Topic-bridged PLSA for Cross-Domain Text Classification

Gui-Rong Xue[1] Wenyuan Dai[1] Qiang Yang[2] Yong Yu[1]

[1]Shanghai Jiao Tong University
800 Dongchuan Road, Shanghai 200240 China
{grxue, dwyak, yyu}@apex.sjtu.edu.cn

[2]Hong Kong University of Science and Technology
Clearwater Bay, Kowloon, Hong Kong
qyang@cse.ust.hk

ABSTRACT

In many Web applications, such as blog classification and newsgroup classification, labeled data are in short supply. It often happens that obtaining labeled data in a new domain is expensive and time consuming, while there may be plenty of labeled data in a related but different domain. Traditional text classification approaches are not able to cope well with learning across different domains. In this paper, we propose a novel cross-domain text classification algorithm which extends the traditional *probabilistic latent semantic analysis* (PLSA) algorithm to integrate labeled and unlabeled data, which come from different but related domains, into a unified probabilistic model. We call this new model *Topic-bridged PLSA*, or TPLSA. By exploiting the common topics between two domains, we transfer knowledge across different domains through a *topic-bridge* to help the text classification in the target domain. A unique advantage of our method is its ability to maximally mine knowledge that can be transferred between domains, resulting in superior performance when compared to other state-of-the-art text classification approaches. Experimental evaluation on different kinds of datasets shows that our proposed algorithm can improve the performance of cross-domain text classification significantly.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval: *Classification and Clustering.*

General Terms

Algorithms, Performance, Experimentation.

Keywords

Topic-bridged PLSA, Cross-Domain, Text Classification

1. INTRODUCTION

In the traditional supervised learning framework, a classification task is to first train a classification model on a labeled training data. Then, the learned model is used to classify a test data set. Generally speaking, under such a framework, the learning algorithm relies on the availability of a large amount of labeled data. In practice, high-quality labeled data are often hard to come by, especially for learning tasks in a new domain. Labeling data in a new domain involves much human labor and is time-consuming. Fortunately, there are often plenty of labeled data from a related but different domain. This may be the case when the labeled data are out-of-date, but the new data are obtained from fast evolving information sources such as the Web blogs. The situation is more serious in current dynamically-changing Web environment. Unfortunately, traditional learning approaches cannot cope well with such a situation. For example, as described in [16], a text classification model trained on a Yahoo! directory performs poorly on a Weblog classification problem, since the distribution of terms may differ significantly. Another example is newsgroup classification, where the news items for financial news may have a different distribution from the sporting events. It is often true that the performance of such classification decreases dramatically along the time dimension. This poses a new learning problem, because we often have to spend much effort in labeling the old documents, but it will cost us much to label the new ones. It would be a waste to throw away the old labeled documents entirely. In such a situation, how to accurately classify the new test data using as much of the old data as possible becomes a critical problem.

In this paper, we focus on the problem of cross-domain text classification. We are given two data sets D_L and D_U, which are from two related but different domains. Here D_L is a labeled data set from old domain, while D_U is from a new domain and needs to be classified. We assume that, the class labels in D_L and the labels to be predicted in D_U are drawn from the same class-label set C. Under such circumstances, our objective is to accurately classify the documents in D_U by fully utilizing the old domain data D_L and their labels.

We propose a novel approach called *Topic-bridged PLSA* (or TPLSA for short) for the cross-domain text classification problem. Our intuition is derived from the observation that the data in two domains may share some common topics, since the two domains are assumed to be relevant. Our key idea is to extend PLSA [8] to build a *topic-bridge* and then transfer the common topics between two domains. Using our TPLSA model, the common knowledge between two domains can be extracted as a prior knowledge in the model, and then can be transferred to the test domain through the bridge with respect to common latent topics. Overall, we combine this knowledge and the new knowledge learned from the unlabeled data for text classification in the test domain, even when the test domain has a different distribution from the training domain.

In particular, we first extend the probabilistic latent semantic analysis (PLSA) [8] to incorporate both labeled and unlabeled data. Our extension allows us to use the hidden variables in PLSA as topics (or classes in classification setting) to bridge the documents in training and test domains, and learn under a joint probabilistic model. Such a new model is based on a simultaneous decomposition of the contingency tables associated with term occurrence knowledge in documents from both training and test domains, which identifies the principal topics of the training data as well as documents in the test data that support those topics. These topics are then taken as a bridge between the training and test domains. In addition, more prior knowledge from the training data is encoded in a set of constraints imposed between docu-

SIGIR'08, July 20-24, 2008, Singapore.
Copyright 2008 ACM 978-1-60558-164-4/08/07...$5.00.

ments, including the must-link constraints for documents to belong to the same cluster and cannot-link constraints for documents to belong to the different clusters. This prior knowledge is learned and applied for learning a classifier on the test data from different domains. According to the above algorithm, an object function is given involving both the likelihood on all the data and the pair-wise constraints on the labeled training data. The EM algorithm is applied to iteratively maximizing such object function to acquire the final categories for the test data.

A major advantage of our work is that by extending the PLSA model for data from both training and test domains, we are able to delineate nicely parts of the knowledge through TPLSA that is constant between different domains and parts that are specific to each data set. This allows the transferring of the learned knowledge to be naturally done *even when the domains are different between training and test data*. We conduct experiments to test the performance of TPLSA on 11 different datasets and compare our PLSA-based cross-domain text classification algorithm with other state-of-the-art algorithms. Experimental results prove that topic-bridged PLSA can achieve higher performance than other algorithms.

The paper is organized as follows. In Section 2, we give a brief review of related work. In Section 3, topic-bridged PLSA is proposed for cross-domain text classification. The evaluation results are shown in Section 4. Section 6 concludes with a summary and suggestion for future work.

2. RELATED WORK

The traditional classification learning assumes that the class labels are given for training data under the same distribution as the test data. Two schemes are generally considered, including supervised learning and semi-supervised learning. Supervised learning focuses on the case where the labeled data are sufficient. Naive Bayes classifiers [13] and support vector machines [11] are known as two of the most effective methods for supervised text classification. Semi-supervised learning [25] addresses the problem that the labeled data are too few to build a good classifier. It makes use of a large amount of unlabeled data, together with a small amount of the labeled data to enhance the classification. Examples for semi-supervised learning include EM-based methods [17], transductive learning [12] etc. Both supervised and semi-supervised classification requires the labeled and unlabeled data should be under the same distribution. However, in our problem, the labeled and unlabeled data come from different domains, and their underlying distributions are often different from each other. This violates the basic assumption of traditional classification learning.

Correcting sample selection bias [24] is another branch of work related to cross-domain classification. If we assume the domain discrepancy is only caused by *sample selection bias*, while other facts are ignored, we can simply apply the theory of correcting sample selection bias to solve the cross-domain classification problem. Sample selection bias [7], the Nobel-prize winning work in Economics in 2000, assumes that the distribution difference is resulted from using non-randomly selected samples from the universe. Sample selection bias was firstly introduced in the econometrics [7], and then came into the field of machine learning [24]. Zadrozny [24] proposed a two-step approach for correcting sample selection. The probability density is estimated to model the selection of training instances, and then the sample selection bias is corrected based on the estimated density. Several researches addressed the selection probability density estimation, e.g. kernel density estimation [20] and kernel mean matching [9]. Blitzer et al. [2] also analyzed the learning bounds for cross-domain learning based on instance weighting. However, those algorithms did not investigate the rich structure of the test data.

Learning from auxiliary data [21] sources addresses classification using small amount of high quality base training data and large amount of low quality *auxiliary* training data. For cross-domain learning, we can consider the base and auxiliary training data come from different domains. In this area, Wu et al. [21] proposed an SVM-based algorithm using both base and auxiliary training data. They demonstrated some improvement by using the auxiliary data. After that, several other researches emerged to improve this learning task, e.g. active learning [14], and boosting [5]. Different from these works, in our problem, there are not any base labeled data.

The probabilistic latent semantic analysis [8] is a popular method for document clustering and related tasks. PLSA was derived from the well-known latent semantic analysis (LSA) for text analysis, and provides solid statistical foundation. In this model, each document is considered as the convex combination of several topics, where these topics or latent semantic variables are obtained using the maximum-likelihood principle. An expansion of PLSA is proposed in [4], which integrates document content and hypertext connectivity for document clustering. Similar to other clustering methods, PLSA does not need labeled information, and thus does not consider the available prior knowledge of the domain. We adapt this model to include the labeled information on topics, thus arriving at a new model we call topic-bridged PLSA (TPLSA) that uses the training documents as source to extract the prior knowledge. By making use of this prior knowledge, our algorithm is able to find documents that relevant for each topic even when these documents are from different domains.

In this paper, we will incorporate the must-link and cannot-link constraints into our TPLSA model, which borrowed the idea from semi-supervised clustering. Semi-supervised clustering [1] builds clusters under some additional constraints provided by a few labeled data, in the form of must-links (two examples must in the same cluster) and cannot-links (two examples cannot in the same cluster). It finds a balance between satisfying these constraints and optimizing the original clustering objective function. Several semi-supervised clustering algorithms have been proposed, including [1][3][10]. Our algorithm is essentially a classification algorithm in which the constraints given by the training data provide a class structure. It will be shown theoretically and empirically that our algorithm works well for cross-domain classification. In addition, we are interested in using the class-label knowledge gained from source domain training data to help classify documents in the target domain, which is not solvable by traditional semi-supervised clustering algorithms alone.

3. TOPIC-BRIDGED PLSA

3.1 Problem Definition

In our problem, each training instance d is a text document and is assigned to a unique output label from a topic set $C = \{c_1, \ldots, c_k\}$. A vocabulary of words $W = \{w_1, \ldots, w_v\}$ is given, allowing each input document to be represented as a “bag-of-words” vector via term frequency. These labeled documents are $D_L = \{d_1, \ldots, d_m\}$, where each $d_l \in D_L$ is assigned with label $c_i \in C$. The test data are $D_U = \{d_1, \ldots, d_n\}$, which are the unlabeled documents for prediction. In this paper, we assume the training dataset D_L are from the related but different domain with test dataset D_U. Our objective is to assign the labels $c_i \in C$ to $d_u \in D_U$ as accurately as possible using the training data D_L in another domain.

3.2 Learning under the Topic-bridged PLSA

Since the documents in D_L and D_U are generally composed of terms, we can decompose two parts of the TPLSA on D_L and D_U separately. According to the observation of D_L and W, we can perform a PLSA on $D_L \times W$ as:

$$\Pr(d_l \mid w) = \sum_z \Pr(d_l \mid z) \Pr(z \mid w) \tag{1}$$

where $d_l \in D_L$ is the document in training set.

For the test data D_U, according to the observation of D_U and W, we can perform the PLSA on $D_U \times W$:

$$\Pr(d_u \mid w) = \sum_z \Pr(d_u \mid z) \Pr(z \mid w) \tag{2}$$

where $d_u \in D_U$ is the document in test set. Note that, D_L and D_U come from different domains.

3.2.1 *Bridging Multiple Domains with PLSA*

Analogous to [4], our key observation is that even if the domains are different between the training and test datasets, they are related and still share similar topics from the terms. Rather than applying two separate PLSAs, it is reasonable to integrate the two models into a joint probabilistic model, and use the hidden variable z to bridge the training and test domains. This is the essential part of TPLSA as shown in Figure 1.

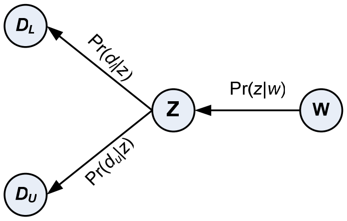

Figure 1. TPLSA Model for Bridging Training and Test Data

By Equations (1) and (2), both decompositions share the same term-specific mixing part Pr(z | w). They relate the conditional probabilities for training documents and test documents: each topic has some probability Pr(d_l | z) of generating a training document d_l with one distribution as well as some probability Pr(d_u | z) of generating a test document d_u with other distribution. As a result, we can merge training documents and test documents into an integrated model. Since the mixing topics are shared, the learned decompositions must be *consistent* with training and test data. We define such mixing topic z as a bridge, and the knowledge transferred between training and test domains should be *consistent* between two domains as well. In the knowledge transferring application, such a bridge allows the model to take evidences about training data structure into account when predicting the categories of test data. Meanwhile, the structure of test data is also be exploited simultaneously under the coupling model. Thus, the model can capture the inherent distribution of test data.

After the decomposition is learned on both training and test data jointly, the class information of the test data can be acquired through the categories of the training data. We integrate the two models using a relative weight parameter $\lambda \in (0, 1)$, which is introduced to represent the trade-off of the weighting between training data and test data. Clearly, if λ is near 0, we trust the test data more and training data less. If λ approaches to 1, we rely heavily on the training data. After the integration, we can maximize the following log-likelihood function with the relative weight λ.

$$L = \sum_w \left[\lambda \sum_{d_l} n(w, d_l) \log \sum_z \Pr(d_l \mid z) \Pr(z \mid w) + (1-\lambda) \sum_{d_u} n(w, d_u) \log \sum_z \Pr(d_u \mid z) \Pr(z \mid w) \right] \tag{3}$$

3.2.2 *Topic Constraints on PLSA*

To fully utilize the knowledge in the training domain, we apply the idea of must-link constraints and cannot-link constraints used in semi-supervised clustering [3] to our model: for all d_l^i, d_l^j in a same topic, there is a must-link constraint for all pairs (d_l^i, d_l^j); for all d_l^i, d_l^j in two different topics, there must have a cannot-link constraint. We encode these two kinds of constraints as follows:

- To enforce the must-link constraint for (d_l^i, d_l^j), we add the following penalty term:

  $$f_s(d_l^i, d_l^j) = \log \sum_z \Pr(d_l^i \mid z) \Pr(d_l^j \mid z) \tag{4}$$

  Note that Pr(d_l^i | z) Pr(d_l^j | z) represents the probability that two documents d_l^i and d_l^j generated by same topic z, and the penalty term f_s(d_l^i, d_l^j) denotes the log-probability that two documents d_l^i and d_l^j are on the same topic.
- In the opposite way, the cannot-link constraint for two documents (d_l^i, d_l^j) can be enforced as the penalty terms of f_d(d_l^i, d_l^j) as

  $$f_d(d_l^i, d_l^j) = \log \sum_{z_i \neq z_j} \Pr(d_l^i \mid z_i) \Pr(d_l^j \mid z_j) \tag{5}$$

Considering the penalty terms of f_s(d_l^i, d_l^j) and f_d(d_l^i, d_l^j), together with the log-likelihood function L, the objective function can be expressed as follows.

$$L_c = L + \beta_1 \sum f_s(d_l^i, d_l^j) + \beta_2 \sum f_d(d_l^i, d_l^j) \tag{6}$$

Recall that L is the log-likelihood of the observed data. The second term of Equation (6) denotes that all documents known to be on the same topic should be grouped into same cluster, while the third term denotes that all documents in different topic should be grouped into different clusters. Here β_1 and β_2 are the parameters to set the weights for the must-link and cannot-link constraints during estimation. According to our experiments, the performance is not very sensitive to the value $\beta 1$ and β_2.

3.2.3 *Optimization for Objective Function L_c*

Since the objective function L_c is non-convex, EM algorithm [6] is used to find a local optimal solution of L_c in Equation (6).

For the E-step, we use the following formulae for the posterior probabilities of the latent variables associated with each observation:

$$\Pr(z \mid d_l, w) = \frac{\Pr(d_l \mid z) \Pr(z \mid w)}{\Pr(d_l \mid w)} \tag{7}$$

$$\Pr(z \mid d_u, w) = \frac{\Pr(d_u \mid z) \Pr(z \mid w)}{\Pr(d_u \mid w)} \tag{8}$$

As the optimization on constraints f_s(d_l^i, d_l^j) and f_d(d_l^i, d_l^j), we use the expression for C(d_l, w, z) to represent the similar meaning to Pr(z | d, w) in the traditional PLSA. For all pairs d_l^i, d_l^j under the same category z, $C_s(d_l^i, d_l^j, z)$ can be interpreted as the probability of a topic z on the condition that two documents belong to same topic. Similarly, for all pairs d_l^i, d_l^j in two different categories, $C_d(d_l^i, d_l^j, z_i, z_j)$ is the probability that two documents belong to two different topics z_i and z_j. The terms C_s and C_d correspond to the must-link and cannot-link constraints, respectively. Specifically, $C(d_l, w, z)$, $C_s(d_l^i, d_l^j, z)$ and $C_d(d_l^i, d_l^j, z_i, z_j)$ are estimated by:

$$C(d_l, w, z) = \frac{\Pr(z \mid w)\Pr(d \mid z)}{\sum_z \Pr(z \mid w)\Pr(d \mid z)} \tag{9}$$

$$C_s(d_l^i, d_l^j, z) = \frac{\Pr(d_l^i \mid z)\Pr(d_l^j \mid z)}{\sum_z \Pr(d_l^i \mid z)\Pr(d_l^j \mid z)} \tag{10}$$

$$C_d(d_l^i, d_l^j, z_i, z_j) = \frac{\Pr(d_l^i \mid z_i)\Pr(d_l^j \mid z_j)}{\sum_{z_i \neq z_j} \Pr(d_l^i \mid z_i)\Pr(d_l^j \mid z_j)} \tag{11}$$

In the M-step, the probabilities with respect to unlabeled data d_u can be estimated in the similar ways to traditional PLSA, so that

$$\Pr(d_u \mid z) \propto \sum_w n(w, d_u)\Pr(z \mid d_u, w) \tag{12}$$

For labeled data d_l, after $C(d_l, w, z)$, $C_s(d_l^i, d_l^j, z)$ and $C_d(d_l^i, d_l^j, z_i, z_j)$ are determined, the new values of parameters $\Pr(d_l \mid z)$ can be calculated by

$$\begin{aligned}\Pr(d_l \mid z) &\propto \sum_w n(w, d_l)C(d_l, w, z) \\ &+ \beta_1 \sum_{d \in D_L} C_s(d_l, d, z) + \beta_2 \sum_{d \in D_L} \sum_{z': z' \neq z} C_d(d_l, d, z, z')\end{aligned} \tag{13}$$

where β_1 and β_2 are the weight parameters for these constraints. Hence, the estimation of $\Pr(d_l \mid z)$ is for the objective function L_c.

While considering w is co-occurrence with documents in training dataset D_L and with the documents in test dataset D_U, $\Pr(z \mid w)$ is estimated with the mixing properties:

$$\begin{aligned}\Pr(z \mid w) &\propto \lambda \sum_{d_l} n(w, d_l)\Pr(z \mid d_l, w) \\ &+ (1-\lambda)\sum_{d_u} n(w, d_u)\Pr(z \mid d_u, w)\end{aligned} \tag{14}$$

The principal of EM algorithm ensures that the value of the objective function L_c as defined in Equation (6) increases monotonically and *converges* to a local optimization [6]. When the EM algorithm converges, the values of parameters $\Pr(d_u \mid z)$ are used to assign each document to a category.

3.3 Algorithm for TPLSA

We now put the entire model together as topic-bridged PLSA (TPLSA). Our proposed algorithm is an integration on the unified model and the constraints to maximize L_c, $\Pr(d_u \mid z)$, $\Pr(z \mid w)$, and $\Pr(d_l \mid z)$, respectively.

As notations, we use c as the function to represent the category of the document or a hidden factor, $c(d)$ as the category of document d, and $c(z)$ as the category represented by a hidden factor. Our TPLSA algorithm is given in Algorithm 1. Here $L_c(i)$ is the constrained likelihood function at ith iteration.

As described in [8], the time complexity of the standard PLSA is $O(k \cdot N)$, where k is the total number of categories and N is the number of total term-document co-occurrence. In the worst case, N is equal to $v \cdot (m + n)$ where v is the total number of terms while m and n is the total number of documents in training dataset and test dataset. According to Equations (12), (13), and (14), the total time cost for unified model is still equal to $O(k \cdot v \cdot (m + n))$. The time cost of calculating the constraints for Equations (9) to (11) is $O(k \cdot m^2)$. As a result, the total time cost for topic-bridged PLSA is $O(k \cdot m^2 + k \cdot v \cdot (m + n))$.

Algorithm 1 Topic-bridged PLSA

```
Input: Document-term matrices $D_L \times W$ and $D_U \times W$, and the
       predefined topic $c$ for each $d \in D_L$;
Output: Topic $c$ for each $d \in D_U$.
Initialize $\Pr(d_l \mid z)$, $\Pr(d_u \mid z)$, , and $\Pr(z \mid w)$ randomly.
while $L_c$ has not converged to a pre-specified value do
    Estimate the values of $\Pr(z|d_l, w)$, $\Pr(z|d_u, w)$, $C(d_l, w, z)$, $C_s(d_l^i, d_l^j, z)$ and $C_d(d_l^i, d_l^j, z_i, z_j)$ in Equations (7),
        (8), (9), (10) and (11), respectively.
    Maximize the values of $\Pr(d_u \mid z)$, $\Pr(d_l \mid z)$ and $\Pr(z \mid w)$,
    in Equations (12), (13), and (14), respectively.
end while
for each $z$ do
    $c(z) = \arg\max_c |\{d \in D_L \mid c(d) = z\}|$.
end for
for each $d \in D_U$ do
    $c(d) = c(\arg\max_z \Pr(z \mid d))$.
end for
```

4. EXPERIMENTS

In this section we empirically evaluate our algorithm for TPLSA for cross-domain text classification and compare it with other state-of-the-art algorithms.

4.1 Datasets

In order to evaluate the properties of our framework, we conducted experiments with three different text corpora: UseNet news articles (20 Newsgroups[1]), SRAA[2] and Newswire articles (Reuters-21578[3]). Since these three data sets are not originally designed for evaluating cross-domain classification, we split the original data in a way to make the domains of the training and test data different, as follows.

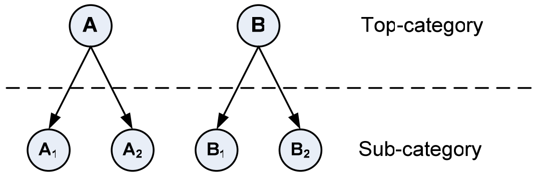

Figure 2. Example of Data Generation

First, we observe that all three data sets have hierarchical structures. For example, 20 Newsgroups corpus contain seven top categories. Under the top categories, there are 20 sub-categories. We define the tasks as top-category classification problems, where our goal is to classify the documents according to the top-one level categories. When we split the data to generate training and test datasets, the data are split based on sub-categories instead of based on random splitting. As shown in Figure 2, for example, **A** and **B** are two top categories, each having two subcategories. Consider a classification task to distinguish the test instances

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

[2] http://www.cs.umass.edu/mccallum/data/sraa.tar.gz

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

between **A** and **B**. Under **A**, there are two sub-categories $\mathbf{A_1}$ and $\mathbf{A_2}$, while $\mathbf{B_1}$ and $\mathbf{B_2}$ are two sub-categories under **B**. We split the data set in such a way that $\mathbf{A_1}$ and $\mathbf{B_1}$ are used as training data, and $\mathbf{A_2}$ and $\mathbf{B_2}$ are used as the test data. Then, the training and test sets contain data in different sub-categories. Their domains also differ as a result.

4.1.1 20 Newsgroups

The 20 Newsgroups is a text collection of approximately 20,000 newsgroup documents, partitioned across 20 different newsgroups nearly evenly. Six different datasets are generated from 20 Newsgroups for evaluating cross-domain classification algorithms. Each data set contains two top categories: one as positive and the other as negative classes. Then, we split the data based on sub-categories, as shown in Table 1. In Table 1, *comp* vs. *sci* indicates that the top category *comp* is treated as positive class and *sci* is as negative. Four sub-categories *comp.graphics*, *comp.os.ms-windows.misc*, *sci.crypt* and *sci.electronics* are chosen to be the training data, while five sub-categories *comp.sys.ibm.pc.hardware*, *comp.sys.mac. hardware*, *comp.windows.x*, *sci.med*, and *sci.space* to be the test data. The other five data sets are organized in the same way.

Table 1. Six Datasets Generated from 20 Newsgroups

Data Set	Traning Data	Test Data
comp vs sci	comp.graphics comp.os.ms-window-misc sci.crypt sci.electronics	comp.sys.ibm.pc.hardware comp.sys.mac.hardware comp.windows.x sci.med sci.space
rec vs talk	rec.autos rec.motocycles talk.politics.guns talk.politics.misc	rec.sport.baseball rec.sport.hockey talk.politics.mideast talk.religion.misc
rec vs sci	rec.autos rec.sport.baseball sci.med sci.space	rec.motocycles rec.sport.hockey sci.crypt sci.electronics
sci vs talk	sci.electronics sci.med talk.politics.misc talk.religion.misc	sci.crypt sci.space talk.politics.guns talk.politics.mideast
comp vs rec	comp.graphics comp.sys.ibm.pc.hardware comp.sys.mac.hardware rec.motocycles rec.sport.hockey	comp.os.ms-window-misc comp.windows.x rec.autos rec.sport.baseball
comp vs talk	comp.graphics comp.sys.mac.hardware comp.windows.x talk.politics.mideast talk.religion.misc	comp.sys.ibm.pc.hardware comp.sys.mac.hardware talk.politics.guns talk.politics.misc

Table 2. Two Datasets Generated from SRAA

Data Set	Traning Data	Test Data
auto vs aviation	sim-auto & sim-aviation	real-auto & real-aviation
real vs simulated	real-aviation & sim-aviation	real-auto & sim-auto

4.1.2 SRAA

SRAA is a Simulated/Real/Aviation/Auto UseNet data set for document classification. 73,218 UseNet articles are collected from four discussion groups about simulated autos (*sim-auto*), simulated aviation (*sim-aviation*), real autos (*real-auto*) and real aviation (*real-aviation*).

Consider a task that aims to predict labels of instances between *real* and *simulated*. We use the documents in *real-auto* and *sim-auto* as training data, while *real-aviation* and *sim-aviation* as test data. Since all the data in the training set are about autos, while all the data in the test set are about aviation, the distributions of feature spaces in the training and test sets are different from each other. The *auto* vs *aviation* data set is generated in the similar way.

4.1.3 Reuters-21578

Reuters-21578 is one of the most used test collections for evaluating automatic text-categorization techniques. It contains five top categories. Among these categories, *orgs*, *people* and *places* are three big ones. For the category *places*, we removed all the documents about the USA to make the three categories nearly even.

Reuters-21578 corpus also has hierarchical structure. The data sets are generated for cross-domain classification in the similar ways as what we have done on the 20 Newsgroups and SRAA corpora. Three datasets *orgs* vs. *people*, *orgs* vs. *places* and *peoples* vs. *places* are generated for cross-domain classification. Since there are too many sub-categories, we do not list the details description here.

4.1.4 Data Preprocessing

Some preprocessing has been applied to the raw text data. First, we perform the Porter stemmer [18] on terms. Then, stop words were removed. A simple feature selection method, Document Frequency (DF) Thresholding [23], is used to cut down the number of features, in order to speed up the classification. Based on [23], DF thresholding is suggested, as the method, which has comparable performance with Information Gain or CHI, is simplest with lowest cost in computation. In our experiments, we set the DF threshold to 3. Finally, TFIDF [19] is used for feature weighting. After that, all the document vectors are converted to a unit vector dividing by its length.

4.1.5 Data Distribution

The first three columns of Table 3 show the statistical properties of the data sets. The first two data sets are from the SRAA corpus. The next six are generated using 20 Newsgroups data set. The last three are from Reuters-21578 test collection. Kullback-Leibler divergence values [15] between the training and test sets in each test are presented in the second column in the table, sorted in decreasing order from top down. The next column titled "Documents" show the size of the data sets used.

Table 3. Datasets for Cross-Domain Classification, including the Performance by SVM and NBC on the Measurement of Accuracy

Data Set	Kullback-Leibler Divergence	Documents		SVM		NBC	
		$\|D_L\|$	$\|D_U\|$	D_L-D_U	D_U-CV	D_L-D_U	D_U-CV
real vs simulated	1.161	8,000	8,000	0.734	0.968	0.741	0.965
auto vs aviation	1.126	8,000	8,000	0.772	0.967	0.85	0.969
rec vs talk	1.102	3,669	3,561	0.767	0.997	0.765	0.994
rec vs sci	1.021	3,961	3,965	0.788	0.993	0.835	0.992
comp vs talk	0.967	4,482	3,652	0.897	0.995	0.976	0.996
comp vs sci	0.874	3,930	4,900	0.683	0.988	0.793	0.979
comp vs rec	0.866	4,904	3,949	0.835	0.992	0.928	0.993
sci vs talk	0.854	3,374	3,828	0.774	0.991	0.774	0.989
orge vs places	0.329	1,079	1,080	0.546	0.915	0.623	0.753
people vs places	0.307	1,239	1,210	0.734	0.887	0.784	0.883
orgs vs people	0.303	1,016	1,046	0.703	0.894	0.711	0.871

KL divergence [15] is calculated as:

$$KL(D_L \| D_U) = \sum_w \Pr_L(w) \log \frac{\Pr_L(w)}{\Pr_U(w)} \qquad (15)$$

where $\Pr_L(w)$ is the estimation of feature w on D_L and $\Pr_U(w)$ is the estimation of feature w on D_U. It can be seen that the Kullback-Leibler divergence (KL) values for all the data sets are much larger than the case when we simply split the same data set into test and training data, which has a KL value of nearly zero.

An example of co-occurrence for document and word distribution on *real* vs. *simulated* dataset is shown in Figure 3. The first 8000 documents are training documents and next 8000 documents are test documents. Among 8000 documents, the first 4000 documents are labeled with *real* and the next 4000 documents are labeled with *simulated*. As shown in the figure, training data has different distribution with the test data. Furthermore, there are still large commonness among them. In this work, we propose to enlarge such commonness to improve the classification over the domains with different distributions.

We further show the performance on these datasets with the other supervised classification algorithms that are typically used for text classification, to illustrate the effect of different domains. As shown in Figure 3, the column $D_U - D_U$ shows the performance when we use the labeled training data D_L to train the model for classification and test the new data D_U while column D_U–CV shows the best case of performing 10-fold cross-validation on D_U. Note that in obtaining the best case for each classifier, the training data are obtained from the labeled data from D_U and the test part is also from D_U while hiding the labels in different folds. Thus, this best case is the case for classification with the same domain, which gives the *best possibly result* for that classifier. As shown in the table, we can find that using a learned model from the different domain data to classify the test data will significantly decrease the performance. There exists a big gap between the worst and best cases for each row.

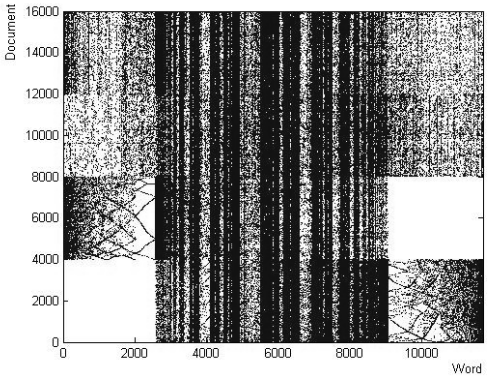

Figure 3. Co-occurrence of Documents and Words for Dataset "*real* vs. *simulated*"

4.2 Compared Algorithms

Three different types of algorithms are employed for comparison with proposed topic-bridged PLSA algorithm. The first two are supervised learning algorithms including NBC, SVM, semi-supervised learning algorithms TSVM [12] and Kernel Density Estimation (KDE) algorithm [24].

Naïve Bayesian classifier (NBC) is a well-known supervised learning algorithm, and SVM has been successfully applied in many applications like text classification [11]. For algorithms that use SVM^{light}, we used a linear kernel function as done in [11].

We further compared topic-bridged PLSA with semi-supervised learning. For semi-supervised learning, we implemented the Transductive Support Vector Machines (TSVM) [12]. For TSVM, we used a linear kernel function as in [12] for the semi-supervised transductive learning.

We also compare to the algorithm which corrects sample selection bias using Kernel Density Estimation [24]. The basic classifier for KDE is based on NB and SVM with linear. We call them as KDE-NB and KDE-SVM, respectively.

4.3 Evaluation Metric

There exist many evaluation metrics for measuring the performance of classification. In this paper, we employ the metric accuracy for comparing different algorithms by considering that it is binary classification. Assume that T is function which maps from document d to its true class label $c = T(d)$, and L be the function which maps from document d to its prediction label $c = L(d)$ by the classifiers. According to the definition in [22], the accuracy is defined as: $Accuracy = |\{d \mid d \in D_U \wedge T(d) = L(d)\}| \,/\, |D_U|$.

4.4 Overall Performance

In this section, two compared experiments are conducted for four algorithms. For the proposed topic-bridged PLSA algorithm, we set the parameters λ, β_1 and β_2 with 0.5, 50 and 15, respectively. The iteration times is set to 100. These parameters will be studied in parameter tuning section.

4.4.1 Performance on Different Datasets

The first experiment is conducted on 11 datasets listed in Table 3. The experimental results are listed in Table 4 with the measure accuracy. As shown in the table, TPLSA can achieve better performance than other algorithms. Unlike the aforementioned algorithms, our algorithm incorporates the training data and test data into a unified model, which can better exploit the knowledge of the training dataset and the inherent structure of test dataset.

Table 4. Performance Comparison for Different Datasets

Data Set	NB	SVM	KDE_NB	KDE-SVM	TSVM	TPLSA
real vs simulated	0.741	0.734	0.764	0.743	0.87	**0.889**
auto vs aviation	0.85	0.772	0.85	0.772	0.898	**0.947**
rec vs talk	0.765	0.767	0.765	0.786	0.96	**0.977**
rec vs sci	0.835	0.788	0.855	0.788	0.938	**0.951**
comp vs talk	0.976	0.897	0.976	0.897	0.903	**0.977**
comp vs sci	0.793	0.683	0.801	0.684	0.817	**0.989**
comp vs rec	0.928	0.835	0.928	0.835	0.902	**0.951**
sci vs talk	0.774	0.774	0.786	0.79	0.892	**0.962**
orgs vs places	0.623	0.546	0.623	0.551	0.564	**0.653**
people vs places	0.784	0.734	0.784	0.742	0.769	**0.805**
orgs vs people	0.711	0.703	0.712	0.704	0.703	**0.763**
Average	0.798	0.748	0.804	0.754	0.838	**0.897**

Since supervised approaches including SVM and NBC do not consider the domain difference on training and test datasets, there algorithms get worse performance. Semi-supervised approach TSVM can achieve higher performance than supervised methods, which prove that inspecting the testing data will achieve higher performance. However, these methods assume that the training data and test data are from the same domain. These methods do not fully utilize the structure information contained in test data with different domain. As a result, the semi-supervised algorithms can not be good enough. Correcting sample selection bias [24], the columns under "Kernel Density Estimation" (KDE), performs only slightly better than NB and SVM classification algorithms, respectively. We believe it is because distribution difference cannot cover all the issues in cross-domain text classification.

As mentioned, the performance of supervised learning algorithms including SVM and NBC will be affected according to the distribution of training data and testing data since they are from different domains. After performing topic-bridged PLSA, we can exploit training data and test data simultaneously. The improvement over the supervised methods is shown in Figure 4. The X-axis represents different datasets while the KL divergence is decreased from 1 to 11. The Y-axis represents the improvement for our proposed algorithm over SVM and NBC. Generally, we can find that the lower KL divergence is, the less improvement will be achieved. It can be explained the general supervised algorithms can also perform good enough on the datasets with lower divergence.

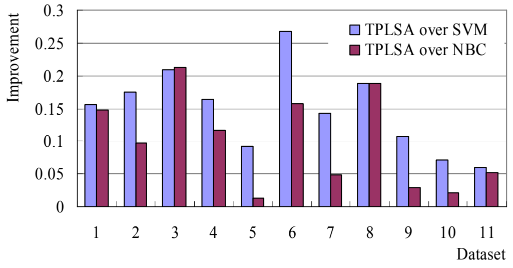

Figure 4. Performance for Different KL divergence

4.4.2 *Performance on Different Size of Training Data*

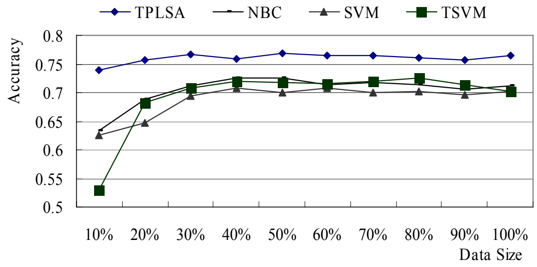

Figure 5. Performance Comparison for Different Data Size

For most of learning algorithms, the density of training data will affect the performance. In this experiment, we conduct experiments on the "*org* vs. *people*" dataset to empirically analyze how classification accuracy evolves when the size of training data is changed from 10%, 20%, …, and 100%. We simulate the different sizes of training data by randomly extracting from the training data. The experimental results are shown in Figure 5. We can find that the accuracy curve of TPLSA is above over the curves of other three algorithms. TPLSA can achieve higher performance over other three algorithms on different size of training data, specifically on few training data. Furthermore, as shown in the figure, our proposed algorithm is not very sensitive to the data size. This also confirms that using the structure information of unlabeled data can achieve better performance.

According to our calculating, the KL divergence between different training data and test data is decreased according to the increasing of training data size. As shown in Figure 5, we can also find that the higher KL divergence is, the more improvement will be achieved.

4.5 Parameters Sensitivity

In this section, the experiments are conducted to tune the parameters and to show that our proposed TPLSA algorithm is *not sensitive* to these parameters. Each parameter is tuned by fixing other parameters.

4.5.1 *Parameters for Must-link and Cannot-link*

The main two parameters of TPLSA are: the coefficients β_1 and β_2 for the penalty terms. The coefficient β_1 represents the degree of enforcement that the documents with same category in labeled data should be in same cluster, while the coefficient β_2 regulates the degree of enforcement that the documents with different categories in training data should be in different clusters. To reveal the effect of β_1 and β_2 on the performance, we fix the value of one of them and vary the other to show the change in performance. The experiment for tuning is conducted on *orgs* vs. *people* dataset.

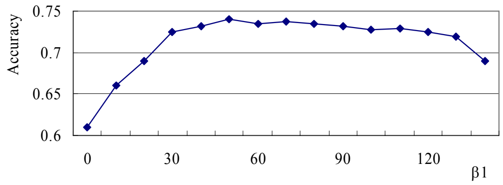

Figure 6. Performance for Different β_1

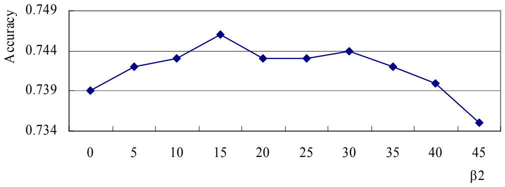

Figure 7. Performance for Different β_2

Figure 6 and Figure 7 show the impact of β_1 and β_2 on the performance of TPLSA. One character of the curves in Figure 6 is that when β_1 is increased from 0 to 150 with interval 10, the performance will be increased firstly and then decreased. This indicates that setting β_1 properly will give a better performance. When the value of β_1 is decreased to close to 0, the impact of the penalty terms for the must-links is removed. As a result, the performance will be decreased, which indicates that the must-link constraints have a strong impact on the performance. Furthermore, as shown in the Figure 6, the value β_1 is relatively stable in the interval [30, 120]. We believe the TPLSA algorithm is not sensitive to the value β_1 since the interval is large enough. As shown in Figure 7, changing β_2 have little impact on the performance. The performance of TPLSA is not sensitive to the value β_2. In our experiments, $\beta_1 = 50$ and $\beta_2 = 15$.

4.5.2 *Performance on Different λ*

As shown in Equation (3), the parameter λ is to tune the weight between labeled training data and unlabeled test data. Here we show the different performance affected by the parameter λ to inspect the sensitiveness of TPLSA. λ is tuned on 11 different datasets in table 3. The experimental results are shown in Figure 8. In the figure, X-axis shows the change of parameter λ which is tuned from 0.1 to 1. As shown in the figure, the performance will first increase and then decrease when λ is increased from 0.1 to 1. It is shown that setting λ with a proper value will achieve higher performance for TPLSA. Furthermore, as shown in the Figure, if the value of λ is changed in the interval from 0.4 to 0.8, the performance of TPLSA will be less impacted across different datasets. It shows that our proposed algorithm is not sensitive to the parameter λ. In this work, we set λ to 0.5 for other experiments.

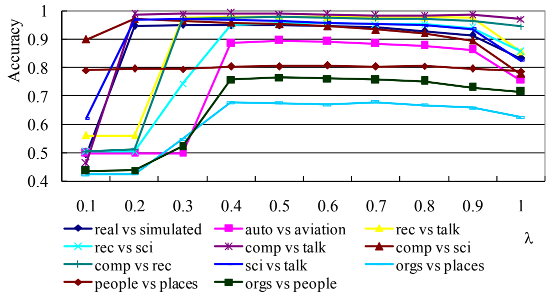

Figure 8. Performance for Different λ

4.6 Convergence

The method that we used to find the optimal objective function L_c as defined in Equation (6) is based on the EM algorithm, which is an iterative process that will converge to a local optimum. Figure 9 shows the change of performance with respect to the number of iterations. We observe that the performance grows faster during the first 60 iterations. The performance is nearly constant when more than 100 iterations are performed. This proves that our algorithm will be converged in about 100 iterations.

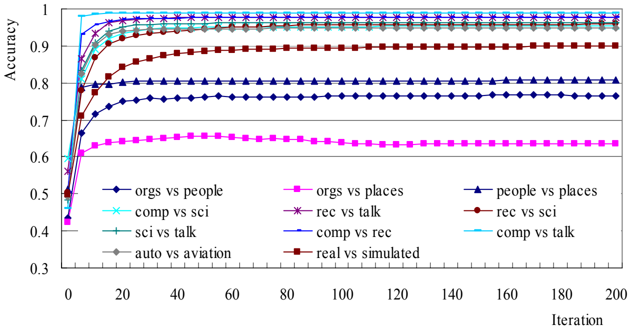

Figure 9. Performance for Iteration

5. CONCLUSION AND FUTURE WORK

In this paper, we proposed a novel algorithm called *Topic-bridged PLSA* to handle the cross-domain text classification problem by allowing knowledge learned from documents in one domain to be effectively transferred to another. The algorithm extends the traditional Probabilistic Latent Semantic Analysis (PLSA) to integrate the labeled training data and unlabeled test data under a joint probabilistic model with the common topic as bridge. We conducted experimental evaluation on 11 datasets and the results show that the proposed algorithm achieves better performance than other state-of-the-art classification algorithms.

As a future work, we will consider other learning methods to acquire the parameters used in the TPLSA model, and consider other related classification tasks such as multi-class classification. Moreover, we plan to do further investigation to inspect the inherent relations among text datasets with different but related domains.

6. REFERENCES

[1] Basu, S., Banerjee, A., and Mooney, R. J. Semi-Supervised Clustering by Seeding. In *ICML*, 2002.

[2] Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., and Wortman, J. Learning Bounds for Domain Adaptation. In *NIPS* 2007.

[3] Cohn, D., Caruana, R., and McCallum, A. Semi-Supervised Clustering with User Feedback.Technical Report TR2003-1892, Cornell University, 2003.

[4] Cohn, D., and Hofmann, T. The Missing Link - a Probabilistic Model of Document Content and Hypertext Connectivity. In *NIPS*, 2001.

[5] Dai, W., Yang, Q., Xue, G.-R., and Yu, Y, Boosting for Transfer Learning. In *ICML*, 2007.

[6] Dempster, A., Laird, N., and Rubin, D. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of Royal Statistical Society*, Series B, 39(1): 1–38, 1977.

[7] Heckman, J. J. Sample Selection Bias as a Specification Error. *Econometrica* 47:153–161, 1979.

[8] Hofmann, T. Probabilistic Latent Semantic Analysis. In *SIGIR*, 1999.

[9] Huang, J., Smola, A., Gretton, A., Borgwardt, K. M., and Schölkopf, B. Correcting Sample Selection Bias by Unlabeled Data. In *NIPS*, 2007.

[10] Ji, X., Xu, W., and Zhu, S. Document Clustering with Prior Knowledge. In *SIGIR*, 2006.

[11] Joachims, T. Text Categorization with Support Vector Machines: Learning with Many Relevant Features. In *ICML*, 1998.

[12] Joachims, T. Transductive Inference for Text Classification using Support Vector Machines. In *ICML,* 1999.

[13] Lewis, D. D. Representation and Learning in Information Retrieval. PhD thesis, Amherst, MA, USA, 1992.

[14] Liao, X., Xue, Y., and Carin, L. Logistic Regression with an Auxiliary Data Source. In *ICML*, 2005.

[15] Kullback, S. and Leibler, R. A. On Information and Sufficiency. *Annals of Mathematical Statistics,* 22(1):79-86, 1951.

[16] Ni, X., Xue, G.-R., Ling, X., Yu, Y., Yang, Q. Exploring in the Weblog Space by Detecting Informative and Affective Articles. In *WWW*, 2007.

[17] Nigam, K., McCallum, A. K., Thrun, S., and Mitchell, T. Text Classification from Labeled and Unlabeled Documents using EM. Machine Learning, 39(2-3):103–134, 2000.

[18] Porter, M. F. An Algorithm for Suffix Stripping. Program 14, 3, pp 130–137, 1980.

[19] Robertson, S. E., Walker, S., Beaulieu, M. M., Gatford, M., and Payne, A. Okapi at TREC-4. In TREC-4, 73-96. 1996.

[20] Shimodaira, H. Improving Predictive Inference under Covariate Shift by Weighting the Log-likelihood Function. *Journal of Statistical Planning and Inference*, 2000.

[21] Wu, P., and Dietterich, T. G. Improving SVM Accuracy by Training on Auxiliary Data Sources. In *ICML*, 2004.

[22] Yang, Y. An Evaluation of Statistical Approaches to Text Categorization. *Journal of Information Retrieval*, Vol. 1, No. 1/2, 67–88, 1999.

[23] Yang, Y. and Pedersen, J.P. A Comparative Study on Feature Selection in Text Categorization. In *ICML*, 1997.

[24] Zadrozny, B. Learning and Evaluating Classifiers under Sample Selection Bias. In *ICML*, 2004.

[25] Zhu, X. Semi-Supervised Learning Literature Survey. CS TR 1530, University of Wisconsin-Madison, 2006.