

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

Fuzhen Zhuang ^{†¶} Ping Luo [‡] Hui Xiong [§] Qing He [†] Yuhong Xiong [‡]
Zhongzhi Shi [†]

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 nonnegative 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.

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 re-collect the needed training data. It would be nice to reduce the need and effort to re-collect the training data. This leads to the research of *cross-domain learning** [1, 2, 3, 4, 5, 6, 7, 8]. 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, on the course-related pages the terms describing the concept of *reading materials* can be “required reading list”, “textbooks”, “reference” and so on. Since linguistic habits in expressing a concept are different in different domains, the phrases for the same concept may have different probabilities in different domains (universities in this example). Thus, features on raw terms are not reliable for text classification, especially in cross-domain learning. However, the concept behind the phrases may have the same effect to indicate the class labels of the documents from different domains. In this example, a page is more probable to be course-related if it contains the concept of *reading materials*. In other words, only concepts behind raw words are stable in indicating taxonomy, thus the association between word clusters and document classes is independent of data domains. Therefore, we can use it as bridge to transfer knowledge cross different domains.

Motivated by this observation, in this study, we explicitly consider the stable associations between concepts (expressed by word clusters) and document classes across data domains by the nonnegative matrix factorization. The basic formula of matrix tri-factorization is

[†]The Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, {zhuangfz, heq, shizz}@ics.ict.ac.cn.

[‡]Hewlett Packard Labs China, {ping.luo, Yuhong.Xiong}@hp.com.

[§]MSIS Department, Rutgers University, hxiong@rutgers.edu.

[¶]Graduate University of Chinese Academy of Sciences.

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

as follows,

$$(1.1) \quad X_{m \times n} = F_{m \times k_1} S_{k_1 \times k_2} G_{n \times k_2}^T,$$

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

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

2 Preliminaries and Problem Formulation

In this section, we first introduce some basic concepts and mathematical notations used throughout this paper, and then formulate the matrix tri-factorization based classification framework.

2.1 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 upper case, such as X and Y . Also, $X_{(ij)}$ indicates the i -th row and j -th column element of matrix 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 nonnegative real numbers respectively.

DEFINITION 1. (TRACE OF MATRIX) Given a data matrix $X \in \mathbb{R}^{n \times n}$, the trace of X is computed as

$$(2.2) \quad \text{Tr}(X) = \sum_{i=1}^n X_{(ii)}.$$

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 $\text{Tr}(X) = \sum_{i=1}^m X_{(ii)}$.

DEFINITION 2. (FROBENIUS NORM OF MATRIX) Given a data matrix $X \in \mathbb{R}^{m \times n}$, the frobenius norm of X is computed as

$$(2.3) \quad \|X\|^2 = \sum_{i=1}^m \sum_{j=1}^n X_{(ij)}^2.$$

Additionally, we give some properties of the trace and frobenius norm, which will be used in Section 3 and 4.

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

$$(2.4) \quad \text{Tr}(X^T X) = \text{Tr}(X X^T).$$

PROPERTY 2. Given matrixes $X, Y \in \mathbb{R}^{m \times n}$, then

$$(2.5) \quad \text{Tr}(a \cdot X + b \cdot Y) = a \cdot \text{Tr}(X) + b \cdot \text{Tr}(Y).$$

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

$$(2.6) \quad \|X\|^2 = \text{Tr}(X^T X) = \text{Tr}(X X^T).$$

2.2 Problem Formulation

For the joint probability matrix $X_s \in \mathbb{R}_+^{m \times n_s}$ 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,

$$(2.7) \quad \begin{aligned} & \min_{F_s, S_s, G_s} \|X_s - F_s S_s G_s^T\|^2 + \frac{\alpha}{n_s} \|G_s - G_0\|^2, \\ & \text{s.t.} \quad \sum_{j=1}^{k_1} F_{s(ij)} = 1, \sum_{j=1}^{k_2} G_{s(ij)} = 1, \end{aligned}$$

where α is the trade-off parameter, G_0 contains the true label information in the source-domain. Specifically, when the i -th instance belongs to class j , then $G_{0(ij)} = 1$; and $G_{0(ik)} = 0$ for $k \neq j$. In this formulation G_0 is used as the supervised information by requiring G_s is similar to G_0 . After minimizing Equation (2.7) we obtain F_s, G_s, S_s , where

• $F_s \in \mathbb{R}_+^{m \times k_1}$ represents the information of word clusters, and $F_{s(ij)}$ is the probability that the i -th word belongs to the j -th word cluster.

- $G_s \in \mathbb{R}_+^{n_s \times k_2}$ represents the information of document clusters, and $G_{s(ij)}$ is the probability that the i -th document belongs to the j -th document cluster.
- $S_s \in \mathbb{R}_+^{k_1 \times k_2}$ represents the associations between word clusters and document clusters.

Then, for the joint probability matrix $X_t \in \mathbb{R}_+^{m \times n_t}$ 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,

$$(2.8) \quad \min_{F_t, G_t} \|X_t - F_t S_0 G_t^T\|^2$$

$$s.t. \quad \sum_{j=1}^{k_1} F_{t(ij)} = 1, \sum_{j=1}^{k_2} G_{t(ij)} = 1,$$

where S_0 is the output from Equation (2.7). In this formulation 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 Equation (2.8) we obtain F_t, G_t . Their explanations are similar to those for F_s, G_s respectively. Then, the class label of the i -th document in the target domain is output as

$$(2.9) \quad index_i = \arg \max_j G_{t(ij)}.$$

Finally, we can combine the two sequential optimization problems in Equation (2.7) and (2.8) into a joint optimization formulation as follows,

$$(2.10) \quad \min_{F_s, G_s, S, F_t, G_t} \|X_s - F_s S G_s^T\|^2 + \frac{\alpha}{n_s} \cdot \|G_s - G_0\|^2$$

$$+ \beta \cdot \|X_t - F_t S G_t^T\|^2,$$

$$s.t. \quad \sum_{j=1}^{k_1} F_{s(ij)} = 1, \sum_{j=1}^{k_2} G_{s(ij)} = 1,$$

$$\sum_{j=1}^{k_1} F_{t(ij)} = 1, \sum_{j=1}^{k_2} G_{t(ij)} = 1,$$

where $\alpha \geq 0$ and $\beta \geq 0$ are the trade-off factors. In this formulation S is shared in the matrix factorizations of the source and target domains. This way 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 Equation (2.10), which can cover both the two sub-problems in Equation (2.7) and (2.8).

3 Solution to the Optimization Problem

In this section, we develop an alternately iterative algorithm to solve the problem in Equation (2.10).

According to the preliminary knowledge in Section 2.1, we know that the minimization of Equation (2.10) is equivalent to minimizing the following equation,

$$(3.11) \quad \mathcal{L}(F_s, G_s, S, F_t, G_t)$$

$$= Tr(X_s^T X_s - 2X_s^T F_s S G_s^T + G_s S^T F_s^T F_s S G_s^T)$$

$$+ \frac{\alpha}{n_s} \cdot Tr(G_s G_s^T - 2G_s G_0^T + G_0 G_0^T)$$

$$+ \beta \cdot Tr(X_t^T X_t - 2X_t^T F_t S G_t^T + G_t S^T F_t^T F_t S G_t^T),$$

$$s.t. \quad \sum_{j=1}^{k_1} F_{s(ij)} = 1, \sum_{j=1}^{k_2} G_{s(ij)} = 1,$$

$$\sum_{j=1}^{k_1} F_{t(ij)} = 1, \sum_{j=1}^{k_2} G_{t(ij)} = 1.$$

The partial differential of \mathcal{L} is as follows,

$$\frac{\partial \mathcal{L}}{\partial F_s} = -2X_s G_s S^T + 2F_s S G_s^T G_s S^T,$$

$$\frac{\partial \mathcal{L}}{\partial G_s} = -2X_s^T F_s S + 2G_s S^T F_s^T F_s S$$

$$+ \frac{2\alpha}{n_s} \cdot (G_s - G_0),$$

$$\frac{\partial \mathcal{L}}{\partial S} = -2F_s^T X_s G_s + 2F_s^T F_s S G_s^T G_s$$

$$- 2\beta \cdot F_t^T X_t G_t + 2\beta \cdot F_t^T F_t S G_t^T G_t,$$

$$\frac{\partial \mathcal{L}}{\partial F_t} = -2\beta \cdot X_t G_t S^T + 2\beta \cdot F_t S G_t^T G_t S^T,$$

$$\frac{\partial \mathcal{L}}{\partial G_t} = -2\beta \cdot X_t^T F_t S + 2\beta \cdot G_t S^T F_t^T F_t S.$$

Since \mathcal{L} is not concave, it is hard to obtain the global solution by applying the latest non-linear 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

$$(3.12) \quad F_{s(ij)} \leftarrow F_{s(ij)} \cdot \sqrt{\frac{(X_s G_s S^T)_{(ij)}}{(F_s S G_s^T G_s S^T)_{(ij)}}},$$

$$(3.13) \quad G_{s(ij)} \leftarrow G_{s(ij)} \cdot \sqrt{\frac{(X_s^T F_s S + \frac{\alpha}{n_s} \cdot G_0)_{(ij)}}{(G_s S^T F_s^T F_s S + \frac{\alpha}{n_s} \cdot G_s)_{(ij)}}},$$

$$(3.14) \quad F_{t(ij)} \leftarrow F_{t(ij)} \cdot \sqrt{\frac{(X_t G_t S^T)_{(ij)}}{(F_t S G_t^T G_t S^T)_{(ij)}}},$$

$$(3.15) \quad G_{t(ij)} \leftarrow G_{t(ij)} \cdot \sqrt{\frac{(X_t^T F_t S)_{(ij)}}{(G_t S^T F_t^T F_t S)_{(ij)}}},$$

Then, we normalize F_s, G_s, F_t, G_t to satisfy the equality constrains. The normalization formulas are as follows,

$$(3.16) \quad F_{s(i)} \leftarrow \frac{F_{s(i)}}{\sum_{j=1}^{k_1} F_{s(ij)}},$$

$$(3.17) \quad G_{s(i)} \leftarrow \frac{G_{s(i)}}{\sum_{j=1}^{k_2} G_{s(ij)}},$$

$$(3.18) \quad F_{t(i)} \leftarrow \frac{F_{t(i)}}{\sum_{j=1}^{k_1} F_{t(ij)}},$$

$$(3.19) \quad G_{t(i)} \leftarrow \frac{G_{t(i)}}{\sum_{j=1}^{k_2} G_{t(ij)}}.$$

Next, using the normalized F_s, G_s, F_t, G_t we update S as follows,

$$(3.20) \quad S_{(ij)} \leftarrow S_{(ij)} \cdot \sqrt{\frac{(F_s^T X_s G_s + \beta \cdot F_t^T X_t G_t)_{(ij)}}{(F_s^T F_s S G_s^T G_s + \beta \cdot F_t^T F_t S G_t^T G_t)_{(ij)}}}.$$

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

4 Analysis of Algorithm Convergence

To investigate the convergence of iterating rules in Equations (3.12) through (3.20), we first check the convergence of F_s when G_s, S, F_t, G_s are fixed. For this optimization problem with constraints we formulate the following Lagrangian function,

$$(4.21) \quad \mathcal{G}(F_s) = \|X_s - F_s S G_s^T\|^2 + \text{Tr}[\lambda(F_s \mathbf{u}^T - \mathbf{v}^T)(F_s \mathbf{u}^T - \mathbf{v}^T)^T],$$

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 $\|X_s - F_s S G_s^T\|^2 = \text{Tr}(X_s^T X_s - 2X_s^T F_s S G_s^T + G_s S^T F_s^T F_s S G_s^T)$. Then,

$$(4.22) \quad \frac{\partial \mathcal{G}}{\partial F_s} = -2X_s G_s S^T + 2F_s S G_s^T G_s S^T + 2\lambda F_s \mathbf{u}^T \mathbf{u} - 2\lambda \mathbf{v}^T \mathbf{u}.$$

LEMMA 1. *Using the update rule (4.23), Equation (4.21) will monotonously decrease.*

$$(4.23) \quad F_{s(ij)} \leftarrow F_{s(ij)} \cdot \sqrt{\frac{(X_s G_s S^T + \lambda \mathbf{v}^T \mathbf{u})_{(ij)}}{(F_s S G_s^T G_s S^T + \lambda F_s \mathbf{u}^T \mathbf{u})_{(ij)}}}.$$

Algorithm 1 The Matrix Tri-factorization based Classification (MTrick) Algorithm

Input: The joint probability matrix $X_s \in \mathbb{R}_+^{m \times n_s}$ on labeled source-domain; the true label information G_0 of source-domain; the joint probability matrix $X_t \in \mathbb{R}_+^{m \times n_t}$ on unlabeled target-domain; and the trade-off factors α, β ; the error threshold $\varepsilon > 0$; and the maximal iterating number max .

Output: F_s, F_t, G_s, G_t and S .

1. Initialize the matrix variables as $F_s^{(0)}, F_t^{(0)}, G_s^{(0)}, G_t^{(0)}$ and $S^{(0)}$. The initialization method will be detailed in the experimental section.
 2. Calculate the initial value $\mathcal{L}^{(0)}$ of Equation (3.11).
 3. $k := 1$.
 4. Update $F_s^{(k)}$ based on Equation (3.12), and normalize $F_s^{(k)}$ based on Equation (3.16).
 5. Update $G_s^{(k)}$ based on Equation (3.13), and normalize $G_s^{(k)}$ based on Equation (3.17).
 6. Update $F_t^{(k)}$ based on Equation (3.14), and normalize $F_t^{(k)}$ based on Equation (3.18).
 7. Update $G_t^{(k)}$ based on Equation (3.15), and normalize $G_t^{(k)}$ based on Equation (3.19).
 8. Update $S^{(k)}$ based on Equation (3.20).
 9. Calculate the value $\mathcal{L}^{(k)}$ of Equation (3.11). If $|\mathcal{L}^{(k)} - \mathcal{L}^{(k-1)}| < \varepsilon$, then turn to Step 11.
 10. $k := k + 1$. If $k \leq max$, then turn to Step 4.
 11. Output $F_s^{(k)}, F_t^{(k)}, G_s^{(k)}, G_t^{(k)}$ and $S^{(k)}$.
-

Proof. To prove Lemma 1 we describe the definition of auxiliary function [9] as follows.

DEFINITION 3. (AUXILIARY FUNCTION) A function $H(Y, \tilde{Y})$ is called an auxiliary function of $\mathcal{T}(Y)$ if it satisfies

$$(4.24) \quad H(Y, \tilde{Y}) \geq \mathcal{T}(Y), H(Y, Y) = \mathcal{T}(Y),$$

for any Y, \tilde{Y} .

Then, define

$$(4.25) \quad Y^{(t+1)} = \arg \min_Y H(Y, Y^{(t)}).$$

Through this definition,

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

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

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

$$(4.26) \quad \begin{aligned} H(F_s, F'_s) = & -2 \sum_{ij} (X_s G_s S^T)_{(ij)} F'_{s(ij)} \left(1 + \log \frac{F_{s(ij)}}{F'_{s(ij)}}\right) \\ & + \sum_{ij} (F'_s S G_s^T G_s S^T)_{(ij)} \frac{F_{s(ij)}^2}{F'_{s(ij)}} \\ & + \sum_{ij} (\lambda F'_s \mathbf{u}^T \mathbf{u})_{(ij)} \frac{F_{s(ij)}^2}{F'_{s(ij)}} \\ & - 2 \sum_{ij} (\lambda \mathbf{v}^T \mathbf{u})_{(ij)} F'_{s(ij)} \left(1 + \log \frac{F_{s(ij)}}{F'_{s(ij)}}\right). \end{aligned}$$

Obviously, when $F'_s = F_s$ the equality $H(F_s, F'_s) = \mathcal{G}(F_s)$ holds. Also we can prove the inequality $H(F_s, F'_s) \geq \mathcal{G}(F_s)$ holds using the similar proof approach in [10]. Then, while fixing F'_s , we minimize $H(F_s, F'_s)$.

$$(4.27) \quad \begin{aligned} \frac{\partial H(F_s, F'_s)}{\partial F_{s(ij)}} = & -2 (X_s G_s S^T)_{(ij)} \frac{F'_{s(ij)}}{F_{s(ij)}} \\ & + 2 (F'_s S G_s^T G_s S^T + \lambda F'_s \mathbf{u}^T \mathbf{u})_{(ij)} \frac{F_{s(ij)}}{F'_{s(ij)}} \\ & - 2 (\lambda \mathbf{v}^T \mathbf{u})_{(ij)} \frac{F'_{s(ij)}}{F_{s(ij)}}. \end{aligned}$$

$$(4.28) \quad \text{Let } \frac{\partial H(F_s, F'_s)}{\partial F_{s(ij)}} = \mathbf{0},$$

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

Thus, the update rule (4.23) 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 F_s is 1. Here we propose a simple normalization technique to satisfy the constrains regardless of λ . Specifically, in each iteration we use Equation (3.16) to normalize F_s . After normalization, the two constants of $\lambda F'_s \mathbf{u}^T \mathbf{u}$ and $\lambda \mathbf{v}^T \mathbf{u}$ are equal. Thus, the effect of Equation (3.12) and Equation (3.16) can be approximately equivalent to Equation (4.23) when only considering the convergence. In other words, we adopt the approximate update rule of Equation (3.12) by omitting the items which depend

on λ in Equation (4.23). We can use the similar method to analyze the convergence of the update rules for G_s, F_t, G_t, S in Equation (3.13), (3.14), (3.15), (3.20) respectively.

THEOREM 1. (CONVERGENCE) *After each round of iteration in Algorithm 1 the objective function in Equation (2.10) will not increase.*

According to the lemmas for the convergence analysis on the update rules for F_s, G_s, F_t, G_t, S , and the Multiplicative Update Rules [9], each update step in Algorithm 1 will not increase Equation (2.10) and the objective has a lower bounded by zero, which guarantee the convergence. Thus, the above theorem holds.

5 Experimental Validation

In this section, we show experiments to validate the effectiveness of the proposed algorithm. To simplify the discussion, we only focus on the binary classification problems (the number of document clusters is two) in the experiments, while the algorithm can be naturally applied for multi-class cases.

5.1 Data Preparation

20Newsgroup[†] 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*, the top category *talk* contains four subcategories *talk.politics.guns*, *talk.politics.mideast*, *talk.politics.misc* and *talk.religion.misc*, and the top category *rec* also contains four subcategories *rec.autos*, *rec.motorcycles*, *rec.sport.baseball* and *rec.sport.hockey*. For the top categories *sci*, *talk* and *rec*, any two top categories can be selected to construct 2-class classification problems. In the experimental setting, we only randomly select two data sets *sci vs. talk* and *rec vs. sci*.

For the data set *sci vs. 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

[†]<http://people.csail.mit.edu/jrennie/20Newsgroups/>

is suitable for cross-domain learning due to the facts that 1) the training and test data are from different distributions since they are from different subcategories; 2) 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 vs. talk*, we totally construct 144 ($P_4^2 \cdot P_4^2$) classification tasks. The data set *rec vs. sci* is constructed similarly with *sci vs. talk*.

To further validate our algorithm, we also perform experiments on the data set *Reuters-21578*[‡], 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].

5.2 Baseline Methods and Evaluation Metric

We compare MTrick with some baseline classification methods, including the supervised algorithms of Logistic Regression (LG) [11] and Support Vector Machine (SVM) [12], and the semi-supervised algorithm of Transductive Support Vector Machine (TSVM) [13], also the cross-domain methods of Co-clustering based Classification (CoCC) [5] and the Local Weighted Ensemble (LWE) [2]. Additionally, the two-step optimization approach using Equation (2.7) and (2.8) is adopted as baseline (denoted as MTrick0). The prediction accuracy on the unlabeled target-domain data is the evaluation metric.

5.3 Implementation Details

In MTrick, F_s, F_t, G_s, G_t, S are initialized as follows,

1. F_s and F_t are initialized as the word clustering results by PLSA [14]. Specifically, $F_{s(ij)}$ and $F_{t(ij)}$ 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[§] in the experiments.
2. G_s is initialized as the true class information in the source-domain.
3. G_t is initialized as the predicted results of any supervised classifier, which is trained based on the source-domain data. In this experiment Logistic Regression is adopted to give these initial results.
4. S is initialized as follows: each entry is assigned with the same value and the sum of values in each row satisfies $\sum_j S_{(ij)} = 1$.

[‡]<http://www.daviddlewis.com/resources/testcollections/reuters21578/>

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

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 Y , which is then transformed to the joint probability matrix 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 LG is implemented by the package[¶], SVM and TSVM are given by SVM^{light}^{||}. The parameter settings of CoCC and LWE are the same with those in their original papers, and the value of α in Equation (2.7) is set to 1 after careful investigation for MTrick0.

5.4 Experimental Results

Next, we present detailed experimental results.

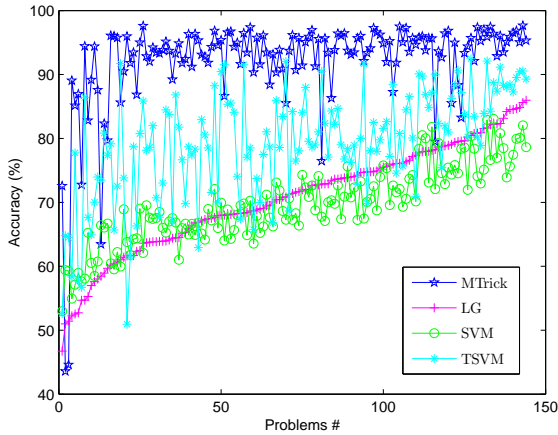
5.4.1 A Comparison of LR, SVM, TSVM, CoCC, MTrick0 and MTrick

We compare these classification approaches on the data set *sci vs. talk* and *rec vs. sci*, and all the results are recorded in Figure 1 and Figure 2. The 144 problems of each data set are sorted by increasing order of the performance of LG. Thus, the x -axis in these two figures can also indicate the degree of difficulty in knowledge transferring.

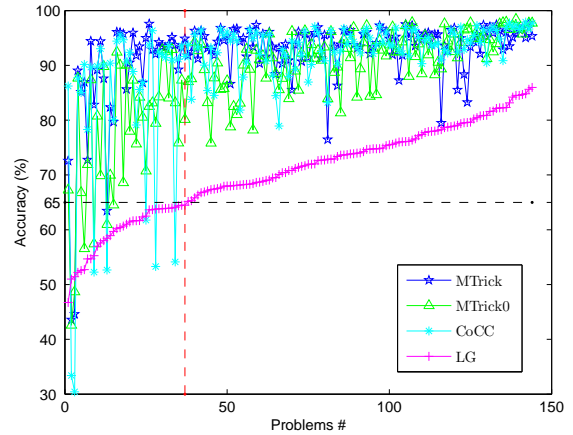
From the results, we have the following observations: 1) Figure 1(a) and Figure 2(a) show that MTrick is significantly better than the supervised learning algorithms LG and SVM, which indicates that the traditional supervised learning approaches can not perform well on the cross-domain learning tasks. 2) Also, MTrick is also much better than the semi-supervised method of TSVM. 3) In Figure 1(b) and Figure 2(b), the left side of red-dotted line represents the results when the accuracy of LG lower than 65%, while the right represents the results when the accuracy of LG higher than 65%. It is shown that when LG achieves accuracy higher than this threshold, MTrick and CoCC perform similarly. However, when the accuracy of LG is lower than it, MTrick performs much better than CoCC. These results indicate that MTrick has the stronger ability to transfer knowledge when the labeled source domain can not provide enough auxiliary information. 4) MTrick is also better than MTrick0, which shows that the joint optimization can achieve a better solution than the sep-

[¶]<http://research.microsoft.com/~minka/papers/logreg/>

^{||}<http://svmlight.joachims.org/>



(a) MTrick vs. LG, SVM, TSVM on data set *sci vs. talk*



(b) MTrick vs. MTrick0, CoCC on data set *sci vs. talk*

Figure 1: The Performance Comparison among LR, SVM, TSVM, CoCC, MTrick0 and MTrick on data set *sci vs. talk*

arate 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 1 (L and R denote the average results when the accuracy of LG lower and higher than 65%, respectively, while $Total$ represents the average results on all 144 problems). These results again show that MTrick is an effective approach for cross-domain learning, and has stronger ability to transfer knowledge.

5.4.2 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** are depicted in Table 2. The experimental results are recorded in Table 3 (We adopt the evaluation results of TSVM and LWE on the three problems in [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.

Table 2: 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 vs. people	document from	document from a
orgs vs. place	a set of	different set
people vs. place	sub-categories	of sub-categories

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

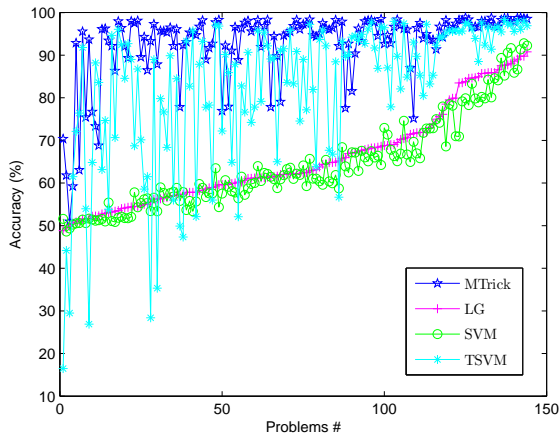
5.4.3 Analysis of the Output F_s and F_t

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 F_s and F_t respectively. In other words, the words in source domain and target domain are all grouped into k_1 clusters after optimization. By the following calculating 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 i -th ($1 \leq i \leq k_1$) cluster in source domain and target domain respectively, and \mathcal{C}_i be the sets of representative words for the i -th word cluster output by PLSA. Then, we define two measures as follows,

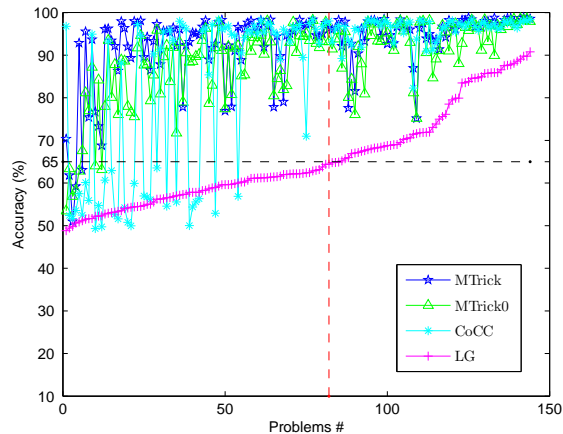
$$(5.29) \quad r_1 = \frac{1}{k_1} \sum_{i=1}^{k_1} \frac{|\mathcal{I}_i|}{|\mathcal{C}_i|},$$

$$(5.30) \quad r_2 = \frac{1}{k_1} \sum_{i=1}^{k_1} \frac{|\mathcal{U}_i \cap \mathcal{C}_i|}{|\mathcal{C}_i|},$$

where $\mathcal{I}_i = \mathcal{A}_i \cap \mathcal{B}_i$ and $\mathcal{U}_i = \mathcal{A}_i \cup \mathcal{B}_i$. For each problem constructed from the data set *sci vs. talk* we record these two values and the results are shown in Figure 3. 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



(a) MTrick vs. LG, SVM, TSVM on data set *rec vs. sci*



(b) MTrick vs. MTrick0, CoCC on data set *rec vs. sci*

Figure 2: The Performance Comparison among LR, SVM, TSVM, CoCC, MTrick0 and MTrick on data set *rec vs. sci*

Table 1: Average Performances (%) on 144 Problem Instances of Each Data Set

Data Sets		LG	SVM	TSVM	CoCC	MTrick0	MTrick
<i>sci vs. talk</i>	<i>L</i>	59.09	62.88	72.13	81.09	76.90	86.52
	<i>R</i>	74.21	71.70	81.58	93.41	91.28	93.71
	<i>Total</i>	70.64	69.62	79.35	90.50	87.88	92.01
<i>rec vs. sci</i>	<i>L</i>	57.42	56.78	75.73	79.69	85.39	90.44
	<i>R</i>	75.76	73.48	91.66	96.18	93.50	95.53
	<i>Total</i>	65.57	64.20	82.81	87.02	88.99	92.70

Table 3: The Performance Comparison Results (%) among LG, SVM, TSVM, CoCC, LWE and MTrick

Data Sets	LG	SVM	TSVM	CoCC	LWE	MTrick
orgs vs. people	74.92	74.25	73.80	79.79	79.67	80.80
orgs vs. place	71.91	69.99	69.89	74.18	73.04	76.77
people vs. place	58.03	59.05	58.43	66.94	68.52	69.02

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

5.4.4 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 demonstrate

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 10 randomly selected problems of data set *sci vs. talk*. 10 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 4. The 12th and 13th row denote the variance and mean of 10 parameter settings for each problem, respectively. The last row represents the performance using the default parameters adopted in this paper.

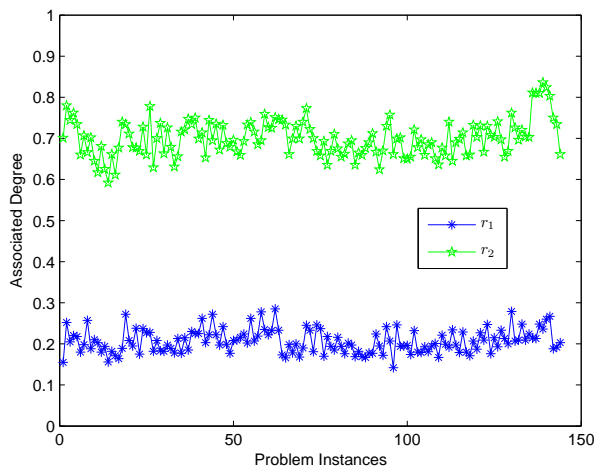


Figure 3: The Values of r_1 and r_2 on All the Problems of Data Set *sci vs. talk*.

From Table 4, 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.

5.4.5 Algorithm Convergence

Here, we also empirically check the convergence property of the proposed iterative algorithm. For 9 randomly-selected problems of *sci vs. talk* the results are shown in Figure 4, where the x -axis represents the number of iterations, and the left and right y -axis denote the prediction accuracy and the logarithm of the objective value in Equation (3.11), 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.

6 Related Work

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

6.1 Nonnegative Matrix Factorization

Since our algorithm framework is based on the nonnegative matrix factorization, so here we will introduce some works about nonnegative matrix factorization (NMF). NMF has been shown to be widely used for many applications, such as dimensionality reduction, pattern recognition, clustering and classifica-

tion [10, 9, 15, 16, 17, 18, 19, 20] etc.

Lee et al. [9] proposed the nonnegative matrix factorization 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. Guillaumet et al. [16] extended the NMF to a weighted nonnegative 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. [10] provided an analysis of the relationship between 2-facts and 3-facts NMF, and proposed an orthogonal nonnegative matrix tri-factorization for clustering. They empirically showed that the bi-orthogonal nonnegative matrix tri-factorization based approach can simultaneously cluster rows and columns of the input data matrix effectively. Wang et al. [18] developed a novel matrix factorization based approach for semi-supervised clustering and extended it to different kinds of constrained co-clusterings. The probabilistic topic models, such as PLSA [14] and LDA [21], can also be considered as a method of nonnegative matrix tri-factorization [22]. 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* [14]. 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] proposed to transfer label information from source domain to target domain by sharing the information of word clusters for the task of sentiment classification. However, for a general cross-domain classification problem the two corresponding 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. [20] 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

Table 4: The Parameter Effect for Performance (%) of Algorithm MTrick

Sampling ID	α	β	k_1	Problem ID									
				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

value of this collaborative optimization by the experimental results.

6.2 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 re-weighted 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 [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 NLP tasks.

Feature selection based approaches aim to find a common feature space which is useful to cross-domain learning. Jiang [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. 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. The proposed algorithm in this

paper can also be regarded as the feature selection based approach for cross-domain learning.

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 matrix tri-factorization based classification framework (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 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.

8 Acknowledgments

This work is supported by the National Science Foundation of China (No.60675010, 60933004, 60975039), 863 National High-Tech Program (No.2007AA01Z132), National Basic Research Priorities Programme (No.2007CB311004) and National Science and Technology Support Plan (No.2006BAC08B06).

References

- [1] W. Y. Dai, Y. Q. Chen, G. R. Xue, Q. Yang, and Y. Yu. Translated learning: Transfer learning across

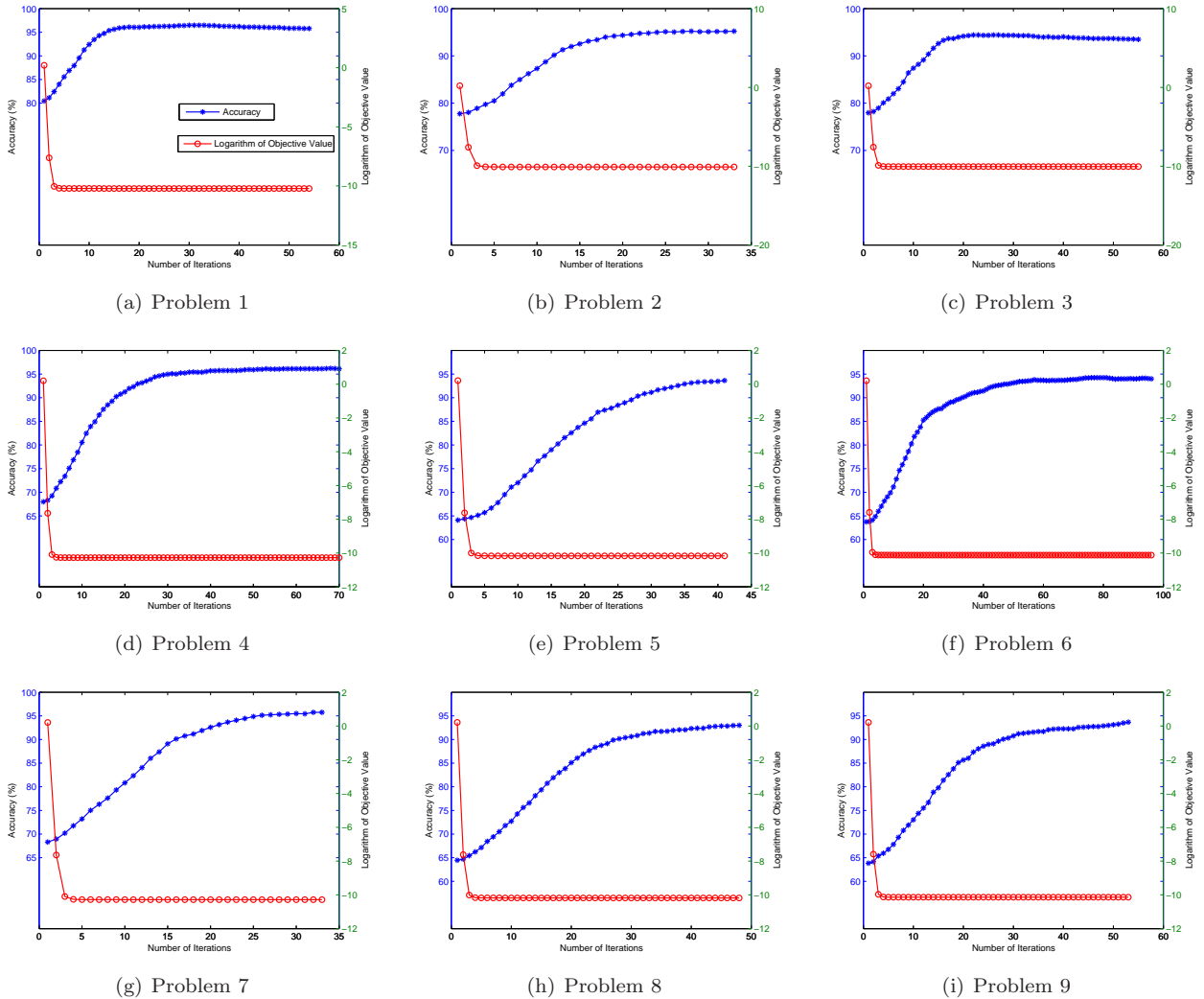


Figure 4: Number of Iterations vs. the Performance of MTrick and Objective Value.

- 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*, pages 283–291, 2008.
- [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*, 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, Napa Valley, California, USA*, pages 103–112, 2008.
- [7] 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.
- [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*, pages 716–717, 2009.
- [9] D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *Proceedings of the 15th NIPS, Vancouver, British Columbia, Canada, 2001*.
- [10] C. Ding, T. Li, W. Peng, and H. Park. Orthogonal nonnegative matrix tri-factorizations for clustering. In

- Proceedings of the 12th ACM SIGKDD, Philadelphia, USA, ACM Press*, pages 126–135, 2006.
- [11] David Hosmer and Stanley Lemeshow. *Applied Logistic Regression*. Wiley, New York, 2000.
- [12] B. E. Boser, I. Guyou, and V. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the 5th AWCLT*, 1992.
- [13] T. Joachims. Transductive inference for text classification using support vector machines. In *Proceedings of the 16th ICML*, 1999.
- [14] T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Journal of Machine Learning*, pages 177–196, 2001.
- [15] D. Guillamet and J. Vitrià. Non-negative matrix factorization for face recognition. In *Proceedings of the 5th CCAI*, pages 336–344, 2002.
- [16] D. Guillamet, J. Vitrià, and B. Schiele. Introducing a weighted non-negative matrix factorization for image classification. *Pattern Recognition Letters*, 24:2447–2454, 2003.
- [17] 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*, pages 1041–1048, 2003.
- [18] F. Wang, T. Li, and C. S. Zhang. Semi-supervised clustering via matrix factorization. In *Proceedings of the 8th SDM*, 2008.
- [19] T. Li, C. Ding, Y. Zhang, and B. Shao. Knowledge transformation from word space to document space. In *Proceedings of the 31st SIGIR, Singapore*, pages 187–194, 2008.
- [20] B. Li, Q. Yang, and X. Y. Xue. Can movies and books collaborate? cross-domain collaborative filtering for sparsity reduction. In *Proceedings of the 21st IJCAI*, pages 2052–2057, 2009.
- [21] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, pages 993–1022, 2003.
- [22] E. Gaussier and C. Goutte. Relation between pls and nmf and implications. In *Proceedings of the 28th SIGIR, Salvador, Brazil*, pages 601–602, 2005.
- [23] J. Jiang and C. X. Zhai. Instance weighting for domain adaptation in nlp. In *Proceedings of the 45th ACL*, pages 264–271, 2007.
- [24] J. Jiang and C. X. Zhai. A two-stage approach to domain adaptation for statistical classifiers. In *Proceedings of the 16th CIKM*, pages 401–410, 2007.
- [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.