Exploiting Associations between Word Clusters and Document Classes for Cross-domain Text Categorization†

Fuzhen Zhuang[1,2*], Ping Luo[3], Hui Xiong[4], Qing He[1], Yuhong Xiong[5] and Zhongzhi Shi[1]

[1]*The Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences*

[2]*Graduate University of Chinese Academy of Sciences, Beijing 100039, China*

[3]*Hewlett Packard Labs China*

[4]*MSIS Department, Rutgers University*

[5]*Innovation Works*

Received 26 April 2010; revised 7 October 2010; accepted 27 October 2010
DOI:10.1002/sam.10099
Published online 30 November 2010 in Wiley Online Library (wileyonlinelibrary.com).

Abstract: Cross-domain text categorization targets on adapting the knowledge learnt from a labeled source domain to an unlabeled target domain, where the documents from the source and target domains are drawn from different distributions. However, in spite of the different distributions in raw-word features, the associations between word clusters (conceptual features) and document classes may remain stable across different domains. In this paper, we exploit these unchanged associations as the bridge of knowledge transformation from the source domain to the target domain by the non-negative matrix tri-factorization. Specifically, we formulate a joint optimization framework of the two matrix tri-factorizations for the source- and target-domain data, respectively, in which the associations between word clusters and document classes are shared between them. Then, we give an iterative algorithm for this optimization and theoretically show its convergence. The comprehensive experiments show the effectiveness of this method. In particular, we show that the proposed method can deal with some difficult scenarios where baseline methods usually do not perform well. © 2010 Wiley Periodicals, Inc. Statistical Analysis and Data Mining 4: 100–114, 2011

Keywords: cross-domain learning; domain adaption; transfer learning; text categorization

1. INTRODUCTION

Many learning techniques work well only under a common assumption: the training and test data are drawn from the same feature space and the same distribution. When the features or distribution change, most statistical models need to be rebuilt from scratch using newly collected training data. However, in many real-world applications it is expensive or impossible to recollect the needed training data. It would be nice to reduce the need and effort to recollect the training data. This leads to the research of *cross-domain learning*[1] [1–9]. In this paper, we study the problem of cross-domain learning for text categorization. We assume that the documents from the source and target domains share the same space of word feature, also, they share the same set of document labels. Under these assumptions, we study how to accurately predict the class labels of the documents in the target domain with a different data distribution.

In cross-domain learning for text categorization it is quite often that different domains use different phrases to express the same concept. For instance, the words indicating

Correspondence to: Fuzhen Zhuang (zhuangfz@ics.ict.ac.cn)
† This is an invited submission from the Best of SDM 2010.

[1] Previous works often refer this problem as *transfer learning* or *domain adaption*.

the concept of *computer science* can be 'hardware', 'software', 'program', 'programmer', 'disks', 'rom', and so on. However, the frequencies of these words are different in different domains. In the news about a hardware company the high-frequency words may be 'hardware', 'disks', 'rom', etc., while the words like 'software', 'program' and 'programmer' are the high-frequency ones in the domain of software companies. Thus, features on raw words are not reliable for text classification, especially in cross-domain learning. On the other side, the concept behind the words may have the same effect to indicate the class labels of the documents from different domains. In this example, a page is more likely to be *computer*-related if it contains the concept of *computer science*. In other words, only concepts behind raw words are reliable in indicating taxonomy. Additionally, this reliable association between word concepts and document classes is also stable across data domains. Therefore, we can use it as bridge to transfer knowledge cross different domains.

Intuitive Example. Since a word concept can be expressed as a group of related words, word concept and word cluster are interchangeable in the following. To intuitively show the motivation behind this work we borrow the example introduced in Ref. [10]. In panel (a) of Fig. 1, there are four synthetic documents, specifically, D1 and D2 belong to the class of information retrieval (IR), and D3 and D4 belong to the class of computer vision (CV). Let D1 and D3 be the data with labels in the source domain, and D2 and D4 without labels in the target domain. Panel (b) of Fig. 1 gives the vector representation of data set based on raw words (the data set includes six distinct words after removing stop words). We find that if these words in the data set can be grouped into three word clusters, the data set can also be represented based on concepts shown in Panel (c) of Fig. 1. Clearly, the features over the word clusters are more useful in classification than the raw-word features. Furthermore, as shown in Panel (d) we compute the co-occurrence matrixes of the concept cluster and document class for the source and target domains, respectively. We can see that in this example these two matrixes are the same. Through this manual-built example we show that in general the association between word cluster and document class remains stable across data domains. Thus, it can be used as the bridge to transfer knowledge.

Motivated by this observations, in this study, we explicitly consider the stable associations between word concepts and document classes across data domains by the non-negative matrix factorization (NMF). The basic formula of matrix tri-factorization is as follows:

$$\boldsymbol{X}_{m\times n} = \boldsymbol{F}_{m\times k_1}\boldsymbol{S}_{k_1\times k_2}\boldsymbol{G}^{\mathrm{T}}_{n\times k_2}, \tag{1}$$

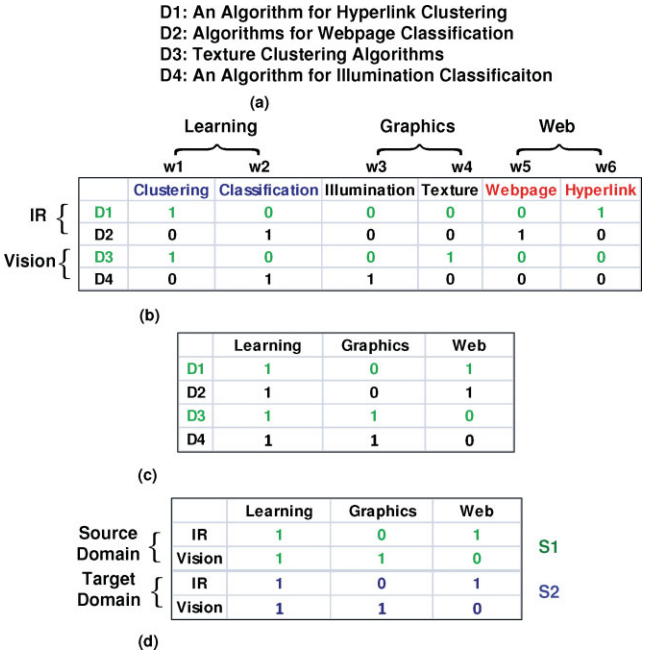

Fig. 1 An intuitive example. Concept based on classification is more stable than raw word based, and the same association of word cluster and document class may be shared by different domains. (a) A synthetic data set; (b) vector representation for data set based on raw words; (c) vector representation for data set based on concepts; (d) The co-occurence of word cluster and document class on source and target domains.

where $\boldsymbol{X}$ is the joint probability matrix for a given word-document matrix $\boldsymbol{Y}$ ($\boldsymbol{X} = \frac{\boldsymbol{Y}}{\sum_{i,j}\boldsymbol{Y}_{i,j}}$), and m, n, k_1, k_2 are the numbers of words, documents, word clusters, and document clusters, respectively. Conceptually, $\boldsymbol{F}$ denotes the word clustering information, $\boldsymbol{G}$ denotes the document clustering information, and $\boldsymbol{S}$ denotes the association between word clusters and document clusters. Later, we will detail the meaning of $\boldsymbol{F}$, $\boldsymbol{S}$, and $\boldsymbol{G}$, and argue that only $\boldsymbol{S}$ is stable for different domains, while $\boldsymbol{F}$ and $\boldsymbol{G}$ can be different in different domains.

Therefore, we propose a matrix tri-factorization-based classification framework (MTrick) for cross-domain learning. Indeed, we conduct a joint optimization for the two matrix tri-factorizations on the source- and target-domain data, respectively, where $\boldsymbol{S}$, denoting the association between word clusters and document clusters, is shared in these two tri-factorizations as the bridge of knowledge transformation. Additionally, the class label information of the source-domain data is injected into the matrix $\boldsymbol{G}$ for the source domain to supervise the optimization process. Then, we develop an alternately iterative algorithm to solve this joint optimization problem, and theoretically prove its convergence. Experimental results show the effectiveness of MTrick for cross-domain learning.

Contributions. We conclude the contributions of this work as follows:

(1) For the problem of cross-domain classification, we observe that though the distributions in raw-word features are different, the associations between word clusters (conceptual features) and document classes may remain stable across different domains for classification.

(2) Along this line, we proposed to simultaneously tri-factorize two matrixes on the source and target domains, while using the association between word clusters and document clusters as bridge to transfer knowledge.

(3) To solve the joint optimization problem, we develop an iterative algorithm and theoretically analyze its convergence.

(4) Finally, we conduct systemic experiments to show the effectiveness of the proposed algorithm, including binary classification and multiple class cases. Experimental results note that our algorithm can gain significant improvement over baseline methods, and also can perform well on some difficult scenarios.

In our previous work [11], we propose a MTrick for cross-domain learning under the observation that though the distributions are different cross different data domains, the association between word cluster and document class may retain the same and be independent of data domains. Then we develop an alternately iterative algorithm to solve the optimization problem and theoretically analyze its convergence. Finally, the experiments on two-class classification tasks show the advantage of the proposed method.

In this paper, we further give an intuitive example to make more clear of the motivation for our work. Although the distributions are different cross different domains, the association between word cluster and document class may retain the same. Thus we can explicitly exploit the stable association cross different domains for cross-domain learning. We also conduct much more experiments to further validate the effectiveness of the proposed method. The new experiments include 4×100 problem instances on four data sets for three-class text classification, and 96 problem instances for two-class text classification. All these results again validate the superiority of the proposed method, and MTrick can also perform very well even when the difficulty degree of transfer learning is great (see Section 6.4).

Overview. The remainder of this paper is organized as follows. In Section 2, we present related work, then we introduce the framework of MTrick in Section 3. Section 4 presents the optimization solution. In Section 5, we provide a theoretical analysis of the convergence of the proposed iterative method. Section 6 gives the experimental evaluation to show the effectiveness of MTrick. Finally, Section 7 concludes the paper.

2. RELATED WORK

In this section, we introduce some previous works which are closely related to our work.

2.1. Non-negative Matrix Factorization

Since our algorithm framework is based on the NMF, so here we introduce some works about NMF. NMF has been shown to be widely used for many applications, such as dimensionality reduction, pattern recognition, clustering, and classification [12–18]. Lee and Seung [13] proposed the NMF to decompose the multivariate data, and gave two different multiplicative algorithms for NMF. Moreover, they applied an auxiliary function to prove the monotonic convergence of both algorithms. After this pioneering work researchers extended this model and apply them to different applications. Guillamet *et al*. [15] extended the NMF to a weighted non-negative matrix factorization (WNMF) to improve the capabilities of representations. Experimental results show that WNMF achieves a great improvement in the image classification accuracy compared with NMF and principal component analysis (PCA). Ding *et al*. [12] provided an analysis of the relationship between 2-facts and 3-facts NMF, and proposed an orthogonal non-negative matrix tri-factorization for clustering. They empirically showed that the bi-orthogonal non-negative matrix tri-factorization-based approach can simultaneously cluster rows and columns of the input data matrix effectively. Wang *et al*. [17] developed a novel matrix factorization-based approach for semisupervised clustering and extended it to different kinds of constrained coclusterings. The probabilistic topic models, such as probabilistic latent semantic analysis (PLSA) [19] and latent dirichlet allocation (LDA) [20], can also be considered as a method of non-negative matrix tri-factorization [21]. They are different from the proposed model of MTrick in that: the word clusters and document clusters in topic models share the same semantic space, actually the space of *latent topics* [19]. However, in MTrick, the word clusters and document clusters have different semantic spaces, and the associations between word clusters and document clusters are explicitly expressed.

Researchers also leverage NMF for transfer learning tasks. Li et al. [8,22] proposed two stage methods

for sentiment classification based on constrained nonnegative matrix tri-factorizations. Both of them first transfer document-side sentiment into word-level sentiment on the source-domain data, and then transfers word sentiment learnt in the first stage to documents in the target domain. Also the method in Ref. [22] needs some labeled data. While our method focuses on modeling the association of word concepts and document classes, which is domain-independent. Also our method does not need the human effort to label some data in target domain. As shown by the example in Section 1, different domains may use different words to express the same word concept, so we think that the word cluster in two domains may be similar, but not exactly the same due to the distribution difference. Thus, in this paper, we propose to share only the association between word clusters and document classes. Li *et al*. [18] developed a novel approach for cross-domain collaborative filtering, in which a *codebook* (referred as the association between word clusters and document clusters in our paper) is shared. In the above two papers, they dealt with two separate tasks of matrix factorization: first on the source domain, and then on the target domain. Additionally, the shared information is the output from the first step, and also the input of the second step. However, in our work, we combine the two factorizations into a collaborative optimization task, and show the extra value of this collaborative optimization by the experimental results.

2.2. Cross-domain Learning

Recent years have witnessed numerous research in cross-domain learning. In general, cross-domain learning for classification can be grouped into two categories, namely instance weighting-based and feature selection-based cross-domain learning methods.

Instance weighting-based approaches focus on the reweighted strategy that increases the weight of instances which are close to the target domain in data distribution and decreases the weight of instances which are far from the target domain. Dai *et al*. [7] 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 and Zhai [23] also dealt with the domain adaptation from the view of instance weighting. They found that the difference of the joint distributions between the source domain and target domain is the cause of the domain adaptation problem, and proposed a general instance weighting framework, which has been validated to work well on natural language processing (NLP) tasks.

Feature selection-based approaches aim to find a common feature space which is useful to cross-domain learning. Jiang and Zhai [24] developed a two-phase feature selection framework for domain adaptation. In that approach, they first selected the features called generalizable features which are emphasized while training a general classifier. Then they leveraged unlabeled data from target domain to pick up features that are specifically useful for the target domain. Dai *et al*. [5] proposed a coclustering-based approach for this problem. In this method, they identified the word clusters among the source and target domains, via which the class information and knowledge propagated from source domain to target domain. Pan *et al*. [25] proposed a dimensionality reduction approach, in which they can find out the latent feature space which can be regarded as the bridged knowledge between the source domain and the target domain. Si *et al*. [26] 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. Si *et al*. [27] also proposed a transfer subspace learning framework, which includes two items. The first one is the general subspace learning framework, which can apply to various dimensionality reduction algorithms; while the second one minimizes the Bregman divergence between the distribution of training data and testing data in the selected subspace. The proposed algorithm in this paper can also be regarded as the feature selection-based approach for cross-domain learning.

3. PRELIMINARIES AND PROBLEM FORMULATION

In this section, we first introduce some basic concepts and mathematical notations used throughout this paper, and then formulate the MTrick.

3.1. Basic Concepts and Notations

In this paper, we use bold letters, such as $\mathbf{u}$ and $\mathbf{v}$, to represent vectors. Data matrixes are written in bold upper case, such as $\boldsymbol{X}$ and $\boldsymbol{Y}$. Also, $X_{(ij)}$ indicates the ith row and jth column element of matrix $\boldsymbol{X}$. Calligraphic letters, such as $\mathcal{A}$ and $\mathcal{D}$, are used to represent sets. Finally, we use $\mathbb{R}$ and $\mathbb{R}_+$ to denote the set of real numbers and non-negative real numbers respectively.

DEFINITION 1: *(Trace of matrix): Given a data matrix* $\boldsymbol{X} \in \mathbb{R}^{n\times n}$*, the trace of* $\boldsymbol{X}$ *is computed as*

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

Actually, the trace of matrix can also be computed when the matrix is not a square matrix. Without losing any generality, let $m < n$ and $\boldsymbol{X} \in \mathbb{R}^{m \times n}$, then $\mathrm{Tr}(\boldsymbol{X}) = \sum_{i=1}^{m} X_{(ii)}$.

DEFINITION 2: *(Frobenius norm of matrix): Given a data matrix $\boldsymbol{X} \in \mathbb{R}^{m \times n}$, the Frobenius norm of $\boldsymbol{X}$ is computed as*

$$||\boldsymbol{X}||^2 = \sum_{i=1}^{m} \sum_{j=1}^{n} X_{(ij)}^2. \tag{3}$$

Additionally, we give some properties of the trace and Frobenius norm, which will be used in Sections 4 and 5.

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

$$\mathrm{Tr}(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}) = \mathrm{Tr}(\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}). \tag{4}$$

Property 2 *Given matrixes $\boldsymbol{X}, \boldsymbol{Y} \in \mathbb{R}^{m \times n}$, then*

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

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

$$||\boldsymbol{X}||^2 = \mathrm{Tr}(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}) = \mathrm{Tr}(\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}). \tag{6}$$

3.2. Problem Formulation

For the joint probability matrix $\boldsymbol{X}_{\mathrm{s}} \in \mathbb{R}_{+}^{m \times n_{\mathrm{s}}}$ in the source-domain data (where m is the number of words, and n_{s} is the number of documents in the source domain), we formulate the following constrained optimization problem:

$$\begin{aligned} &\min_{\boldsymbol{F}_{\mathrm{s}}, \boldsymbol{S}_{\mathrm{s}}, \boldsymbol{G}_{\mathrm{s}}} ||\boldsymbol{X}_{\mathrm{s}} - \boldsymbol{F}_{\mathrm{s}}\boldsymbol{S}_{\mathrm{s}}\boldsymbol{G}_{\mathrm{s}}^{\mathrm{T}}||^2 + \frac{\alpha}{n_{\mathrm{s}}} \cdot ||\boldsymbol{G}_{\mathrm{s}} - \boldsymbol{G}_0||^2, \\ &\text{s.t.} \quad \sum_{j=1}^{k_1} F_{\mathrm{s}_{(ij)}} = 1, \sum_{j=1}^{k_2} G_{\mathrm{s}_{(ij)}} = 1, \end{aligned} \tag{7}$$

where α is the trade-off parameter, $\boldsymbol{G}_0$ contains the true label information in the source domain. Specifically, when the ith instance belongs to class j, then $G_{0_{(ij)}} = 1$; and $G_{0_{(ik)}} = 0$ for $k \neq j$. In this formulation, $\boldsymbol{G}_0$ is used as the supervised information by requiring $\boldsymbol{G}_{\mathrm{s}}$ is similar to $\boldsymbol{G}_0$. After minimizing Eq. (7), we obtain $\boldsymbol{F}_{\mathrm{s}}, \boldsymbol{G}_{\mathrm{s}}, \boldsymbol{S}_{\mathrm{s}}$, where

- $\boldsymbol{F}_{\mathrm{s}} \in \mathbb{R}_{+}^{m \times k_1}$ represents the information of word clusters, and $F_{\mathrm{s}(ij)}$ is the probability that the ith word belongs to the jth word cluster.
- $\boldsymbol{G}_{\mathrm{s}} \in \mathbb{R}_{+}^{n_{\mathrm{s}} \times k_2}$ represents the information of document clusters, and $G_{\mathrm{s}(ij)}$ is the probability that the ith document belongs to the jth document cluster.
- $\boldsymbol{S}_{\mathrm{s}} \in \mathbb{R}_{+}^{k_1 \times k_2}$ represents the associations between word clusters and document clusters.

Then, for the joint probability matrix $\boldsymbol{X}_{\mathrm{t}} \in \mathbb{R}_{+}^{m \times n_{\mathrm{t}}}$ in the target-domain data (n_{t} is the number of documents in the target domain), we can also formulate the following constrained optimization problem:

$$\begin{aligned} &\min_{\boldsymbol{F}_{\mathrm{t}}, \boldsymbol{G}_{\mathrm{t}}} ||\boldsymbol{X}_{\mathrm{t}} - \boldsymbol{F}_{\mathrm{t}}\boldsymbol{S}_0\boldsymbol{G}_{\mathrm{t}}^{\mathrm{T}}||^2 \\ &\text{s.t.} \quad \sum_{j=1}^{k_1} F_{\mathrm{t}_{(ij)}} = 1, \sum_{j=1}^{k_2} G_{\mathrm{t}_{(ij)}} = 1, \end{aligned} \tag{8}$$

where $\boldsymbol{S}_0$ is the optimal value for $\boldsymbol{S}_{\mathrm{s}}$ resulting from solving the problem in Eq. (7). In this formulation, $\boldsymbol{S}_0$ is used as the supervised information for the optimization process. This is motivated by the analysis that the source and target domain may share the same associations between word clusters and document clusters. After minimizing Eq. (8), we obtain $\boldsymbol{F}_{\mathrm{t}}, \boldsymbol{G}_{\mathrm{t}}$. Their explanations are similar to those for $\boldsymbol{F}_{\mathrm{s}}, \boldsymbol{G}_{\mathrm{s}}$, respectively. Then, the class label of the ith document in the target domain is output as

$$\mathrm{index}_i = \arg\max_j G_{\mathrm{t}_{(ij)}}. \tag{9}$$

Finally, we can combine the two sequential optimization problems in Eqs. (7) and (8) into a joint optimization formulation as follows:

$$\begin{aligned} &\min_{\boldsymbol{F}_{\mathrm{s}}, \boldsymbol{G}_{\mathrm{s}}, \boldsymbol{S}, \boldsymbol{F}_{\mathrm{t}}, \boldsymbol{G}_{\mathrm{t}}} ||\boldsymbol{X}_{\mathrm{s}} - \boldsymbol{F}_{\mathrm{s}}\boldsymbol{S}\boldsymbol{G}_{\mathrm{s}}^{\mathrm{T}}||^2 + \frac{\alpha}{n_{\mathrm{s}}} \cdot ||\boldsymbol{G}_{\mathrm{s}} - \boldsymbol{G}_0||^2 \\ &\qquad + \beta \cdot ||\boldsymbol{X}_{\mathrm{t}} - \boldsymbol{F}_{\mathrm{t}}\boldsymbol{S}\boldsymbol{G}_{\mathrm{t}}^{\mathrm{T}}||^2, \\ &\text{s.t.} \quad \sum_{j=1}^{k_1} F_{\mathrm{s}_{(ij)}} = 1, \sum_{j=1}^{k_2} G_{\mathrm{s}_{(ij)}} = 1, \\ &\qquad \sum_{j=1}^{k_1} F_{\mathrm{t}_{(ij)}} = 1, \sum_{j=1}^{k_2} G_{\mathrm{t}_{(ij)}} = 1, \end{aligned} \tag{10}$$

where $\alpha \geq 0$ and $\beta \geq 0$ are the trade-off factors. In this formulation, $\boldsymbol{S}$ is shared in the matrix factorizations of the source and target domains. This way $\boldsymbol{S}$ is used as the bridge of knowledge transformation from the source domain to the target domain. Next we focus only on how to solve the joint optimization problem in Eq. (10), which can cover both the subproblems in Eqs. (7) and (8).

4. SOLUTION TO THE OPTIMIZATION PROBLEM

In this section, we develop an alternate iterative algorithm to solve the problem in Eq. (10). According to the

preliminary knowledge in Section 3.1, we know that the minimization of Eq. (10) is equivalent to minimizing the following equation:

$$\begin{aligned}
&\mathcal{L}(\boldsymbol{F}_{\mathrm{s}}, \boldsymbol{G}_{\mathrm{s}}, \boldsymbol{S}, \boldsymbol{F}_{\mathrm{t}}, \boldsymbol{G}_{\mathrm{t}}) \\
&\quad = \mathrm{Tr}(\boldsymbol{X}_{\mathrm{s}}^{\mathrm{T}}\boldsymbol{X}_{\mathrm{s}} - 2\boldsymbol{X}_{\mathrm{s}}^{\mathrm{T}}\boldsymbol{F}_{\mathrm{s}}\boldsymbol{S}\boldsymbol{G}_{\mathrm{s}}^{\mathrm{T}} + \boldsymbol{G}_{\mathrm{s}}\boldsymbol{S}^{\mathrm{T}}\boldsymbol{F}_{\mathrm{s}}^{\mathrm{T}}\boldsymbol{F}_{\mathrm{s}}\boldsymbol{S}\boldsymbol{G}_{\mathrm{s}}^{\mathrm{T}}) \\
&\qquad + \frac{\alpha}{n_{\mathrm{s}}} \cdot \mathrm{Tr}(\boldsymbol{G}_{\mathrm{s}}\boldsymbol{G}_{\mathrm{s}}^{\mathrm{T}} - 2\boldsymbol{G}_{\mathrm{s}}\boldsymbol{G}_0^{\mathrm{T}} + \boldsymbol{G}_0\boldsymbol{G}_0^{\mathrm{T}}) \\
&\qquad + \beta \cdot \mathrm{Tr}(\boldsymbol{X}_{\mathrm{t}}^{\mathrm{T}}\boldsymbol{X}_{\mathrm{t}} - 2\boldsymbol{X}_{\mathrm{t}}^{\mathrm{T}}\boldsymbol{F}_{\mathrm{t}}\boldsymbol{S}\boldsymbol{G}_{\mathrm{t}}^{\mathrm{T}} + \boldsymbol{G}_{\mathrm{t}}\boldsymbol{S}^{\mathrm{T}}\boldsymbol{F}_{\mathrm{t}}^{\mathrm{T}}\boldsymbol{F}_{\mathrm{t}}\boldsymbol{S}\boldsymbol{G}_{\mathrm{t}}^{\mathrm{T}}), \\
&\text{s.t.} \quad \sum_{j=1}^{k_1} F_{\mathrm{s}(ij)} = 1, \sum_{j=1}^{k_2} G_{\mathrm{s}(ij)} = 1, \\
&\qquad \sum_{j=1}^{k_1} F_{\mathrm{t}(ij)} = 1, \sum_{j=1}^{k_2} G_{\mathrm{t}(ij)} = 1.
\end{aligned} \tag{11}$$

where $\mathcal{L}$ is the objective function. The partial differential of $\mathcal{L}$ is as follows:

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \boldsymbol{F}_{\mathrm{s}}} &= -2\boldsymbol{X}_{\mathrm{s}}\boldsymbol{G}_{\mathrm{s}}\boldsymbol{S}^{\mathrm{T}} + 2\boldsymbol{F}_{\mathrm{s}}\boldsymbol{S}\boldsymbol{G}_{\mathrm{s}}^{\mathrm{T}}\boldsymbol{G}_{\mathrm{s}}\boldsymbol{S}^{\mathrm{T}}, \\
\frac{\partial \mathcal{L}}{\partial \boldsymbol{G}_{\mathrm{s}}} &= -2\boldsymbol{X}_{\mathrm{s}}^{\mathrm{T}}\boldsymbol{F}_{\mathrm{s}}\boldsymbol{S} + 2\boldsymbol{G}_{\mathrm{s}}\boldsymbol{S}^{\mathrm{T}}\boldsymbol{F}_{\mathrm{s}}^{\mathrm{T}}\boldsymbol{F}_{\mathrm{s}}\boldsymbol{S} \\
&\quad + \frac{2\alpha}{n_{\mathrm{s}}} \cdot (\boldsymbol{G}_{\mathrm{s}} - \boldsymbol{G}_0), \\
\frac{\partial \mathcal{L}}{\partial \boldsymbol{S}} &= -2\boldsymbol{F}_{\mathrm{s}}^{\mathrm{T}}\boldsymbol{X}_{\mathrm{s}}\boldsymbol{G}_{\mathrm{s}} + 2\boldsymbol{F}_{\mathrm{s}}^{\mathrm{T}}\boldsymbol{F}_{\mathrm{s}}\boldsymbol{S}\boldsymbol{G}_{\mathrm{s}}^{\mathrm{T}}\boldsymbol{G}_{\mathrm{s}} \\
&\quad - 2\beta \cdot \boldsymbol{F}_{\mathrm{t}}^{\mathrm{T}}\boldsymbol{X}_{\mathrm{t}}\boldsymbol{G}_{\mathrm{t}} + 2\beta \cdot \boldsymbol{F}_{\mathrm{t}}^{\mathrm{T}}\boldsymbol{F}_{\mathrm{t}}\boldsymbol{S}\boldsymbol{G}_{\mathrm{t}}^{\mathrm{T}}\boldsymbol{G}_{\mathrm{t}}, \\
\frac{\partial \mathcal{L}}{\partial \boldsymbol{F}_{\mathrm{t}}} &= -2\beta \cdot \boldsymbol{X}_{\mathrm{t}}\boldsymbol{G}_{\mathrm{t}}\boldsymbol{S}^{\mathrm{T}} + 2\beta \cdot \boldsymbol{F}_{\mathrm{t}}\boldsymbol{S}\boldsymbol{G}_{\mathrm{t}}^{\mathrm{T}}\boldsymbol{G}_{\mathrm{t}}\boldsymbol{S}^{\mathrm{T}}, \\
\frac{\partial \mathcal{L}}{\partial \boldsymbol{G}_{\mathrm{t}}} &= -2\beta \cdot \boldsymbol{X}_{\mathrm{t}}^{\mathrm{T}}\boldsymbol{F}_{\mathrm{t}}\boldsymbol{S} + 2\beta \cdot \boldsymbol{G}_{\mathrm{t}}\boldsymbol{S}^{\mathrm{T}}\boldsymbol{F}_{\mathrm{t}}^{\mathrm{T}}\boldsymbol{F}_{\mathrm{t}}\boldsymbol{S}.
\end{aligned}$$

Since $\mathcal{L}$ is not concave, it is hard to obtain the global solution by applying the latest nonlinear optimization techniques. In this study, we develop an alternately iterative algorithm, which can converge to a local optimal solution.

In each round of iteration these matrixes are updated as

$$F_{\mathrm{s}(ij)} \leftarrow F_{\mathrm{s}(ij)} \cdot \sqrt{\frac{(X_{\mathrm{s}}G_{\mathrm{s}}S^{\mathrm{T}})_{(ij)}}{(F_{\mathrm{s}}SG_{\mathrm{s}}^{\mathrm{T}}G_{\mathrm{s}}S^{\mathrm{T}})_{(ij)}}}, \tag{12}$$

$$G_{\mathrm{s}(ij)} \leftarrow G_{\mathrm{s}(ij)} \cdot \sqrt{\frac{(X_{\mathrm{s}}^{\mathrm{T}}F_{\mathrm{s}}S + \frac{\alpha}{n_{\mathrm{s}}} \cdot G_0)_{(ij)}}{(G_{\mathrm{s}}S^{\mathrm{T}}F_{\mathrm{s}}^{\mathrm{T}}F_{\mathrm{s}}S + \frac{\alpha}{n_{\mathrm{s}}} \cdot G_{\mathrm{s}})_{(ij)}}}, \tag{13}$$

$$F_{\mathrm{t}(ij)} \leftarrow F_{\mathrm{t}(ij)} \cdot \sqrt{\frac{(X_{\mathrm{t}}G_{\mathrm{t}}S^{\mathrm{T}})_{(ij)}}{(F_{\mathrm{t}}SG_{\mathrm{t}}^{\mathrm{T}}G_{\mathrm{t}}S^{\mathrm{T}})_{(ij)}}}, \tag{14}$$

$$G_{\mathrm{t}(ij)} \leftarrow G_{\mathrm{t}(ij)} \cdot \sqrt{\frac{(X_{\mathrm{t}}^{\mathrm{T}}F_{\mathrm{t}}S)_{(ij)}}{(G_{\mathrm{t}}S^{\mathrm{T}}F_{\mathrm{t}}^{\mathrm{T}}F_{\mathrm{t}}S)_{(ij)}}}. \tag{15}$$

Then, we normalize $\boldsymbol{F}_{\mathrm{s}}, \boldsymbol{G}_{\mathrm{s}}, \boldsymbol{F}_{\mathrm{t}}, \boldsymbol{G}_{\mathrm{t}}$ to satisfy the equality constraints. The normalization formulas are as follows:

$$F_{\mathrm{s}(i\cdot)} \leftarrow \frac{F_{\mathrm{s}(i\cdot)}}{\sum_{j=1}^{k_1} F_{\mathrm{s}(ij)}}, \tag{16}$$

$$G_{\mathrm{s}(i\cdot)} \leftarrow \frac{G_{\mathrm{s}(i\cdot)}}{\sum_{j=1}^{k_2} G_{\mathrm{s}(ij)}}, \tag{17}$$

$$F_{\mathrm{t}(i\cdot)} \leftarrow \frac{F_{\mathrm{t}(i\cdot)}}{\sum_{j=1}^{k_1} F_{\mathrm{t}(ij)}}, \tag{18}$$

$$G_{\mathrm{t}(i\cdot)} \leftarrow \frac{G_{\mathrm{t}(i\cdot)}}{\sum_{j=1}^{k_2} G_{\mathrm{t}(ij)}}. \tag{19}$$

Next, using the normalized $\boldsymbol{F}_{\mathrm{s}}, \boldsymbol{G}_{\mathrm{s}}, \boldsymbol{F}_{\mathrm{t}}, \boldsymbol{G}_{\mathrm{t}}$ we update $\boldsymbol{S}$ as follows:

$$S_{(ij)} \leftarrow S_{(ij)} \cdot \sqrt{\frac{(F_{\mathrm{s}}^{\mathrm{T}}X_{\mathrm{s}}G_{\mathrm{s}} + \beta \cdot F_{\mathrm{t}}^{\mathrm{T}}X_{\mathrm{t}}G_{\mathrm{t}})_{(ij)}}{(F_{\mathrm{s}}^{\mathrm{T}}F_{\mathrm{s}}SG_{\mathrm{s}}^{\mathrm{T}}G_{\mathrm{s}} + \beta \cdot F_{\mathrm{t}}^{\mathrm{T}}F_{\mathrm{t}}SG_{\mathrm{t}}^{\mathrm{T}}G_{\mathrm{t}})_{(ij)}}}. \tag{20}$$

The detailed procedure of this iterative computation is given in Algorithm 1.

5. ANALYSIS OF ALGORITHM CONVERGENCE

To investigate the convergence of iterating rules in Eqs. (12)–(20), we first check the convergence of $\boldsymbol{F}_s$ when $\boldsymbol{G}_{\mathrm{s}}, \boldsymbol{S}, \boldsymbol{F}_{\mathrm{t}}, \boldsymbol{G}_{\mathrm{s}}$ are fixed. For this optimization problem with constraints we formulate the following Lagrangian function:

$$\begin{aligned}
\mathcal{G}(\boldsymbol{F}_{\mathrm{s}}) &= ||\boldsymbol{X}_{\mathrm{s}} - \boldsymbol{F}_{\mathrm{s}}\boldsymbol{S}\boldsymbol{G}_{\mathrm{s}}^{\mathrm{T}}||^2 \\
&\quad + \mathrm{Tr}[\lambda(\boldsymbol{F}_{\mathrm{s}}\mathbf{u}^{\mathrm{T}} - \mathbf{v}^{\mathrm{T}})(\boldsymbol{F}_{\mathrm{s}}\mathbf{u}^{\mathrm{T}} - \mathbf{v}^{\mathrm{T}})^{\mathrm{T}}],
\end{aligned} \tag{21}$$

where $\lambda \in \mathbb{R}^{m\times m}$, $\mathbf{u} \in \mathbb{R}^{1\times k_1}$, $\mathbf{v} \in \mathbb{R}^{1\times m}$ (the entry values of $\mathbf{u}$ and $\mathbf{v}$ are all equal to 1), and $||\boldsymbol{X}_{\mathrm{s}} - \boldsymbol{F}_{\mathrm{s}}\boldsymbol{S}\boldsymbol{G}_{\mathrm{s}}^{\mathrm{T}}||^2 = \mathrm{Tr}(\boldsymbol{X}_{\mathrm{s}}^{\mathrm{T}}\boldsymbol{X}_{\mathrm{s}} - 2\boldsymbol{X}_{\mathrm{s}}^{\mathrm{T}}\boldsymbol{F}_{\mathrm{s}}\boldsymbol{S}\boldsymbol{G}_{\mathrm{s}}^{\mathrm{T}} + \boldsymbol{G}_{\mathrm{s}}\boldsymbol{S}^{\mathrm{T}}\boldsymbol{F}_{\mathrm{s}}^{\mathrm{T}}\boldsymbol{F}_{\mathrm{s}}\boldsymbol{S}\boldsymbol{G}_{\mathrm{s}}^{\mathrm{T}})$. Then,

$$\frac{\partial \mathcal{G}}{\partial \boldsymbol{F}_{\mathrm{s}}} = -2\boldsymbol{X}_{\mathrm{s}}\boldsymbol{G}_{\mathrm{s}}\boldsymbol{S}^{\mathrm{T}} + 2\boldsymbol{F}_{\mathrm{s}}\boldsymbol{S}\boldsymbol{G}_{\mathrm{s}}^{\mathrm{T}}\boldsymbol{G}_{\mathrm{s}}\boldsymbol{S}^{\mathrm{T}} + 2\lambda\boldsymbol{F}_{\mathrm{s}}\mathbf{u}^{\mathrm{T}}\mathbf{u} - 2\lambda\mathbf{v}^{\mathrm{T}}\mathbf{u}. \tag{22}$$

LEMMA 1: Using the update rule given in Eq. (23), Eq. (21) will monotonously decrease.

$$F_{\mathrm{s}(ij)} \leftarrow F_{\mathrm{s}(ij)} \cdot \sqrt{\frac{(X_{\mathrm{s}}G_{\mathrm{s}}S^{\mathrm{T}} + \lambda\mathbf{v}^{\mathrm{T}}\mathbf{u})_{(ij)}}{(F_{\mathrm{s}}SG_{\mathrm{s}}^{\mathrm{T}}G_{\mathrm{s}}S^{\mathrm{T}} + \lambda F_{\mathrm{s}}\mathbf{u}^{\mathrm{T}}\mathbf{u})_{(ij)}}}. \tag{23}$$

Proof: To prove Lemma 1 we describe the definition of auxiliary function [13] as follows. ■

Algorithm 1 The MTrick Algorithm

Input: The joint probability matrix $\boldsymbol{X}_s \in \mathbb{R}_+^{m\times n_s}$ on labeled source domain; the true label information $\boldsymbol{G}_0$ of source domain; the joint probability matrix $\boldsymbol{X}_t \in \mathbb{R}_+^{m\times n_t}$ on unlabeled target domain; and the trade-off factors α, β; the error threshold $\varepsilon > 0$; and the maximal iterating number max.
Output: The information of word clusters $\boldsymbol{F}_s$ and $\boldsymbol{F}_t$, the information of document clusters $\boldsymbol{G}_s$ and $\boldsymbol{G}_t$, the association between word clusters and document clusters $\boldsymbol{S}$.

1. Initialize the matrix variables as $\boldsymbol{F}_s^{(0)}$, $\boldsymbol{F}_t^{(0)}$, $\boldsymbol{G}_s^{(0)}$, $\boldsymbol{G}_t^{(0)}$ and $\boldsymbol{S}^{(0)}$. The initialization method will be detailed in the experimental section.
2. Calculate the initial value $\mathcal{L}^{(0)}$ of Eq. (11).
3. $k := 1$.
4. Update $\boldsymbol{F}_s^{(k)}$ based on Eq. (12), and normalize $\boldsymbol{F}_s^{(k)}$ based on Eq. (16).
5. Update $\boldsymbol{G}_s^{(k)}$ based on Eq. (13), and normalize $\boldsymbol{G}_s^{(k)}$ based on Eq. (17).
6. Update $\boldsymbol{F}_t^{(k)}$ based on Eq. (14), and normalize $\boldsymbol{F}_t^{(k)}$ based on Eq. (18).
7. Update $\boldsymbol{G}_t^{(k)}$ based on Eq. (15), and normalize $\boldsymbol{G}_t^{(k)}$ based on Eq. (19).
8. Update $\boldsymbol{S}^{(k)}$ based on Eq. (20).
9. Calculate the value $\mathcal{L}^{(k)}$ of Eq. (11). If $|\mathcal{L}^{(k)} - \mathcal{L}^{(k-1)}| < \varepsilon$, then turn to Step 11.
10. $k := k + 1$. If $k \le$ max, then turn to Step 4.
11. Output the word clustering information $\boldsymbol{F}_s^{(k)}$ and $\boldsymbol{F}_t^{(k)}$, the document clustering information $\boldsymbol{G}_s^{(k)}$ and $\boldsymbol{G}_t^{(k)}$, the association between word clusters and document clusters $\boldsymbol{S}^{(k)}$.

DEFINITION 3: *(Auxiliary function): A function $H(\boldsymbol{Y}, \widetilde{\boldsymbol{Y}})$ is called an auxiliary function of $\mathcal{T}(\boldsymbol{Y})$ if it satisfies*

$$H(\boldsymbol{Y}, \widetilde{\boldsymbol{Y}}) \ge \mathcal{T}(\boldsymbol{Y}),\ H(\boldsymbol{Y}, \boldsymbol{Y}) = \mathcal{T}(\boldsymbol{Y}), \tag{24}$$

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

Then, define

$$\boldsymbol{Y}^{(t+1)} = \arg\min_Y H(\boldsymbol{Y}, \boldsymbol{Y}^{(t)}). \tag{25}$$

Through this definition,

$$\mathcal{T}(\boldsymbol{Y}^{(t)}) = H(\boldsymbol{Y}^{(t)}, \boldsymbol{Y}^{(t)}) \ge H(\boldsymbol{Y}^{(t+1)}, \boldsymbol{Y}^{(t)}) \ge \mathcal{T}(\boldsymbol{Y}^{(t+1)}).$$

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

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

$$\begin{aligned} H(\boldsymbol{F}_s, \boldsymbol{F}'_s) = &-2\sum_{ij}(X_s G_s S^T)_{(ij)} F'_{s_{(ij)}} \left(1 + \log\frac{F_{s_{(ij)}}}{F'_{s_{(ij)}}}\right) \\ &+ \sum_{ij}(F'_s S G_s^T G_s S^T)_{(ij)} \frac{F^2_{s_{(ij)}}}{F'_{s_{(ij)}}} \\ &+ \sum_{ij}(\lambda F'_s \mathbf{u}^T\mathbf{u})_{(ij)} \frac{F^2_{s_{(ij)}}}{F'_{s_{(ij)}}} \\ &- 2\sum_{ij}(\lambda \mathbf{v}^T\mathbf{u})_{(ij)} F'_{s_{(ij)}} \left(1 + \log\frac{F_{s_{(ij)}}}{F'_{s_{(ij)}}}\right). \end{aligned} \tag{26}$$

Obviously, when $\boldsymbol{F}'_s = \boldsymbol{F}_s$ the equality $H(\boldsymbol{F}_s, \boldsymbol{F}'_s) = \mathcal{G}(\boldsymbol{F}_s)$ holds. Also we can prove the inequality $H(\boldsymbol{F}_s, \boldsymbol{F}'_s) \ge \mathcal{G}(\boldsymbol{F}_s)$ holds using the similar proof approach in Ref. [12]. Then, while fixing $\boldsymbol{F}'_s$, we minimize $H(\boldsymbol{F}_s, \boldsymbol{F}'_s)$.

$$\begin{aligned} \frac{\partial H(\boldsymbol{F}_s, \boldsymbol{F}'_s)}{\partial F_{s_{(ij)}}} = &-2(X_s G_s S^T)_{(ij)} \frac{F'_{s_{(ij)}}}{F_{s_{(ij)}}} \\ &+ 2(F'_s S G_s^T G_s S^T + \lambda F'_s \mathbf{u}^T\mathbf{u})_{(ij)} \frac{F_{s_{(ij)}}}{F'_{s_{(ij)}}} \\ &- 2(\lambda \mathbf{v}^T\mathbf{u})_{(ij)} \frac{F'_{s_{(ij)}}}{F_{s_{(ij)}}}. \end{aligned} \tag{27}$$

Let $\frac{\partial H(\boldsymbol{F}_s, \boldsymbol{F}'_s)}{\partial F_{s_{(ij)}}} = \mathbf{0}$,

$$\Rightarrow F_{s_{(ij)}} = F'_{s_{(ij)}} \cdot \sqrt{\frac{(X_s G_s S^T + \lambda \mathbf{v}^T\mathbf{u})_{(ij)}}{(F'_s S G_s^T G_s S^T + \lambda F'_s \mathbf{u}^T\mathbf{u})_{(ij)}}}. \tag{28}$$

Thus, the update rule given in Eq. (23) decreases the values of $\mathcal{G}(F_s)$. Then, Lemma 1 holds.

The only obstacle left is the computation of the Lagrangian multiplier λ. Actually, λ in this problem is to drive the solution to satisfy the constrained condition that the sum of the values in each row of $\boldsymbol{F}_s$ is 1. Here we propose a simple normalization technique to satisfy the constraints regardless of λ. Specifically, in each iteration we

use Eq. (16) to normalize $\boldsymbol{F}_s$. After normalization, the two constants of $\lambda \boldsymbol{F}_s \mathbf{u}^T \mathbf{u}$ and $\lambda \mathbf{v}^T \mathbf{u}$ are equal. Thus, the effect of Eqs. (12) and (16) can be approximately equivalent to Eq. (23) when only considering the convergence. In other words, we adopt the approximate update rule of Eq. (12) by omitting the items which depend on λ in Eq. (23). We can use the similar method to analyze the convergence of the update rules for $\boldsymbol{G}_s, \boldsymbol{F}_t, \boldsymbol{G}_t, \boldsymbol{S}$ in Eqs. (13), (14), (15), and (20), 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 $\boldsymbol{F}_s, \boldsymbol{G}_s, \boldsymbol{F}_t, \boldsymbol{G}_t, \boldsymbol{S}$, and the Multiplicative Update Rules [13], each update step in Algorithm 1 will not increase Eq. (10) and the objective has a lower bound zero, which guarantee the convergence. Thus, the above theorem holds.

6. EXPERIMENTAL VALIDATION

In this section, we show experiments to validate the effectiveness of the proposed algorithm. We focus on the two-class and three-class classification problems in the experiments (the number of document clusters are set to two and three, respectively).

6.1. Data Preparation

20Newsgroup[2] is one of the benchmark data sets for text categorization. Since the data set is not originally designed for cross-domain learning, we need to do some data preprocessing. The data set is a collection of approximately 20 000 newsgroup documents, which is partitioned evenly cross 20 different newsgroups. Each newsgroup corresponds to a different topic, and some of the newsgroups are very closely related. Thus, they can be grouped into certain top category. For example, the top category *sci* contains four subcategories *sci.crypt*, *sci.electronics*, *sci.med*, and *sci.space*. Four top categories in *20Newsgroup* are used for our experiments, which are detailed in Table 1.

6.1.1. Two-class classification

We select three top categories *sci*, *talk*, and *rec* to perform two-class classification experiments. Any two top categories can be selected to construct two-class classification problems, and we can construct three data sets *sci versus talk*, *rec versus sci*, and *rec versus talk* in the experimental setting. For the data set *sci versus talk*, we randomly select one subcategory from *sci* and one subcategory from *talk*, which denote the positive and negative data, respectively. The test data set is similarly constructed as the training data set, except that they are from different subcategories. Thus, the constructed classification task is suitable for cross-domain learning due to the facts that (i) the training and test data are from different distributions since they are from different subcategories; (ii) they are also related to each other since the positive (negative) instances in the training and test set are from the same top categories. For the data set *sci versus talk*, we totally construct 144 ($P_4^2 \cdot P_4^2$) classification tasks. The data sets *rec versus sci* and *rec versus talk* are constructed similarly with *sci versus talk*.

Table 1. 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*

6.1.2. Three-class classification

We construct three-classification problems similarly to the two-class case. For the four top categories, we can construct four data sets, *comp versus rec versus sci*, *comp versus rec versus talk*, *comp versus sci versus talk*, and *rec versus sci versus talk*, by randomly selecting three top categories. For each data set, the subcategories from each top categories are selected to form source and target domains, except that they are from different subcategories. Thus we can obtain 1728 ($P_4^2 \cdot P_4^2 \cdot P_4^2$) classification tasks for each data set. In this three-class situation, we only perform the experiments on 100 randomly selected problem instances from each data set.

To further validate our algorithm, we also perform experiments on the data set *Reuters-21578*,[3] which has three top categories *orgs*, *people*, and *place* (each top category also has several subcategories). We evaluate the proposed algorithm on three classification tasks constructed by Gao *et al.* [2].

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

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

6.2. Baseline Methods and Evaluation Metric

We compare MTrick with some baseline classification methods, including the supervised algorithms of logistic regression (LR) [28], LibSVM [29], support vector machine (SVM) [30], and the semisupervised algorithm of transductive support vector machine (TSVM) [31], also the cross-domain methods of coclustering-based classification (CoCC) [5] and the local weighted ensemble (LWE) [2]. Additionally, the two-step optimization approach using Eqs. (7) and (8) is adopted as baseline (denoted as MTrick0). The prediction accuracy on the unlabeled target-domain data is the evaluation metric.

6.3. Implementation Details

In MTrick, $\boldsymbol{F}_{\mathrm{s}}, \boldsymbol{F}_{\mathrm{t}}, \boldsymbol{G}_{\mathrm{s}}, \boldsymbol{G}_{\mathrm{t}}, \boldsymbol{S}$ are initialized as follows:

1. $\boldsymbol{F}_{\mathrm{s}}$ and $\boldsymbol{F}_{\mathrm{t}}$ are initialized as the word clustering results by PLSA [19]. Specifically, $F_{\mathrm{s}_{(ij)}}$ and $F_{\mathrm{t}_{(ij)}}$ are both initialized to be $P(z_j|w_i)$ output by PLSA on the whole data set of the source and target domain. We adopt the Matlab implementation of PLSA[4] in the experiments.
2. $\boldsymbol{G}_{\mathrm{s}}$ is initialized as the true class information in the source domain.
3. $\boldsymbol{G}_{\mathrm{t}}$ is initialized as the predicted results of any supervised classifier, which is trained based on the source-domain data. In this experiment, LR is adopted to give these initial results.
4. $\boldsymbol{S}$ is initialized as follows: each entry is assigned with the same value and the sum of values in each row satisfies $\sum_j S_{(ij)} = 1$.

Note that PLSA has a randomly initialization process. Thus, we perform the experiments three times and the average performance of MTrick is output. The *tf-idf* weights are used as entry values of the word-document matrix $\boldsymbol{Y}$, which is then transformed to the joint probability matrix $\boldsymbol{X}$. The threshold of document frequency with value of 15 is used to decrease the number of features. After some preliminary test, we set the trade-off parameters $\alpha = 1$, $\beta = 1.5$, the error threshold $\varepsilon = 10^{-11}$, the maximal iterating number $\max = 100$, and the number of word clusters $k_1 = 50$.

The baseline methods LR is implemented by the package,[5] SVM and TSVM are given by SVM$^{\mathrm{light}}$.[6]

The parameter settings of CoCC and LWE are the same with those in their original papers, and the value of α in Eq. (7) is set to 1 after careful investigation for MTrick0.

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

[5] http://research.microsoft.com/~minka/papers/logreg/

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

6.4. Experimental Results

Next, we present detailed experimental results. To intuitively show the advantage of our method, the best values of accuracy are marked with bold font in Tables 2, 4 and 5.

6.4.1. Two-class classification

A Comparison of LR, SVM, TSVM, CoCC, MTrick0, and MTrick: We compare these classification approaches on the data set *sci versus talk*, *rec versus sci*, and *rec versus talk*, and all the results of 144×3 problems are recorded in Figs. 2–4. The 144 problems of each data set are sorted by increasing order of the performance of LR. Thus, the x-axes in these figures can also indicate the degree of difficulty in knowledge transferring.

From the results, we have the following observations:

- Figures 2(a), 3(a), and 4(a) show that MTrick is significantly better than the supervised learning algorithms LR and SVM, which indicates that the traditional supervised learning approaches cannot perform well on the cross-domain learning tasks.
- MTrick is also much better than the semisupervised method of TSVM.
- In Figs. 2(b), 3(b), and 4(b), the left side of red-dotted line represents the results when the accuracy of LR lower than 65%, while the right represents the results when the accuracy of LR higher than 65%. It is shown that when LR achieves accuracy higher than this threshold, MTrick and CoCC perform similarly. However, when the accuracy of LR is lower than this threshold, MTrick performs much better than CoCC. These results indicate that MTrick has the stronger ability to transfer knowledge when the labeled source domain cannot provide enough auxiliary information.
- MTrick is also better than MTrick0, which shows that the joint optimization can achieve a better solution than the separate optimization.

Additionally, we compare these classification algorithms by the average performance of all 144 problems from each data set, and the results are listed in Table 2 (L and R denote the average results when the accuracy of LR lower and higher than 65%, respectively, while Total represents the average results on all 144 problems). The t-test with 95% confidence also shows the performance improvement of MTrick by other compared algorithms is statistically significant. All these results again show that MTrick is an effective approach for cross-domain learning, and has stronger ability to transfer knowledge.

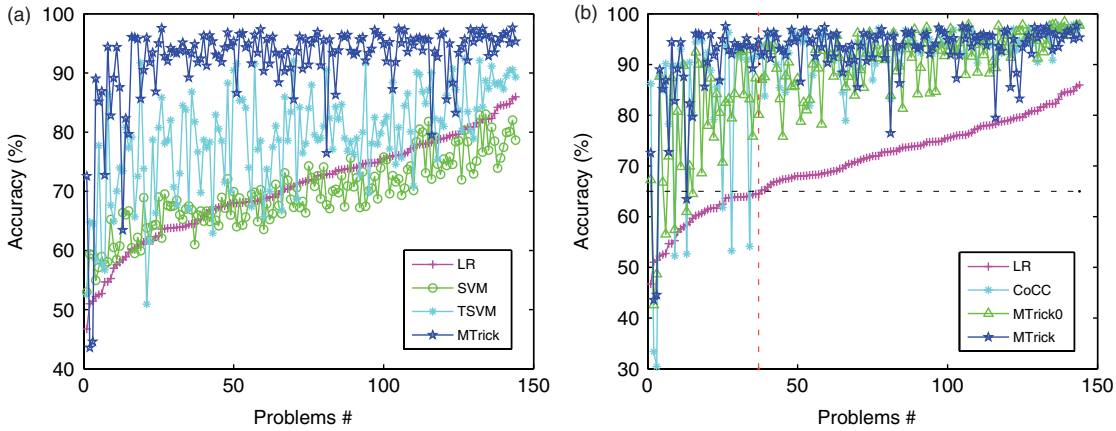

Fig. 2 The performance comparison among LR, SVM, TSVM, CoCC, MTrick0, and MTrick on data set *sci versus talk*. (a) MTrick versus LR, SVM, TSVM on data set *sci versus talk*; (b) MTrick versus MTrick0, CoCC on data set *sci versus talk*.

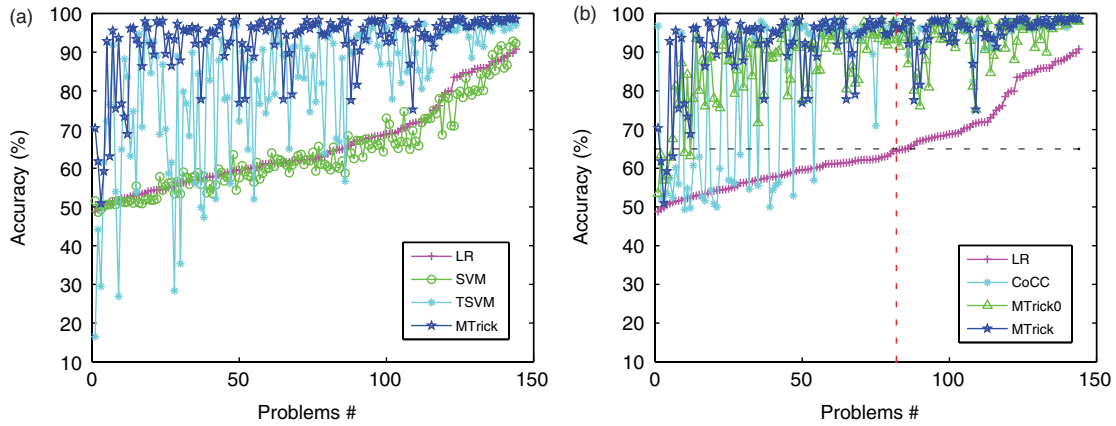

Fig. 3 The performance comparison among LR, SVM, TSVM, CoCC, MTrick0, and MTrick on data set *rec versus sci*. (a) MTrick versus LR, SVM, TSVM on data set *rec versus sci*; (b) MTrick versus MTrick0, CoCC on data set *rec versus sci*.

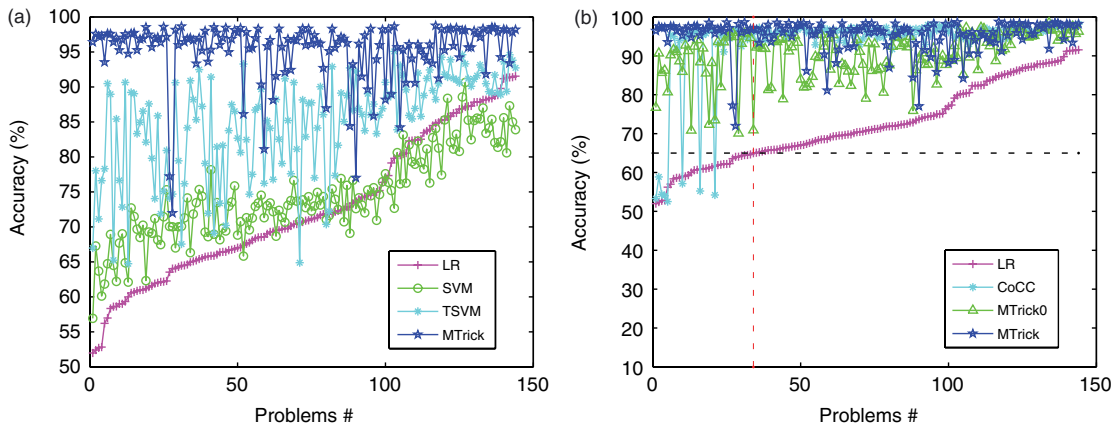

Fig. 4 The performance comparison among LR, SVM, TSVM, CoCC, MTrick0, and MTrick on data set *rec versus talk*. (a) MTrick versus LR, SVM, TSVM on data set *rec versus talk*; (b) MTrick versus MTrick0, CoCC on data set *rec versus talk*.

Table 2. Average performances (%) on 144 problem instances of each data set for two-class classification.

Data sets		LR	SVM	TSVM	CoCC	MTrick0	MTrick
sci versus talk	*L*	59.09	62.88	72.13	81.09	76.90	**86.52**
	R	74.21	71.70	81.58	93.41	91.28	**93.71**
	Total	70.64	69.62	79.35	90.50	87.88	**92.01**
rec versus sci	*L*	57.42	56.78	75.73	79.69	85.39	**90.44**
	R	75.76	73.48	91.66	**96.18**	93.50	95.53
	Total	65.57	64.20	82.81	87.02	88.99	**92.70**
rec versus talk	*L*	60.28	67.64	79.82	85.62	87.62	**95.57**
	R	76.29	76.52	86.52	**96.14**	91.19	95.09
	Total	72.49	74.42	84.94	93.66	90.35	**95.21**

Table 3. The data description for performance comparison among LR, SVM, TSVM, CoCC, LWE, and MTrick.

Data sets	Source domain $\mathcal{D}_s$	Target domain $\mathcal{D}_t$
orgs versus people orgs versus place people versus place	Document from a set of subcategories	Document from a different set of subcategories

Table 4. The performance comparison results (%) among LR, SVM, TSVM, CoCC, LWE, and MTrick.

Data sets	LR	SVM	TSVM	CoCC	LWE	MTrick
orgs versus people	74.92	74.25	73.80	79.79	79.67	**80.80**
orgs versus place	71.91	69.99	69.89	74.18	73.04	**76.77**
people versus place	58.03	59.05	58.43	66.94	68.52	**69.02**

A Comparison of LR, SVM, TSVM, CoCC, LWE, and MTrick: Furthermore, we also compare MTrick with LR, SVM, TSVM, CoCC, and LWE on *Reuters-21578*. The adopted data sets[7] are depicted in Table 3. The experimental results are recorded in Table 4 (we adopt the evaluation results of TSVM and LWE on the three problems in Ref. [2]). We can find that MTrick is better than all the algorithms LR, SVM, TSVM, CoCC, and LWE, which again show the effectiveness of MTrick.

6.4.2. *Multiclass classification*

To further test the superiority of MTrick to deal with multiclass classification problems, we compare it with LR, LibSVM, CoCC, and MTrick0. Note that LR and CoCC are adapted to handling multiclass situation by one versus rest manner.

We conduct three-class classification experiments on four data sets depicted in Section 6.1. All the results are shown in Fig. 5. We can be informed that MTrick gains a remarkable improvement over LR, LibSVM, CoCC, and MTrick0, especially, MTrick can reach the accuracy higher than 80% even LR and LibSVM have the random guess performance in Fig. 5(d). Also we can find that the performance of CoCC on three-class classification is consistent with the results when dealing with two-class classification. When the accuracy of LR is lower than 65%, CoCC seriously suffers from the distribution gap. All these results validate the effectiveness of MTrick to deal with multiclass scenarios.

We also investigate the average performance of 100 problem instances for each data set according to the evaluation metric method in Section 6.4.1, and the results are listed in Table 5. Again these results show MTrick is better than all comparison algorithms.

6.5. Analysis of the Output F_s and F_t

MTrick not only outputs the prediction results for target domain, but also generates the word clusters for the source and target domain data, expressed by $\boldsymbol{F}_s$ and $\boldsymbol{F}_t$, respectively. In other words, the words in source domain and target domain are all grouped into k_1 clusters after optimization. Following, we aim to show that the word clusters from the source and target domains are related to and different from each other. For each cluster we can select t (here $t = 20$) representative words, actually the t most probable words. Let $\mathcal{A}_i$ and $\mathcal{B}_i$ be the sets of representative words for the ith ($1 \le i \le k_1$) cluster in source domain and target domain, respectively, and $\mathcal{C}_i$ be the sets of representative words for the ith word cluster output by PLSA. Then, we define two measures as follows:

$$r_1 = \frac{1}{k_1}\sum_{i=1}^{k_1}\frac{|\mathcal{I}_i|}{|\mathcal{C}_i|}, \tag{29}$$

$$r_2 = \frac{1}{k_1}\sum_{i=1}^{k_1}\frac{|\mathcal{U}_i \bigcap \mathcal{C}_i|}{|\mathcal{C}_i|}, \tag{30}$$

[7] http://ews.uiuc.edu/~jinggao3/kdd08transfer.htm. Gao *et al.* [2] gives the detailed description.

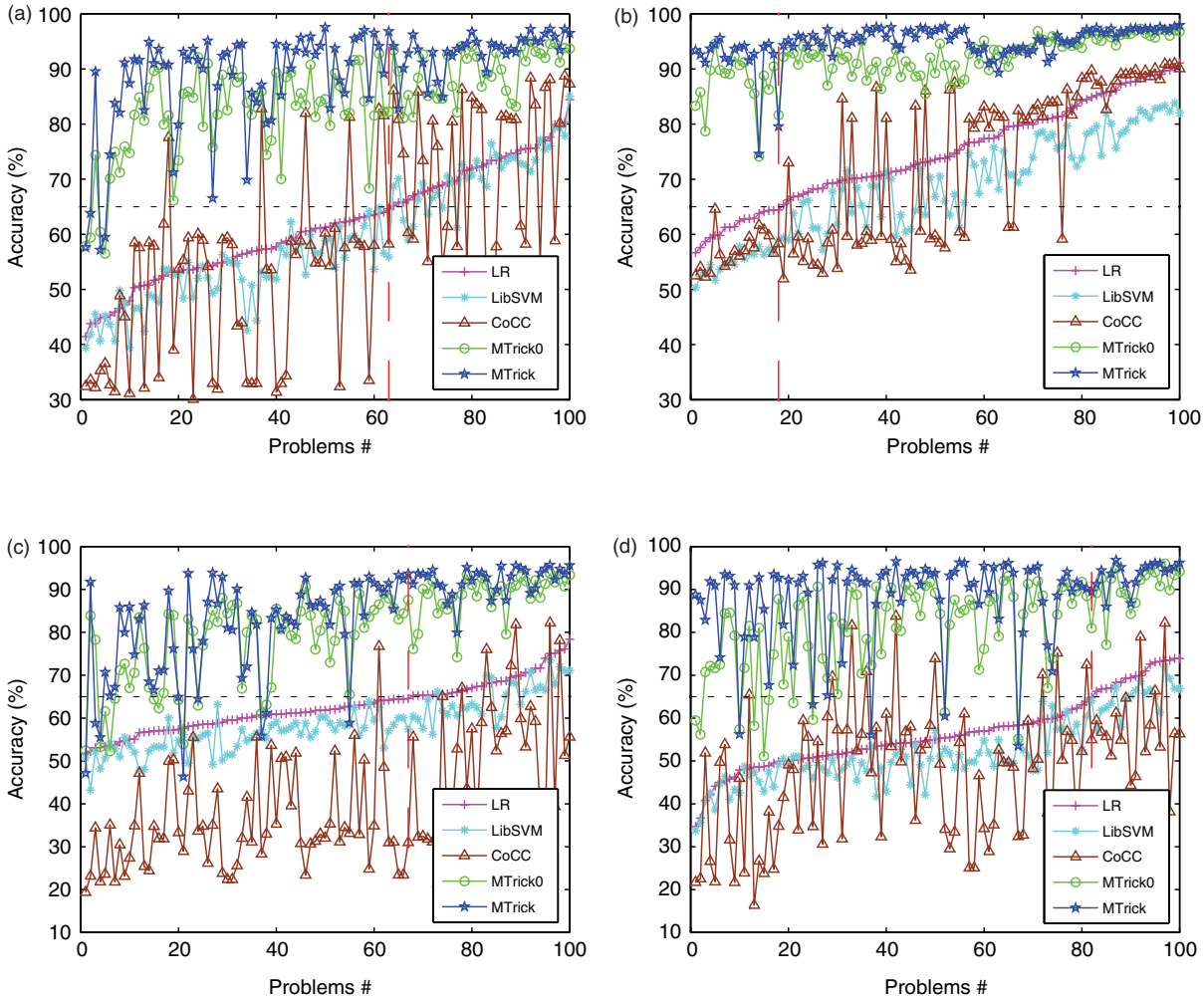

Fig. 5 The performance comparison among LR, LibSVM, MTrick0, and MTrick on four data set for three-class classification. (a) MTrick versus LR, LibSVM, MTrick0 on data set *comp versus rec versus sci*; (b) MTrick versus LR, LibSVM, MTrick0 on data set *comp versus rec versus talk*; (c) MTrick versus LR, LibSVM, MTrick0 on data set *comp versus sci versus talk*; (d) MTrick versus LR, LibSVM, MTrick0 on data set *rec versus sci versus talk*.

Table 5. Average performances (%) on 100 problem instances of each data set for three-class classification.

	comp versus rec versus sci			*comp versus rec versus talk*			*comp versus sci versus sci*			*rec versus sci versus talk*		
	L	*R*	Total	*L*	*R*	Total	*L*	*R*	Total	*L*	*R*	Total
LR	55.53	72.49	61.81	61.54	77.52	74.65	59.44	69.10	62.63	64.88	70.34	56.45
LibSVM	52.35	71.11	59.29	55.17	71.19	68.31	55.51	64.76	58.56	64.03	64.91	52.31
CoCC	50.53	72.46	58.65	56.99	74.47	71.33	35.12	52.32	40.80	46.30	57.58	48.33
MTrick0	81.20	89.53	84.28	87.14	92.91	91.87	76.95	88.64	80.81	81.00	90.78	81.83
MTrick	**87.31**	**93.22**	**89.50**	**91.41**	**95.51**	**94.77**	**80.12**	**92.12**	**84.08**	**89.48**	**93.58**	**88.60**

where $\mathcal{I}_i = \mathcal{A}_i \bigcap \mathcal{B}_i$ and $\mathcal{U}_i = \mathcal{A}_i \bigcup \mathcal{B}_i$. For each problem constructed from the data set *sci versus talk* we record these two values and the results are shown in Fig. 6. The curve of r_1 shows that although the word clusters from the source domain and target domain are different, they are related by sharing some representative words for word clusters. The curve of r_2 shows that the union of the word clusters from the source and target domains is similar to those output by PLSA based on the whole data. In other words the word clusters in the source and target domains not only exhibit their specific characteristics, but also share some general features. These results coincide with our analysis

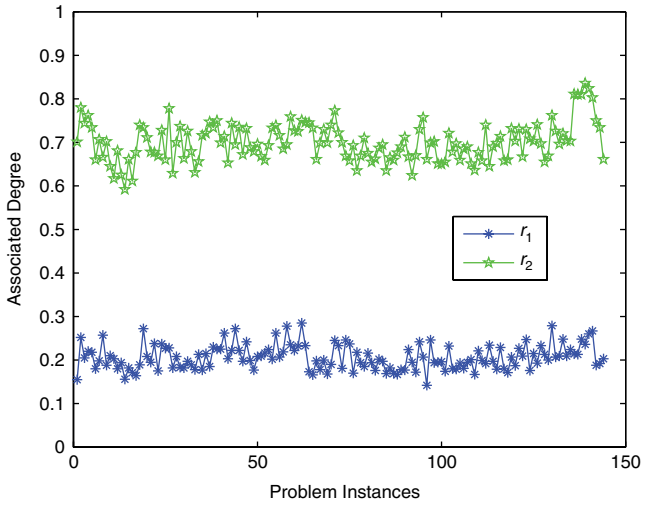

Fig. 6 The values of r_1 and r_2 on all the problems of data set *sci versus talk*.

that different data domains may use different terms in expressing the same concept; however, they are also closely related to each other.

6.6. Parameter Effect

In the problem formulation, we have three parameters, including two trade-off factors α, β and the number of word clusters k_1. Though the optimal combination of these parameters is hard to obtain, we can empirically show the performance of MTrick is not sensitive when the parameters are sampled in some value ranges. We bound the parameters $\alpha \in [1, 10]$, $\beta \in [0.5, 3]$ and $k_1 \in [10, 100]$ after some preliminary test and evaluate them on ten randomly selected problems of data set *sci versus talk*. Ten combinations of parameters are randomly sampled from the ranges, and the results of each problem on each parameter setting and their average performance are shown in Table 6. The 12th and 13th rows denote the variance and mean of ten parameter settings for each problem, respectively. The last row represents the performance using the default parameters adopted in this paper.

From Table 6, we can find that the average performance of all the parameter settings is almost the same with the results from the default parameters. Furthermore, the variance of all the parameter settings is small. It shows that the performance of MTrick is not sensitive to the parameters when they are sampled from the predefined bounds.

6.7. Algorithm Convergence

Here, we also empirically check the convergence property of the proposed iterative algorithm. For nine randomly-selected problems of *sci versus talk*, the results are shown in Fig. 7, where the x-axis represents the number of iterations, and the left and right y-axes denote the prediction accuracy and the logarithm of the objective value in Eq. (11), respectively. In each figure, it can be seen that the value of objective function decreases along with the iterating process, which agrees with the theoretic analysis.

7. CONCLUDING REMARKS

In this paper, we studied how to exploit the associations between word clusters and document clusters for cross-domain learning. Along this line, we proposed a MTrick which simultaneously deals with the two tri-factorizations for the source- and target-domain data. To capture the features in the conceptual level for classification, in MTrick, the associations between word clusters and document clusters remain the same in both source and target domains. Then, we developed an iterative algorithm for the proposed

Table 6. The parameter effect for performance (%) of algorithm MTrick on data set *sci versus talk*.

				Problem ID									
Sampling ID	α	β	k_1	1	2	3	4	5	6	7	8	9	10
1	2.44	2.39	58	92.34	94.28	95.37	88.47	94.99	92.43	95.24	92.04	91.69	95.32
2	7.45	1.69	83	93.05	94.35	97.00	88.47	95.28	92.69	94.91	91.76	92.33	95.30
3	6.92	0.96	38	95.92	94.70	97.33	90.90	95.01	89.32	94.47	90.45	89.99	95.63
4	2.67	1.65	15	94.39	95.53	96.02	90.53	95.42	92.59	94.55	90.92	90.02	95.28
5	5.61	2.45	72	91.58	95.07	94.79	87.83	95.34	93.17	94.99	91.24	91.75	95.28
6	3.63	2.32	32	93.59	94.12	94.98	89.98	95.57	92.90	94.49	91.83	91.24	95.09
7	2.30	1.57	21	92.72	94.46	96.47	89.77	94.84	92.43	94.49	91.34	91.46	96.23
8	7.53	0.72	52	95.80	94.12	97.52	91.13	95.40	89.55	94.35	89.71	90.12	95.47
9	1.88	1.50	26	95.57	94.14	96.90	90.70	95.71	92.69	94.93	91.53	90.08	95.08
10	7.95	1.18	92	94.54	95.02	97.51	89.75	95.28	92.18	94.55	91.38	92.28	95.70
Variance				2.351	0.236	1.089	1.370	0.073	1.897	0.085	0.496	0.913	0.119
Mean				93.95	94.58	96.39	89.75	95.28	92.00	94.70	91.22	91.10	95.44
This paper	1	1.5	50	93.77	94.42	94.99	90.33	95.05	93.12	95.96	93.84	90.90	95.66

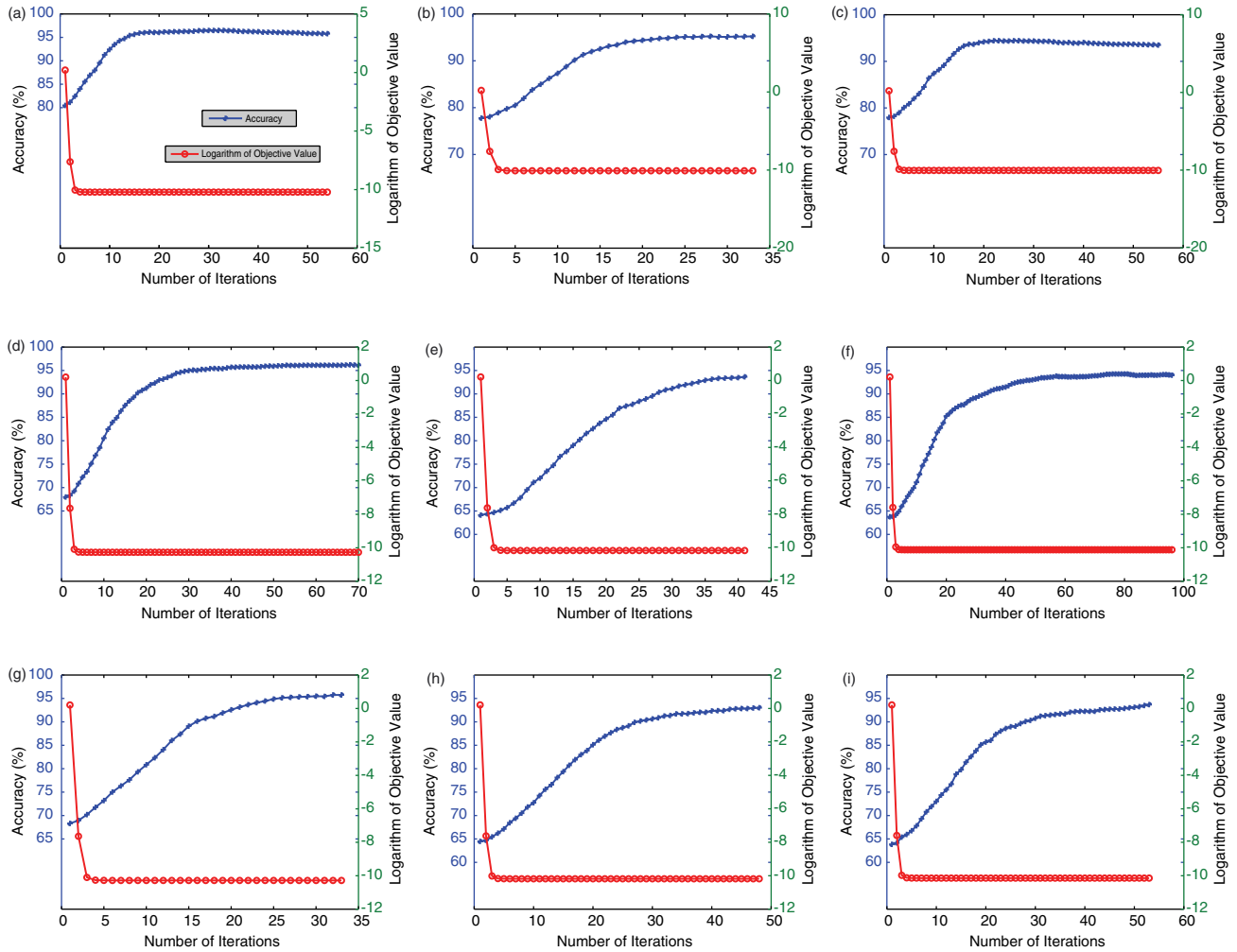

Fig. 7 Number of iterations versus the performance of MTrick and objective value. (a) Problem 1; (b) problem 2; (c) problem 3; (d) problem 4; (e) problem 5; (f) problem 6; (g) problem 7; (h) problem 8; (i) problem 9.

optimization problem, and also provided the theoretic analysis as well as some empirical evidences to show its convergence property. Finally, the experimental results show that MTrick can significantly improve the performance of cross-domain learning for text categorization. Note that, although MTrick was developed in the context of text categorization, it can be applied to more broad classification problems with dyadic data, such as the word-document matrix.

ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of China (Nos. 60933004, 60975039, 61035003, 60903141, 61072085), National Basic Research Priorities Programme (No. 2007CB311004), and National Science and Technology Support Plan (No. 2006BAC 08B06).

REFERENCES

[1] 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, Vancouver, British Columbia, Canada, 2008.

[2] 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, Las Vegas, Nevada, USA, (2008): 283–291.

[3] 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, Pairs, France, 2009.

[4] J. Jiang, Domain Adaptation in Natural Language Processing, PhD thesis, Computer Science in the Graduate College of the University of Illinois at Urbana-Champaign, 2008.

[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, San Jose, California, 2007, 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 Proceedings of the 17th ACM CIKM, Napa Valley, California, USA, 2008, 103–112.
[7] W. Y. Dai, Q. Yang, G. R. Xue, and Y. Yu, Boosting for transfer learning, In Proceedings of the 24th ICML, 2007, 193–200.
[8] T. Li, V. Sindhwani, C. Ding, and Y. Zhang, Knowledge transformation from for cross-domain sentiment classification, In Proceedings of the 32st SIGIR, Boston, Massachusetts, USA, 2009, 716–717.
[9] F. Z. Zhuang, P. Luo, H. Xiong, Y. H. Xiong, Q. He, and Z. Z. Shi, Cross-domain learning from multiple sources: a consensus regularization perspective, IEEE Trans Knowl Data Eng 22(12) (2010), 1664–1678.
[10] T. Li, C. Ding, Y. Zhang, and B. Shao, Knowledge transformation from word space to document space, In Proceedings of the 31st SIGIR, Singapore, 2008, 187–194.
[11] 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 SIAM SDM, 2010, 13–24.
[12] C. Ding, T. Li, W. Peng, and H. Park, Orthogonal nonnegative matrix tri-factorizations for clustering, Proceedings of the 12th ACM SIGKDD, Philadelphia, USA, ACM Press, 2006, 126–135.
[13] D. D. Lee and H. S. Seung, Algorithms for non-negative matrix factorization, In Proceedings of the 15th NIPS, Vancouver, British Columbia, Canada, 2001.
[14] D. Guillamet and J. Vitrià, Non-negative matrix factorization for face recognition, In Proceedings of the 5th CCAI, 2002, 336–344.
[15] D. Guillamet, J. Vitrià, and B. Schiele, Introducting a weighted non-negative matrix factrorization for image classification, Pattern Recognit Lett 24 (2003), 2447–2454.
[16] F. Sha, L. K. Saul, and D. D. Lee, Multiplicative updates for nonnegative quadratic programming in support vector machines, In Proceedings of the 17th NIPS, Vancouver, British Columbia, Canada, 2003, 1041–1048.
[17] F. Wang, T. Li, and C. S. Zhang, Semi-supervised clustering via matrix factorization, In Proceedings of the 8th SDM, 2008.
[18] B. Li, Q. Yang, and X. Y. Xue, Can movies and books collaborate? Cross-domain collaborative filtering for sparsity reduction, In Proceedings of the 21rd IJCAI, 2009, 2052–2057.
[19] T. Hofmann, Unsupervised learning by probabilistic latent semantic analysis, J Mach Learn 42(1/2) (2001), 177–196.
[20] D. M. Blei, A. Y. Ng, and M. I. Jordan, Latent Dirichlet allocation, J Mach Learn Res 3 (January, 2003), 993–1022.
[21] E. Gaussier and C. Goutte, Relation between plsa and nmf and implications, In Proceedings of the 28th SIGIR, Salvador, Brazil, 2005, 601–602.
[22] T. Li, V. Sindhwani, C. Ding, and Y. Zhang, Bridge domains with words: Opinion analysis with matrix tri-factorizations, In Proceedings of the 10th SIAM SDM, Columbus, Ohio, USA, 2010, 293–302.
[23] J. Jiang and C. X. Zhai, Instance weighting for domain adaptation in nlp, In Proceedings of the 45th ACL, 2007, 264–271.
[24] J. Jiang and C. X. Zhai, A two-stage approach to domain adaptation for statistical classifiers, In Proceedings of the 16th CIKM, 2007, 401–410.
[25] S. J. Pan, J. T. Kwok, and Q. Yang, Transfer learning via dimensionality reduction, In Proceedings of the 23rd AAAI, 2008, 677–682.
[26] S. Si, D. C. Tao, and K. P. Chan, Evolutionary cross-domain discriminative hessian eigenmaps, IEEE Trans Image Process 19(4) (2010), 1075–1086.
[27] S. Si, D. C. Tao, and B. Geng, Bregman divergence-based regularization for transfer subspace learning, IEEE Trans Knowl Data Eng 22(7) (2010), 929–942.
[28] D. Hosmer and S. Lemeshow, Applied Logistic Regression, New York, Wiley, 2000.
[29] C. C. Chang and C. J. Lin, LIBSVM: a library for support vector machines, 2001. Software available at http://www.csie.ntu.edu.tw/cjlin/libsvm.
[30] B. E. Boser, I. Guyou, and V. Vapnik, A training algorithm for optimal margin classifiers, In Proceedings of the 5th AWCLT, 1992.
[31] T. Joachims, Transductive inference for text classification using support vector machines, In Proceedings of the 16th ICML, 1999.