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

Fuzhen Zhuang
Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences
zhuangfz@ics.ict.ac.cn

Ping Luo
Hewlett-Packard Labs, China
ping.luo@hp.com

Changying Du[*]
Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences
ducy@ics.ict.ac.cn

Qing He
Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences
heq@ics.ict.ac.cn

Zhongzhi Shi
Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences
shizz@ics.ict.ac.cn

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 distribution with respect to the raw features. Some recent studies argued that the high-level concepts (e.g. word clusters) can help model the data distribution difference, and thus are more appropriate for classification. Specifically, 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, besides these shared concepts each domain may have its own distinct concepts. To address this point, we propose a general transfer learning framework based on non-negative matrix tri-factorization 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 may lead to better classification accuracy. To solve the proposed optimization problem we develop an iterative algorithm and also theoretically analyze its convergence. Finally, extensive experiments show the significant superiority of our model over the baseline methods. In particular, we show that our method works much better in the more challenging tasks when distinct concepts may exist.

[*]Changying Du is also with the Graduate University of Chinese Academy of Sciences.

WSDM'13, February 4–8, 2013, Rome, Italy.
Copyright 2013 ACM 978-1-4503-1869-3/13/02 ...$15.00.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning–Machine Learning

Keywords

Triplex Transfer Learning; Distribution Mismatch; Non-negative Matrix Tri-factorization; Common Concept; Distinct Concept

1. 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 efforts 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, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]. 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 key words 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 key words 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. In the other hand, we refer to the association between the concept 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 motivated 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 exist in the data domains. For example, there may be some concepts in a text corpus, which are totally irrelevant to the content of another corpus. 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. 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'.*

Table 1: The Three Kinds of Concepts

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

These three kinds of concepts are summarized in Table 1. 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] models the identical concepts only. MTrick [9] explores the associations between word cluster and document class for cross-domain classification, thus actually considers the alike concepts only. DKT [11] adopts the similar idea with MTrick for cross-language web page classification. Recently, DTL (Dual Transfer Learning) [12] is proposed to model alike and identical concepts together. Therefore, an ideal model should handle the alike, identical, and distinct concepts simultaneously. Motivated by this observation, we propose a general framework based on non-negative matrix tri-factorization (NMTF) techniques, which considers 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 (i.e. TriTL for short). For the sake of clarity, the differences of the four previous methods and our model are summarized in Table 2.

Table 2: The Comparison of Models

	Alike	Identical	Distinct
CoCC [5]		√	
MTrick [9]	√		
DKT [11]	√		
DTL [12]	√	√	
TriTL	√	√	√

To highlight the contributions of this work, we summarize them as follows.

1. We deeply analyze the commonalities and distinctions between the source and target domains, and find that there are also some distinct concepts in each of data domain.

2. We introduce distinct concepts into transfer learning. Together with alike and identical concepts we propose a triplex transfer learning model to model them simultaneously based on non-negative matrix tri-factorization. An iterative algorithm is developed to solve the proposed matrix factorization problem, and its theoretical analysis on algorithm convergence is also provided.

3. We conduct the systematic experiments to show the superiority of TriTL over the compared methods. In particular, we show that our method works much better in the more challenging tasks when distinct concepts may exist.

The rest of this paper is organized as follows, Section 2 briefly introduces the preliminary knowledge and notations, followed by the detailed formalization and solution of the proposed model TriTL in Section 3. Section 4 gives the experimental results. We summarize the related works in Section 5, and Section 6 concludes. Finally, the theoretical analysis of the iterative algorithm is given in Appendix.

2. PRELIMINARY KNOWLEDGE

In this section, we first give the notations used throughout this paper, and then briefly introduce the non-negative matrix tri-factorization (NMTF) technique and its notions.

2.1 Notations

We use calligraphic letters to represent sets, such as $\mathcal{D}$ is used to denote data set. The data matrices are written in upper case, such as X and Y, and $X_{[i,j]}$ indicates the i-th row and j-th column element of matrix X. Also, we use $\mathbb{R}$ and $\mathbb{R}_+$ to 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 1. For clarity, the frequently-used notations and denotations are summarized in Table 3.

2.2 Non-negative Matrix Factorization

Non-negative matrix factorization (NMF) technique has been widely used for text and image classification in the last

Table 3: The 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
$\top$	Denotes the transposition of matrix

decade [13, 14, 15, 16]. Our model is based on the non-negative matrix tri-factorization, and the basic formula is

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

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 on word clusterings. G denotes the document labeling information, and S denotes the association between word clusterings 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 3 and Appendix.

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

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

Actually, 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 $tr(X) = \sum_{i=1}^{m} X_{(ii)}$.

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*

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

The properties of the trace and frobenius norm are as follows,

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

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

Property 2. *Given 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*

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

3. TRIPLEX TRANSFER LEARNING

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

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

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

3.1 Problem Formalization

Supposed 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 sub-index 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 sub-index 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 sub-index 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 sub-index of r).

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

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

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

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

Here, the constraints inform that the sum of the entries in each column of F equals to 1 and the sum of the entries in each row of G equals to 1. 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 the different document classes.

3.2 The Solution to TriTL

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

$$\begin{aligned}\mathcal{L} &= \sum_{r=1}^{s+t} ||X_r - [F^1, {F^2}_r, {F^3}_r] \left[\begin{array}{c} S^1 \\ S^2 \\ {S^3}_r \end{array} \right] G_r^\top ||^2 \\ &= \sum_{r=1}^{s+t} tr(X_r^\top X_r - 2 \cdot X_r^\top [F^1, {F^2}_r, {F^3}_r] \left[\begin{array}{c} S^1 \\ S^2 \\ {S^3}_r \end{array} \right] G_r^\top \\ &\quad + G_r \left[\begin{array}{c} S^1 \\ S^2 \\ {S^3}_r \end{array} \right]^\top [F^1, {F^2}_r, {F^3}_r]^\top [F^1, {F^2}_r, {F^3}_r] \left[\begin{array}{c} S^1 \\ S^2 \\ {S^3}_r \end{array} \right] G_r^\top) \\ &= \sum_{r=1}^{s+t} tr(X_r^\top X_r - 2 \cdot X_r^\top A_r - 2 \cdot X_r^\top B_r - 2 \cdot X_r^\top C_r \\ &\quad + G_r {S^1}^\top {F^1}^\top A_r + G_r {S^2}^\top {F^2}_r^\top B_r + G_r {S^3}_r^\top {F^3}_r^\top C_r \\ &\quad + 2 \cdot G_r {S^1}^\top {F^1}^\top B_r + 2 \cdot G_r {S^1}^\top {F^1}^\top C_r + 2 \cdot G_r {S^2}^\top {F^2}_r^\top C_r) \\ &\quad s.t. \sum_{i=1}^{m} {F^1}_{[i,j]} = 1, \sum_{i=1}^{m} {F^2}_{r\,[i,j]} = 1, \\ &\quad\quad \sum_{i=1}^{m} {F^3}_{r\,[i,j]} = 1, \sum_{j=1}^{c} G_{r\,[i,j]} = 1,\end{aligned} \tag{11}$$

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 non-linear optimization techniques. In this work we develop an alternately iterative algorithm, which can converge to a local optimal solution.

In each round of iteration these matrices are updated as

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

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

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

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

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

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

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

After the calculation of each round of iteration, F^1, ${F^2}_r$, ${F^3}_r$, G_r are normalized using Eq.(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 $(1 \leq r \leq s)$ are assigned as the true label information. Specifically, $G_{r[i,u]} = 1$ if the i-th document belongs to the u-th 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 [17]. 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 $(s+1 \leq r \leq s+t)$. The convergence analysis of Algorithm 1 can be referred in Appendix.

Algorithm 1 Triplex Transfer Learning (TriTL) Algorithm

Input: The source domains $\mathcal{D}_r = \{x_i^{(r)}, y_i^{(r)}\}|_{i=1}^{n_r}$ $(1 \leq r \leq s)$, target domains $\mathcal{D}_r = \{x_i^{(r)}\}|_{i=1}^{n_r}$ $(s+1 \leq r \leq s+t)$, 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 $(1 \leq r \leq s)$ 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, ${S^3}_r$ $(1 \leq r \leq s+t)$, and G_r $(s+1 \leq r \leq s+t)$.

```
1. Initialization: The initializations of $F^{1(0)}$, ${F^2}_r^{(0)}$,
   ${F^3}_r^{(0)}$ are detailed in Section 3.2; $S^{1(0)}$, $S^{2(0)}$, ${S^3}_r^{(0)}$
   are randomly assigned, and $G_r^{(0)}$ $(s+1 \leq r \leq s+t)$
   are initialized as the probabilistic output by supervised
   learning models, such as Logistic Regression (LR) [18]
   in the experiments.
2. $k := 1$.
3. Update $F^{1(k)}$ according to Eq.(19);
4. For $r := 1 \to s+t$
       Update ${F^2}_r^{(k)}$ according to Eq.(20) and ${F^3}_r^{(k)}$
       according to Eq.(21);
5. end
6. Update $S^{1(k)}$ according to Eq.(22) and $S^{2(k)}$ according
   to Eq.(23);
7. For $r := 1 \to s+t$
       Update ${S^3}_r^{(k)}$ according to Eq.(24);
8. end
9. For $r := s+1 \to s+t$
       Update $G_r^{(k)}$ according to Eq.(25);
10. end
11. Normalize $F^{1(k)}$, ${F^2}_r^{(k)}$, ${F^3}_r^{(k)}$, $G_r^{(k)}$ according to
    Eq.(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
    $G_r^{(k)}$.
```

4. EXPERIMENTAL EVALUATION

In this section, we systemically demonstrate the effectiveness of the proposed transfer learning framework TriTL. 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 multi-class classification problems and multiple source and target domains, i.e., $s > 1$ and $t > 1$.

4.1 Data Preparation

20Newsgroups[2] is one of the benchmark data sets for evaluate transfer learning algorithms, which is widely used in previous works [1, 3, 19, 20]. 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 4.

Firstly, We construct the transfer learning tasks using the approach in [9]. For example, for the data set *rec vs. 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 data set *rec vs. 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 vs. 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.

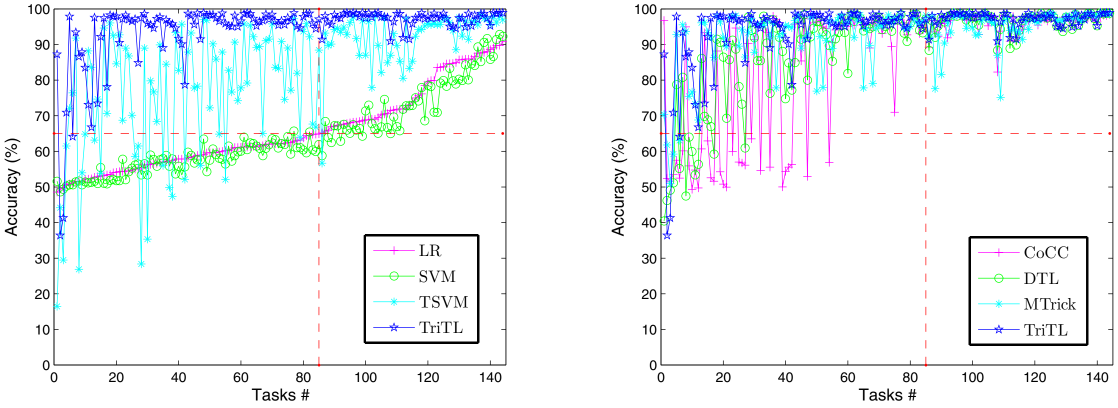
This way, we can construct additional 384 ($144 \div 3 \times 8$) classification tasks. Among all these 384 tasks we first run the supervised learning model Logistic Regression (LR) [18] on each of them. Then, we select the ones whose accuracies from LR are from 50% to 55% (this is only slightly better than the random classification, thus they might be much difficult). In summary, we have 65 such tasks for the experiments.

4.2 Experimental Setting

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

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

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



(a) TriTL vs. LR, SVM, TSVM

(b) TriTL vs. CoCC, DTL, MTrick

Figure 1: The Performance Comparison among LR, SVM, TSVM, CoCC, DTL, MTrick and TriTL on data set *rec vs. sci*

Table 4: The 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*

- The supervised algorithms: Logistic Regression (LR) [18], Support Vector Machine (SVM) [21];
- The semi-supervised algorithm: Transductive Support Vector Machine (TSVM) [22];
- The cross-domain methods: Co-clustering 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 $T = 100$. The baseline methods LR is implemented by Matlab[3], SVM and TSVM are given by SVM^{light4}. 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. Experimental results in Section 4.3 show that, this slight change of normalization results in significant improvement of DTL.

We use the classification accuracy as the evaluation metric,

$$auc = \frac{|\{d|d \in \mathcal{D} \wedge f(d) = y\}|}{n}, \tag{27}$$

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

4.3 Experimental Results

4.3.1 *Comparison on the Traditional Transfer Learning Tasks*

We compare TriTL with LR, SVM, TSVM, CoCC, DTL and MTrick on the data set *rec vs. sci*, and all the results of the 144 classification tasks are recorded in Figure 1 and Table 5. In figure 1, the 144 tasks are sorted by the increase order of the performance of LR. The lower accuracy of LR indicates that it is harder to transfer the knowledge from source domain to target domain. Also, these classification tasks are separated into two parts, the left side of red dotted line in figure 1 represents the problems with accuracy of LR lower than 65%, while the right higher than 65%. Table 5 lists the corresponding average performance.

From these results, we have the following findings.

- TriTL is significantly better than the supervised learning algorithms LR and SVM, and the semi-supervised method TSVM.
- TriTL significantly outperforms all the compared transfer learning algorithms CoCC, MTrick and DTL with the statistical test. In Talble 5, TriTL achieves the best results in term of the average performances, no matter the classification tasks with accuracy of LR lower or higher than 65%. This improvement might be due to the synthesized effectiveness in modeling all the three concepts. When the accuracy of LR is lower than 65%, 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 higher than 65% the data distributions of the source and target domains might be similar. Thus, modeling the identical concepts may work. Therefore, our model TriTL is much flexible under different situations.
- When the accuracy of LR is lower than 65%, 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.

[3] http://research.microsoft.com/∼minka/papers/logreg/.
[4] http://svmlight.joachims.org/.

Table 5: Average Performances (%) on 144 Tasks of Data Set *rec vs. sci*

Data Set		LR	SVM	TSVM	CoCC	DTL	MTrick	TriTL
rec vs. sci	*Lower*	57.41	56.78	75.73	79.69	84.29	90.44	**92.23**
	Higher	75.77	73.48	91.66	96.18	96.56	95.53	**97.19**
	Total	65.57	64.20	82.81	87.02	89.75	92.70	**94.43**

- When the accuracy of LR higher than 65%, the compared transfer learning algorithms perform similarly. However, we still can find that DTL is slightly better than CoCC, and CoCC slightly outperforms MTrick. The reason might be that modeling the identical concepts in DTL and CoCC improves the performance when the data distributions of the source and target domains are similar.

4.3.2 Comparison on the New Type of Transfer Learning Tasks

To further validate the effectiveness of TriTL, we construct the other 65 transfer learning tasks which are detailed in Section 4.1. The average accuracy values of these 65 tasks using all the methods are given in Table 6. From this table, it can be found that TriTL once more obtains the best results. In Table 6, MTrick is better than DTL, and DTL outperforms CoCC. These results are consistent to the analysis in Section 4.3.1 when the accuracy of LR is lower than 65%.

Table 6: Average Performances (%) on 65 Much Harder Transfer Learning Tasks

LR	SVM	TSVM	CoCC	DTL	MTrick	TriTL
52.45	51.81	74.32	69.66	75.34	78.45	**80.93**

4.4 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 10 randomly selected tasks from the 144 classification problems of *rec vs. sci*. We randomly sample 10 combinations of parameters, and all the results are shown in Table 7. The 12th and 13th row respectively represents the average accuracy and variance of each tasks under the 10 combinations of parameters. The last row is the result using the default parameters adopted in this paper.

It is obvious that in Table 7, the mean performance of the 10 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.

4.5 Algorithm Convergence

In this section, we also empirically check the convergence of the iterative algorithm to TriTL. We randomly choose 6 tasks from the data set *rec vs. sci*, and the results are shown in Figure 2. 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 Eq.(10), respectively. Both prediction accuracy and objective value can converge within 100 iterations, and the value of objective function in Eq.(10) decreases along with the iterating process, which coincides with the theoretic analysis.

5. RELATED WORKS

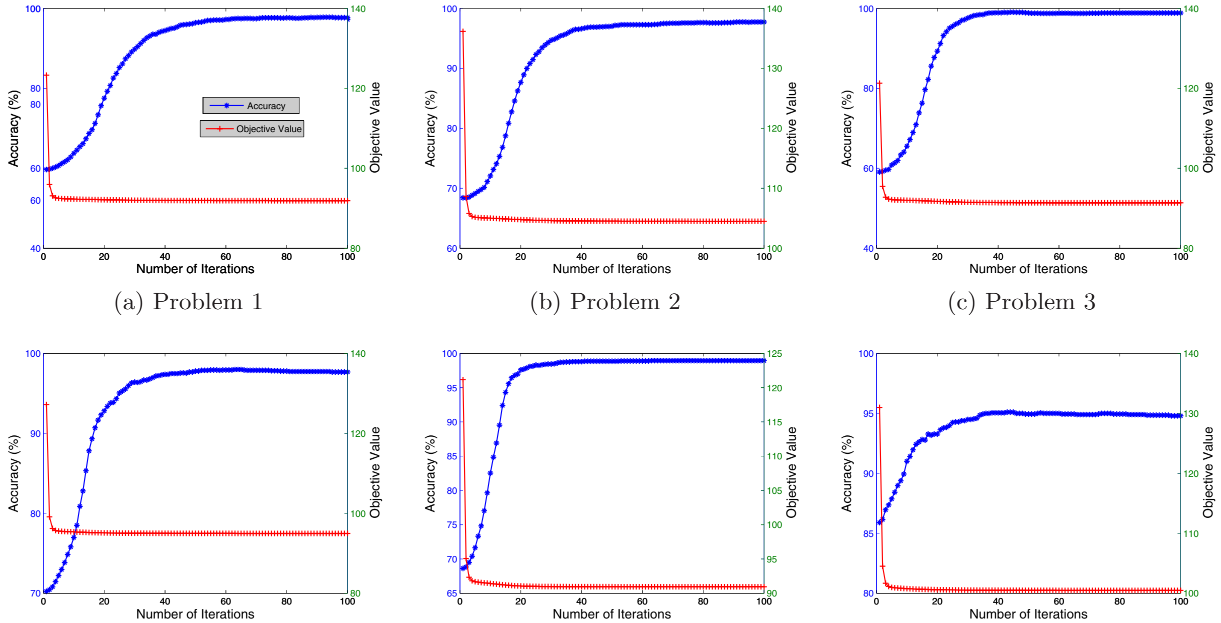
This section we summarize the related works of transfer learning, which has aroused large amounts of interest and research in recent years. Here we group the previous works of transfer learning into three categories, i.e., feature based, instant weighing based and model combination based transfer learning.

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 [23, 5, 24]. Jiang et al. [23] 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. [24] presented a novel method to identify variant and invariant features between two data sets 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 [25, 26, 27]. Pan et al. [25] proposed a dimensionality reduction approach to find out this latent feature space, in which supervised learning algorithms can be applied to train classification models. Gu et al. [26] 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. Gupta et al. [28] 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 re-weight 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 [20, 29, 30]. Dai et al. [20] 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. [29] proposed a general instance weighting framework, which has been validated to work well on NLP tasks. Wan et al. [30] first aligned the feature spaces in both domains utilizing some

Table 7: The 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) Problem 1 (b) Problem 2 (c) Problem 3

(d) Problem 4 (e) Problem 5 (f) Problem 6

Figure 2: Number of Iterations vs. the Performance of TriTL and Objective Value.

on-line 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, 31]. 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 [31] developed a new multi-domain 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 model them together. This work belongs to the feature based methods, and simultaneously model 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.

6. CONCLUSIONS

In this paper, we systemically analyze the three kinds of concepts, namely identical, alike and distinct concepts, among the source and target domains. By considering them altogether we propose a general model model TriTL based on nonnegative matrix tri-factorization. Then, an alternately iterative algorithm is developed to solve the proposed optimization problem. Finally, we construct two types of transfer learning tasks, on which we conduct the systematic experiments. It shows that TriTL always significantly outperforms the compared methods under different situations of the source and target domain.

It is worth mentioning that TriTL is a general model, which can tackle multiple source domains, multiple target domains and multi-class classification problems. Furthermore, we can easily incorporate unlabeled source domain data and labeled target domain data into this model.

7. ACKNOWLEDGEMENTS

The work is supported by the National Natural Science Foundation of China (No. 61175052, 60975039, 61203297, 60933004, 61035003), National High-tech R&D Program of China (863 Program) (No.2012AA011003).

8. REFERENCES

[1] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 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 *Proceedings of the 22nd NIPS*, 2008.

[3] J. Gao, W. Fan, J. Jiang, and J. W. Han. Knowledge transfer via multiple model local structure mapping. In *Proceedings of the 14th ACM SIGKDD*, pages 283–291, 2008.

[4] J. Gao, W. Fan, Y. Z. Sun, and J. W. Han. Heterogeneous source consensus learning via decision propagation and negotiation. In *Proceedings of the 15th ACM SIGKDD*, pages 339–348, 2009.

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

[6] P. Luo, F. Z. Zhuang, H. Xiong, Y. H. Xiong, and Q. He. Transfer learning from multiple source domains via consensus regularization. In *Proceedings of the 17th ACM CIKM*, pages 103–112, 2008.

[7] G. R. Xue, W. Y. Dai, Q. Yang, and Y. Yu. Topic-bridged plsa for cross-domain text classification. In *Proceedings of the 31st ACM SIGIR*, pages 627–634, 2008.

[8] W. Y. Dai, O. Jin, G. R. Xue, Q. Yang, and Y. Yu. Eigen transfer: a unified framework for transfer learning. In *Proceedings of the 26th ICML*, pages 193–200, 2009.

[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 *Proceedings of the 10th SIAM SDM*, pages 13–24, 2010.

[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 *Proceedings of the 19th ACM CIKM*, pages 359–368. ACM, 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 *Proceedings of the 34th ACM SIGIR*, pages 933–942. ACM, 2011.

[12] M. Long, J. Wang, G. Ding, W. Cheng, X. Zhang, and W. Wang. Dual transfer learning. In *Proceedings of the 12th SIAM SDM*, 2012.

[13] C. Ding, T. Li, W. Peng, and H. Park. Orthogonal nonnegative matrix t-factorizations for clustering. In *Proceedings of the 12th ACM SIGKDD*, pages 126–135. ACM, 2006.

[14] T. Li, V. Sindhwani, C. Ding, and Y. Zhang. Knowledge transformation for cross-domain sentiment classification. In *Proceedings of the 32nd ACM SIGIR*, pages 716–717. ACM, 2009.

[15] F. Wang, T. Li, and C. Zhang. Semi-supervised clustering via matrix factorization. In *Proceedings of The 8th SIAM SDM*, pages 1041–1048, 2008.

[16] T. Li, C. Ding, Y. Zhang, and B. Shao. Knowledge transformation from word space to document space. In *Proceedings of the 31st ACM SIGIR*, pages 187–194. ACM, 2008.

[17] T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Journal of Machine Learning*, pages 177–196, 2001.

[18] David Hosmer and Stanley Lemeshow. *Applied Logistic Regression*. Wiley, New York, 2000.

[19] D. Zhang, J. He, Y. Liu, L. Si, and R.D. Lawrence. Multi-view transfer learning with a large margin approach. In *Proceedings of the 17th ACM SIGKDD*, pages 1208–1216. ACM, 2011.

[20] W. Y. Dai, Q. Yang, G. R. Xue, and Y. Yu. Boosting for transfer learning. In *Proceedings of the 24th ICML*, pages 193–200, 2007.

[21] B. E. Boser, I. Guyou, and V. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the 5th AWCLT*, 1992.

[22] T. Joachims. Transductive inference for text classification using support vector machines. In *Proceedings of the 16th ICML*, 1999.

[23] J. Jiang and C. X. Zhai. A two-stage approach to domain adaptation for statistical classifiers. In *Proceedings of the 16th ACM CIKM*, pages 401–410, 2007.

[24] S. Uguroglu and J. Carbonell. Feature selection for transfer learning. *Machine Learning and Knowledge Discovery in Databases*, pages 430–442, 2011.

[25] S. J. Pan, J. T. Kwok, and Q. Yang. Transfer learning via dimensionality reduction. In *Proceedings of the 23rd AAAI*, pages 677–682, 2008.

[26] Q. Q. Gu and J. Zhou. Learning the shared subspace for multi-task clustering and transductive transfer classification. In *Proceedings of 9th ICDM*, pages 159–168. IEEE, 2009.

[27] S.J. Pan, I.W. Tsang, J.T. Kwok, and Q. Yang. Domain adaptation via transfer component analysis. *IEEE TRANSACTIONS ON NEURAL NETWORKS*, 22(2):199–210, 2011.

[28] S.K. Gupta, D. Phung, B. Adams, T. Tran, and S. Venkatesh. Nonnegative shared subspace learning and its application to social media retrieval. In *Proceedings of the 16th ACM SIGKDD*, pages 1169–1178. ACM, 2010.

[29] J. Jiang and C. X. Zhai. Instance weighting for domain adaptation in nlp. In *Proceedings of the 45th ACL*, pages 264–271, 2007.

[30] C. Wan, R. Pan, and J. Li. Bi-weighting domain adaptation for cross-language text classification. In *Proceedings of 22nd IJCAI*, pages 1535–1540, 2011.

[31] M. Dredze, A. Kulesza, and K. Crammer. Multi-domain learning by confidence-weighted parameter combination. *Machine Learning*, 79(1):123–149, 2010.

[32] L. Lee and D. Seung. Algorithms for non-negative matrix factorization. volume 13, pages 556–562, 2001.

[33] C. Ding, T. Li, W. Peng, and H. Park. Orthogonal nonnegative matrix tri-factorizations for clustering. In *Proceedings of the 12th ACM SIGKDD*, pages 126–135, 2006.

[34] D.D. Lee and H.S. Seung. Learning the parts of

objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

APPENDIX

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

$$\begin{aligned}\mathcal{G}(F^1) =&\sum_{r=1}^{s+t}||X_r - F_rS_rG_r^\top||^2 \\ &+ tr[\boldsymbol{\lambda}(F^{1^\top}\mathbf{1}_m - \mathbf{1}_{k_1})(F^{1^\top}\mathbf{1}_m - \mathbf{1}_{k_1})^\top],\end{aligned} \tag{28}$$

where $\boldsymbol{\lambda} \in \mathbb{R}^{k_1 \times k_1}$ is a diagonal matrix. Omitting the items, which are independent of F^1, Eq. (28) becomes

$$\begin{aligned}\mathcal{G}(F^1) =&\sum_{r=1}^{s+t} tr(-2 \cdot X_r^\top F^1 S^1 G_r^\top + G_r S^{1^\top} F^{1^\top} A_r \\ &+ 2 \cdot G_r S^{1^\top} F^{1^\top} B_r + 2 \cdot G_r S^{1^\top} F^{1^\top} C_r) \\ &+ tr[\boldsymbol{\lambda}(F^{1^\top}\mathbf{1}_m\mathbf{1}_m^\top F^1 - 2 \cdot \mathbf{1}_{k_1}\mathbf{1}_m^\top F^1)],\end{aligned} \tag{29}$$

Then the differential is

$$\begin{aligned}\frac{\partial \mathcal{G}}{\partial F^1} =&\sum_{r=1}^{s+t}(-2 \cdot X_r G_r S^{1^\top} + 2 \cdot A_r G_r S^{1^\top} + 2 \cdot B_r G_r S^{1^\top} \\ &+ 2 \cdot C_r G_r S^{1^\top}) + 2 \cdot \mathbf{1}_m({\mathbf{1}_m}^\top F^1 - {\mathbf{1}_{k_1}}^\top)\boldsymbol{\lambda},\end{aligned} \tag{30}$$

Lemma 1. *Using the update rule (31), Equation (29) will monotonously decrease.*

$$F^1{}_{[i,j]} \leftarrow F^1{}_{[i,j]} \cdot \sqrt{\frac{[\sum_{r=1}^{s+t} X_r G_r S^{1^\top} + \mathbf{1}_m {\mathbf{1}_{k_1}}^\top \boldsymbol{\lambda}]_{[i,j]}}{[\sum_{r=1}^{s+t} D_r + \mathbf{1}_m {\mathbf{1}_m}^\top F^1 \boldsymbol{\lambda}]_{[i,j]}}}, \tag{31}$$

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 [32] 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*

$$\mathcal{Q}(Y, \widetilde{Y}) \geq \mathcal{T}(Y), \mathcal{Q}(Y, Y) = \mathcal{T}(Y), \tag{32}$$

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

Then, define

$$Y^{(t+1)} = arg\min_Y \mathcal{Q}(Y, Y^{(t)}). \tag{33}$$

Through this definition,

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

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,

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

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 [33]. Then, while fixing $F^{1'}$, we minimize $\mathcal{Q}(F^1, F^{1'})$. The differential of $\mathcal{Q}(F^1, F^{1'})$ is

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

Let $\frac{\partial \mathcal{Q}(F^1, F^{1'})}{\partial F^1{}_{[i,j]}} = 0$, we can obtain Eq.(31). Thus, the update rule (31) 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, $\boldsymbol{\lambda}$ 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 1. Here we adopt the normalization technology in [34, 9] to satisfy the constrains regardless of $\boldsymbol{\lambda}$. Specifically, in each iteration we use Eq.(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 Eq.(26) and Eq.(19) can be approximately equivalent to Equation (31) when only considering the convergence. In our solution, we adopt the approximate update rule of Eq.(19) by omitting the items which depends on $\boldsymbol{\lambda}$ in Eq.(31). 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, $S^3{}_r$ $(1 \leq r \leq \ s + t)$, G_r $(s + 1 \leq r \leq \ s + t)$ in Eqs. (20), (21), (22), (23), (24), (25), (26) respectively.

Theorem 1 (Convergence). *After each round of iteration in Algorithm 1 the objective function in Eq.(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 [32], each update step in Algorithm 1 will not increase Eq. (10) and the objective has a lower bounded by zero, which guarantee the convergence. Thus, the above theorem holds.