Triplex Transfer Learning: Exploiting Both Shared and Distinct Concepts for Text Classification

Fuzhen Zhuang, Ping Luo, *Member, IEEE,* Changying Du, Qing He, Zhongzhi Shi, *Senior Member, IEEE,* and Hui Xiong, *Senior Member, IEEE*

***Abstract*—Transfer learning focuses on the learning scenarios when the test data from target domains and the training data from source domains are drawn from similar but different data distributions with respect to the raw features. Along this line, some recent studies revealed that the high-level concepts, such as word clusters, could help model the differences of data distributions, and thus are more appropriate for classification. In other words, these methods assume that all the data domains have the same set of shared concepts, which are used as the bridge for knowledge transfer. However, in addition to these shared concepts, each domain may have its own distinct concepts. In light of this, we systemically analyze the high-level concepts, and propose a general transfer learning framework based on nonnegative matrix trifactorization, which allows to explore both shared and distinct concepts among all the domains simultaneously. Since this model provides more flexibility in fitting the data, it can lead to better classification accuracy. Moreover, we propose to regularize the manifold structure in the target domains to improve the prediction performances. To solve the proposed optimization problem, we also develop an iterative algorithm and theoretically analyze its convergence properties. Finally, extensive experiments show that the proposed model can outperform the baseline methods with a significant margin. In particular, we show that our method works much better for the more challenging tasks when there are distinct concepts in the data.**

***Index Terms*—Common concept, distinct concept, distribution mismatch, nonnegative matrix trifactorization, triplex transfer learning.**

I. Introduction

TRADITIONAL classification algorithms often fail to obtain satisfying performance, since in many emerging real-world applications, new test data usually come from different data sources with different but semantically-related distributions. For example, to build a news portal for any of the Fortune 500 companies, we want to classify the everyday news about this company into some classes, such as product-related, financial report, business and industry analysis, stock review, merger and acquisition related, and so on. The traditional classification model learned from the news of a company may not perform well on the news of another company since these two companies may have different business areas, and thus, the distributions on the raw words in the two news corpora may be different. To reduce the manual effort in labeling the training data in the new domain, leads to a vast amount of studies in transfer learning (also referred to as domain adaptation, cross-domain learning) [1]–[14]. It aims at adapting the classification models trained from the source domains to the target domains with different data distributions.

Although the source and target domains have different data distributions in raw word features, many recent studies exploit the commonality between different domains for knowledge transfer [5], [9], [11], [12]. In these studies, the high-level concepts (i.e., word clusters and topics) are utilized with the observation that different domains may use different keywords to express the same concept while the association between the concepts and the document classes may be stable across domains [11]. In this paper, we refer to the set of keywords in expressing a concept as the extension of this concept, in other words, the extension of a concept can be described as the distribution over words. On the other hand, we refer to the association between the concepts and the document classes as the concept intension, which can also be expressed as the indication to a document class. With these terminologies, the widely used observation actually says that the extension of a concept may be different in different domains while its intension is stable across all the domains. This basic observation motivates these recent studies to use the stable concept intension as the bridge for knowledge transfer.

It is clear that most of the previous works assume that all the data domains share the same set of concepts with their respective stable intensions. However, it is not always true since some distinct concepts may only exist in a text corpus, which are totally irrelevant to the content of another corpus. For example, some company is launching the business combination, and it may apply the keywords consolidation, amalgamation, and so

Manuscript received March 12, 2013; revised July 4, 2013; accepted September 1, 2013. A preliminary version of this work has been published in ACM WSDM 2013 [1]. This work was supported in part by the National Natural Science Foundation of China under Grant 61175052, Grant 61203297, Grant 60933004, and Grant 61035003, in part by the National High-Tech Research and Development Program of China (863 Program) under Grant 2013AA01A606 and Grant 2012AA011003, and in part by the National Program on Key Basic Research Project (973 Program) under Grant 2013CB329502. This paper was recommended by Associate Editor H. Wang.

F. Zhuang, C. Du, Q. He, and Z. Shi are with the Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100864, China (e-mail: zhuangfz@ics.ict.ac.cn; ducy@icc.ict.ac.cn; heq@ics.ict.ac.cn; shizz@ics.ict.ac.cn).

P. Luo is with the Hewlett-Packard Laboratories, Beijing 100084, China (e-mail: ping.luol@hp.com).

H. Xiong is with the Management Science and Information Systems Department, Rutgers Business School, Rutgers University, Newark, NJ 08901 USA (e-mail: hxiong@rutgers.edu).

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

Digital Object Identifier 10.1109/TCYB.2013.2281451

TABLE I
THREE KINDS OF CONCEPTS

		Extension	Intension
Shared	Identical Concepts	*same*	*same*
	Alike Concepts	*different*	*same*
Non-shared	Distinct Concepts	*different*	*different*

TABLE II
COMPARISON OF MODELS

	Alike	Identical	Distinct
CoCC [6]		√	
MTrick [10]	√		
DKT [12]	√		
DTL [13]	√	√	
TriTL	√	√	√

on. Thus, these distinct concepts in Definition 1 have both different extensions and different intensions.

Definition 1 (Distinct Concepts): A concept is distinct when it has both different extension and different intension with any other concepts.

Additionally, all the shared concepts can be further divided into two groups, namely, alike concepts and identical concepts, defined as follows. The alike concepts have the same intension but different extension with others'. They are actually widely used in previous works [9]. Meanwhile, there may be some concepts with both the same intension and the same extension with others' as shown in [12]. They are the identical concepts.

Definition 2 (Alike Concepts): A concept is alike to some other ones when it has the same intension but different extension with others'.

Definition 3 (Identical Concepts): A concept is identical with some other ones when it has both the same intension and the same extension with others'.

These three kinds of concepts are summarized in Table I. They may all exist in the multiple corpora. However, all the previous works never consider these three kinds of concepts together for classification, and only address them separately or partially. For example, CoCC [5] modeled the identical concepts only. MTrick [9] exploited the associations between word clusters and document classes for cross-domain classification, thus actually considered the alike concepts only. DKT [11] adopted the similar idea with MTrick for cross-language web page classification. Recently, dual transfer learning (DTL) [12] was proposed to model alike and identical concepts together. Therefore, an ideal model should handle the identical, alike, and distinct concepts simultaneously. Motivated by this observation, we propose a general framework based on nonnegative matrix trifactorization (NMTF) techniques, which consider all these concepts jointly. We believe that the more flexibility in modeling the data may improve the classification accuracy. Since our model considers the three kinds of concepts, we call it Triplex transfer learning (TriTL). For the sake of clarity, the differences of the four previous methods and our model are summarized in Table II.

In our previous work [15], we analyzed the commonality and distinction of the source and target domains, and revealed that there might also exist some distinct concepts in each of the data domain. In that paper, we introduced distinct concepts together with alike and identical concepts into transfer learning, and developed a TriTL model to exploit them simultaneously based on nonnegative matrix trifactorization. Along this line, an iterative algorithm was developed to solve the proposed matrix factorization problem, and the theoretical analysis of the algorithm convergence was also provided. Finally, we conducted extensive experiments to show the superiority of TriTL over the compared methods. In particular, we showed that our method works much better in the more challenging tasks when distinct concepts exist.

Indeed, in this paper, we further exploit the intrinsic structure of the target domains, and propose to regularize the manifold structure to enhance the prediction performances of TriTL. The experiments show the additional improvements of accuracy compared with TriTL. In summary, we have the following contributions in this paper.

1) First, we further consider the clustering assumption of the manifold structure, and propose to regularize triplex transfer learning (RTriTL) as shown in Section III-C.
2) Second, we theoretically analyze the computational complexity of the proposed iterative algorithm to show the efficiency of TriTL as shown in Section III-D.
3) Third, we conduct much more experiments to demonstrate the effectiveness of TriTL and RTriTL. These experiments include additional 269 new transfer learning tasks and three classification problems constructed from Reuters-21578 data set, compared with the ones in [15]. It is worth noting that RTriTL can further significantly improve the classification accuracy compared with TriTL as shown in Section IV-C2.
4) Finally, three types of word concepts captured by TriTL are empirically evaluated, and the experimental results show that our TriTL model can effectively identify the shared and distinct concepts for knowledge transfer as shown in Section IV-F.

Overview. The rest of this paper is organized as follows. Section II briefly introduces the preliminary knowledge and math notations. In Section III, we show the proposed model TriTL and the enhanced version of TriTL. Section IV gives the experimental results. In Section V, we summarize the related works. Section VI concludes this paper. Finally, in the Appendix, we provide the theoretical analysis of the iterative algorithm.

II. PRELIMINARY KNOWLEDGE

In this section, we first give the notations used throughout this paper, and, then, briefly introduce the nonnegative matrix trifactorization (NMTF) technique and its notions.

A. Notations

We use calligraphic letters to represent sets, such as $\mathcal{D}$ is used to denote dataset. The data matrices are written in uppercase, such as X and Y, and $X_{[i,j]}$ indicates the ith row and jth column element of matrix X. Also, we use $\mathbb{R}$ and $\mathbb{R}_+$ to

TABLE III
NOTATION AND DENOTATION

$\mathcal{D}$	The data set
X	The word-document co-occurrence matrix from a domain
m	The number of words
n	The number of documents
c	The number of document classes
r	The index of domain
s	The number of source domains
t	The number of target domains
k_1	The number of identical concepts
k_2	The number of alike concepts
k_3	The number of distinct concepts
F	The matrix for the word clusterings
S	The matrix for the association between word clusters and document classes
G	The matrix for the document labeling
$\top$	The transposition of a matrix

denote the set of real numbers and nonnegative real numbers, respectively. Finally, $\mathbf{1}_m$ is used to represent a column vector with size m, and its elements are all equal to one. For clarity, the frequently-used notations and denotations are summarized in Table III.

B. Nonnegative Matrix Factorization

Nonnegative matrix factorization (NMF) technique has been widely used for text and image classification in the last decade [16]–[19]. Our model is based on the nonnegative matrix trifactorization, and the basic formula is

$$X_{m\times n} = F_{m\times k} S_{k\times c} G_{n\times c}^{\top} \tag{1}$$

where X is the word-document matrix, and m, n, k, c are the numbers of words, documents, word clusters, and document classes, respectively, $G^{\top}$ is the transposition of G. Conceptually, the matrix of F contains the information of word clusterings. G denotes the document labeling information, and S denotes the association between word clusters and document classes [9]. In this paper, each column of F refers to a concept and each row of G refers to a document. The details on these matrices will be addressed later.

Here, we also introduce some concepts about NMF, which are used in Section III and the Appendix.

Definition 4 (Trace of Matrix): Given a data matrix $X \in \mathbb{R}^{n\times n}$, the trace of X is computed as

$$tr(X) = \sum_{i=1}^{n} X_{[i,i]}. \tag{2}$$

In fact, the trace of matrix can also be computed when the matrix is not a phalanx. Without losing any generality, let $m < n$ and $X \in \mathbb{R}^{m\times n}$, then $tr(X) = \sum_{i=1}^{m} X_{[i,i]}$.

Definition 5 (Frobenius Norm of Matrix): Given a data matrix $X \in \mathbb{R}^{m\times n}$, the Frobenius norm of X is computed as

$$||X|| = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} X_{[i,j]}^2}. \tag{3}$$

The properties of the trace and Frobenius norm are as follows:

Property 1: Given a matrix $X \in \mathbb{R}^{m\times n}$, then

$$tr(X^T X) = tr(X X^T). \tag{4}$$

Property 2: Given two matrices $X, Y \in \mathbb{R}^{m\times n}$, then

$$tr(a \cdot X + b \cdot Y) = a \cdot tr(X) + b \cdot tr(Y). \tag{5}$$

Property 3: Given a matrix $X \in \mathbb{R}^{m\times n}$, then

$$||X||^2 = tr(X^T X) = tr(X X^T). \tag{6}$$

III. TRIPLEX TRANSFER LEARNING

Motivated by the observation on the three kinds of concepts, we divide F and S into three parts, respectively. Namely, $F = [F_{m\times k_1}^1, F_{m\times k_2}^2, F_{m\times k_3}^3]$ $(k_1 + k_2 + k_3 = k)$, where F^1 refers to the word clustering information for the identical concepts, F^2 refers to the word clustering information for the alike concepts, and F^3 refers to the word clustering information for the distinct concepts. Correspondingly, the association S can be denoted as

$$S = \begin{bmatrix} S_{k_1\times c}^1 \\ S_{k_2\times c}^2 \\ S_{k_3\times c}^3 \end{bmatrix}$$

where S^1 refers to the association between the identical concepts and document classes, S^2 refers to the association between the alike concepts and document classes, and S^3 refers to the association between the distinct concepts and document classes. Thus, (1) can be rewritten as

$$\begin{aligned} X_{m\times n} &= F_{m\times k} S_{k\times c} G_{n\times c}^{T} \\ &= [F_{m\times k_1}^1, F_{m\times k_2}^2, F_{m\times k_3}^3] \begin{bmatrix} S_{k_1\times c}^1 \\ S_{k_2\times c}^2 \\ S_{k_3\times c}^3 \end{bmatrix} G_{n\times c}^{\top}. \end{aligned} \tag{7}$$

Based on (7), we will formulate the transfer learning framework in the following.

A. Problem Formalization

Suppose, we have $s + t$ data domains, denoted as $\mathcal{D} = (\mathcal{D}_1, \cdots, \mathcal{D}_s, \mathcal{D}_{s+1}, \cdots, \mathcal{D}_{s+t})$. Without loss of generality, we assume the first s domains are source domains with the document labels, i.e., $\mathcal{D}_r = \{x_i^{(r)}, y_i^{(r)}\}|_{i=1}^{n_r}$ $(1 \le r \le s)$, and the left t domains are target domains without any label information, i.e., $\mathcal{D}_r = \{x_i^{(r)}\}|_{i=1}^{n_r}$ $(s{+}1 \le r \le s{+}t)$. n_r is the number of documents in data domain $\mathcal{D}_r$. Let $X = (X_1, \cdots, X_s, X_{s+1}, \cdots, X_{s+t})$ be the word-document co-occurrence matrices of $s + t$ domains, then the objective function is formulated as follows:

$$\mathcal{L} = \sum_{r=1}^{s+t} ||X_r - F_r S_r G_r^{\top}||^2 \tag{8}$$

where $X_r \in \mathbb{R}_+^{m\times n_r}$, $F_r \in \mathbb{R}_+^{m\times k}$, $S_r \in \mathbb{R}_+^{k\times c}$ and $G_r \in \mathbb{R}_+^{n_r\times c}$.

As described earlier, we divide the word clustering matrix F_r into three parts $F_r = [F^1, {F^2}_r, {F^3}_r]$ $(F^1 \in \mathbb{R}_+^{m\times k_1}, {F^2}_r \in \mathbb{R}_+^{m\times k_2}, {F^3}_r \in \mathbb{R}_+^{m\times k_3}, k_1 + k_2 + k_3 = k)$. Here, since F^1 refers to the word clusterings on the identical concepts, it is shared in all the domains (note that F^1 does not have the subindex of r). While ${F^2}_r$ and ${F^3}_r$ refers to the word clusterings on the alike and distinct concepts, they are different in different domains (note that ${F^2}_r$ and ${F^3}_r$ do have the subindex of r).

Similarly, S_r can be expressed as $S_r = \begin{bmatrix} S^1 \\ S^2 \\ {S^3}_r \end{bmatrix}$ ($S^1 \in \mathbb{R}_+^{k_1 \times c}$, $S^2 \in \mathbb{R}_+^{k_2 \times c}$, ${S^3}_r \in \mathbb{R}_+^{k_3 \times c}$). Here, S^1 (S^2) are the associations between the identical (alike) concepts and document classes. Thus, they are shared in all the domains (note that S^1 and S^2 do not have the subindex of r). However, ${S^3}_r$ represents the association between distinct concepts and document classes. Thus, it is domain dependent (note that ${S^3}_r$ does have the subindex of r).

Therefore, the objective function in (8) can be rewritten as follows:

$$\begin{aligned} \mathcal{L} &= \sum_{r=1}^{s+t} ||X_r - F_r S_r G_r^\top||^2 \\ &= \sum_{r=1}^{s+t} ||X_r - [F^1, {F^2}_r, {F^3}_r] \begin{bmatrix} S^1 \\ S^2 \\ {S^3}_r \end{bmatrix} G_r^\top ||^2. \end{aligned} \tag{9}$$

Considering the constraints to F_r and G_r, we come to the optimization problem as

$$\begin{aligned} &\min_{F_r, S_r, G_r} \mathcal{L} \\ &s.t. \sum_{i=1}^{m} {F^1}_{[i,j]} = 1, \sum_{i=1}^{m} {F^2}_{r[i,j]} = 1, \\ &\sum_{i=1}^{m} {F^3}_{r[i,j]} = 1, \sum_{j=1}^{c} G_{r[i,j]} = 1. \end{aligned} \tag{10}$$

Here, the constraints enforce that the sum of the entries in each column of F equals to one and the sum of the entries in each row of G equals to one. In other words, each column of F refers to the word distribution of a concept while each row of G refers to the probabilities that a document belongs to different document classes.

B. Solution to TriTL

To solve the optimization problem in (10), we derive an iterative algorithm. According to the properties of the trace and Frobenius norm, the minimization of (10) is equal to minimize the following objective function:

$$\begin{aligned} \mathcal{L} = &\textstyle\sum_{r=1}^{s+t} ||X_r - [F^1, {F^2}_r, {F^3}_r] \begin{bmatrix} S^1 \\ S^2 \\ {S^3}_r \end{bmatrix} G_r^\top ||^2 \\ = &\sum_{r=1}^{s+t} tr(X_r^\top X_r - 2 \cdot X_r^\top [F^1, {F^2}_r, {F^3}_r] \begin{bmatrix} S^1 \\ S^2 \\ {S^3}_r \end{bmatrix} G_r^\top \\ &+ G_r \begin{bmatrix} S^1 \\ S^2 \\ {S^3}_r \end{bmatrix}^\top [F^1, {F^2}_r, {F^3}_r]^\top [F^1, {F^2}_r, {F^3}_r] \begin{bmatrix} S^1 \\ S^2 \\ {S^3}_r \end{bmatrix} G_r^\top) \\ = &\sum_{r=1}^{s+t} tr(X_r^\top X_r - 2 \cdot X_r^\top A_r - 2 \cdot X_r^\top B_r - 2 \cdot X_r^\top C_r \\ &+ G_r {S^1}^\top {F^1}^\top A_r + G_r {S^2}^\top {F^2}_r^\top B_r + G_r {S^3}_r^\top {F^3}_r^\top C_r \\ &+ 2 \cdot G_r {S^1}^\top {F^1}^\top B_r + 2 \cdot G_r {S^1}^\top {F^1}^\top C_r + 2 \cdot G_r {S^2}^\top {F^2}_r^\top C_r) \end{aligned} \tag{11}$$

$$\begin{aligned} s.t. &\sum_{i=1}^{m} {F^1}_{[i,j]} = 1, \sum_{i=1}^{m} {F^2}_{r[i,j]} = 1 \\ &\sum_{i=1}^{m} {F^3}_{r[i,j]} = 1, \sum_{j=1}^{c} G_{r[i,j]} = 1 \end{aligned}$$

where $A_r = F^1 S^1 G_r^\top$, $B_r = {F^2}_r S^2 G_r^\top$, $C_r = {F^3}_r {S^3}_r G_r^\top$. The partial differentials of $\mathcal{L}$ are as follows:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial F^1} = &\sum_{r=1}^{s+t} (-2 \cdot X_r G_r {S^1}^\top + 2 \cdot A_r G_r {S^1}^\top \\ &+ 2 \cdot B_r G_r {S^1}^\top + 2 \cdot C_r G_r {S^1}^\top) \end{aligned} \tag{12}$$

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial {F^2}_r} = &- 2 \cdot X_r G_r {S^2}^\top + 2 \cdot B_r G_r {S^2}^\top \\ &+ 2 \cdot A_r G_r {S^2}^\top + 2 \cdot C_r G_r {S^2}^\top \end{aligned} \tag{13}$$

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial {F^3}_r} = &- 2 \cdot X_r G_r {S^3}_r^\top + 2 \cdot C_r G_r {S^3}_r^\top \\ &+ 2 \cdot A_r G_r {S^3}_r^\top + 2 \cdot B_r G_r {S^3}_r^\top \end{aligned} \tag{14}$$

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial S^1} = &\sum_{r=1}^{s+t} (-2 \cdot {F^1}^\top X_r G_r + 2 \cdot {F^1}^\top A_r G_r \\ &+ 2 \cdot {F^1}^\top B_r G_r + 2 \cdot {F^1}^\top C_r G_r) \end{aligned} \tag{15}$$

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial S^2} = &\sum_{r=1}^{s+t} (-2 \cdot {F^2}_r^\top X_r G_r + 2 \cdot {F^2}_r^\top B_r G_r \\ &+ 2 \cdot {F^2}_r^\top A_r G_r + 2 \cdot {F^2}_r^\top C_r G_r) \end{aligned} \tag{16}$$

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial {S^3}_r} = &- 2 \cdot {F^3}_r^\top X_r G_r + 2 \cdot {F^3}_r^\top C_r G_r \\ &+ 2 \cdot {F^3}_r^\top A_r G_r + 2 \cdot {F^3}_r^\top B_r G_r \end{aligned} \tag{17}$$

$$\frac{\partial \mathcal{L}}{\partial G_r} = -2 \cdot X_r^\top F_r S_r + 2 \cdot G_r S_r^\top F_r^\top F_r S_r. \tag{18}$$

Note that when $r = \{1, \cdots, s\}$, G_r is the true label information, so we just need to solve G_r when $r = \{s+1, \cdots, s+t\}$. Since $\mathcal{L}$ is not concave, it is hard to obtain the global solution by applying the latest nonlinear optimization techniques. In this paper, we develop an alternately iterative algorithm, which can converge to a local optimal solution.

In each round of iteration these matrices are updated as

$$\begin{aligned} {F^1}_{[i,j]} \leftarrow &{F^1}_{[i,j]} \\ &\cdot \sqrt{\frac{[\sum_{r=1}^{s+t} X_r G_r {S^1}^\top]_{[i,j]}}{[\sum_{r=1}^{s+t} (A_r G_r {S^1}^\top + B_r G_r {S^1}^\top + C_r G_r {S^1}^\top)]_{[i,j]}}} \end{aligned} \tag{19}$$

$${F^2}_{r[i,j]} \leftarrow {F^2}_{r[i,j]} \cdot \sqrt{\frac{[X_r G_r {S^2}^\top]_{[i,j]}}{[B_r G_r {S^2}^\top + A_r G_r {S^2}^\top + C_r G_r {S^2}^\top]_{[i,j]}}} \tag{20}$$

$${F^3}_{r[i,j]} \leftarrow {F^3}_{r[i,j]} \cdot \sqrt{\frac{[X_r G_r {S^3}_r^\top]_{[i,j]}}{[C_r G_r {S^3}_r^\top + A_r G_r {S^3}_r^\top + B_r G_r {S^3}_r^\top]_{[i,j]}}} \tag{21}$$

$$S^1{}_{[i,j]} \leftarrow S^1{}_{[i,j]} \cdot \sqrt{\frac{[\sum_{r=1}^{s+t} F^{1\top} X_r G_r]_{[i,j]}}{[\sum_{r=1}^{s+t}(F^{1\top} A_r G_r + F^{1\top} B_r G_r + F^{1\top} C_r G_r)]_{[i,j]}}} \tag{22}$$

$$S^2{}_{[i,j]} \leftarrow S^2{}_{[i,j]} \cdot \sqrt{\frac{[\sum_{r=1}^{s+t} F^{2}{}_r^{\top} X_r G_r]_{[i,j]}}{[\sum_{r=1}^{s+t}(F^{2}{}_r^{\top} B_r G_r + F^{2}{}_r^{\top} A_r G_r + F^{2}{}_r^{\top} C_r G_r))]_{[i,j]}}} \tag{23}$$

$$S^3{}_{r[i,j]} \leftarrow S^3{}_{r[i,j]} \cdot \sqrt{\frac{[F^{3}{}_r^{\top} X_r G_r]_{[i,j]}}{[F^{3}{}_r^{\top} C_r G_r + F^{3}{}_r^{\top} A_r G_r + F^{3}{}_r^{\top} B_r G_r]_{[i,j]}}} \tag{24}$$

$$G_{r[i,j]} \leftarrow G_{r[i,j]} \cdot \sqrt{\frac{[X_r^{\top} F_r S_r]_{[i,j]}}{[G_r S_r^{\top} F_r^{\top} F_r S_r]_{[i,j]}}}. \tag{25}$$

After the calculation of each round of iteration, F^1, $F^2{}_r$, $F^3{}_r$, G_r are normalized using (26) to satisfy the equality constraints

$$\begin{aligned} F^1{}_{[i,j]} &\leftarrow \frac{F^1{}_{[i,j]}}{\sum_{i=1}^{m} F^1{}_{[i,j]}}, \; F^2{}_{r[i,j]} \leftarrow \frac{F^2{}_{r[i,j]}}{\sum_{i=1}^{m} F^2{}_{r[i,j]}} \\ F^3{}_{r[i,j]} &\leftarrow \frac{F^3{}_{r[i,j]}}{\sum_{i=1}^{m} F^3{}_{r[i,j]}}, \; G_{r[i,j]} \leftarrow \frac{G_{r[i,j]}}{\sum_{j=1}^{c} G_{r[i,j]}}. \end{aligned} \tag{26}$$

The detailed procedure of this iterative algorithm is described in Algorithm 1. In this algorithm, the data matrices are normalized such that $X_{r[i,j]}=\frac{X_{r[i,j]}}{\sum_{i=1}^{m} X_{r[i,j]}}$, G_r $(1 \le r \le s)$ are assigned as the true label information. Specifically, $G_{r[i,u]} = 1$ if the ith document belongs to the uth class, else $G_{r[i,v]} = 0$ $(v \neq u)$. F^1 and $F^2{}_r$ are initialized as the word clustering results by PLSA [20]. Specifically, we combine all the data from source and target domains, and conduct the PLSA implemented by Matlab.[1] We set the number of topics as $(k_1 + k_2)$, and obtain the word clustering information $W \in \mathbb{R}_+^{m\times(k_1+k_2)}$. W is divided into two parts $W = [W^1, W^2]$ $(W^1 \in \mathbb{R}_+^{m\times k_1}$, $W^2 \in \mathbb{R}_+^{m\times k_2})$, then F^1 is initialized as W^1 and $F^2{}_r$ is assigned as W^2. Finally, $F^3{}_r$ is randomly initialized, and $F^3{}_{r[i,j]} = \frac{F^3{}_{r[i,j]}}{\sum_{i=1}^{m} F^3{}_{r[i,j]}}$. After the computation of Algorithm 1, we can conduct the classification of target domain data according to G_r $(s + 1 \le r \le s + t)$. The convergence analysis of Algorithm 1 can be referred in the Appendix.

Algorithm 1 Triplex Transfer Learning (TriTL) Algorithm

Input: The source domains $\mathcal{D}_r = \{x_i^{(r)}, y_i^{(r)}\}|_{i=1}^{n_r}$ $(1 \le r \le s)$, target domains $\mathcal{D}_r = \{x_i^{(r)}\}|_{i=1}^{n_r}$ $(s + 1 \le r \le s + t)$, and the corresponding data matrices $X_1, \cdots, X_s, X_{s+1}, \cdots, X_{s+t}$. The data matrices are normalized such that $X_{r[i,j]} = \frac{X_{r[i,j]}}{\sum_{i=1}^{m} X_{r[i,j]}}$, G_r $(1 \le r \le s)$ are assigned as the true label information. The parameters k_1, k_2, k_3, and the number of iterations T.
Output: F^1, $F^2{}_r$, $F^3{}_r$, S^1, S^2, $S^3{}_r$ $(1 \le r \le s + t)$, and G_r $(s + 1 \le r \le s + t)$.

```
1) Initialization: The initializations of $F^{1(0)}$, $F^2{}_r{}^{(0)}$, $F^3{}_r{}^{(0)}$ are detailed in Section III-B; $S^{1(0)}$, $S^{2(0)}$, $S^3{}_r{}^{(0)}$ are randomly assigned, and $G_r{}^{(0)}$ $(s+1 \le r \le s+t)$ are initialized as the probabilistic output by supervised learning models, such as Logistic Regression (LR) [21] in the experiments.
2) $k := 1$.
3) Update $F^{1(k)}$ according to (19);
4) For $r := 1 \to s + t$
       Update $F^2{}_r{}^{(k)}$ according to (20) and $F^3{}_r{}^{(k)}$ according to (21);
5) end
6) Update $S^{1(k)}$ according to (22) and $S^{2(k)}$ according to (23);
7) For $r := 1 \to s + t$
       Update $S^3{}_r{}^{(k)}$ according to (24);
8) end
9) For $r := s + 1 \to s + t$
       Update $G_r{}^{(k)}$ according to (25);
10) end
11) Normalize $F^{1(k)}$, $F^2{}_r{}^{(k)}$, $F^3{}_r{}^{(k)}$, $G_r{}^{(k)}$ according to (26);
12) $k := k + 1$. If $k < T$, then turn to Step 3.
13) Output $F^{1(k)}$, $F^2{}_r{}^{(k)}$, $F^3{}_r{}^{(k)}$, $S^{1(k)}$, $S^{2(k)}$, $S^3{}_r{}^{(k)}$ and $G_r{}^{(k)}$.
```

C. Regularized TriTL

In this section, we further consider the clustering assumption that neighboring samples in geometric structure may share the similar class labels. In other word, the predicted labels of any two examples in target domains are required to be similar when they are neighbors. We hope the regularization of manifold structure in target domains would lead to the performance improvement.

Let M_r be the adjacence matrix of the documents in target domain $\mathcal{D}_r$ $(s + 1 \le r \le s + t)$

$$M_{r[i,j]} = \begin{cases} 1, & \text{if } x_j \text{ belongs to the } N \text{ nearest neighbors of } x_i, \\ 0, & \text{otherwise} \end{cases} \tag{27}$$

where x_i is the column vector. The similarity degree between x_i and x_j can be calculated by cosine measure

$$cos(x_i, x_j) = \frac{x_i{}^{\top} x_j}{\sqrt{x_i{}^{\top} x_i} \cdot \sqrt{x_j{}^{\top} x_j}}. \tag{28}$$

[1] http://www.kyb.tuebingen.mpg.de/bs/people/pgehler/code /index.html.

Let $D_r = \mathrm{diag}(\sum_j M_{r[i,j]})$ and $L_r = D_r - M_r$ be the Laplacian matrix, then incorporating the regularized manifold structure to (10), the optimization problem becomes

$$\begin{aligned}\mathcal{L} &= \sum_{r=1}^{s+t} ||X_r - F_r S_r G_r^\top||^2 + \frac{\gamma}{2}\sum_{r=s+1}^{s+t}\sum_{i,j} M_{r[i,j]}||G_{r[i,\cdot]} - G_{r[j,\cdot]}||^2,\\ &= \sum_{r=1}^{s+t} ||X_r - F_r S_r G_r^\top||^2 + \gamma \cdot \sum_{r=s+1}^{s+t} tr(G_r^\top L_r G_r),\\ s.t. &\sum_{i=1}^{m} F^1{}_{[i,j]} = 1, \sum_{i=1}^{m} F^2{}_{r[i,j]} = 1,\\ &\sum_{i=1}^{m} F^3{}_{r[i,j]} = 1, \sum_{j=1}^{c} G_{r[i,j]} = 1\end{aligned} \tag{29}$$

where $G_{r[i,\cdot]}$ denotes the ith row of G_r. Obviously, the regularization item only depends on the variable G_r, thus we can easily acquire the solution of optimization problem in (29) based on Algorithm 1. Specifically, the update formula of G_r in (25) is replaced as

$$G_{r[i,j]} \leftarrow G_{r[i,j]} \cdot \sqrt{\frac{[X_r^\top F_r S_r]_{[i,j]}}{[G_r S_r^\top F_r^\top F_r S_r + L_r G_r]_{[i,j]}}}. \tag{30}$$

For succinctness, this regularized triplex transfer learning algorithm is denoted as RTriTL.

D. Computational Complexity of the Iterative Algorithm

To show the efficiency of the proposed iterative algorithm in Algorithm 1, here, we analyze its computational complexity. Let $n = \sum_r n_r$ be the total number of documents from all source and target domains, $k = k_1 + k_2 + k_3$ be the total number of concepts, including the numbers of identical, alike and distinct concepts, for each round of iteration in Algorithm 1, the computational complexity of (19) to calculate F^1 is $O(7mnc + mkc(s+t) + 4mk_1c(s+t) + mk_1)$. Generally, $k \ll m$, $c \ll m$, $k \ll n$ and $c \ll n$, so the computational complexity of (19) can be rewritten as $O(mnc)$. Similarly, the computational complexities of (20), (21), (22), (23), (24), and (25) are, respectively, $O(mn_rc)$, $O(mn_rc)$, $O(mnc+mnk_1)$, $O(mnc + mnk_2)$, $O(mn_rc + mnk_3)$, and $O(mn_rc + mn_rk)$.

Given the number of iterations T, the maximal computational intensity is $O(Tmnc + Tmnk)$, which is linear to the number of words and documents. In fact, the matrixes are always very sparse, and we implement the iterative algorithm by sparse matrix computation in Matlab, thus the computational intensity can be dramatically reduced.[2] Note that the regularization item in RTriTL almost does not increase the computation complexity. Therefore, the theoretical analysis guarantees the efficiency of the proposed iterative algorithm.

IV. Experimental Evaluation

In this section, we systemically demonstrate the effectiveness of the proposed transfer learning framework TriTL and RTriTL. In the experiments, we only focus on binary text classification and there are only one source domain and one target domain, i.e., $s = 1$ and $t = 1$. Note that TriTL is a general model, which can handle multiclass classification problems and multiple source and target domains, i.e., $s > 1$ and $t > 1$.

A. Data Preparation

20Newsgroups[3] is one of the benchmark data sets for evaluating transfer learning algorithms, which is widely used in previous works [1], [3], [22], [23]. This corpus has approximately 20 000 newsgroup documents, which are evenly divided into 20 subcategories. Some similar subcategories are grouped into a top category, e.g., the four subcategories sci.crypt, sci.electronics, sci.med, and sci.space belong to the top category sci. The four top categories and their subcategories are depicted in Table IV.

Firstly, we construct the transfer learning tasks using the approach in [9]. For example, for the dataset rec versus sci, we randomly select a subcategory from rec as positive class and a subcategory from sci as negative class to produce the source domain. The target domain is similarly constructed, thus in totally 144 ($P_4^2 \cdot P_4^2$) classification tasks are generated for dataset rec versus sci. However, in this traditional setting, the source and target domains are both drawn from the same top categories, thus they may tend to share all the concepts.

Secondly, to validate our model TriTL can effectively exploit the distinct concepts, we further construct another type of classification tasks. For example, for the classification task generated from rec versus sci from the above approach, we replace one subcategory from the target domain as another subcategory from the top category comp or talk. In this new type of classification tasks, the source and target domains are not drawn from the same top categories, thus they would have their own distinct concepts.

In this way, we can construct additional 384 ($144 \div 3 \times 8$) classification tasks. Among all these 384 tasks, we first run the supervised learning model logistic regression (LR) [21] on each of them, and then select the 334 ones whose accuracies from LR are higher than 50%.[4]

In summary, we have 144 traditional transfer learning tasks and 334 new transfer learning problems generated from the 20Newsgroups dataset.

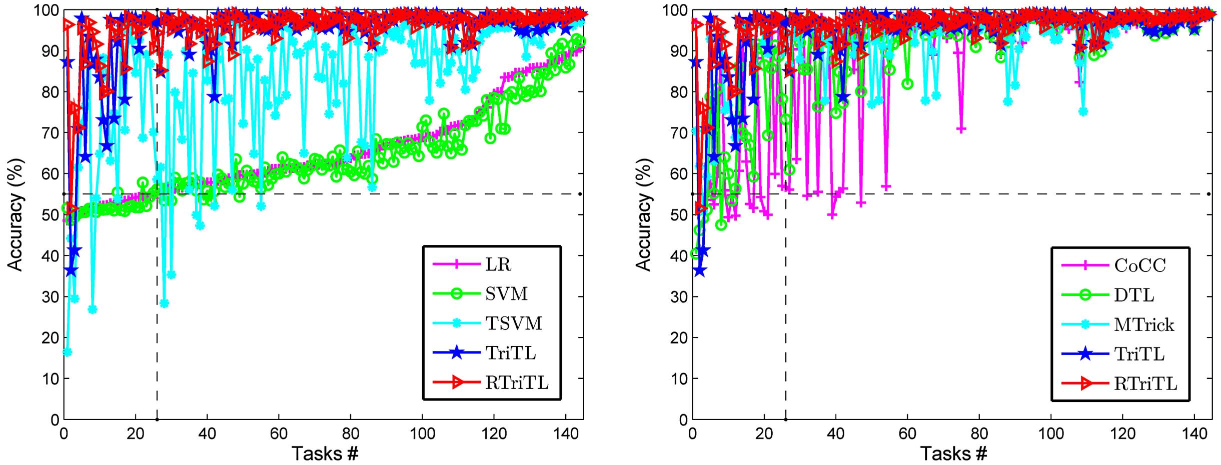
The dataset Reuters-21578,[5] which has three top categories orgs, people, and place (Each top category also has several subcategories), is also adopted to validate our algorithm in the experiments. We directly use the three classification tasks constructed by Gao *et al.* [3].

[2] Experimental results show that the iterative algorithm can finish our task in about 15 seconds under the default setting of parameters. There are about 8,000 words and 4000 documents in each task. The configuration of computing platform: Intel Core i7-3770 CPU 3.4GHz, RAM 4.0GB.

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

[4] In this paper, we don't consider the classification tasks whose accuracies from LR are lower than 50%, since they are much more challenging and vulnerable to negative transfer.

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



(a)

(b)

Fig. 1. Performance comparison among LR, SVM, TSVM, CoCC, DTL, MTrick and TriTL, RTriTL on dataset rec versus sci. (a) RTriTL, TriTL versus LR, SVM, TSVM. (b) RTriTL, TriTL versus CoCC, DTL, MTrick.

TABLE IV

TOP CATEGORIES AND THEIR SUBCATEGORIES

Top Categories	Subcategories
comp	*comp.*{*graphics, sys.mac.hardware*} *comp.sys.ibm.pc.hardware* *comp.os.ms-windows.misc*

rec	*rec.*{*autos, motorcycles*}
	rec.sport.{*baseball, hockey*}

sci	*sci.*{*crypt, med, electronics, space*}
talk	*talk.politics.*{*guns, mideast, misc*} *talk.religion.misc*

B. Experimental Setting

Compared algorithms: We compare our models TriTL, RTriTL with some state-of-the-art baselines, including:

1) the supervised algorithms: LR [21], support vector machine (SVM) [24];
2) the semisupervised algorithm: Transductive support vector machine (TSVM) [25];
3) the cross-domain methods: Coclustering-based classification (CoCC) [5], MTrick [9], and dual transfer learning [12].

Parameter setting: In TriTL, we set $k_1 = 20$, $k_2 = 20$, $k_3 = 10$ and $T = 100$. In RTriTL, the parameters k_1, k_2 and k_3 are set the same as the ones in TriTL, the number of nearest neighbors $N = 50$, the tradeoff parameter $\gamma = 1000$, and $T = 200$ to achieve better convergence quality. The baseline methods LR is implemented by Matlab,[6] SVM and TSVM are given by SVM ^{light}.[7] The parameters of CoCC, MTrick, and DTL are set as the default ones in their original papers, except that for DTL, we normalize the data matrix the same as this paper, i.e., $X_{[i,j]} = \frac{X_{[i,j]}}{\sum_{i=1}^{m} X_{[i,j]}}$, rather than $X_{[i,j]} = \frac{X_{[i,j]}}{\sum_{i=1}^{m}\sum_{j=1}^{n} X_{[i,j]}}$ in their paper. Preliminary tests show that, this slight change of normalization results in significant improvement of DTL.

We use the classification accuracy as the evaluation metric

$$accuracy = \frac{|\{d|d \in \mathcal{D} \wedge f(d) = y\}|}{n} \tag{31}$$

where y is the true label of document d, n is number of documents, and the function $f(d)$ gives d a prediction label.

C. Experimental Results

1) *Comparison on the Traditional Transfer Learning Tasks:* We compare TriTL, RTriTL with LR, SVM, TSVM, CoCC, DTL, and MTrick on the dataset rec versus sci, and all the results of the 144 classification tasks are recorded in Fig. 1 and Table V. In Fig. 1, the 144 tasks are sorted by the increasing order of the performance of LR. The lower accuracy of LR indicates that it is more difficult to transfer the knowledge from source domain to target domain. Also, these classification tasks are separated into two parts, the left side of black dotted line in Fig. 1 represents the problems with accuracy of LR lower than 55%, while the right higher than 55%. Table V lists the corresponding average performance.

From these results, we have the following findings.

a) TriTL is significantly better than the supervised learning algorithms LR and SVM, and the semisupervised method TSVM. This show that the traditional learning algorithms may fail in the transfer learning tasks.
b) TriTL significantly outperforms all the compared transfer learning algorithms CoCC, MTrick, and DTL with the statistical test. In Table V, TriTL achieves the best results in term of the average performances, no matter the classification tasks with accuracy of LR lower or higher than 55%. This improvement might be due to the synthesized effectiveness in modeling all the three concepts. When the accuracy of LR is lower than 55%, the degree of distribution difference between source and target domains might be large. Thus, modeling the distinct concepts in TriTL may improve the performance. On the other hand, when the accuracy of LR is higher than 55%, the data distributions of the source and target

[6] http://research.microsoft.com/∼minka/papers/logreg/.

[7] http://svmlight.joachims.org/.

TABLE V

AVERAGE PERFORMANCES (%) ON 144 TRADITIONAL TRANSFER LEARNING TASKS AND 334 NEW TRANSFER LEARNING ONES

		LR	SVM	TSVM	CoCC	DTL	MTrick	TriTL	RTriTL
Traditional Learning Tasks	*Lower than 55%*	52.45	51.82	69.88	64.09	68.62	83.00	**84.05**	**88.45**
	Higher than 55%	68.33	66.80	85.52	91.84	94.18	94.74	**96.61**	**96.87**
	Total	65.57	64.20	82.81	87.02	89.75	92.70	**94.43**	**95.41**
New Learning Tasks	*Lower than 55%*	52.45	51.81	74.32	69.66	75.34	78.45	**80.93**	**82.24**
	Higher than 55%	70.03	68.62	86.68	91.76	94.44	93.97	**95.24**	**97.40**
	Total	66.61	65.35	84.27	87.46	90.72	90.95	**92.45**	**94.45**

TABLE VI

PERFORMANCES (%) ON THREE TRANSFER LEARNING TASKS FROM *Reuters*-21578 DATASET

Data Sets	LR	SVM	TSVM	CoCC	DTL	MTrick	TriTL	RTriTL
orgs vs. people	74.92	74.25	73.80	79.79	79.69	80.80	**81.24**	**81.88**
orgs vs. place	71.91	69.99	69.89	74.18	76.51	76.77	**78.04**	**78.95**
people vs. place	58.03	59.05	58.43	66.94	68.52	**69.02**	68.35	**69.68**
Average	68.29	67.76	67.37	73.64	74.91	75.53	**75.88**	**76.84**

domains might be similar. Thus, modeling the identical and alike concepts may work. Therefore, our model TriTL is much flexible under different situations.

c) When the accuracy of LR is lower than 55%, MTrick is better than DTL, and DTL is better than CoCC, which coincide with our expectation. In these difficult tasks, the degree of distribution difference between source and target domains might be large. There might not be any identical concepts shared in the source and target domains. Thus, modeling the identical concepts in DTL and CoCC might deteriorate the performance.

d) When the accuracy of LR is higher than 55%, the compared transfer learning algorithms perform similarly, and they all outperform the traditional learning methods. This time the transfer learning algorithms all consider the identical or alike concepts, since the degree of distribution difference between source and target domains might be small.

e) RTriTL gains the performance improvement against TriTL in terms of the average accuracy, which indicates the positive effect of the regularized manifold structure in the target domains.

2) *Comparison on the New Type of Transfer Learning Tasks:* To further validate the effectiveness of TriTL and RTriTL, we construct the other 334 transfer learning tasks in which the distinct concepts may exist. The average accuracy values of these 334 tasks using all the methods are given in Table V. In this table, we also divide these tasks into two groups, whose accuracies from LR are lower or higher than 55%. From this table, it can be found that TriTL, RTriTL once more obtain the best results. MTrick is better than DTL, and DTL outperforms CoCC when the average accuracy of LR is lower than 55%, which are consistent to the analysis in Section IV-C1.

3) *Comparison on Reuters-21578:* The Reuters-21578 dataset is also used to verify the superiority of TriTL and RTriTL, and we directly adopt the constructed transfer learning tasks from Gao *et al.* [3]. All the results of each task and their average performances are listed in Table VI. The results show that TriTL and RTriTL can also perform very well on these three tasks, and they outperform all the baselines in terms of average accuracy.

D. Parameter Sensitivity

Here, we investigate the parameter sensitivity of our model TriTL. There are three parameters in TriTL, including the number of identical concepts k_1, the number of alike concepts k_2, and the number of distinct concepts k_3. To verify that TriTL is not sensitive to the parameter setting, we relax the sampling ranges of these three parameters. Specifically, after some preliminary test, we bound the parameters $k_1 \in [15, 25]$, $k_2 \in [15, 25]$, and $k_3 \in [5, 15]$, and evaluate them on ten randomly selected tasks from the 144 classification problems of rec versus sci. We randomly sample ten combinations of parameters, and all the results are shown in Table VII. The 12th and 13th rows, respectively, represent the average accuracy and variance of each tasks under the ten combinations of parameters. The last row is the result using the default parameters adopted in this paper.

It is obvious that in Table VII, the mean performance of the ten combinations of parameters for each task is almost the same as the one using the default parameters, and the variance is very small. These results show that TriTL is not sensitive to the parameter setting when they are sampled from some predefined bounds.

In RTriTL, we adopt the same parameters of k_1, k_2, and k_3 as in TriTL, i.e., $k_1 = 20$, $k_2 = 20$, and $k_3 = 10$. In the following, we study the parameter sensitivity of the number of nearest neighbors N and the tradeoff factor γ to RTriTL. We randomly select 24 tasks from the dataset rec versus sci and 65 ones (whose accuracies from LR are lower than 55%) from the new transfer learning problems to, respectively, study the parameter affection on N and γ. The average results over all selected problems under different parameter settings are reported in Fig. 2. According to these results, we set $N = 50$ and $\gamma = 1000$ as the default values in this paper to get stable and outstanding performance.

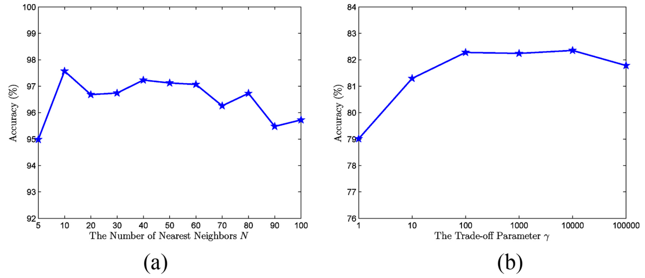
E. Algorithm Convergence

In this section, we also empirically check the convergence of the iterative algorithm to TriTL. We randomly choose six

TABLE VII

PARAMETER EFFECT FOR PERFORMANCE (%) OF ALGORITHM TRITL

Sampling ID	k_1	k_2	k_3	Problem ID									
				1	2	3	4	5	6	7	8	9	10
1	20	22	15	97.11	97.37	98.01	98.72	98.86	96.81	97.76	98.91	91.54	94.91
2	16	22	15	97.24	97.37	97.96	98.65	98.86	97.01	97.64	98.93	91.59	95.03
3	19	24	10	97.23	97.22	97.91	98.79	98.86	96.89	97.59	98.91	91.76	94.84
4	19	21	8	97.06	97.14	97.98	98.77	98.88	97.01	97.71	98.89	90.94	95.11
5	18	17	10	97.35	97.22	97.71	98.69	98.86	96.99	97.66	98.89	91.93	94.94
6	15	22	9	96.94	97.39	97.59	98.74	98.91	97.03	97.66	98.89	91.88	94.93
7	18	25	14	97.24	97.53	97.60	98.62	98.91	96.79	97.66	98.93	91.71	94.73
8	24	24	10	96.96	97.41	97.82	98.64	98.86	97.08	97.71	98.94	90.92	94.98
9	19	17	9	97.13	97.12	97.84	98.71	98.86	96.99	97.64	98.89	92.10	94.86
10	24	20	9	97.06	96.99	97.87	98.76	98.86	96.96	97.69	98.89	91.41	95.06
Mean				97.13	97.28	97.83	98.71	98.87	96.96	97.67	98.91	91.58	94.94
Variance				0.017	0.027	0.023	0.003	0.000	0.009	0.002	0.000	0.158	0.013
This paper	20	20	10	97.21	97.43	97.82	98.71	98.88	97.02	97.67	98.91	91.49	94.90



(a) (b)

Fig. 2. Parameter affection of N and γ to RTriTL ($k_1 = 20$, $k_2 = 20$, $k_3 = 10$). (a) N ($\gamma = 1000$). (b) γ ($N = 50$).

tasks from the dataset rec versus sci, and the results are shown in Fig. 3. In these figures, the x-axis denotes the number of iterations, and the left and right y-axis denotes the prediction accuracy and the objective value in (10), respectively. Both prediction accuracy and objective value can converge within 100 iterations, and the value of objective function in (10) decreases along with the iterating process, which coincides with the theoretic analysis.

F. Visualization of Word Clusters

To show the effectiveness of TriTL, in this section, we also empirically demonstrate TriTL can simultaneously capture the distinct, alike, and identical concepts. In details, τ keywords (e.g., $\tau = 20$ is adopted in the experiments.) are selected to express each topic according to the word clustering information F. Table VIII lists some word clusters from the classification task, i.e., source domain: rec.autos versus sci.space, and target domain: rec.sport.hockey versus talk.politics. mideast. From these results, it can be indicated that TriTL can capture the identical and alike concepts applying the same/different keywords about the same topic recreation among source and target domains (From the first row to sixth row in Table VIII). Furthermore, TriTL can effectively capture the distinct concepts using different keywords (From the seventh row to tenth row in Table VIII), e.g., the keywords planet, mars, jpl, nasa describe the topic science about space belonging to the source domain, while the keywords israelis, istanbul, gaza describe the topic politics about mideast belonging to the target domain.

V. RELATED WORKS

In this section, we summarize the related works of transfer learning, which has aroused large amounts of interest and research in recent years. Here, we group the previous works of transfer learning into three categories, i.e., feature-based, instant weighing-based, and model combination-based transfer learning.

Feature-based methods can further be divided into two categories, i.e., feature selection and feature mapping. Feature selection-based methods are to identify the common features (at the level of raw words) between source and target domains, which are useful for transfer learning [5], [26], [27]. Jiang *et al.* [26] argued that the features highly related to class labels should be assigned to large weights in the learnt model, thus they developed a two-step feature selection framework for domain adaptation. They first selected the general features to build a general classifier, and then considered the unlabeled target domain to select specific features for training target classifier. Uguroglu *et al.* [27] presented a novel method to identify variant and invariant features between two datasets for transfer learning. Feature space mapping-based methods are to map the original high-dimensional features into a low-dimensional feature space, under which the source and target domains comply with the same data distribution [13], [28]–[32]. Pan *et al.* [28] proposed a dimensionality reduction approach to find out this latent feature space, in which supervised learning algorithms can be applied to train classification models. Si *et al.* [31] presented the cross-domain discriminative Hessian Eigenmaps to find a subspace, in which the distributions of training and test data are similar; also both the local geometry and the discriminative information can be well passed from the training domain to test domain. Gu *et al.* [29] learnt the shared subspace among multiple domains for clustering and transductive transfer classification. In their problem formulation, all the domains have the same cluster centroid in the shared subspace. The label information can also be injected for classification tasks in this method. Tian *et al.* [13] proposed a sparse transfer learning algorithm, in which both the user's search intention and sample distribution knowledge are considered, for interactive video search reranking.

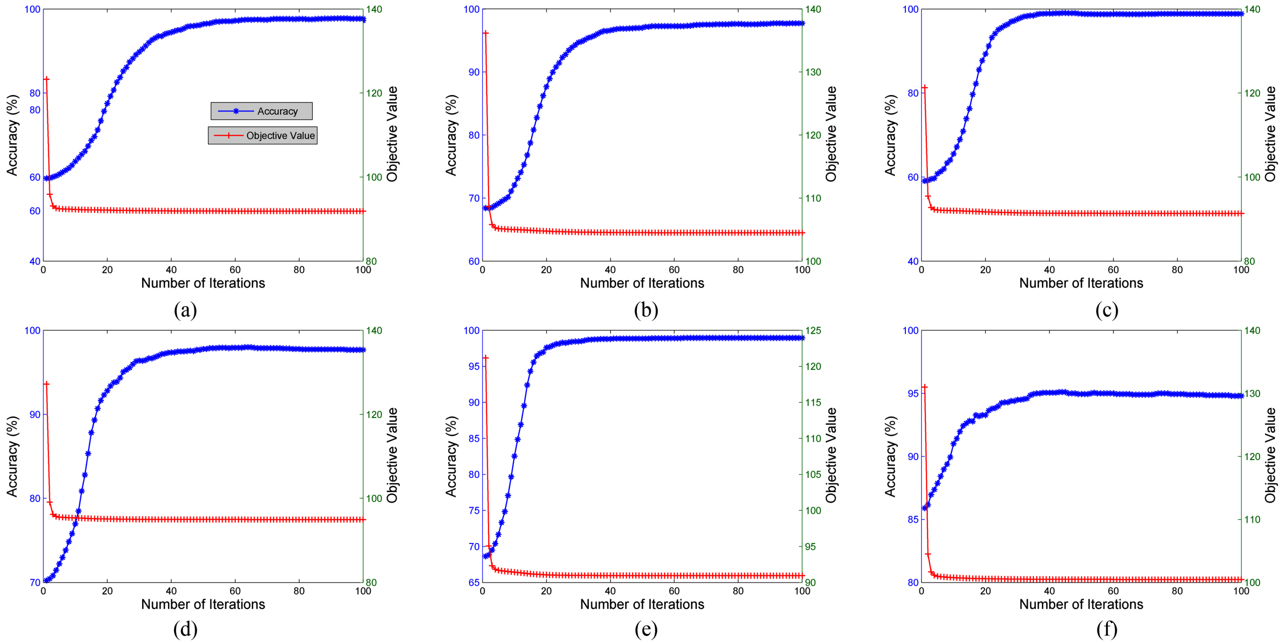


Fig. 3. Number of iterations versus the Performance of TriTL and Objective Value. (a) Problem 1. (b) Problem 2. (c) Problem 3. (d) Problem 4. (e) Problem 5. (f) Problem 6.

TABLE VIII

IDENTICAL, ALIKE, AND DISTINCT CONCEPTS CAPTURED BY TRITL (SOURCE DOMAIN (r = 1): *rec.autos* VERSUS *sci.space*, TARGET DOMAIN (r = 2): *rec.sport.hockey* VERSUS *talk.politics.mideast*)

Identical Concepts	F^1	Topic 1	will,there,european,turn,snow,hot,all,att,particularly, **cars**,de,any,canada,**oil**,red,**car**,ericsson,are,se,change
		Topic 2	**games**,news,buffalo,next,mailing,san,**hockey**,bay,pick,current, june,lu,year,city,st,list,radio,will,sweden,contact
Alike Concepts	Topic 1	${F^2}_1$	**cars**,drew,brakes,centerline,tek,brake,**car**,**speed**,uokmax,com,bird, ford,**clutch**,virginia,convertible,wv,sho,uoknor,taurus,callison
		${F^2}_2$	show,coverage,andrew,msu,eos,baltimore,**play**,ca,tom,pat,**ice**,**game**, caps,francis,**baseball**,overtime,night,stats,jagr,**espn**
	Topic 2	${F^2}_1$	police,rocks,chintan,amin,**road**,**vw**,**gas**,purdue,**gt**,cactus,lehigh, **driving**,**accident**,**mph**,wagon,**auto**,uiuc,insurance,**car**,**cars**
		${F^2}_2$	sweden,**sport**,emotional,ca,blues,friedman,skins,next,prism,kevin, jersey,mask,gatech,gtd,**goalie**,hrivnak,capitals,**fan**,mike,go
Distinct Concepts	${F^3}_1$	Topic 1	**planet**,observations,teflon,tommy,cacs,srl,baalke,**mars**,gov,higgins, **jpl**,**nasa**,**temperature**,**planets**,kelvin,dseg,ti,smiley,**jupiter**,**hst**
		Topic 2	glen,oz,kelvin,**planetary**,mercury,**saturn**,**nasa**,radiation,ti,phil, mccall,gov,fraering,sun,**jpl**,**mars**,ron,**jupiter**,fnal,baalke
	${F^3}_2$	Topic 1	**israelis**,ncsu,mcrcim,igc,sexual,shostack,brad,marc,quote, davidsson,**istanbul**,dog,cute,idf,favors,das,bu,**gaza**,pro,cpr
		Topic 2	hernlem,hasan,**isreal**,**civilians**,**istanbul**,**hamas**,mcgill,**lebanese**,elias, diesel, wagon, nissan, mileage, byte, saturn, toyota, si, cars, car, db

Si *et al.* [32] developed a transfer subspace learning framework, which can be applicable to various dimensionality reduction algorithms and minimize the Bregman divergence between the distribution of training data and testing data in the selected subspace. Gupta *et al.* [33] proposed a nonnegative shared subspace learning for social media retrieval. However, their algorithm does not consider the alike concepts and can not be directly used for transfer classification.

Instance weighting-based approaches reweight the instances in source domains according to the similarity measure on how they are close to the data in the target domain. Specifically, the weight of an instance is increased if it is close to the data in the target domain, otherwise the weight is decreased [23], [34], [35]. Dai *et al.* [23] extended boosting-style learning algorithm to cross-domain learning, in which the training instances with different distribution from the target domain are less weighted for data sampling, while the training instances with the similar distribution to the target domain are more weighted. Jiang *et al.* [34] proposed a general instance weighting framework, which has been validated to work well on NLP tasks. Wan *et al.* [35] first aligned the feature spaces in both domains utilizing some online translation service, and then proposed an iterative feature and instance weighting (Bi-Weighting) method for cross-language text classification.

Model combination-based methods aim at giving different weights to the classification models in an ensemble [3], [36]. Gao *et al.* [3] proposed a dynamic model weighting method for each test example according to the similarity between the model and the local structure of the test example in the target domain. Dredze [36] developed a new multidomain online learning framework based on parameter combination from multiple classifiers for a new target domain.

However, there has not yet transfer learning algorithm systemically analyzes the commonalities and speciality between source and target domains, and models them together. This paper belongs to the feature-based methods, and simultaneously models the three commonalities and specific characteristic between source and target domains. Moreover, we design a new type of experiments to validate the effectiveness of our model.

VI. Conclusion

In this paper, we systemically study the problem of transfer learning when there are three types of concepts, namely, identical, alike, and distinct concepts, among the source and target domains. By considering them altogether, we propose a general model TriTL based on nonnegative matrix trifactorization. Then, an alternately iterative algorithm is developed to solve the proposed optimization problem. Moreover, we propose to regularize the manifold structure in target domains to further improve the prediction performances. Finally, we construct two types of transfer learning tasks. The experimental results show that the proposed algorithms TriTL and RTriTl could significantly outperform the compared methods under various situations of the source and target domains.

Appendix

To study the convergence of update rules in (19)–(26), we first check the convergence of F^1 when the rest parameters are fixed. According to (10), we formulate the optimization problem with constraints as the following Lagrangian function

$$\begin{aligned}\mathcal{G}(F^1) =& \sum_{r=1}^{s+t} ||X_r - F_r S_r G_r^\top||^2 \\ &+ tr[\boldsymbol{\lambda}({F^1}^\top \mathbf{1}_m - \mathbf{1}_{k_1})({F^1}^\top \mathbf{1}_m - \mathbf{1}_{k_1})^\top]\end{aligned} \tag{32}$$

where $\boldsymbol{\lambda} \in \mathbb{R}^{k_1 \times k_1}$ is a diagonal matrix. Omitting the items, which are independent of F^1, (32) becomes

$$\begin{aligned}\mathcal{G}(F^1) =& \sum_{r=1}^{s+t} tr(-2 \cdot X_r^\top F^1 S^1 G_r^\top + G_r {S^1}^\top {F^1}^\top A_r \\ &+ 2 \cdot G_r {S^1}^\top {F^1}^\top B_r + 2 \cdot G_r {S^1}^\top {F^1}^\top C_r) \\ &+ tr[\boldsymbol{\lambda}({F^1}^\top \mathbf{1}_m \mathbf{1}_m^\top F^1 - 2 \cdot \mathbf{1}_{k_1} \mathbf{1}_m^\top F^1)].\end{aligned} \tag{33}$$

Then, the differential is

$$\begin{aligned}\frac{\partial \mathcal{G}}{\partial F^1} =& \sum_{r=1}^{s+t} (-2 \cdot X_r G_r {S^1}^\top + 2 \cdot A_r G_r {S^1}^\top + 2 \cdot B_r G_r {S^1}^\top \\ &+ 2 \cdot C_r G_r {S^1}^\top) + 2 \cdot \mathbf{1}_m ({\mathbf{1}_m}^\top F^1 - {\mathbf{1}_{k_1}}^\top) \boldsymbol{\lambda}.\end{aligned} \tag{34}$$

Lemma 1: Using the update rule (35), (33) will monotonously decrease

$${F^1}_{[i,j]} \leftarrow {F^1}_{[i,j]} \cdot \sqrt{\frac{[\sum_{r=1}^{s+t} X_r G_r {S^1}^\top + \mathbf{1}_m {\mathbf{1}_{k_1}}^\top \boldsymbol{\lambda}]_{[i,j]}}{[\sum_{r=1}^{s+t} D_r + \mathbf{1}_m {\mathbf{1}_m}^\top F^1 \boldsymbol{\lambda}]_{[i,j]}}} \tag{35}$$

where $D_r = A_r G_r {S^1}^\top + B_r G_r {S^1}^\top + C_r G_r {S^1}^\top$.

Proof: To prove Lemma 1 we describe the definition of auxiliary function [37] as follows.

Definition 6 (Auxiliary function): A function $\mathcal{Q}(Y, \widetilde{Y})$ is called an auxiliary function of $\mathcal{T}(Y)$ if it satisfies

$$\mathcal{Q}(Y, \widetilde{Y}) \geq \mathcal{T}(Y),\, \mathcal{Q}(Y, Y) = \mathcal{T}(Y) \tag{36}$$

for any Y, $\widetilde{Y}$.
Then, define

$$Y^{(t+1)} = arg\, \min_Y \mathcal{Q}(Y, Y^{(t)}). \tag{37}$$

Through this definition

$$\mathcal{T}(Y^{(t)}) = \mathcal{Q}(Y^{(t)}, Y^{(t)}) \geq \mathcal{Q}(Y^{(t+1)}, Y^{(t)}) \geq \mathcal{T}(Y^{(t+1)}). \tag{38}$$

It means that the minimizing of the auxiliary function of $\mathcal{Q}(Y, Y^{(t)})$ ($Y^{(t)}$ is fixed) has the effect to decrease the function of $\mathcal{T}$.

Now, we can construct the auxiliary function of $\mathcal{G}$ as

$$\begin{aligned}&\mathcal{Q}(F^1, {F^1}') = \\ &\quad \sum_{i=1}^{m} \sum_{j=1}^{k_1} \{-2 \cdot (\sum_{r=1}^{s+t} X_r G_r {S^1}^\top)_{[i,j]} {F^1}'_{[i,j]} (1 + \log \frac{{F^1}_{[i,j]}}{{F^1}'_{[i,j]}}) \\ &\quad -2 \cdot (\mathbf{1}_m {\mathbf{1}_{k_1}}^\top \boldsymbol{\lambda})_{[i,j]} {F^1}'_{[i,j]} (1 + \log \frac{{F^1}_{[i,j]}}{{F^1}'_{[i,j]}}) \\ &\quad +(\sum_{r=1}^{s+t} {A_r}' G_r {S^1}^\top + \mathbf{1}_m {\mathbf{1}_m}^\top {F^1}' \boldsymbol{\lambda})_{[i,j]} \frac{{F^1}_{[i,j]} {F^1}_{[i,j]}}{{F^1}'_{[i,j]}} \\ &\quad +[\sum_{r=1}^{s+t} (B_r G_r {S^1}^\top + C_r G_r {S^1}^\top)]_{[i,j]} ({F^1}'_{[i,j]} + \frac{{F^1}_{[i,j]} {F^1}_{[i,j]}}{{F^1}'_{[i,j]}})\}\end{aligned}$$

where ${A_r}' = {F^1}' S^1 G_r^\top$. Obviously, when $F^1 = {F^1}'$ the equality $\mathcal{Q}(F^1, {F^1}') = \mathcal{G}(F^1)$ holds. Also, we can prove the inequality $\mathcal{Q}(F^1, {F^1}') \geq \mathcal{G}(F^1)$ holds using the similar proof approach in [38]. Then, while fixing ${F^1}'$, we minimize $\mathcal{Q}(F^1, {F^1}')$. The differential of $\mathcal{Q}(F^1, {F^1}')$ is

$$\begin{aligned}&\frac{\partial \mathcal{Q}(F^1, {F^1}')}{\partial {F^1}_{[i,j]}} = \\ &\qquad -2 \cdot (\sum_{r=1}^{s+t} X_r G_r {S^1}^\top)_{[i,j]} \frac{{F^1}'_{[i,j]}}{{F^1}_{[i,j]}} \\ &\qquad -2 \cdot (\mathbf{1}_m {\mathbf{1}_{k_1}}^\top \boldsymbol{\lambda})_{[i,j]} \frac{{F^1}'_{[i,j]}}{{F^1}_{[i,j]}}\end{aligned}$$

$$+2 \cdot (\sum_{r=1}^{s+t} A_r{}' G_r S^{1\top} + \mathbf{1}_m \mathbf{1}_m{}^\top F^{1'} \boldsymbol{\lambda})_{[i,j]} \frac{F^1{}_{[i,j]}}{F^{1'}{}_{[i,j]}}$$

$$+2 \cdot [\sum_{r=1}^{s+t} (B_r G_r S^{1\top} + C_r G_r S^{1\top})]_{[i,j]} \frac{F^1{}_{[i,j]}}{F^{1'}{}_{[i,j]}}.$$

Let $\frac{\partial \mathcal{Q}(F^1, F^{1'})}{\partial F^1{}_{[i,j]}} = 0$, we can obtain (35). Thus, the update rule (35) decreases the values of $\mathcal{G}(F^1)$. Then, Lemma 1 holds. ■

The only obstacle left is the calculation of the Lagrangian multipliers $\boldsymbol{\lambda}$. Actually, the role of λ in this problem is to drive the solution to satisfy the constrained condition that the sum of the values in each column of F^1 is one. Here, we adopt the normalization technology in [39] and [9] to satisfy the constrains regardless of $\boldsymbol{\lambda}$. Specifically, in each iteration, we use (26) to normalize F^1. After normalization, $\mathbf{1}_m \mathbf{1}_{k_1}{}^\top \boldsymbol{\lambda}$ is equal to $\mathbf{1}_m \mathbf{1}_m{}^\top F^1 \boldsymbol{\lambda}$ which are both constants; therefore, the effect of (19) and (26) can be approximately equivalent to (35) when only considering the convergence. In our solution, we adopt the approximate update rule of (19) by omitting the items that depends on $\boldsymbol{\lambda}$ in (35). We can use the similar method to analyze the convergence of the update rules for $F^2{}_r$, $F^3{}_r$, S^1, S^2, $S^3{}_r$ $(1 \leq r \leq s+t)$, and G_r $(s+1 \leq r \leq s+t)$ in (20), (21), (22), (23), (24), (25), and (26), respectively.

Theorem 1 (Convergence): After each round of iteration in Algorithm 1, the objective function in (10) will not increase. According to the lemmas for the convergence analysis on the update rules for F^1, $F^2{}_r$, $F^3{}_r$, S^1, S^2, $S^3{}_r$ $(1 \leq r \leq s+t)$, G_r $(s+1 \leq r \leq s+t)$, and the multiplicative update rules [37], each update step in Algorithm 1 will not increase (10) and the objective has a lower bounded by zero, which guarantee the convergence. Thus, the above theorem holds.

References

[1] 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.

[2] W. Y. Dai, Y. Q. Chen, G. R. Xue, Q. Yang, and Y. Yu, "Translated learning: Transfer learning across different feature spaces," in *Proc. 22nd NIPS*, 2008, pp. 353–360.

[3] J. Gao, W. Fan, J. Jiang, and J. W. Han, "Knowledge transfer via multiple model local structure mapping," in *Proc. 14th ACM SIGKDD*, 2008, pp. 283–291.

[4] J. Gao, W. Fan, Y. Z. Sun, and J. W. Han, "Heterogeneous source consensus learning via decision propagation and negotiation," in *Proc. 15th ACM SIGKDD*, 2009, pp. 339–348.

[5] W. Y. Dai, G. R. Xue, Q. Yang, and Y. Yu, "Co-clustering based classification for out-of-domain documents," in *Proc. 13th ACM SIGKDD*, 2007, pp. 210–219.

[6] P. Luo, F. Z. Zhuang, H. Xiong, Y. H. Xiong, and Q. He, "Transfer learning from multiple source domains via consensus regularization," in *Proc. 17th ACM CIKM*, 2008, pp. 103–112.

[7] G. R. Xue, W. Y. Dai, Q. Yang, and Y. Yu, "Topic-bridged PLSA for cross-domain text classification," in *Proc. 31st ACM SIGIR*, 2008, pp. 627–634.

[8] W. Y. Dai, O. Jin, G. R. Xue, Q. Yang, and Y. Yu, "Eigen transfer: A unified framework for transfer learning," in *Proc. 26th ICML*, 2009, pp. 193–200.

[9] F. Z. Zhuang, P. Luo, H. Xiong, Q. He, Y. H. Xiong, and Z. Z. Shi, "Exploiting associations between word clusters and document classes for cross-domain text categorization," in *Proc. 10th SIAM SDM*, 2010, pp. 13–24.

[10] 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 *Proc. 19th ACM CIKM*, 2010, pp. 359–368.

[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 ACM SIGIR*, 2011, pp. 933–942.

[12] M. Long, J. Wang, G. Ding, W. Cheng, X. Zhang, and W. Wang, "Dual transfer learning," in *Proc. 12th SIAM SDM*, 2012, pp. 540–551.

[13] X. Tian, D. C. Tao, and Y. Rui, "Sparse transfer learning for interactive video search reranking," *ACM Trans. Multimedia Comput., Commun., Applicat.*, vol. 8, no. 3, pp. 26:1–26:19, 2012.

[14] B. Geng, D. Tao, and C. Xu, "DAML: Domain adaptation metric learning," *Trans. Img. Proc.*, vol. 20, no. 10, pp. 2980–2989, 2011.

[15] F. Zhuang, P. Luo, C. Du, Q. He, and Z. Shi, "Triplex transfer learning: Exploiting both shared and distinct concepts for text classification," in *Proc. 6th ACM WSDM*, 2013, pp. 425–434.

[16] C. Ding, T. Li, W. Peng, and H. Park, "Orthogonal nonnegative matrix t-factorizations for clustering," in *Proc. 12th ACM SIGKDD*, 2006, pp. 126–135.

[17] T. Li, V. Sindhwani, C. Ding, and Y. Zhang, "Knowledge transformation for cross-domain sentiment classification," in *Proc. 32nd ACM SIGIR*, 2009, pp. 716–717.

[18] F. Wang, T. Li, and C. Zhang, "Semi-supervised clustering via matrix factorization," in *Proc. 8th SIAM SDM*, 2008, pp. 1041–1048.

[19] T. Li, C. Ding, Y. Zhang, and B. Shao, "Knowledge transformation from word space to document space," in *Proc. 31st ACM SIGIR*, 2008, pp. 187–194.

[20] T. Hofmann (2001). "Unsupervised learning by probabilistic latent semantic analysis," *J. Mach. Learn.*, vol. 42, no. 1–2, pp. 177–196.

[21] D. Hosmer and S. Lemeshow, *Applied Logistic Regression*. New York, NY, USA: Wiley, 2000.

[22] D. Zhang, J. He, Y. Liu, L. Si, and R. Lawrence, "Multiview transfer learning with a large margin approach," in *Proc. 17th ACM SIGKDD*, 2011, pp. 1208–1216.

[23] W. Y. Dai, Q. Yang, G. R. Xue, and Y. Yu, "Boosting for transfer learning," in *Proc. 24th ICML*, 2007, pp. 193–200.

[24] B. E. Boser, I. Guyou, and V. Vapnik, "A training algorithm for optimal margin classifiers," in *Proc. 5th AWCLT*, 1992, pp. 144–152.

[25] T. Joachims, "Transductive inference for text classification using support vector machines," in *Proc. 16th ICML*, 1999, pp. 200–209.

[26] J. Jiang and C. X. Zhai, "A two-stage approach to domain adaptation for statistical classifiers," in *Proc. 16th ACM CIKM*, 2007, pp. 401–410.

[27] S. Uguroglu and J. Carbonell, "Feature selection for transfer learning," in *Proc. Mach. Learn. Knowl. Discovery Databases*, 2011, pp. 430–442.

[28] S. J. Pan, J. T. Kwok, and Q. Yang, "Transfer learning via dimensionality reduction," in *Proc. 23rd AAAI*, 2008, pp. 677–682.

[29] Q. Q. Gu and J. Zhou, "Learning the shared subspace for multitask clustering and transductive transfer classification," in *Proc. 9th IEEE ICDM*, Dec. 2009, pp. 159–168.

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

[31] S. Si and D. C. Tao, "Evolutionary cross-domain discriminative Hessian eigenmaps journal," *Trans. Imag. Proc.*, vol. 19, no. 4, pp. 1075–1086, 2010.

[32] S. Si, D. C. 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.

[33] S. Gupta, D. Phung, B. Adams, T. Tran, and S. Venkatesh, "Nonnegative shared subspace learning and its application to social media retrieval," in *Proc. 16th ACM SIGKDD*, 2010, pp. 1169–1178.

[34] J. Jiang and C. X. Zhai, "Instance weighting for domain adaptation in NLP," in *Proc. 45th ACL*, 2007, pp. 264–271.

[35] C. Wan, R. Pan, and J. Li, "Bi-weighting domain adaptation for cross-language text classification," in *Proc. 22nd IJCAI*, 2011, pp. 1535–1540.

[36] M. Dredze, A. Kulesza, and K. Crammer, "Multidomain learning by confidence-weighted parameter combination," *Mach. Learn.*, vol. 79, no. 1, pp. 123–149, 2010.

[37] L. Lee and D. Seung, "Algorithms for nonnegative matrix factorization," vol. 13, pp. 556–562, 2001.

[38] C. Ding, T. Li, W. Peng, and H. Park, "Orthogonal nonnegative matrix tri-factorizations for clustering," in *Proc. 12th ACM SIGKDD*, 2006, pp. 126–135.

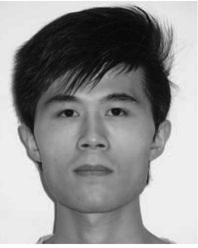
[39] D. Lee and H. Seung, "Learning the parts of objects by nonnegative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.

Fuzhen Zhuang is currently an Assistant Professor with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China. He has published several papers in some prestigious referred journals and conference proceedings, such as the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, *Information Sciences*, IJCAI, ACM CIKM, ACM WSDM, SIAM SDM and *IEEE ICDM*. His current research interests include transfer learning, machine learning, data mining, parallel classification, and clustering.

Ping Luo (M'07) received the Ph.D. degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China.

He is currently a Research Scientist with the Hewlett-Packard Labs, Beijing, China. He has published several papers in some prestigious refereed journals and conference proceedings, such as the IEEE TRANSACTIONS ON INFORMATION THEORY, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, *Journal of Parallel and Distributed Computing*, ACM SIGKDD, ACM CIKM, IJCAI. His current research interests include knowledge discovery and machine learning.

Dr. Luo was a recipient of the Doctoral Dissertation Award, China Computer Federation, in 2009. He is a member of the ACM.



Changying Du received the B.S. degree from the Department of Mathematics, Central South University, Changsha, China, in 2008, and is currently pursuing the Ph.D. degree at the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China.

His current research interests include machine learning, data mining, distributed classification and clustering, natural language processing, and information retrieval.

Qing He received the B.S. degree from Hebei Normal University, Shijiazhang, China, in 1985, the M.S. degree from Zhengzhou University, Zhengzhou, China, in 1987, both in mathematics, and the Ph.D. degree in fuzzy mathematics and artificial intelligence from Beijing Normal University, Beijing, China, in 2000.

He is currently a Professor with the Institute of Computing Technology, Chinese Academy of Science (CAS), Beijing, China, and a Professor with the Graduate University of Chinese Academy of Sciences, Beijing, China. Since 1987 to 1997, he has been with Hebei University of Science and Technology, Hebei, China. He is currently a Doctoral Tutor at the Institute of Computing and Technology, CAS. His current research interests include data mining, machine learning, classification, and fuzzy clustering.

Zhongzhi Shi (SM'98) is a Professor with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, leading the Research Group of Intelligent Science. His current research interests include intelligence science, multiagent systems, semantic web, machine learning, and neural computing.

Mr. Shi was a recipient of the 2nd-Grade National Award at Science and Technology Progress of China, in 2002, two 2nd-Grade Awards at Science and Technology Progress of the Chinese Academy of Sciences, in 1998 and 2001, respectively. He is a member of the AAAI and ACM, Chair for the WG 12.2 of IFIP. He serves as Vice President for Chinese Association of Artificial Intelligence.

Hui Xiong (SM'07) received the B.E. degree from the University of Science and Technology of China, Anhui, China, the M.S. degree from the National University of Singapore, Singapore, and the Ph.D. degree from the University of Minnesota, Minneapolis, MN, USA.

He is currently an Associate Professor in the Management Science and Information Systems Department, Rutgers University, Newark, NJ, USA. He has published over 90 technical papers in peer-reviewed journals and conference proceedings. His current research interests include data and knowledge engineering, with a focus on developing effective and efficient data analysis techniques for emerging data intensive applications.

Dr. Xiong was the recipient of the 2008 IBM ESA Innovation Award, the 2009 Rutgers University Board of Trustees Research Fellowship for Scholarly Excellence, the 2007 Junior Faculty Teaching Excellence Award, and the 2008 Junior Faculty Research Award at the Rutgers Business School. He is a Co-Editor of *Clustering and Information Retrieval* (Kluwer Academic, 2003) and a Co-Editor-in-Chief of *Encyclopedia of GIS* (Springer, 2008). He is an Associate Editor of the *Knowledge and Information Systems* journal and has served regularly on the organization committees and the program committees of a number of international conferences and workshops. He is a Senior Member of the ACM.