Mining Distinction and Commonality across Multiple Domains Using Generative Model for Text Classification

Fuzhen Zhuang, Ping Luo, *Member*, *IEEE Computer Society*, Zhiyong Shen, Qing He, Yuhong Xiong, Zhongzhi Shi, *Senior Member*, *IEEE*, and Hui Xiong, *Senior Member*, *IEEE*

Abstract—The distribution difference among multiple domains has been exploited for cross-domain text categorization in recent years. Along this line, we show two new observations in this study. First, the data distribution difference is often due to the fact that different domains use different index words to express the same concept. Second, the association between the conceptual feature and the document class can be stable across domains. These two observations actually indicate the *distinction* and *commonality* across domains. Inspired by the above observations, we propose a generative statistical model, named Collaborative Dual-PLSA (CD-PLSA), to simultaneously capture both the domain distinction and commonality among multiple domains. Different from Probabilistic Latent Semantic Analysis (PLSA) with only one latent variable, the proposed model has two latent factors y and z, corresponding to word concept and document class, respectively. The shared *commonality* intertwines with the *distinctions* over multiple domains, and is also used as the bridge for knowledge transformation. An Expectation Maximization (EM) algorithm is developed to solve the CD-PLSA model, and further its distributed version is exploited to avoid uploading all the raw data to a centralized location and help to mitigate privacy concerns. After the training phase with all the data from multiple domains we propose to refine the immediate outputs using only the corresponding local data. In summary, we propose a two-phase method for cross-domain text classification, the first phase for collaborative training with all the data, and the second step for local refinement. Finally, we conduct extensive experiments over hundreds of classification tasks with multiple source domains and multiple target domains to validate the superiority of the proposed method over existing state-of-the-art methods of supervised and transfer learning. It is noted to mention that as shown by the experimental results CD-PLSA for the collaborative training is more tolerant of distribution differences, and the local refinement also gains significant improvement in terms of classification accuracy.

Index Terms—Statistical generative models, cross-domain learning, distinction and commonality, classification

1 Introduction

To build a learning model, traditional learning methods usually yield to the fundamental assumption that the data from different information sources are drawn from the same data distribution. However, in many emerging real-world applications, new test data usually come from fast evolving information sources with different but semantically related distributions. For example, to build an enterprise news portal, we need to classify the news about a certain company into some predefined categories, such as "merger and acquisition," "product announcement," "financial scandal," and so on. This classification model may be trained from the news about one company, and may fail on the news for another company since the business areas for the two companies may be different. To deal with this change of data distributions, one solution is to include more labeled data in the new domains into the training set. However, it is often expensive or not practical to recollect the required amount of new training data. Indeed, it is highly desirable to reduce the need and efforts to label new data. This leads to the research of *cross-domain learning* (often referred to as *transfer learning* or *domain adaption*) [2], [3], [4], [5], [6], [7], [8], [9], [10]. In this paper, the training data and test data are also referred to source domain and target domain, respectively.

Unlike previous approaches, which consider the distribution of the low-level features of raw words, we exploit high-level *word concepts*. Here, any word concept y can be represented by a multinomial distribution $p(w|y)$ over words, and this distribution is often domain dependent. Let us take the word concept "products" as an example, if this concept is within the domain of the HP company, which makes printers, the values of $p(\text{``}printer\text{''}|\text{``}products\text{''})$ and $p(\text{``}LaserJet\text{''}|\text{``}products\text{''})$ are large within the domain of HP. If we change the domain to IBM, the representative

- *F. Zhuang, Q. He, and Z. Shi are with the Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Kexueyuan Nanlu #6, Zhongguan Cun, Haidian District, Beijing 100190, China. E-mail: {zhuangfz, heq, shizz}@ics.ict.ac.cn.*
- *P. Luo and Z. Shen are with the Hewlett-Packard Labs, 5/F, Block A, SP Tower, No. 1 Zhong Guan Cun, East Road, Haidian District, Beijing 100084, China. E-mail: {ping.luo, zhiyongs}@hp.com.*
- *Y. Xiong is with the Lashou.com, 10th Floor, Building H, Time Fortune Center, Chaoyang District, Beijing 10028, China. E-mail: yhxiong@yahoo.com.*
- *H. Xiong is with the Management Science and Information Systems Department, Rutgers Business School—Newark and New Brunswick, Rutgers, The State University of New Jersey, Washington Park, Washington Street, Newark, NJ 07102. E-mail: hxiong@rutgers.edu.*

Manuscript received 28 Sept. 2010; revised 21 Feb. 2011; accepted 16 May 2011; published online 23 June 2011.
Recommended for acceptance by X. Zhou.
For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number TKDE-2010-09-0522.
Digital Object Identifier no. 10.1109/TKDE.2011.143.

TABLE 1
The Relationship of the Terminologies

Domain Distinction		Domain Commonality	
Extension of *word* concept $p(w\|y)$	Extension of *document* concept $p(d\|z)$	Intension of *word* concept $p(y,z)$	Intension of *document* concept $p(y,z)$

words of this concept turn to be some IBM product names, and $p(\text{``}printer\text{''}|\text{``}products\text{''})$ and $p(\text{``}LaserJet\text{''}|\text{``}products\text{''})$ will have a very small value within the domain of IBM. Indeed, Table 6 in the experimental section also lists three word concepts with the corresponding key words for each of the four domains. In the table, we can observe that different domains may use different words to express and describe the same concept.

Moreover, we observe that, wherever a word concept exists, it has the same implication to the class of the document which contains this concept. Let us consider the word concept "products." If a news contains the word concept "products," no matter where it comes from, it is more likely to be a news about "product announcement" rather than about "financial scandal." In other words, the association between word concept y and document class z, represented by their joint probability $p(y,z)$, is usually stable across domains.

In the above example, $p(w|y)$ and $p(y,z)$ corresponds to the two sides of a word concept y, *extension* and *intension*, respectively. *In general, the extension of a concept is just the collection of individual objects to which it is correctly applied, while the intension of a concept is the set of features which are shared by everything to which it applies.*[1] Following the general definitions of concept extension and intension their definitions for word concept are as follows.

Definition 1 (Extension of Word Concept). *The extension of a word concept y is the degree of applicability of that concept for each word w, denoted by $p(w|y)$.*

That is to say, when $p(w|y)$ is large, w is a typical object to which the word concept y can be applied.

Definition 2 (Intension of Word Concept). *The intension of a word concept y is expressed by its association with each document class z, denoted by their joint probability $p(y,z)$ in this study.*

For a word concept y, the values of $p(y,z)$ over different document classes z can be considered as the intrinsic features of concept y.

In a similar way, we can also define the extension and intension of *document concept* z as $p(d|z)$ (a multinomial distribution over document d) and $p(y,z)$, respectively. Since we consider each document class for classification as a document concept here, document class and document concept are interchangeable in this paper. To make the terminologies "Distinction," "Commonality," "Extension," and "Intension" defined in this paper more clear, their relationships are summarized in Table 1. The domain distinction includes the extension of word concept and document concept, while domain commonality contains the intension of word concept and document concept. Let's revisit the example of enterprise news classification, the distinctions among data domains HP and IBM are: 1) the key words describing the word concept "products" are different; 2) the documents in document class "product announcement" are different. On the other hand, the domain commonality is the shared association between word concept and document class, e.g., $p(\text{``products''}, \text{``product announcement''})$.

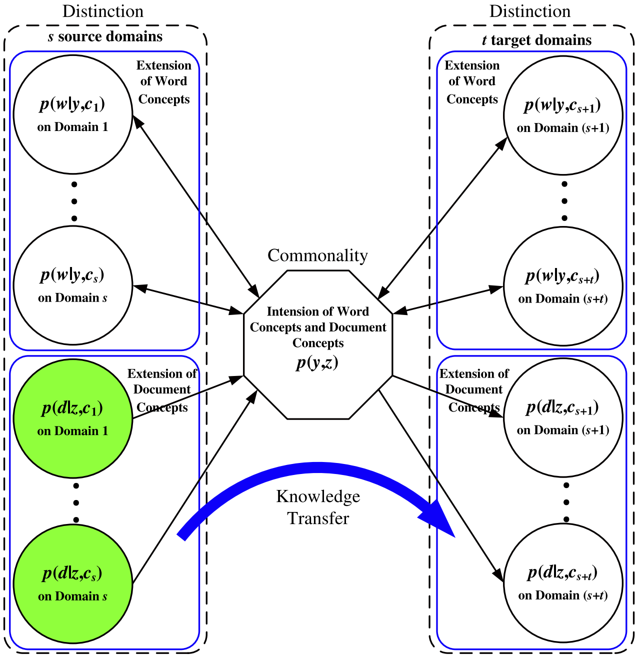

Distinction
s source domains
Extension of Word Concepts
$p(w|y,c_1)$ on Domain 1
$p(w|y,c_s)$ on Domain *s*
Extension of Document Concepts
$p(d|z,c_1)$ on Domain 1
$p(d|z,c_s)$ on Domain *s*
Commonality
Intension of Word Concepts and Document Concepts $p(y,z)$
Knowledge Transfer
Distinction
t target domains
Extension of Word Concepts
$p(w|y,c_{s+1})$ on Domain (*s*+1)
$p(w|y,c_{s+t})$ on Domain (*s*+*t*)
Extension of Document Concepts
$p(d|z,c_{s+1})$ on Domain (*s*+1)
$p(d|z,c_{s+t})$ on Domain (*s*+*t*)

Fig. 1. Extension and intension of concepts.

With the above definitions, we further argue that the extension of any word concept or document concept is often domain-dependent, while its intension is often stable across different domains. Thus, we propose to exploit the distinction and commonality across data domains for text categorization.

In the first phase, we develop a generative statistical model, Collaborative Dual-PLSA (CD-PLSA), to simultaneously capture both domain distinction and commonality. The main idea of this model is illustrated in Fig. 1. In this figure, we have s source domains and t target domains (s and t can be any positive integers), represented by the dashed rectangle on the left and right, respectively. In each dashed rectangle there are two solid rectangles at the above and below, bounding the extensions of word concepts and document concepts, respectively. All these extensions, as the distinction for each domain, share the same intensions of word and document concepts as their commonality (the polygon in the middle). Since we know the class label of each document in the source domains, we actually know the extensions of the document concepts in the source domains. Thus, these observed extensions of the document concepts (the filled circles) are used as the supervision information, which is transferred through the bridge of concept intensions (the polygon in the middle) to the other parts of the model (the unfilled circles). We employ an EM

[1] http://www.philosophypages.com/lg/e05.htm.

solution to learn the CD-PLSA model. Also in order to handle the situation where the data domains are geographically separated from each other, we provide a distributed solution to the CD-PLSA model. In this distributed version only some intermediate statistics are transmitted, rather than communicating and exposing the raw data, which can alleviate the privacy concerns to some degree.

In the second phase, we further exploit the intrinsic structure of the target domains. After solving the CD-PLSA model, we can obtain the intensions of word and document concepts $p(y,z)$, which are shared by all data domains. Indeed, the output intensions $p(y,z)$ from CD-PLSA model may not be the exact ones for target domains. Thus, we propose to refine the outputs from CD-PLSA model with only the local data in target domains.

In summary, we propose a two-phase cross-domain approach for text classification. In the first phase, we collaboratively train a generative model (CD-PLSA) based on all the domain data to generate the commonality $p(y,z)$ and distinction $p(w|y)$, $p(d|z)$. In the second phase, we further refine the outputs only with the local data corresponding to each target domain. Thus, the whole method is called Refined CD-PLSA (RCD-PLSA for short).

Finally, we conduct extensive experiments to evaluate the effectiveness of CD-PLSA and RCD-PLSA on binary classification problems as well as multiclass classification tasks with multiple source and target domains. Experimental results show that CD-PLSA (in the first training phase) is more tolerant to distribution differences, and RCD-PLSA with the local refinement in the second phase further gains significant improvement in terms of overall classification accuracy.

Overview. The remainder of this paper is organized as follows: Section 2 introduces the related work. In Section 3, we review some preliminaries and then give the problem formulation. Its solution by EM and the two-phase method are followed in Section 4. Next, a distributed solution to CD-PLSA is described in Section 5 and the experimental results to validate our algorithm are described in Section 6. Finally, conclusions are drawn in Section 7.

2 Related Works and Discussions

In this section, we will survey some related work, and then give some discussions on generative and discriminative classifiers for cross-domain learning.

2.1 Cross-Domain Learning

Cross-domain Learning has attracted great attention in recent years, and the works in this field can be grouped into four categories based on the different types of techniques used for knowledge transfer, namely feature selection based, feature space mapping, weight based, and model combination-based methods.

Feature selection-based methods are to identify the common features (at the level of raw words) between source and target domains, which are useful for transfer learning. Jiang and Zhai [11] argued that the features highly related to class labels should be assigned to large weights in the learned model, thus they developed a two-step feature selection framework for domain adaptation. They first selected the general features to build a general classifier, and then considered the unlabeled target domain to select specific features for training target classifier. Zhuang et al. [12] formulated a joint optimization framework of the two matrix trifactorizations for the source and target domain data, respectively, in which the associations between word clusters and document classes are shared between them for knowledge transfer. Although the basic assumption of this method is similar to our method, it lacks the probabilistic explanation of the model and is not easy to be extended to handle the tasks with multiple source and target domains. Dai et al. [6] 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.

Feature space mapping-based methods are to map the original high-dimensional features into a low-dimensional feature space, under which the source and target domains comply with the same data distribution. Pan et al. [13] proposed a dimensionality reduction approach to find out this latent feature space, in which supervised learning algorithms can be applied to train classification models. Gu and Zhou [14] learned the shared subspace among multiple domains for clustering and transductive transfer classification. In their problem formulation, all the domains have the same cluster centroid in the shared subspace. The label information can also be injected for classification tasks in this method. Xie et al. [15] tried to fill up those missing values of disjoint features to drive the marginal distributions of two domains closer, and then found the comparable substructures in the latent space where both marginal and conditional distribution are similar. In this latent space, given an unlabeled instances in the target domain the most similar labeled instances are retrieved for classification.

Weight-based methods can be further grouped into two kinds, i.e., the instance weighting based and model weighting-based methods. Instance weighting-based approaches reweight the instances in source domains according to the similarity measure on how they are close to the data in the target domain. Specifically, the weight of an instance is increased if it is close to the data in the target domain, otherwise the weight is decreased. Jiang and Zhai [16] proposed a general instance weighting framework, which has been validated to work well on NLP tasks. Dai et al. [8] 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. On the other side model weighting-based methods give different weights to the classification models in an ensemble. Gao et al. [3] proposed a dynamic model weighting method for each test example according to the similarity between the model and the local structure of the test example in the target domain.

Model combination-based methods, considering the situation of multiple source domains, integrate the source-domain local models according to certain criterion. Luo et al. [7] proposed the regularization framework which maximizes not only the posteriori in each source domain, but also the consensus degree of these models' prediction results on the target domain. Dredze et al. [17] proposed a online model update method for each coming instance, which guarantee that after each iteration the combined model yields a correct prediction for the current instance with high probability while also making the smallest change from the existing models from the source domains.

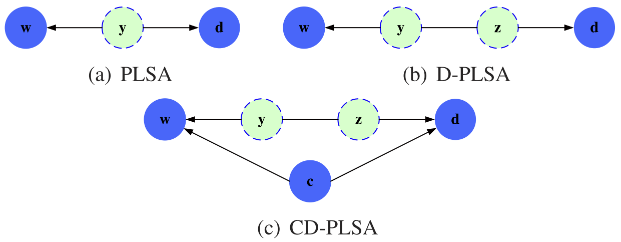
The most related works are [18], [9]. The work of Zhai et al. [18] connects the variations of a topic under different contexts by leveraging the same background for this topic. Our work can also use this technique to explore possible improvements. In this sense, their work is orthogonal to ours. Xue et al. [9] proposed the model of topic-bridged PLSA for cross-domain text categorization, and the basic assumption of this work is that the source and target domains share the same topics. Specifically, they conduct two topic modelings over the source and target domains jointly, and induce the supervision of the labeled source domain data by the pairwise constraints, similar to the must-link and cannot-link constraints used in semi-supervised clustering. Different from topic-bridged PLSA, our model explicitly explores the commonality (concept intension) and distinction (concept extension) of the topics across multiple domains rather than assume that these topics are exactly the same. Additionally, since our model has two latent variables for word concept and document class, it can naturally include the supervision from the source domain, rather than add a penalty of the pairwise constraints to the original log-likelihood function.

2.2 Discussion on Generative versus Discriminative Classifiers for Transfer Learning

Given the observed data x and their labels y, we can formulate the learning of a classifier as calculating the posterior distribution $p(y|x)$. A discriminative classifier models this distribution directly while a generative classifier models the joint probability $p(x, y)$, after which $p(y|x)$ is calculated via Bays rules. There is a widely held belief in the literatures that discriminative classifiers are preferred to generative ones in practice. For example, Vapnik articulated in [19] that

> One should solve the classification problem directly and never solve a more general problem as an intermediate step such as modeling $p(x|y)$.

However, when learning and applying discriminative classifiers, we essentially assume that all the data instances are generated from the identical distribution. This assumption may not hold when data are from different sources. Ideally, the conditional probability $p(y|x)$ may be the same across different domains, however, the marginal probability $p(x)$ on each domain is prone to be different. The problem is that since the training of $p(y|x)$ based on the data in a source domain is biased toward the local marginal probability $p(x)$ it is difficult to achieve the ideal $p(y|x)$ by discriminative models even using the data from all the source domains. On the other hand, the generative classifiers, like CD-PLSA proposed here, provide us facilities to explicitly model the data distribution differences across domains. Thus, it may introduce extra values in prediction. Therefore, we argue that generative models may be suited for transfer learning.



(a) PLSA

(b) D-PLSA

(c) CD-PLSA

Fig. 2. The graphical models.

3 Preliminaries and Problem Formulation

In this section, we first briefly review Probabilistic Latent Semantic Analysis (PLSA), and then introduce an extension of PLSA, Dual-PLSA. Finally, we formulate our problem for cross-domain classification.

3.1 A Review of PLSA

Probabilistic Latent Semantic Analysis [20] is a statistical model to analyze co-occurrence data by a mixture decomposition. Specifically, given the word-document co-occurrence matrix O whose element $O_{w,d}$ represents the frequency of word w appearing in document d, PLSA models O by using a mixture model with latent topics (each topic is denoted by y) as follows:

$$p(w, d) = \sum_y p(w, d, y) = \sum_y p(w|y)p(d|y)p(y). \tag{1}$$

Fig. 2a shows the graphical model for PLSA. The parameters of $p(w|y), p(d|y), p(y)$ over all w, d, y are obtained by the EM solution to the maximum likelihood problem.

3.2 The Dual-PLSA Model

In the PLSA model, the documents and words share the same latent variable y. However, documents and words usually exhibit different organizations and structures. Specifically, they may have different kinds of latent topics, denoted by y for word concept and z for document concept. Its graphical model is shown in Fig. 2b. Since there are two latent variables in this model we call it Dual-PLSA (D-PLSA for short) in this paper.

Given the word-document co-occurrence O, we can similarly arise a mixture model like (1),

$$p(w, d) = \sum_{y,z} p(w, d, y, z) = \sum_{y,z} p(w|y)p(d|z)p(y, z). \tag{2}$$

And the parameters of $p(w|y), p(d|z), p(y, z)$ over all w, d, y, z can also be obtained by the EM solution. In these parameters $p(w|y)$ and $p(d|z)$ are actually the extensions of the word concept y and the document concept z, respectively, while $p(y, z)$ is actually their intension.

This model was proposed in [21] for the clustering problem. In this paper, we find that since the word topic and document topic are separated in this model we can inject the label information into $p(d|z)$ when d is a labeled instance and z is actually a document class. This way this model can also be used for semi-supervised classification. We will detail this in Section 6.1.2.

3.3 The Collaborative Dual-PLSA Model

Based on D-PLSA, we propose a statistical generative model for text classification cross multiple domains. Supposed we have $s+t$ data domains, denoted as $\mathcal{D} = (\mathcal{D}_1, \ldots, \mathcal{D}_s, \mathcal{D}_{s+1}, \ldots, \mathcal{D}_{s+t})$. Without loss of generality, we assume the first s domains are source domains with label information and the left t domains are target domains without any label information. Simply, for each domain we can generate its own extensions and intensions of word and document concepts. However, this simple method generates $s+t$ different sets of concept intensions. To obtain only one set of concept intensions, the variables y and z for word concept and document concept, respectively, must be independent of the variable c for the data domain. Therefore, we propose the graphical model in Fig. 2c to catch the requirements that 1) y and z are independent of c; 2) the word w is dependent of both y and c; 3) the document d is dependent of both z and c. Given this graphical model the joint probability over all the variables is

$$p(w,d,y,z,c) = p(w|y,c)p(d|z,c)p(y,z)p(c). \tag{3}$$

The word-document co-occurrence matrix in the cth domain is denoted by $\boldsymbol{O}_c$, whose element $O_{w,d,c}$ represents the co-occurrence frequencies of the triple (w,d,c). If we denote the two latent variables y,z as $\mathbf{Z}$, given the whole data X from different domains we formulate the problem of maximum log likelihood as

$$\log p(\boldsymbol{X}|\theta) = \log \sum_{\boldsymbol{Z}} p(\boldsymbol{Z}, \boldsymbol{X}|\theta), \tag{4}$$

where θ includes the parameters of $p(y,z)$, $p(w|y,c)$, $p(d|z,c)$ and $p(c)$.

We have to mention that although the extensions of the same word concept y on different domains are different, these extensions are semantically related to a certain degree. The reason is that they are trained collaboratively by sharing the same intension of $p(y,z)$. By the experimental results in Section 6.4 we will intuitively show the difference and relatedness among the extensions, which corresponds to the same word concept, on the multiple domains. In this sense we call our model Collaborative Dual-PLSA. Next, we develop an EM solution to the problem in (4).

4 An EM Solution to the Collaborative Dual-PLSA Model

An Expectation-Maximization (EM) algorithm [22], [23] is to maximize the lower bound (via Jensen's inequality) $\mathcal{L}_0$ of (4)

$$\mathcal{L}_0 = \sum_{\boldsymbol{Z}} q(\boldsymbol{Z}) \log \left\{ \frac{p(\boldsymbol{Z}, \boldsymbol{X}|\theta)}{q(\boldsymbol{Z})} \right\}, \tag{5}$$

where $q(\boldsymbol{Z})$ could be arbitrary. We set $q(\boldsymbol{Z}) = p(\boldsymbol{Z}|\boldsymbol{X};\theta^{\text{old}})$ and substitute into (5)

$$\begin{aligned}\mathcal{L}_0 &= \underbrace{\sum_{\boldsymbol{Z}} p(\boldsymbol{Z}|\boldsymbol{X};\theta^{\text{old}}) \log p(\boldsymbol{Z},\boldsymbol{X}|\theta)}_{\mathcal{L}} \\ &\quad \underbrace{- \sum_{\boldsymbol{Z}} p(\boldsymbol{Z}|\boldsymbol{X};\theta^{\text{old}}) \log p(\boldsymbol{Z}|\boldsymbol{X};\theta^{\text{old}})}_{\text{const}} \\ &= \mathcal{L} + \text{const}.\end{aligned} \tag{6}$$

4.1 E Step: Constructing $\mathcal{L}$

According to the derivation in Appendix, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TKDE.2011.143, for the problem setting of CD-PLSA we have

$$\begin{aligned}\mathcal{L} = &\sum_{y,z,w,d,c} O_{w,d,c} p(y,z|w,d,c;\theta^{\text{old}}) \\ &\cdot \log \left[p(y,z)p(w|y,c)p(d|z,c)p(c) \right],\end{aligned} \tag{7}$$

where

$$p(y,z|w,d,c;\theta^{\text{old}}) = \frac{p(y,z)p(w|y,c)p(d|z,c)p(c)}{\sum_{y,z} p(y,z)p(w|y,c)p(d|z,c)p(c)}. \tag{8}$$

4.2 M Step: Maximizing $\mathcal{L}$

Now we maximize $\mathcal{L}$ with its parameters by Lagrangian Multiplier method and extract the terms containing $p(w|y,c)$. Then, we have

$$\mathcal{L}_{[p(w|y,c)]} = \sum_{y,z,w,d,c} O_{w,d,c} p(y,z|w,d,c;\theta^{\text{old}}) \cdot \log p(w|y,c). \tag{9}$$

Applying the constraint $\sum_w p(w|y,c) = 1$ into the following equation:

$$\frac{\partial \left[\mathcal{L}_{[p(w|y,c)]} + \lambda (1 - \sum_w p(w|y,c)) \right]}{\partial p(w|y,c)} = 0, \tag{10}$$

then

$$p(w|y,c) = \frac{\sum_{z,d} O_{w,d,c}\, p(y,z|w,d,c;\theta^{\text{old}})}{\lambda}. \tag{11}$$

Considering the constraint $\sum_w p(w|y,c) = 1$,

$$1 = \sum_w p(w|y,c) = \frac{\sum_w \sum_{z,d} O_{w,d,c}\, p(y,z|w,d,c;\theta^{\text{old}})}{\lambda}, \tag{12}$$

$$\Rightarrow \lambda = \sum_w \sum_{z,d} O_{w,d,c}\, p(y,z|w,d,c;\theta^{\text{old}}). \tag{13}$$

Finally, the update formula of $p(w|y,c)$ can be obtained,

$$\hat{p}(w|y,c) = \frac{\sum_{z,d} O_{w,d,c}\, p(y,z|w,d,c;\theta^{\text{old}})}{\sum_{z,w,d} O_{w,d,c}\, p(y,z|w,d,c;\theta^{\text{old}})}. \tag{14}$$

Similarly,

$$\hat{p}(d|z,c) = \frac{\sum_{y,w} O_{w,d,c}\, p(y,z|w,d,c;\theta^{\text{old}})}{\sum_{y,w,d} O_{w,d,c}\, p(y,z|w,d,c;\theta^{\text{old}})}, \tag{15}$$

$$\hat{p}(y,z) = \frac{\sum_{w,d,c} O_{w,d,c}\, p(y,z|w,d,c;\theta^{\text{old}})}{\sum_{y,z,w,d,c} O_{w,d,c}\, p(y,z|w,d,c;\theta^{\text{old}})}, \tag{16}$$

$$\hat{p}(c) = \frac{\sum_{y,z,w,d} O_{w,d,c}\, p(y,z|w,d,c;\theta^{\text{old}})}{\sum_{y,z,w,d,c} O_{w,d,c}\, p(y,z|w,d,c;\theta^{\text{old}})}. \tag{17}$$

4.3 CD-PLSA to Cross-Domain Classification

In this section, we introduce how to leverage the proposed EM algorithm for cross-domain classification. We need to figure out two subtasks: 1) how to inject the label information in source domains to supervise the EM optimization; 2) how to assign the class label to the instances in the target domains based on the output from the EM algorithm.

For the first task we inject the supervising information (the class label of the instances in the source domains) into the probability $p(d|z,c)$ $(1 \leq c \leq s)$. Specifically, let $L^c \in [0,1]^{n_c \times m}$ be the true label information of the cth domain, where n_c is the number of instances in it, m is the number of document classes. If instance d belongs to document class z_0, then $L^c_{d,z_0} = 1$, otherwise $L^c_{d,z} = 0$ $(z \neq z_0)$. We normalize L^c to satisfy the probabilistic condition so that the sum of the entries in each column equals to 1,

$$N^c_{d,z} = \frac{L^c_{d,z}}{\sum_d L^c_{d,z}}. \tag{18}$$

Then $p(d|z,c)$ is initialized as $N^c_{d,z}$. Note that since this initial value is from the true class label we do not change the value of $p(d|z,c)$ (for $1 \leq c \leq s$) during the iterative process.

For the unlabeled target domains, $p(d|z,c)$ $(s+1 \leq c \leq s+t)$ can be initialized similarly. This time the label information L^c used can be obtained by any supervised classifier (Logistic Regression is used in this paper). Note that since this classifier may output the wrong class label we do change the value of $p(d|z,c)$ (for $s+1 \leq c \leq s+t$) during the iterative process.

After the EM iteration we obtain all the parameters of $p(d|z,c)$, $p(w|y,c)$, $p(y,z)$, $p(c)$, based on which we compute the posteriori probability $p(z|d,c)$ as follows,

$$\begin{aligned} p(z|d,c) &= \frac{p(z,d,c)}{p(d,c)} \propto p(z,d,c) = p(d|z,c)p(z,c) \\ &= p(d|z,c)p(z)p(c) = p(d|z,c)p(c)\sum_y p(y,z) \\ &\propto p(d|z,c)\sum_y p(y,z). \end{aligned} \tag{19}$$

Then, the class label of any document d in a target domain c is predicted to be

$$\arg\max_z p(z|d,c). \tag{20}$$

The detailed procedure of CD-PLSA for cross-domain classification is depicted in Algorithm 1. Note that our algorithm can deal with the situations where there are multiple source domains and multiple target domains.

Algorithm 1. CD-PLSA for Cross-Domain Classification

Input: Given $(s+t)$ data domains $\mathcal{D}_1, \ldots, \mathcal{D}_s$, $\mathcal{D}_{s+1}, \ldots, \mathcal{D}_{s+t}$, where the first s domains are source domains while the left are target domains. T, the number of iterations. Y, the number of word clusters.

Output: the class label of each document d in the target domain.

```
1) Initialization. $p^{(0)}(w|y,c)$ is set to the output $p(w|y)$ from
   PLSA. The initialization of $p^{(0)}(d|z,c)$ is detailed in
   Section 4.3. $p^{(0)}(y,z)$ is set randomly.
2) $k := 1$.
3) for $c := 1 \to s+t$
       Update $p^{(k)}(y,z|w,d,c)$ according to
       Equation (8) in E-step;
4) end.
5) for $c := 1 \to s+t$
       Update $p^{(k)}(w|y,c)$ according to Equation (14)
       in M-step;
6) end.
7) for $c := s+1 \to s+t$
       Update $p^{(k)}(d|z,c)$ according to Equation (15) in
       M-step;
8) end.
9) Update $p^{(k)}(y,z)$ according to Equation (16) in M-step.
10) Update $p^{(k)}(c)$ according to Equation (17) in M-step.
11) $k := k+1$, if $k < T$, turn to Step 3.
12) The class label of any document $d$ in a target domain $c$ is
    predicted by Equation (20).
```

4.4 Refined CD-PLSA

CD-PLSA can output the extension of word concepts $p(w|y,c)$ and document concepts $p(d|z,c)$ for each domain c, and the intensions of word and document concepts $p(y,z)$ which are shared by all data domains. In CD-PLSA, we assume that the intensions of word and document concepts $p(y,z)$ remain the same across different domains. However, since different target domains may have their own characteristics with respect to data distribution, the shared intensions may not be the exact ones for the specific target domains. Thus, we further propose to refine the outputs of CD-PLSA model. Specifically, in this step, based on only the local data from each target domain c we update the variables $p(y,z|w,d,c)$, $p(w|y,c)$, $p(d|z,c)$, $p(y,z|c)$ $(s+1 \leq c \leq s+t)$ separately by (8), (14), (15), (16). Additionally, their initial values are assigned with the outputs from CD-PLSA. The experiments show that this local refinement step further improve CD-PLSA in terms of classification accuracy.

5 A Distributed Implementation of the CD-PLSA Model

Here, we extend the proposed EM-based algorithm into a distributed version, which can work in the situation that the source domains $\mathcal{D}_1, \ldots, \mathcal{D}_s$ and the target domains $\mathcal{D}_{s+1}, \ldots, \mathcal{D}_{s+t}$ are geographically separated. This distributed implementation of the CD-PLSA model helps when the data are separated in multiple source.

In this distributed setting, we need a central node, denoted by mn, as the *master node*, and all the nodes for the data domains are used as *slave nodes*, denoted by $sn^{(1)}, \ldots, sn^{(s+t)}$. We find that 1) $p(y,z|w,d,c;\theta^{\text{old}})$, $p(w|y,c)$ and $p(d|z,c)$ in (8), (14) and (15) can computed locally on $sn^{(c)}$; 2) $p(y,z)$ can be computed locally on the master node. Specifically, let

$$\triangle^{(c)}_{y,z} = \sum_{w,d} O_{w,d,c}\, p(y,z|w,d,c;\theta^{\text{old}}), \tag{21}$$

$$\mathcal{V}^{(c)} = \sum_{y,z,w,d} O_{w,d,c}\, p(y,z|w,d,c;\theta^{\text{old}}). \tag{22}$$

Then,

$$p(y,z) = \frac{\sum_c \triangle^{(c)}_{y,z}}{\sum_{y,z,c} \triangle^{(c)}_{y,z}}, \quad p(c) = \frac{\mathcal{V}^{(c)}}{\sum_c \mathcal{V}^{(c)}}. \tag{23}$$

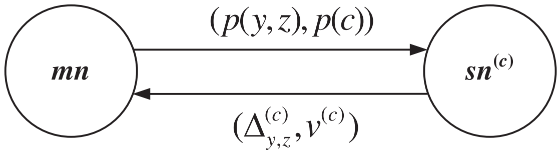


Fig. 3. The statistics transmitted between the master and slave nodes.

In each iteration, the master node first sends the values of $p(y,z)$ and $p(c)$ to each slave node. Then, each slave node $sn^{(c)}$ (for $c \in \{1, \cdots, (s+t)\}$) computes $p(y,z|w,d,c;\theta^{\text{old}})$, $p(w|y,c)$, $p(d|z,c)$, $\triangle_{y,z}^{(c)}$ and $\mathcal{V}^{(c)}$ locally, and sends the local statistics $\triangle_{y,z}^{(c)}$ and $\mathcal{V}^{(c)}$ to the master node. Finally, the master node updates $p(y,z)$ and $p(c)$ according to (23) when receiving all the local statistics from slave nodes, and starts the new round of iteration. It is clear that there are only some statistics, including $\triangle_{y,z}^{(c)}$, $p(y,z)$, $\mathcal{V}^{(c)}$, and $p(c)$, transmitted between the master and slave nodes (depicted in Fig. 3), rather than communicating and exposing the raw data. Let T be the number of iterative rounds, Y be the number of word clusters, C be the number of document classes, then the total communication overhead are $2T \cdot (s+t) \cdot (Y \cdot C + 1)$ (the size of both $p(y,z)$ and $\triangle_{y,z}^{(c)}$ are $Y \cdot C$). Therefore, this distributed algorithm is communication-efficient and also alleviate the privacy concerns to some degree.

6 Experimental Results

In this section, we provide experiments to demonstrate the effectiveness of the proposed models. For the two-class classification problems, each of which involves with four domains: one source domain plus three target domains or three source domains plus one target domain. Also, we perform three-class classification experiments to show that our models can easily handle multiclass problems. The classification accuracy is the evaluation metric in this work. The code of CD-PLSA is available through the web site http://www.intsci.ac.cn/users/zhuangfuzhen/CD_PLSA. htm.

6.1 The Experimental Setup

6.1.1 Data Preparation

20-Newsgroup [2] is one of the widely used data set for cross-domain learning. This corpus has approximately 20,000 newsgroup documents, which are evenly divided into 20 subcategories. Some related subcategories are grouped into a top category, which is used as document class. The corpus contains four top categories *comp*, *rec*, *sci*, and *talk*, which all have four subcategories. Their corresponding subcategories are listed in Table 2. In the experiments, we can construct six data sets for binary classification by randomly selecting two top categories (one for positive and the other one for negative) from the four top categories. They are *rec versus sci*, *comp versus sci*, *sci versus talk*, *comp versus rec*, *comp versus talk*, and *rec versus talk*. Then we construct a two-class cross-domain classification problem as follows: for two top categories A, B (e.g., A is positive and B for negative.) and their four subcategories are denoted as A_1, A_2, A_3, A_4, and B_1, B_2, B_3, B_4, respectively. We select (without replacement) a subcategory from A (e.g., A_2) and a subcategory from B (e.g., B_3) to form a data domain. We repeat the selection four times to get the four data domains. If we select any one of the generated four domains as source domain and the left three domains as target domains, in this way we can generate totally 96 ($4 \cdot P_4^4$) problems of cross-domain classification with one source domain and three target domains. Similarly, we can construct 96 problems with three source domains and one target domain.

TABLE 2
The Top Categories and Their Subcategories

Top Categories	Subcategories
comp	*comp.graphics, comp.os.ms-windows.misc* *comp.sys.ibm.pc.hardware, comp.sys.mac.hardware*
rec	*rec.autos, rec.motorcycles* *rec.sport.baseball, rec.sport.hockey*
sci	*sci.crypt, sci.med, sci.electronics, sci.space*
talk	*talk.politics.guns, talk.politics.mideast* *talk.politics.misc, talk.religion.misc*

For three-class classification four data sets, including *comp versus rec versus sec*, *comp versus rec versus talk*, *comp versus sci versus talk*, and *rec versus sci versus talk*, are generated by randomly selecting three top categories. We construct the three-class classification problems similarly with binary classification, thus we can obtain 2,304 ($4 \cdot P_4^4 \cdot P_4^4$) classification problems (e.g., including three source domains and one target domain for each problem) for each data set. In this three-class situation, we only perform the experiments on 100 randomly selected problem instances from each data set. All the experimental results are detailed in Sections 6.2 and 6.3. The value of 15 is used as the threshold of document frequency to cut down the number of words used in the co-occurrence matrices.

6.1.2 The Baseline Methods

We compare our models CD-PLSA, RCD-PLSA with several baseline classification methods, including

- The supervised learning algorithm Logistic Regression (LG) [24] and LibSVM [25].
- The state-of-the-art cross-domain learning approaches coclustering-based Classification (CoCC)[3] [6], Local Weighted Ensemble (LWE) [3], and the Bridged-Refinement transfer learning (BR) [4].
- Additionally, the algorithm D-PLSA (depicted in Section 3.2) is also used as the baseline. Since there are not domain labels in D-PLSA all the instances appear as if they are from the same domain. In other words the source of each instance is ignored in D-PLSA. Our experiments will show that ignoring this information results in the significant performance sacrifice.

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

[3] We thank the author for providing the codes.

Note that all the classification methods, which cannot directly deal with multiple source domains are adapted as follows: e.g., CoCC, for each source domain and the target domain we train a CoCC model, and then combine these m models by voting with equal weights. The other methods LG (note that LG achieves the similar performance when trained on the merged data of all source domains), LibSVM, BR are adapted to handle multiple source domains similarly with CoCC.

6.1.3 *Implementation Details*

Since the models of D-PLSA and CD-PLSA have the random initialization process, we conduct the experiments three times and the average results are recorded for these two algorithms. Preliminary test shows that our algorithm is not sensitive to the number Y of word clusters (in the range of $[2^5, 2^8]$), thus we set Y to 64. The number of iteration in D-PLSA, CD-PLSA,[4] and RCD-PLSA is set to 50. The parameters of CoCC, LWE, and BR are set to the same values as the original papers.

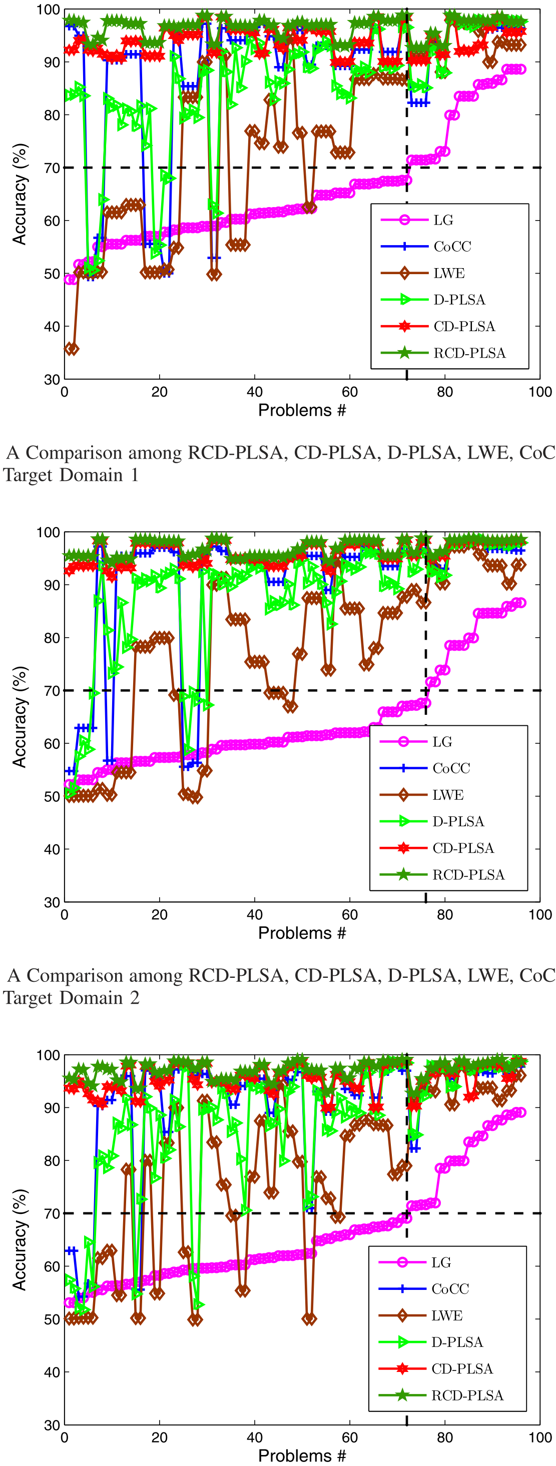
6.2 Results on Two-Class Classification Problems

We compare the proposed models CD-PLSA, RCD-PLSA with baselines LG, CoCC, LWE, and D-PLSA.

6.2.1 *Multiple Target Domains*

Here, we show a comparison of the proposed models CD-PLSA, RCD-PLSA with the baseline methods on the learning tasks with multiple target domains. We have six data sets, and can construct 96 problems for each of them. Here we only list the detailed results of data sets *rec versus sci* and *comp versus sci* in Figs. 4 and 5 since they perform similarly.[5] Each of the two figures have three subfigures, each of which contains the results on one of the three target domains. In each subfigure, the 96 problems are sorted by the increasing order of the accuracy from LG. Thus, the *x-axis* in each figure actually indicates the degree of difficulty in knowledge transformation. From these figures, we can observe that

1. The t-test with 95 percent confidence over all the 192 (96×2) problems shows that CD-PLSA significantly outperforms the other four baseline methods. Furthermore, we find that the improvements of CD-PLSA over the baseline methods are more remarkable when the accuracy of LG is lower than 70 percent. Table 3 records the average results over the corresponding tasks on six data sets. The $Left$ and $Right$ rows represent the average values of the tasks when the accuracy of LG is lower or higher than 70 percent, respectively, while $Total$ denotes the average values over all the 96 problems. We can see that the difference between the average values of CD-PLSA and any baseline method in the $Left$ row is much greater than that in the $Right$ row. That is to say, although the baseline methods may output much lower accuracies when the accuracy of LG is lower than 70 percent, CD-PLSA works still well. The reason may be that the degree of difference in data distributions across domains is too large to be handled by the baseline methods, while our method is more tolerant of distribution differences.



(a) A Comparison among RCD-PLSA, CD-PLSA, D-PLSA, LWE, CoCC, LG on Target Domain 1

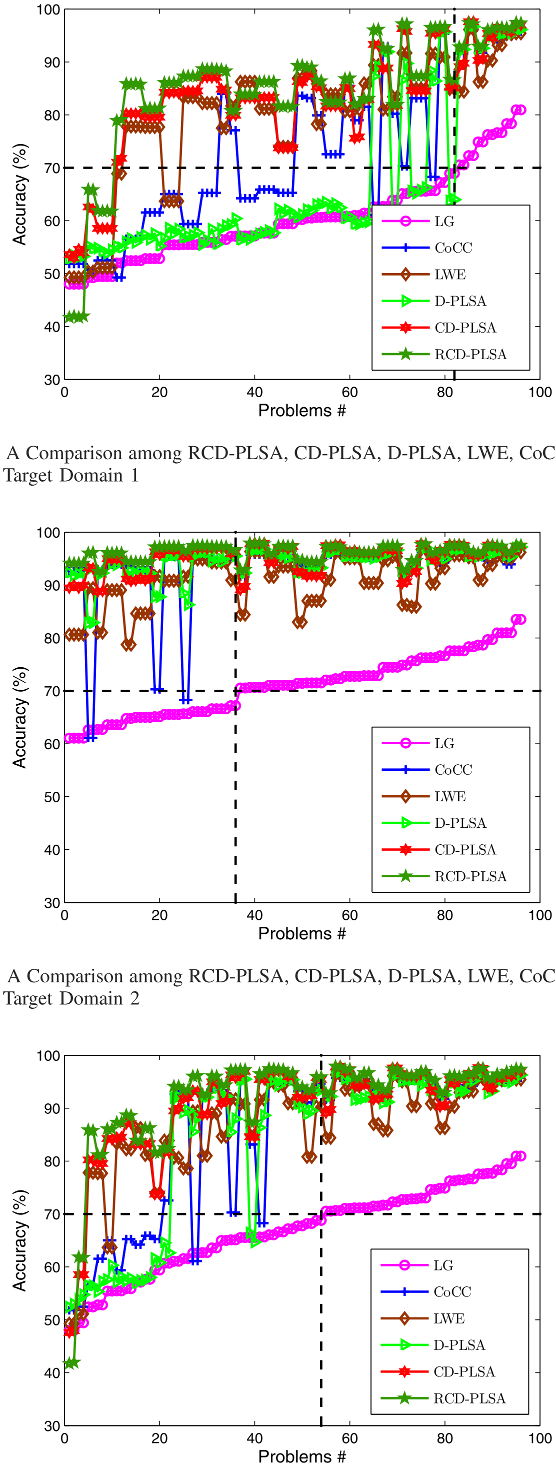
(b) A Comparison among RCD-PLSA, CD-PLSA, D-PLSA, LWE, CoCC, LG on Target Domain 2

(c) A Comparison among RCD-PLSA, CD-PLSA, D-PLSA, LWE, CoCC, LG on Target Domain 3

Fig. 4. The Performance Comparison among RCD-PLSA, CD-PLSA, D-PLSA, LWE, CoCC, and LG on data set *rec versus sci.*

[4] Under these parameters, CD-PLSA can finish our task in 240 seconds. Note that there are about 7,300 features and 7,500 documents in each problem. We will detail the running time in Section 6.5.

[5] All the results are shown in our full version, you can access it through http://www.intsci.ac.cn/users/zhuangfuzhen/.



(a) A Comparison among RCD-PLSA, CD-PLSA, D-PLSA, LWE, CoCC, LG on Target Domain 1

(b) A Comparison among RCD-PLSA, CD-PLSA, D-PLSA, LWE, CoCC, LG on Target Domain 2

(c) A Comparison among RCD-PLSA, CD-PLSA, D-PLSA, LWE, CoCC, LG on Target Domain 3

Fig. 5. The performance comparison among RCD-PLSA, CD-PLSA, D-PLSA, LWE, CoCC, and LG on data set *comp versus sci*.

2. We also observe the advantage of CD-PLSA over D-PLSA in these results. The reasons are as follows: in D-PLSA, the data domain where each instance comes from is ignored, and all the instances are treated as if they come from the same domain. However, the distinction and commonality can only be found by the comparison of at least two domains. Thus, with only one domain our algorithm may sacrifice due to the loss of the information of data domains. On the other hand, we can say that the data domain for each instance introduces significant improvements to our model CD-PLSA.
3. Finally, it is impressed that RCD-PLSA can usually obtain a significant improvement by CD-PSLA in the refinement process. These results indicate the intensions output by CD-PLSA usually are not exactly the ones for target domains, which leaves the space for improvement. Therefore, we can further refine the outputs to enhance the performance in the second step using the local data for each target domain. In a word, all these results validate the effectiveness of CD-PLSA and RCD-PLSA.

6.2.2 *Multiple Source Domains*

Here, we conduct experiments to show that the CD-PLSA model can also work on multiple source domains. We evaluate all the methods on the problem with 3 sources and 1 target. Fig. 6 shows the results. Indeed, similar results can be observed as those in Section 6.2.1, which again show that CD-PLSA outperforms all the compared methods.

We also show Table 4 with the average values over the corresponding 96 problems of the six data sets. The calculation of these values are the same with that in Table 3. Again, these results show that CD-PLSA outperforms the baseline methods on the tasks with multiple source domains, and it can better tolerate the distribution differences. It is obvious to find that RCD-PLSA is consistently better than CD-PLSA in term of the average accuracy, which again verifies the superiority of RCD-PLSA.

We have to mention that in Figs. 6e and 6f, our model CD-PLSA fails in some problems when the baseline method LG performs well (higher than 80 percent). We conjecture that the intension may be very different for these problem instances, thus the independent assumption of CD-PSLA might lead to overfitting and the output $p(y, z)$ bias to the source domains. For this situation, we can use refined CD-PLSA in the second step to make up this little flaw.

6.3 Results on Three-Class Classification Tasks

For three-class classification, we test and compare CD-PSLA and RCD-PLSA models with LG, LibSVM, BR, and D-PLSA. Here the supervised method LibSVM [25] and transfer learning approach BR [4] can directly tackle multiclass classification scenarios, while LG is adapted to handling multiclass situation by one versus rest manner. All the results are exhibited in Fig. 7, and Table 5 records the average performance.

From Figs. 7 and Table 5, we can find that these results are concise with the ones reported in Section 6.2 for binary classification. Our model CD-PLSA is better than all other baselines, except on the data set *comp versus rec versus talk* CD-PLSA is comparable with BR. Specifically, CD-PLSA still can work well on the hard knowledge transfer situations, while the cross-domain methods CD-PLSA and BR are comparable for the much easier problems. Again we observe the additional gains of RCD-PLSA by CD-PLSA, and RCD-PLSA significantly outperforms all the other approaches.

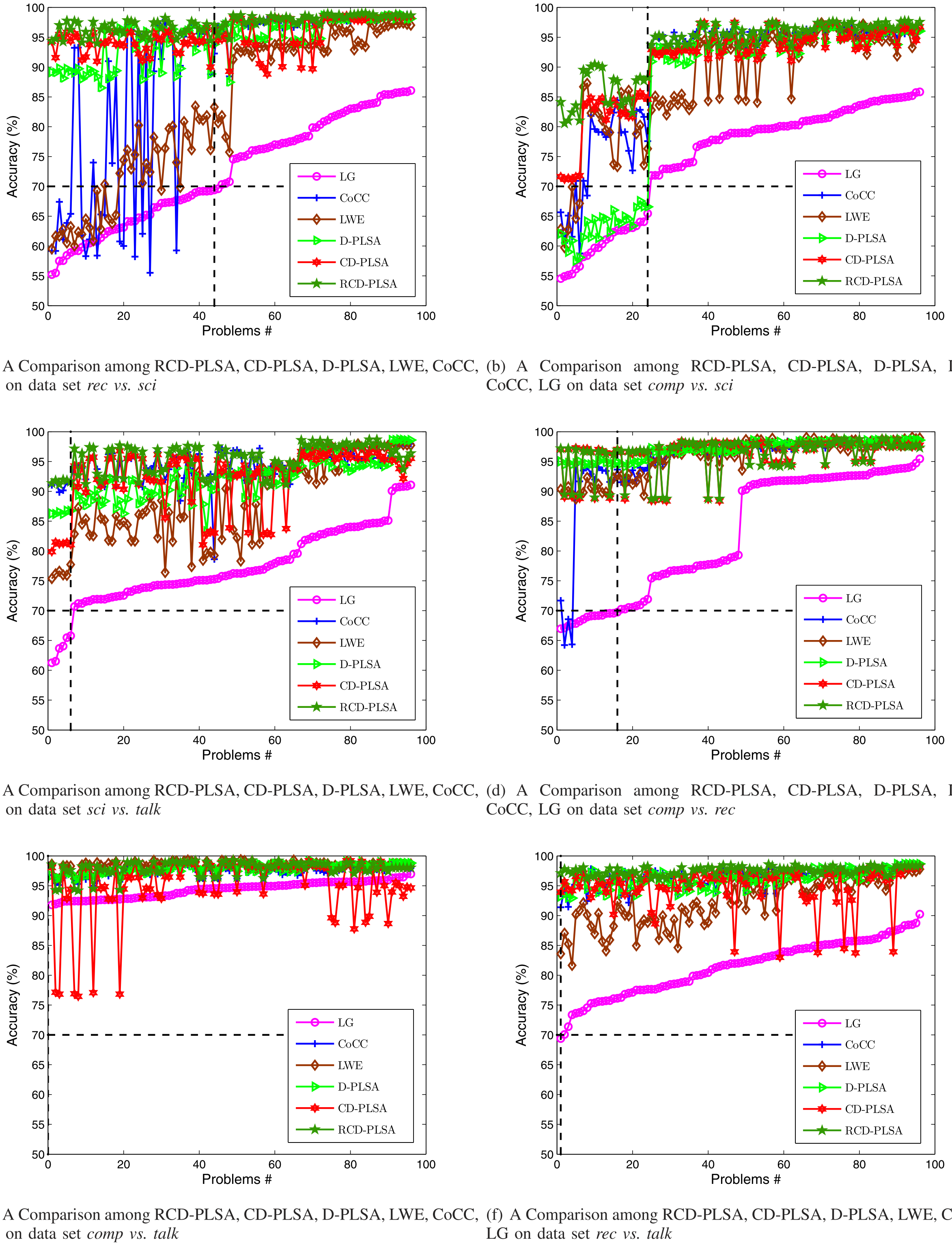
TABLE 3
Average Performances (Percent) on 96 Problems of Each Data Set for Multiple Target Domains for Two-Class Classification

Data Sets			LG	CoCC	LWE	D-PLSA	CD-PLSA	RCD-PLSA
rec vs. sci	Target-1	*Left*	60.33	86.32	69.78	83.16	93.98	**96.59**
		Right	80.82	93.70	93.47	94.31	94.09	**96.47**
		Total	65.46	88.17	75.70	85.95	94.00	**96.56**
	Target-2	*Left*	59.72	89.34	73.24	86.23	95.82	**96.72**
		Right	80.88	96.47	94.29	96.72	97.64	**97.93**
		Total	64.13	90.82	77.62	88.42	96.20	**96.97**
	Target-3	*Left*	61.00	89.62	72.49	84.70	95.39	**96.97**
		Right	81.11	95.55	94.02	95.94	96.32	**97.50**
		Total	66.03	91.10	77.87	87.51	95.62	**97.10**
comp vs. sci	Target-1	*Left*	57.93	69.10	77.64	62.54	80.64	**82.61**
		Right	75.70	94.14	91.56	94.74	94.54	**95.64**
		Total	60.52	72.75	79.67	67.23	82.66	**84.51**
	Target-2	*Left*	64.66	89.52	88.44	92.71	94.08	**95.92**
		Right	74.70	94.95	92.36	95.29	95.65	**96.36**
		Total	70.93	92.91	90.89	94.33	95.06	**96.19**
	Target-3	*Left*	61.02	77.30	82.05	76.86	86.53	**88.40**
		Right	74.36	94.64	92.65	94.30	95.02	**95.99**
		Total	66.86	84.89	86.68	84.49	90.25	**91.72**
sci vs. talk	Target-1	*Left*	63.15	91.38	70.36	89.99	90.40	**95.04**
		Right	78.22	95.95	87.84	95.60	92.39	**96.58**
		Total	72.57	94.24	81.29	93.50	91.64	**96.00**
	Target-2	*Left*	61.35	80.52	69.36	85.49	88.85	**92.83**
		Right	76.42	92.97	87.14	92.29	93.18	**95.76**
		Total	69.52	87.26	78.99	89.17	91.20	**94.41**
	Target-3	*Left*	62.04	85.42	69.89	87.43	89.45	**93.76**
		Right	76.96	94.57	87.57	93.41	93.09	**96.27**
		Total	70.44	90.57	79.84	90.79	91.50	**95.17**
comp vs. rec	Target-1	*Left*	63.26	87.10	93.10	76.71	95.29	**97.26**
		Right	85.15	96.86	90.46	97.71	96.59	**97.85**
		Total	79.68	94.42	91.12	92.46	96.26	**97.70**
	Target-2	*Left*	–	–	–	–	–	–
		Right	84.38	96.81	91.60	98.05	97.21	**98.18**
		Total	84.38	96.81	91.60	98.05	97.21	**98.18**
	Target-3	*Left*	58.02	58.70	92.38	58.11	94.28	**95.75**
		Right	83.67	96.81	90.02	97.90	96.94	**98.05**
		Total	77.26	87.28	90.61	87.95	96.27	**97.47**
comp vs. talk	Target-1	*Left*	–	–	–	–	–	–
		Right	92.51	96.80	97.43	**97.82**	95.00	97.80
		Total	92.51	96.80	97.43	**97.82**	95.00	97.80
	Target-2	*Left*	–	–	–	–	–	–
		Right	92.44	96.40	97.24	**97.95**	96.24	97.81
		Total	92.44	96.40	97.24	**97.95**	96.24	97.81
	Target-3	*Left*	–	–	–	–	–	–
		Right	92.19	96.77	97.50	97.83	96.04	**97.96**
		Total	92.19	96.77	97.50	97.83	96.04	**97.96**
rec vs. talk	Target-1	*Left*	62.63	91.78	64.32	**97.59**	95.07	97.09
		Right	81.34	96.89	88.08	**98.01**	96.11	97.33
		Total	72.77	94.54	77.19	**97.82**	95.64	97.22
	Target-2	*Left*	62.91	88.77	65.66	**97.94**	95.85	97.62
		Right	84.39	96.26	92.53	**97.97**	96.83	97.85
		Total	70.52	91.42	75.17	**97.95**	96.20	97.70
	Target-3	*Left*	63.48	90.02	66.86	**97.79**	95.64	97.37
		Right	83.62	96.61	91.16	**97.90**	96.39	97.62
		Total	71.03	92.49	75.98	**97.83**	95.92	97.46

6.4 Understanding the Extension of a Word Concept over Multiple Domains

Here, we show the difference and relatedness among the extensions of a word concept over multiple domains. Fixing a word concept y and a domain c, we list the top N ($N = 20$ here) words in terms of $p(w|y, c)$. They are actually the representative words for the word concept in a certain domain. The extensions of three word concepts in the four domains are listed in Table 6.

Indeed, the extensions of a word concept on the four domains are related to each other in the sense that their representative words corresponds to the same *semantic*. For example, the third word concept is actually about "Space Science," while the representative words in each extension are different. Specifically, the representative words of this concept in Domain 1 include "rocket," "ESA" (European Space Agency), and "satellite," etc., while those in Domain 2 contain "acceleration," "NASA," and "earth," etc. These results also intuitively show that our model can successfully mine distinction and commonality among multiple domains.



(a) A Comparison among RCD-PLSA, CD-PLSA, D-PLSA, LWE, CoCC, LG on data set *rec vs. sci*

(b) A Comparison among RCD-PLSA, CD-PLSA, D-PLSA, LWE, CoCC, LG on data set *comp vs. sci*

(c) A Comparison among RCD-PLSA, CD-PLSA, D-PLSA, LWE, CoCC, LG on data set *sci vs. talk*

(d) A Comparison among RCD-PLSA, CD-PLSA, D-PLSA, LWE, CoCC, LG on data set *comp vs. rec*

(e) A Comparison among RCD-PLSA, CD-PLSA, D-PLSA, LWE, CoCC, LG on data set *comp vs. talk*

(f) A Comparison among RCD-PLSA, CD-PLSA, D-PLSA, LWE, CoCC, LG on data set *rec vs. talk*

Fig. 6. The performance comparison among RCD-PLSA, CD-PLSA, D-PLSA, LWE, CoCC, and LG on six data sets.

6.5 The Running Time of the CD-PLSA Model

We check the computational performances of CD-PLSA as follows: we randomly select 16 problems from the data set *rec versus sci*, and Fig. 8a shows the running time of CD-PLSA on these 16 problems under different number of word clusters Y. Additionally, the relationship between the average running time (over the problems) and the number of word clusters Y is shown in Fig. 8b, where the y-axis represents the average running time of 16 problems. From these figures, we can find that 1) CD-PLSA runs very fast, and it takes no more than 240 seconds when $Y = 64$ on the data including 7,500 documents and 7,300 features; 2) CD-PLSA runs in linear time with respect to Y.

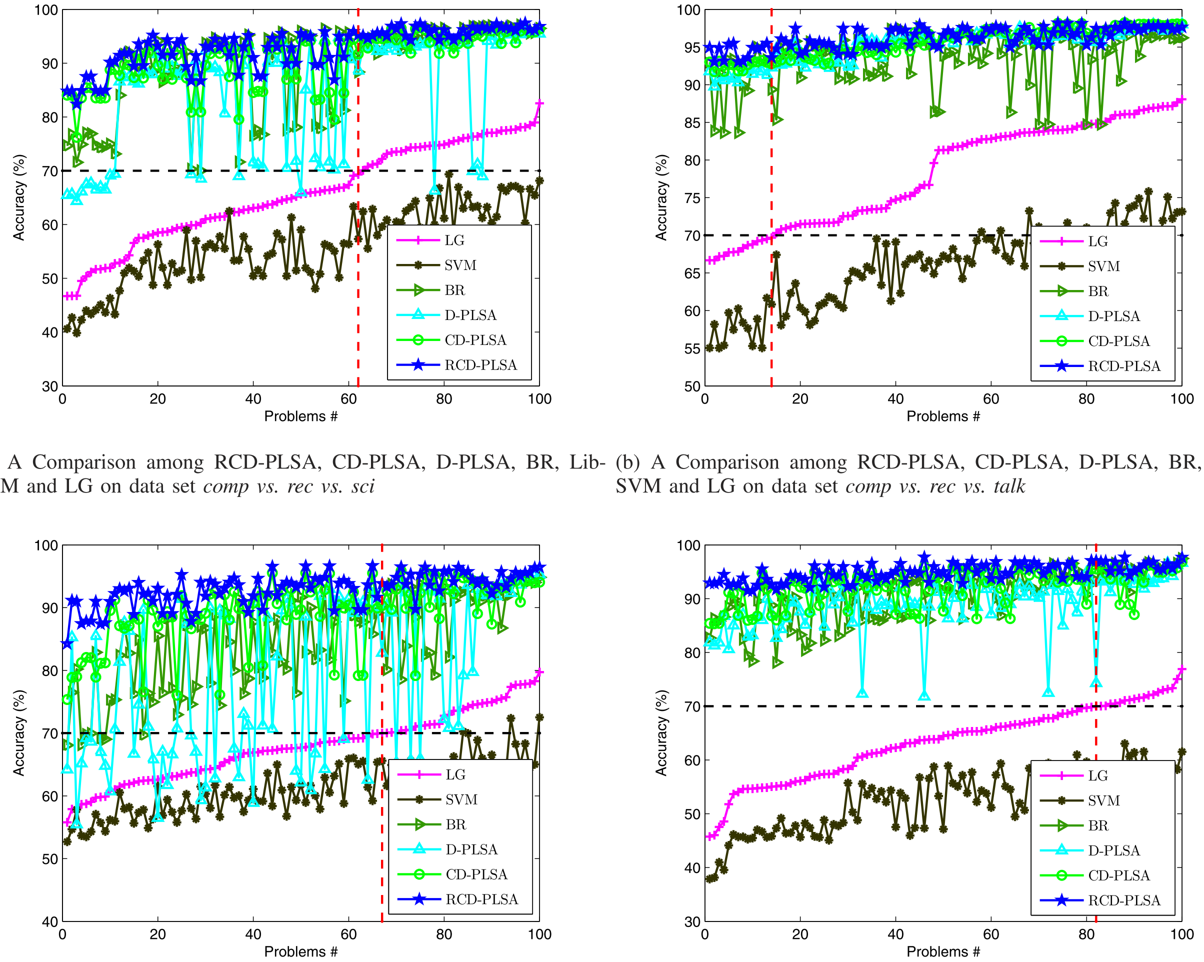
TABLE 4
Average Performances (Percent) on 96 Problems of Each Data Set for Multiple Source Domains

Data Sets		LG	CoCC	LWE	D-PLSA	CD-PLSA	RCD-PLSA
rec vs. sci	*Left*	64.01	80.07	71.41	92.03	94.06	**96.27**
	Right	79.84	97.70	93.62	96.77	96.46	**98.18**
	Total	72.42	89.44	83.21	94.55	95.33	**97.28**
comp vs. sci	*Left*	60.15	74.88	76.77	63.21	80.54	**85.93**
	Right	79.91	95.58	92.14	94.38	94.51	**96.02**
	Total	74.97	90.41	88.30	86.59	91.02	**93.50**
sci vs. talk	*Left*	63.62	91.05	76.30	86.45	81.05	**91.62**
	Right	78.20	94.76	89.78	92.40	92.84	**95.68**
	Total	77.29	94.53	88.94	92.03	92.42	**95.43**
comp vs. rec	*Left*	68.61	86.59	90.46	94.75	94.39	**95.82**
	Right	85.76	97.11	96.90	97.50	96.53	**97.85**
	Total	82.90	95.36	**98.84**	97.04	96.18	98.79
comp vs. talk	*Left*	–	–	–	–	–	–
	Right	94.37	97.41	**98.39**	97.89	95.06	97.96
	Total	94.37	97.41	**98.39**	97.89	95.06	97.96
rec vs. talk	*Left*	69.39	91.39	83.58	93.18	93.93	**97.09**
	Right	81.49	96.60	92.63	96.20	94.98	**97.50**
	Total	81.36	96.56	92.54	96.16	94.96	**97.50**

6.6 Experimental Summary

We summary all the experimental results as follows:

1. CD-PLSA significantly outperforms all the baseline methods. This superiority becomes more remarkable when the accuracy from LG is lower than 70 percent. This indicates that, when the degree of difficulty in knowledge transfer is large, our model still works well while the others may fail. Thus, CD-PLSA is more robust for transfer learning.
2. The CD-PLSA model is further improved by explicitly considering the data domain where each instance comes from. Since the distinction and commonality can only be identified by the comparison of at least two domains, if all the instances are treated as if they come from the same domain, the effectiveness in mining distinction and commonality may compromise. Indeed, the data domain labels on each instance provide a partition of all the data if we group the instances from the same domain into one cluster. Thus, this information is additional supervision to our model.
3. To exploit the intrinsic structure of target domains, we further propose RCD-PLSA to refine the outputs of CD-PLSA for each target domain. And the experiments show that RCD-PLSA usually can gain significant improvement by CD-PLSA.
4. To intuitively understand the extensions of a word concept over different domains, we list the key



(a) A Comparison among RCD-PLSA, CD-PLSA, D-PLSA, BR, LibSVM and LG on data set *comp vs. rec vs. sci*

(b) A Comparison among RCD-PLSA, CD-PLSA, D-PLSA, BR, LibSVM and LG on data set *comp vs. rec vs. talk*

(c) A Comparison among RCD-PLSA, CD-PLSA, D-PLSA, BR, LibSVM and LG on data set *comp vs. sci vs. talk*

(d) A Comparison among RCD-PLSA, CD-PLSA, D-PLSA, BR, LibSVM and LG on data set *rec vs. sci vs. talk*

Fig. 7. The performance comparison among RCD-PLSA, CD-PLSA, D-PLSA, BR, LibSVM, and LG on four data sets.

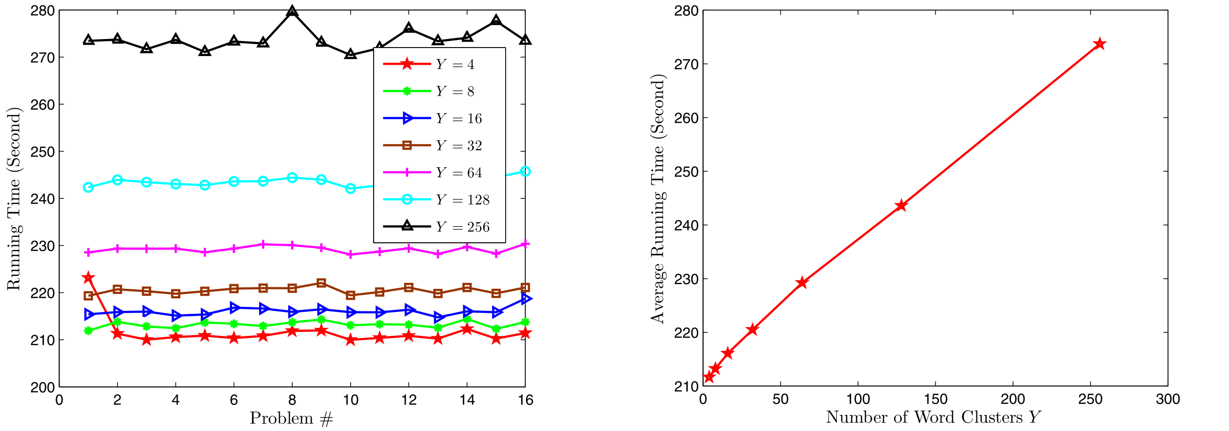
TABLE 5
Average Performances (Percent) on 100 Problem Instances of Each Data Set for Three-Class Classification

	comp vs. rec vs. sci			*comp vs. rec vs. talk*			*comp vs. sci vs. talk*			*rec vs. sci vs. talk*		
	Left	*Right*	*Total*	*Left*	*Right*	*Total*	*Left*	*Right*	*Total*	*Left*	*Right*	*Total*
LG	60.00	75.39	65.85	68.19	80.00	78.31	64.87	73.84	67.83	61.33	72.04	63.26
LibSVM	52.06	62.88	56.17	57.78	67.18	65.86	59.31	64.85	61.14	50.90	57.63	52.12
BR	85.83	95.21	89.40	92.59	96.10	95.61	81.68	90.02	84.43	89.84	95.39	90.84
D-PLSA	81.30	91.30	85.10	91.30	95.87	95.23	76.11	87.65	79.91	88.14	93.80	89.16
CD-PLSA	88.46	94.89	90.90	90.33	93.99	93.48	87.62	92.49	89.23	91.75	94.02	92.16
RCD-PLSA	**91.17**	**96.04**	**93.02**	**94.31**	**96.52**	**96.21**	**92.09**	**94.69**	**92.95**	**94.57**	**96.13**	**94.85**

TABLE 6
Word Concepts with the Corresponding Key Words for Each Domain

Associated with Concept: *Space Science*	Domain 1	rocket, esa, assist, frank, af, thu, helsinki, ron, atlantic, jet, observer, satellite, venus, sei, min, ir, russia, stars, star, ray
	Domain 2	relay, km, rat, pixel, command, elements, arc, acceleration, nasa, earth, fuse, ground, bulletin, pub, anonymous, faq, unix, cit, ir, amplifier
	Domain 3	from, earth, science, word, pictures, years, center, data, national, dale, nasa, gif, reports, mil, planet, field, jpl, ron, smith, unix
	Domain 4	service, archive, unit, magnetic, thousands, technology, information, arc, keys, faq, probes, ir, available, gov, embedded, tens, data, system, unix, mil
Associated with Concept: *Computer Science*	Domain 1	support, astronomer, near, thousands, million, you, vnet, copy, ad, bright, lab, idea, data, hardware, engines, ibm, project, soviet, software, program
	Domain 2	legally, schemes, protected, bytes, mq, disks, patch, registers, machine, pirates, install, card, rom, screen, protection, disk, ram, tape, mb, copy
	Domain 3	discomfort, friend, normal, self, tests, programmer, steve, state, program, lab, you, your, jon, my, headache, trial, she, pain, page, trials
	Domain 4	wcs, cipher, scheme, brute, user, file, encryption, message, serial, decryption, crypto, keys, cryptosystems, skipjack, plaintext, secure, key, encrypted, nsa, des
Associated with Concept: *Car*	Domain 1	saves, power, was, at, disappointment, al, europeans, will, ny, north, their, they, deal, best, year, sports, cs, new, series, gm
	Domain 2	crash, price, vehicle, insurance, handling, gas, xs, dealer, cruiser, leather, buy, latech, fj, paint, ride, buying, bmw, engine, car, honda
	Domain 3	or, value, they, wade, good, car, better, best, three, performance, more, runner, than, average, dl, extra, base, cs, al, year
	Domain 4	dealer, camry, saab, engine, eliot, requests, mazda, liter, mustang, diesel, wagon, nissan, mileage, byte, saturn, toyota, si, cars, car, db

* Domain 1: *rec.sport.hockey* vs. *sci.space*, Domain 2: *rec.motorcycles* vs. *sci.electronics*
Domain 3: *rec.sport.baseball* vs. *sci.med*, Domain 4: *rec.autos* vs. *sci.crypt*.



(a) The Running Time of CD-PLSA on the Data Set *rec vs. sci*

(b) The Relationship between the Running Time and Word Clusters Y

Fig. 8. The running time of CD-PLSA.

words of a concept in different domains. These key words, the biproduct of our model, coincide with our assumption that different domains use different words to describe the same concept.

5. We also investigate the efficiency of the CD-PLSA model. The results indicate CD-PLSA runs very fast, and can always finish when the number of word cluster Y is set to 64 in our experiments. Also, the running time is linear to Y.

7 Concluding Remarks

In this paper, we investigated how to exploit the extension and intension of word and document concepts for cross-domain learning. The extension of word (document) concepts differs in various domain (known as *distinction*), but the intension of word (document) concepts is domain-independent (known as *commonality*). Along this line, we developed a two-phase cross-domain method for text classification. Specifically, a CD-PLSA model is first learned to effectively capture the distinction and commonality across multiple domains in a collaborative way. Then, we further mined the intrinsic structure of target domains by refining the outputs from CD-PLSA (called RCD-PLSA). Finally, extensive experimental results show that CD-PLSA and RCD-PLSA significantly outperform the baseline methods on the tasks with multiple source domains or multiple target domains, and they are more tolerant to distribution differences for transfer learning. Also, the results show the effectiveness of the refinement process for further improving the classification accuracy.

Acknowledgments

The authors Fuzhen Zhuang, Qing He, and Zhongzhi Shi are supported by the National Natural Science Foundation of China (No. 60933004, 60975039, 61035003, 60903141, 61072085), National Basic Research Priorities Programme (No. 2007CB311004) and National Science and Technology Support Plan (No. 2006BAC08B06). A preliminary version of this work has been published in ACM CIKM 2010 [1].

References

[1] F.Z. Zhuang, P. Luo, Z.Y. Shen, Q. He, Y.H. Xiong, Z.Z. Shi, and H. Xiong, "Collaborative Dual-PLSA: Mining Distinction and Commonality across Multiple Domains for Text Classification," *Proc. 19th ACM Conf. Information and Knowledge Management (CIKM '10)*, 2010.

[2] W.Y. Dai, Y.Q. Chen, G.R. Xue, Q. Yang, and Y. Yu, "Translated Learning: Transfer Learning across Different Feature Spaces," *Proc. Advances in Neural Information Processing Systems (NIPS)*, 2008.

[3] J. Gao, W. Fan, J. Jiang, and J.W. Han, "Knowledge Transfer via Multiple Model Local Structure Mapping," *Proc. 14th ACM Int'l Conf. Knowledge Discovery and Data Mining (SIGKDD)*, pp. 283-291, 2008.

[4] D. Xing, W. Dai, G. Xue, and Y. Yu, "Bridged Refinement for Transfer Learning," *Proc. 11th European Conf. Practice of Knowledge Discovery in Databases (PKDD)*, pp. 324-335, 2007.

[5] J. Gao, W. Fan, Y.Z. Sun, and J.W. Han, "Heterogeneous Source Consensus Learning via Decision Propagation and Negotiation," *Proc. 15th ACM Int'l Conf. Knowledge Discovery and Data Mining (SIGKDD)*, pp. 283-291, 2009.

[6] W. Dai, G. Xue, Q. Yang, and Y. Yu, "Co-clustering Based Classification for Out-of-Domain Documents," *Proc. 13th ACM Int'l Conf. Knowledge Discovery and Data Mining (SIGKDD)*, pp. 210-219, 2007.

[7] P. Luo, F.Z. Zhuang, H. Xiong, Y.H. Xiong, and Q. He, "Transfer Learning from Multiple Source Domains via Consensus Regularization," *Proc. 17th ACM Conf. Information and Knowledge Management (CIKM)*, pp. 103-112, 2008.

[8] W. Dai, Q. Yang, G. Xue, and Y. Yu, "Boosting for Transfer Learning," *Proc. 24th Int'l Conf. Machine Learning (ICML)*, pp. 193-200, 2007.

[9] G.R. Xue, W.Y. Dai, Q. Yang, and Y. Yu, "Topic-bridged PLSA for Cross-Domain Text Classification," *Proc. 31st ACM Ann. Int'l Conf. Research and Development in Information Retrieval (SIGIR '08)*, pp. 627-634, 2008.

[10] W.Y. Dai, O. Jin, G.R. Xue, Q. Yang, and Y. Yu, "Eigen Transfer: A Unified Framework for Transfer Learning," *Proc. 26th Ann. Int'l Conf. Machine Learning (ICML)*, pp. 193-200, 2009.

[11] J. Jiang and C.X. Zhai, "A Two-Stage Approach to Domain Adaptation for Statistical Classifiers," *Proc. 16th ACM Conf. Information and Knowledge Management (CIKM)*, pp. 401-410, 2007.

[12] 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," *Proc. SIAM Int'l Conf. Data Mining (SDM)*, pp. 13-24, 2010.

[13] S.J. Pan, J.T. Kwok, and Q. Yang, "Transfer Learning via Dimensionality Reduction," *Proc. 23rd Conf. Artificial Intelligence (AAAI)*, pp. 677-682, 2008.

[14] Q.Q. Gu and J. Zhou, "Learning the Shared Subspace for Multi-task Clustering and Transductive Transfer Classification," *Proc. Int'l Conf. Data Mining (ICDM)*, 2009.

[15] S.H. Xie, W. Fan, J. Peng, O. Verscheure, and J.T. Ren, "Latent Space Domain Transfer between High Dimensional Overlapping Distributions," *Proc. ACM Conf. World Wide Web (WWW)*, pp. 91-100, 2009.

[16] J. Jiang and C.X. Zhai, "Instance Weighting for Domain Adaptation in NLP," *Proc. 45th Ann. Meeting of the Assoc. for Computational Linguistics (ACL)*, pp. 264-271, 2007.

[17] M. Dredze, A. Kulesza, and K. Crammer, "Multi-Domain Learning by Confidence-Weighted Parameter Combination," *J. Machine Learning*, vol. 79, pp. 123-149, 2009.

[18] C.X. Zhai, A. Velivelli, and B. Yu, "A Cross-collection Mixture Model for Comparative Text Mining," *Proc. 10th ACM SIGKDD Conf. Knowledge Discovery and Data Mining (KDD)*, pp. 743-748, 2004.

[19] V.N. Vapnik, *Statictic Learning Theory.* Wiely-Interscience, 1998.

[20] T. Hofmann, "Probabilistic Latent Semantic Analysis," *Proc. 15th Conf. Uncertainty in Artificial Intelligence (UAI)*, pp. 289-296, 1999.

[21] Y. Jiho and S.J. Choi, "Probabilistic Matrix Tri-factorization," *Proc. IEEE Int'l Conf. Acoustics, Speech, and Signal Processing*, pp. 1553-1556, 2009.

[22] A.P. Dempster, N.M. Laird, and D.B. Rubin, "Maximum Likelihood from Incomplete Data via the Em Algorithm," *J. Royal Statistical Soc.*, vol. 39, no. 1, pp. 1-38, 1977.

[23] S. Borman, "The Expectation Maximization Algorithm," http://www.seanborman.com/publications, Technical report, A Short Tutorial, Unpublished Paper, 2004.

[24] D. Hosmer and S. Lemeshow, *Applied Logistic Regression.* Wiley, 2000.

[25] C.C. Chang and C.J. Lin, "LIBSVM: A Library for Support Vector Machines," http://www.csie.ntu.edu.tw/cjlin/libsvm, 2001.

Fuzhen Zhuang is currently working toward the PhD degree in the Institute of Computing Technology, Chinese Academy of Sciences. His research interests include transfer learning, machine learning, data mining, distributed classification and clustering, and natural language processing.

Ping Luo received the PhD degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences. He is currently a research scientist at the Hewlett-Packard Labs, China. He has published several papers in some prestigious refereed journals and conference proceedings, such as the *IEEE Transactions on Information Theory*, *IEEE Transactions on Knowledge and Data Engineering*, *Journal of Parallel and Distributed Computing*, ACM SIGKDD, ACM CIKM, and IJCAI. His research interest include knowledge discovery and machine learning. He is the recipient of the Doctoral Dissertation Award, China Computer Federation, 2009. He is a member of the IEEE Computer Society and the ACM.

Zhiyong Shen received the bachelor's degree in statistics from the Department of Probabilities and Statistics, School of Mathematics Sciences, Peking University, in 2003 and the PhD degree from the State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, in 2009. He is currently working as a postdoctoral fellow at Hewlett-Packard Labs, China. His research interests include data mining and machine learning.

Qing He received the BS degree from Hebei Normal University, Shijiazhang, China, in 1985, and the MS degree from Zhengzhou University, Zhengzhou, China, in 1987, both in mathematics. He received the PhD degree in 2000 from Beijing Normal University in fuzzy mathematics and artificial intelligence, Beijing, China. He is a professor in the Institute of Computing Technology, Chinese Academy of Science (CAS), and at the Graduate University of Chinese (GUCAS). Since 1987 to 1997, he has been with Hebei University of Science and Technology. He is currently a doctoral tutor at the Institute of Computing and Technology, CAS. His research interests include data mining, machine learning, classification, fuzzy clustering.

Yuhong Xiong received the PhD degree in electrical engineering and computer sciences from UC Berkeley and the BS degree in electronic engineering from Tsinghua University in Beijing, China. He is the chief scientist at Lashou.com. His current research interests include data mining applications in e-commerce.

Zhongzhi Shi is a professor in the Institute of Computing Technology, Chinese Academy of Science, leading the Research Group of Intelligent Science. His research interests include intelligence science, multi-agent systems, Semantic Web, machine learning, and neural computing. He won a second-Grade National Award at Science and Technology Progress of China in 2002 and two second-Grade Awards at Science and Technology Progress of the Chinese Academy of Sciences in 1998 and 2001, respectively. He serves as the vice president for the Chinese Association of Artificial Intelligence. He is a senior member of the IEEE, member of AAAI and ACM, chair for the WG 12.2 of IFIP.

Hui Xiong received the BE degree from the University of Science and Technology of China, the MS degree from the National University of Singapore, Singapore, and the PhD degree from the University of Minnesota. He is currently an associate professor in the Management Science and Information Systems Department at Rutgers University. He has published more than 90 technical papers in peer-reviewed journals and conference proceedings. He is a coeditor of *Clustering and Information Retrieval* (Kluwer Academic Publishers, 2003) and a co-editor-in-chief of *Encyclopedia of GIS* (Springer, 2008). He is an associate editor of the *Knowledge and Information Systems* journal and has served regularly on the organization committees and the program committees of a number of international conferences and workshops. His research interests include data and knowledge engineering, with a focus on developing effective and efficient data analysis techniques for emerging data intensive applications. He was the recipient of the 2008 IBM ESA Innovation Award, the 2009 Rutgers University Board of Trustees Research Fellowship for Scholarly Excellence, the 2007 Junior Faculty Teaching Excellence Award, and the 2008 Junior Faculty Research Award at the Rutgers Business School. He is a senior member of the ACM and the IEEE.