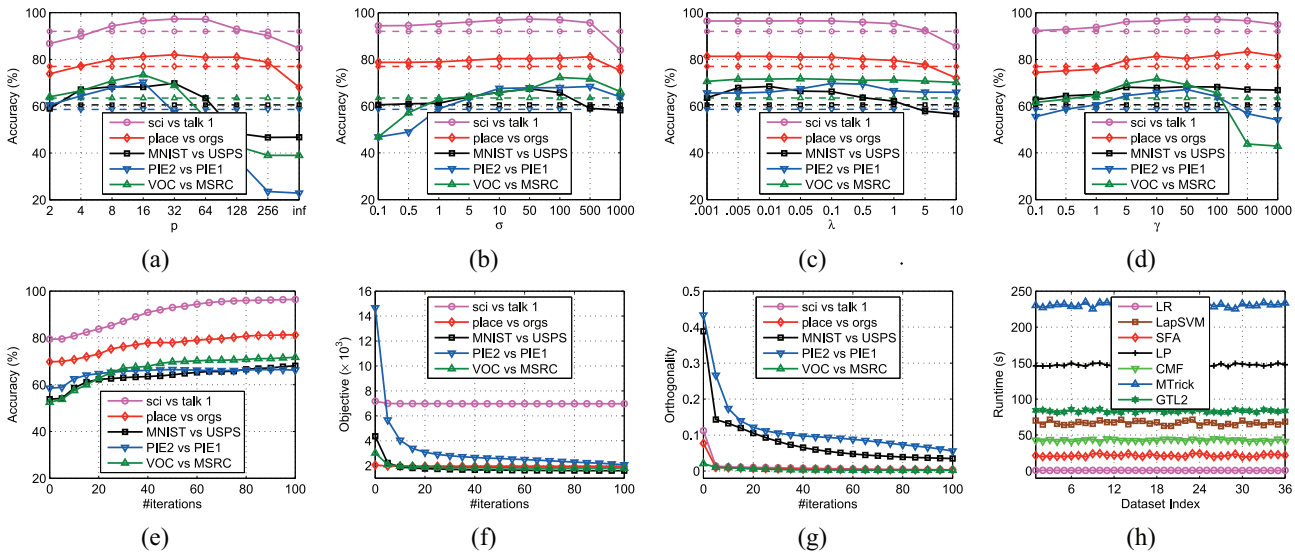


Fig. 5. Parameter sensitivity, convergence, and runtime of GTL_2 (dashed lines show the best baseline results): (a) #nearest neighbors p. (b) Orthogonality reg. σ. (c) Feature graph reg. λ. (d) Example graph reg. γ. (e) Accuracy w.r.t. #iterations. (f) Objective w.r.t. #iterations. (g) Orthogonality w.r.t. #iterations. (h) Runtime of algorithms.

Reuters-21578, USPS & MNIST, PIE, MSRC & VOC2007 respectively to discuss the results.

#Nearest Neighbors p: We run GTL_2 with varying values of p. Intuitively, larger values of p will result in denser nearest-neighbor graphs. To achieve optimal performance, p should be neither too large nor too small, since an extremely dense graph ($p \to \infty$) will connect two examples/features which are not similar at all, while an extremely sparse graph ($p \to 0$) will capture limited similarity information between examples/features. We plot the classification accuracy w.r.t. different values of p in Fig. 5(a), which indicates a wide range $p \in [4, 32]$ for optimal parameter values.

Orthogonality Regularization σ: We run GTL_2 with varying values of σ. Theoretically, σ controls the degree of orthogonality that is satisfied. When $\sigma \to 0$, GTL will be ill-defined and prone to trivial solutions. On the contrary, when $\sigma \to \infty$, GTL will be dominated by the shrinkage regularizer without any fitting. We plot the classification accuracy w.r.t. different values of σ in Fig. 5(b), and choose $\sigma \in [1, 500]$. As a rule of thumb, in practice we can often choose $\sigma \in [\gamma, 10\gamma]$.

Feature Graph Regularization λ: We run GTL_2 with varying values of λ. Theoretically, λ controls the weight of feature graph regularization, and larger values of λ make geometric structure on features more important in GTL. When $\lambda \to \infty$, only the geometric structure is preserved while labeled information is discarded. We plot classification accuracy w.r.t. different values of λ in Fig. 5(c), and choose $\lambda \in [0.001, 1]$. It is noteworthy that the feature graph regularization is much more effective for image data than text data.

Example Graph Regularization γ: We run GTL_2 with varying values of γ. Theoretically, γ controls the weight of example graph regularization, and larger values of γ make geometric structure on examples more important in GTL. When $\gamma \to \infty$, only geometric structure is preserved while labeled information is discarded. We plot classification accuracy w.r.t. different values of γ in Fig. 5(d), and choose $\gamma \in [1, 100]$.

5.6 Convergence Study

Since GTL is an iterative algorithm, we need to check its convergence property empirically by running GTL_2 on the five selected datasets. Fig. 5(e) shows the classification accuracy w.r.t. #iterations. Fig. 5(f) shows the objective function w.r.t. #iterations. From these figures we find that the classification accuracy (objective function) increases (decreases) steadily with more iterations and converges after 100 iterations.

Also, since we adopt a shrinkage methodology to satisfy the orthogonality constraints, we need to check the satisfaction of orthogonality. We run GTL_2 on the selected datasets, and plot the satisfaction of orthogonality, i.e., $\sum_{\pi \in \Pi} \left\| \mathbf{V}_\pi^{\mathrm{T}} \mathbf{V}_\pi - \mathbf{I} \right\|_F^2 / \sum_{\pi \in \Pi} \|\mathbf{I}\|_F^2$, in Fig. 5(g). We find that the orthogonality constraints are iteratively satisfied and converged after 100 iterations.

5.7 Time Complexity

We check the time complexity of all methods empirically by running them on the 36 *comp vs rec* datasets with 25,800 features and 8,000 documents, and show the results in Fig. 5(h). We see that GTL can achieve comparable time complexity as the baseline methods.

6 Conclusion

In this paper, we have proposed a general framework, referred to as Graph Co-Regularized Transfer Learning (GTL), to address the cross-domain learning problems. Specifically, GTL aims to extract common latent factors for

knowledge transfer by preserving the statistical property across domains, and simultaneously, refine the latent factors to alleviate negative transfer by preserving the geometric structure in each domain. An important advantage of GTL is that it can address the shortcomings of most existing transfer learning methods which focus on only one aspect of the data. Furthermore, many matrix factorization methods, e.g., NMF and NMTF, can be readily incorporated into the GTL framework to address transfer learning. Comprehensive experiments on 222 text datasets and 6 image datasets verify that GTL approaches can significantly outperform state-of-the-art transfer learning methods.

ACKNOWLEDGMENTS

This work was supported in part by the National HGJ Key Project (2010ZX01042-002-002), and in part by the National High-Tech R&D Program (2012AA040911), National Basic Research Program (2009CB320700), and National Natural Science Foundation of China (61073005, 61271394). Q. Yang thanks the support of Hong Kong RGC Projects 621211 and 620812. A preliminary version of this paper appeared in AAAI 2012 [1].

REFERENCES

[1] M. Long, J. Wang, G. Ding, D. Shen, and Q. Yang, "Transfer learning with graph co-regularization," in *Proc. 26th AAAI Conf. Artif. Intell.*, Toronto, ON, Canada, 2012.

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

[3] F. Zhuang *et al.*, "Mining distinction and commonality across multiple domains using generative model for text classification," *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 11, pp. 2025–2039, Nov. 2012.

[4] S. J. Pan, X. Ni, J.-T. Sun, Q. Yang, and Z. Chen, "Cross-domain sentiment classification via spectral feature alignment," in *Proc. 19th Int. Conf. WWW*, Raleigh, NC, USA, 2010.

[5] Y. Zhu *et al.*, "Heterogeneous transfer learning for image classification," in *Proc. 25th AAAI Conf. Artif. Intell.*, 2011.

[6] L. Li, K. Zhou, G.-R. Xue, H. Zha, and Y. Yu, "Video summarization via transferrable structured learning," in *Proc. 20th Int. Conf. WWW*, 2011.

[7] B. Li, Q. Yang, and X. Xue, "Transfer learning for collaborative filtering via a rating-matrix generative model," in *Proc. 26th ICML*, Montreal, QC, Canada, 2009.

[8] W. Dai, G.-R. Xue, Q. Yang, and Y. Yu, "Co-clustering based classification for out-of-domain documents," in *Proc. 13th ACM SIGKDD Int. Conf. KDD*, San Jose, CA, USA, 2007.

[9] Z. Wang, Y. Song, and C. Zhang, "Knowledge transfer on hybrid graph," in *Proc. 21st IJCAI*, San Francisco, CA, USA, 2009.

[10] F. Zhuang *et al.*, "Exploiting associations between word clusters and document classes for cross-domain text categorization," in *Proc. 10th SDM*, New York, NY, USA, 2010.

[11] H. Wang, H. Huang, F. Nie, and C. Ding, "Cross-language web page classification via dual knowledge transfer using nonnegative matrix tri-factorization," in *Proc. 34th Int. ACM SIGIR*, Beijing, China, 2011.

[12] M. Long *et al.*, "Dual transfer learning," in *Proc. 12th SDM*, 2012.

[13] X. Ling, W. Dai, G.-R. Xue, Q. Yang, and Y. Yu, "Spectral domain-transfer learning," in *Proc. 14th ACM SIGKDD*, Las Vegas, NV, USA, 2008.

[14] C. Wang and S. Mahadevan, "Manifold alignment without correspondence," in *Proc. 21st IJCAI*, San Francisco, CA, USA, 2009.

[15] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Trans. Neural Netw.*, vol. 22, no. 2, pp. 199–210, Feb. 2011.

[16] C. Wang and S. Mahadevan, "Heterogeneous domain adaptation using manifold alignment," in *Proc. 25th AAAI Conf. Artif. Intell.*, 2011.

[17] X. Shi, Q. Liu, W. Fan, and P. S. Yu, "Transfer across completely different feature spaces via spectral embedding," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 4, pp. 906–918, Apr. 2013.

[18] X. Zhu and J. Lafferty, "Harmonic mixtures: Combining mixture models and graph-based methods for inductive and scalable semi-supervised learning," in *Proc. 22nd ICML*, Bonn, Germany, 2005.

[19] D. Cai, X. He, J. Han, and T. S. Huang, "Graph regularized nonnegative matrix factorization for data representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1548–1560, Aug. 2011.

[20] C. Ding, T. Li, and W. Peng, "Nonnegative matrix factorization and probabilistic latent semantic indexing: Equivalence, chi-square statistic, and a hybrid method," in *Proc. 21st AAAI Conf. Artif. Intell.*, 2006.

[21] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Proc. NIPS 14*, 2000.

[22] C. H. Q. Ding, T. Li, and M. I. Jordan, "Convex and semi-nonnegative matrix factorizations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 1, pp. 45–55, Jan. 2010.

[23] C. Ding, T. Li, W. Peng, and H. Park, "Orthogonal nonnegative matrix tri-factorizations for clustering," in *Proc. 12th ACM SIGKDD*, New York, NY, USA, 2006.

[24] W. Dai, Q. Yang, G.-R. Xue, and Y. Yu, "Boosting for transfer learning," in *Proc. 24th ICML*, Corvallis, OR, USA, 2007.

[25] J. Jiang and C. Zhai, "Instance weighting for domain adaptation in nlp," in *Proc. 45th Annu. Meeting ACL*, Prague, Czech Republic, 2007.

[26] J. Blitzer, R. McDonald, and F. Pereira, "Domain adaptation with structural correspondence learning," in *Proc. 2006 Conf. EMNLP*. Stroudsburg, PA, USA.

[27] S. J. Pan, J. T. Kwok, and Q. Yang, "Transfer learning via dimensionality reduction," in *Proc. 22nd AAAI Conf. Artif. Intell.*, 2008.

[28] S. Si, D. Tao, and B. Geng, "Bregman divergence-based regularization for transfer subspace learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 7, pp. 929–942, Jul. 2010.

[29] T. Li, V. Sindhwani, C. Ding, and Y. Zhang, "Bridging domains with words: Opinion analysis with matrix tri-factorizations," in *Proc. 10th SDM*, 2010.

[30] A. P. Singh and G. J. Gordon, "Relational learning via collective matrix factorization," in *Proc. 14th ACM SIGKDD*, New York, NY, USA, 2008.

[31] G.-R. Xue, W. Dai, Q. Yang, and Y. Yu, "Topic-bridged PLSA for cross-domain text classification," in *Proc. 31st ACM SIGIR*, Singapore, 2008.

[32] D. Cai, X. He, X. Wang, H. Bao, and J. Han, "Locality preserving nonnegative matrix factorization," in *Proc. 21st IJCAI*, 2009.

[33] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," in *Proc. NIPS 15*, 2001.

[34] Q. Gu and J. Zhou, "Co-clustering on manifolds," in *Proc. 15th ACM SIGKDD*, Paris, France, 2009.

[35] Q. Gu, C. Ding, and J. Han, "On trivial solution and scale transfer problems in graph regularized NMF," in *Proc. 22nd IJCAI*, Barcelona, Spain, 2011.

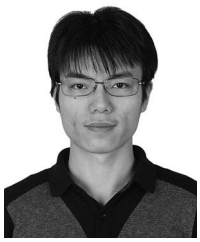
[36] L. Duan, I. W. Tsang, and D. Xu, "Domain transfer multiple kernel learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 3, pp. 465–479, Mar. 2012.

[37] J. Gao, W. Fan, J. Jiang, and J. Han, "Knowledge transfer via multiple model local structure mapping," in *Proc. 14th ACM SIGKDD*, Las Vegas, NV, USA, 2008.

[38] D. Cai, X. He, and J. Han, "Spectral regression: A unified approach for sparse subspace learning," in *Proc. 7th IEEE ICDM*, Omaha, NE, USA, 2007.

[39] A. Vedaldi and B. Fulkerson. (2008). VLFeat: *An Open and Portable Library of Computer Vision Algorithms* [Online]. Available: http://www.vlfeat.org/

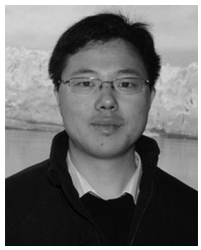
[40] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples," *J. Mach. Learn. Res.*, vol. 7, pp. 2399–2434, Nov. 2006.



Mingsheng Long received the B.S. degree in 2008, from the Department of Electrical Engineering, Tsinghua University, Beijing, China. He is a Ph.D. candidate at the Department of Computer Science and Technology, Tsinghua University. His current research interests include transfer learning, feature learning, sparse learning, and large-scale data mining.

Jianmin Wang graduated from Peking University, Beijing, China, in 1990, and received the M.E. and the Ph.D. degrees in computer software from Tsinghua University, China, in 1992 and 1995, respectively. He is now a Professor in the School of Software, Tsinghua University. His current research interests include unstructured data management, workflow and BPM technology, benchmark for database system, information system security, and big data analytics. He has published over 100 DBLP indexed papers in journals (*TKDE, DMKD, DKE*, and *WWWJ*, etc.) and conferences (SIGMOD, VLDB, ICDE, CVPR, and AAAI, etc). He has led to develop a product data/lifecycle management system, which has been implemented in hundreds of enterprises in China. Now he leads to develop an unstructured data management system, LaUDMS.

Guiguang Ding received the Ph.D. degree in electronic engineering from the University of Xidian, Xi'an, China. He is an Associate Professor in the School of Software, Tsinghua University. His current research interests include the area of multimedia information retrieval and mining, with specific focus on visual object recognition, automatic semantic annotation, image coding and representation, and social media recommendation. He has published over 40 research papers in international conferences and journals and applied for 18 Patent Rights in China.



Dou Shen received the Ph.D. degree in computer science from the Hong Kong University of Science and Technology (HKUST). He is a Director in Baidu Inc., Beijing, China. His current research interests include data mining and machine learning, information retrieval, and computational advertising. During his study in HKUST, he led a team participating in KDDCUP and won all the three prizes. He has published over 40 journal and conference papers and invented 10 patents. He is serving as a Program Committee Member for the major conferences in the field (KDD, SIGIR, WWW, WSDM, AAAI, SDM, and ICDM). He co-organized the data mining and audience intelligence for advertising workshops in conjunction with KDD in 2007, 2008, 2009, and 2010.

Qiang Yang is the Head of Huawei Noah's Ark Research Lab and a Professor in the Department of Computer Science and Engineering, Hong Kong University of Science and Technology. His current research interests include data mining and artificial intelligence including machine learning, planning and activity recognition. He is a Fellow of the IEEE, IAPR, and AAAS. He received the Ph.D. degree from Computer Science Department, University of Maryland, College Park, in 1989. He was an invited speaker at IJCAI 2009, ACL 2009, SDM 2012, WSDM 2013, etc. He was elected as a Vice Chair of ACM SIGART in July 2010. He is the founding Editor-in-Chief of the *ACM Transactions on Intelligent Systems and Technology* (*ACM TIST*). He has served as a PC Co-Chair and General Co-Chair of several international conferences, including ACM KDD 2010 and 2012, ACM RecSys 2013, etc.