Learning with Positive and Unlabeled Examples Using Topic-Sensitive PLSA

Ke Zhou, Gui-Rong Xue, Qiang Yang, *Fellow*, *IEEE*, and Yong Yu

Abstract—It is often difficult and time-consuming to provide a large amount of positive and negative examples for training a classification system in many applications such as information retrieval. Instead, users often find it easier to indicate just a few positive examples of what he or she likes, and thus, these are the only labeled examples available for the learning system. A large amount of unlabeled data are easier to obtain. How to make use of the positive and unlabeled data for learning is a critical problem in machine learning and information retrieval. Several approaches for solving this problem have been proposed in the past, but most of these methods do not work well when only a small amount of labeled positive data are available. In this paper, we propose a novel algorithm called Topic-Sensitive pLSA to solve this problem. This algorithm extends the original probabilistic latent semantic analysis (pLSA), which is a purely unsupervised framework, by injecting a small amount of supervision information from the user. The supervision from users is in the form of indicating which documents fit the users' interests. The supervision is encoded into a set of constraints. By introducing the penalty terms for these constraints, we propose an objective function that trades off the likelihood of the observed data and the enforcement of the constraints. We develop an iterative algorithm that can obtain the local optimum of the objective function. Experimental evaluation on three data corpora shows that the proposed method can improve the performance especially only with a small amount of labeled positive data.

Index Terms—Semisupervised learning, topic-sensitive probabilistic latent semantic analysis, document classification.

1 Introduction

Document classification is an important task in information retrieval, data mining, and Web related areas. As a supervised learning task, document classification learns a classifier from labeled positive and negative data. Its objective is to use the classifier to accurately predict the labels for unseen documents.

In practice, both the positive and negative training data are often labeled manually. Thus, collecting the labeled training data is costly. In today's dynamically changing Web environments, many new classification tasks appear all the time, and the labeled training data may be easily outdated. In order to overcome the problem, semisupervised learning has been introduced to train a classifier, which takes unlabeled data into account to reduce the labeled examples needed. In semisupervised learning, only a few examples are needed for each class. Learning algorithms are then adapted to train an optimal model with respect to both labeled and unlabeled data [1], [2], [3], [4], [5].

Even though traditional semisupervised framework works well for many application domains, we observe that in many information retrieval applications, only positively labeled training data and unlabeled data are available. This is because users are usually interested in examples that satisfy their needs and ignore those that they are not interested in. For example, users may put their favorite Web pages in the bookmarks, but they are usually unwilling to mark boring pages. Another reason is that the positive class is usually more specific than the negative class. People can characterize their interests in detail, but they are often unable to specify what they do not like very well.

For the above reasons, the problem of *partially supervised learning* is studied [6], [7], [8], [9]. In traditional classification problems, the learning algorithm learns from the manually labeled data of both positive and negative classes. However, in partially supervised learning, the learning algorithm is based on only labeled positive data and unlabeled data. The key difference between partially supervised learning and traditional supervised learning is that no labeled negative examples are available. As a consequence, both supervised and semisupervised classification methods are inapplicable to the problem of partially supervised learning. Generally, several existing methods first seek instances from unlabeled data that are reliably negative and then try to find more and more negative data iteratively [7], [8]. Another approach transforms the problem of partially supervised learning to the problem of learning with noise [10]. All the unlabeled data are assumed to be negative and a classifier is learned by taking noisy labels into consideration. Both approaches are proved to be effective for training classifiers with only positive and unlabeled examples. But the underlying assumption of these methods is that the number of labeled positive examples is sufficient to capture the characteristics of the positive class. When this assumption fails, that is, when the number of the labeled positive examples is very small, the performance of these algorithms will be poor, as we will see in Section 4.

- *K. Zhou, G.-R. Xue, and Y. Yu are with the Department of Computer Science and Engineering, Shanghai Jiao-Tong University, No. 800 Dongchuan Road, Shanghai, 200240, China. E-mail: {zhouke, grxue, yyu}@apex.sjtu.edu.cn.*
- *Q. Yang is with the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Hong Kong, China. E-mail: qyang@cse.ust.hk.*

Manuscript received 7 Sept. 2007; revised 17 Apr. 2008; accepted 12 Feb. 2009; published online 18 Feb. 2009.
Recommended for acceptance by D. Gunopulos.
For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number TKDE-2007-09-0449.
Digital Object Identifier no. 10.1109/TKDE.2009.56.

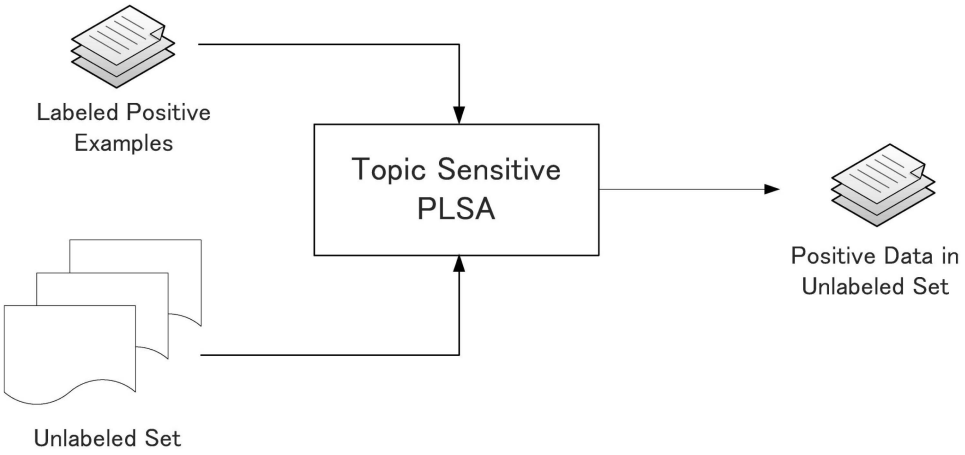


Fig. 1. Framework of topic-sensitive PLSA.

In this paper, we propose a novel method called *topic-sensitive pLSA* to solve the problem of learning from positive and unlabeled data. The proposed topic-sensitive pLSA is based on probabilistic Latent Semantic Analysis (pLSA) [11], which is an unsupervised learning method. Topic-sensitive pLSA enables the original pLSA to take some labeled documents of a specified topic as input and use this information to find documents relevant to the topic from a set of mixed documents that includes also the unlabeled examples (Fig. 1). The supervision from user is encoded into a set of constraints between documents. Two types of constraints are used in our method: *must-link* constraints and *cannot-link* constraints. A must-link constraint represents that the two documents should be grouped into the same cluster. Analogously, a cannot-link constraint means that the two documents should be separated into different clusters. Based on these constraints, we transform the problem of learning from positive and unlabeled data into the one of learning with noisy constraints. In our method, a must-link constraint is applied to each pair of documents in the labeled positive set, and a cannot-link constraint is applied for each pair of documents, where one document is placed in the labeled positive set and the other is in the unlabeled set. Since some documents in the unlabeled set belong to the positive class, the cannot-link constraints are noisy. Thus, the must-link constraints pose heavier penalty if violated than the cannot-link constraints in our methods. This usage of the user input makes our method sensitive to the topic specified by the user, which is why our method is called topic-sensitive pLSA.

Our experimental study shows that when the number of labeled positive examples is very small, the performance of the proposed algorithm is still stable, which makes our method more suitable for the applications with only a small number of positive data and unlabeled mixed data. Many applications of this problem setting can be found. For example, search engines are able to collect the click-through data and perform implicit feedback [12] to refine search results. For the search results, there is a large number of unlabeled documents, which correspond to those that the users do not click on. However, since the users are unlikely to click on many of the search results, there are only a few positive examples available. In this case, many existing algorithms would fail due to the rareness of training data. However, the proposed algorithm can achieve good performance through utilizing the training data effectively.

The main contribution of this paper is that we introduce a novel algorithm for learning with positive and unlabeled data, which extends the pLSA into a semisupervised case. By introducing new constraints to the objective function, we integrate semisupervised learning with clustering. Our method transforms the information of labeled positive data into must-link and cannot-link constraints and performs clustering operations that are restricted by these constraints. Experimental studies show that our method is robust for the situation when there are only a small number of labeled positive examples.

The remainder of this paper is organized as follows: We first review the existing techniques for learning with positive and unlabeled data in Section 2. In Section 3, we describe in detail our topic-sensitive pLSA algorithm. The results of experimental evaluation and analysis of these results are presented in Section 4. Finally, conclusions and future work of this paper are discussed in Section 5.

2 Related Work

2.1 Learning with Positive and Unlabeled Data

Several studies give theoretical foundations of learning from positive and unlabeled data [7], [13]. These studies show the generalization bounds in terms of the sample complexity. These theoretic results show that it is feasible to learning without labeled negative data.

Several practical solutions to the problem of partially supervised learning are designed based on heuristic labeling techniques [7], [9], [8], [14]. These methods first identify a set of reliably negative examples from the unlabeled data and then apply the traditional supervised learning algorithms to train a final classifier. For example, the work of Liu et al. [7] represents a heuristic technique Spy-EM based on the Naive Bayes classifier. This method estimates the parameters of the negative model by a subset of unlabeled examples that are highly reliable to be negative and performs the EM operations to infer the labels of the other unlabeled examples. The negative examples are selected by mixing a subset of positive examples called *spy documents* with the unlabeled examples.

A drawback of Spy-EM is that it works well only when the number of positive examples is large enough, because it needs to split the set of labeled positive examples into the set of positive training examples and the set of spy documents. A similar method, called Positive Examples-Based Learning (PEBL), is proposed in [9]. This method starts from training an SVM classifier with the positive data and an initial set of negative data. The method then uses the obtained model to find more negative examples. The new negative examples are then used to train a new SVM classifier, etc. During each iteration, new negative examples are identified from the unlabeled data set. As a result, more and more negative examples are identified while retaining the positively labeled examples that are correctly labeled. Although these methods work well in practice, their performance depends largely on the reliability of the extracted negative examples, but these initial negative examples are usually identified by heuristical rules or weak classifiers. In fact, these methods are unreliable in the general cases, which limits their application in practice. In contrast, our method is model-based and the objective function is theoretically well-defined.

In [10] and [6], the problem of learning with positive and unlabeled data is transformed into a problem of learning with noise by considering all unlabeled examples as negative and then dealing with the noisy labels by setting different weights in loss function. Specifically, the value of the loss function is large when a labeled positive instance is misclassified to be negative, while the weights are small when a negative instance is misclassified as positive. These methods are based on discriminative supervised classification methods such as logistic regression and SVM. Therefore, their performance largely depends on the number of labeled data. The similar formulation is also used in our proposed method, but our method is generative in nature and based on semisupervised learning. Therefore, our method works well even when the number of labeled examples is relatively small.

Another method related to partially supervised learning is One-class SVM (OSVM) [15]. OSVM transforms the training examples into the feature space and estimates the distribution of positive class from only labeled positive examples. Therefore, in order to train an OSVM classifier, only labeled positive examples are required. Since there is no information about the distribution of the negative class, the number of positive examples should be sufficiently large so that the boundary of the positive class can be induced precisely. In partially supervised learning, the distribution of negative data can be revealed by taking unlabeled data into consideration and the number of labeled positive examples can be reduced.

2.2 Unsupervised and Semisupervised Clustering

Another closely related research field is unsupervised and semisupervised document clustering. Document clustering is an unsupervised learning process that groups documents in a given set into clusters such that documents in the same cluster are more similar to each other than documents in other clusters. There are many algorithms for solving this problem such as k-means, spectral methods [16], [17], and pLSA [11]. These clustering techniques aim to minimize some criteria defined over all possible assignment of clusters. For example, the k-means algorithm minimizes the sum of the squared distance between the data points and the center of the clusters. The pLSA is a popular method for document clustering. Each document is considered as a convex combination of several topics, which are obtained by the maximum likelihood principle. It can be shown that pLSA is equivalent to minimizing the Kullback-Leibler divergence between the empirical distribution of words in a document and the model distribution. Similar to other unsupervised clustering methods, pLSA does not need supervision from the user. In contrast, our proposed topic-sensitive pLSA method uses a few sample documents in the topic of interests as supervision. By making use of this supervision, our algorithm is able to obtain documents relevant to topics specified by the user much better than pLSA alone.

There have also been research works that modify the traditional unsupervised document clustering techniques. An example is semisupervised learning. In [16], semisupervised k-means algorithm is proposed. In [18], [19], the authors proposed extensions of the k-means algorithm that allow soft constraints. A semisupervised normalized cut is introduced by Ji and Xu [20]. In [21], the authors model these two types of constraints in a Bayesian framework. Specifically, the constraints are represented by edges of a Markov random field.

3 Topic-Sensitive pLSA

In this section, we present a document classification method that enables the user to specify the topic of interests by selecting related documents. The documents related to this topic can be identified from a set of unlabeled documents. A novel semisupervised pLSA, called topic-sensitive pLSA, is derived.

The proposed algorithm provides an approach to incorporate the supervision from users into pLSA, which is an unsupervised learning method. This supervision is provided in the form of indicating that several documents belong to the topic of interest. Specifically, these documents are used as positive labeled examples. Through learning from both positively labeled and unlabeled documents, the algorithm is able to identify the documents related to the topic of interest in the unlabeled set. Unlike the previous methods of learning from positive and unlabeled data, our method works well when only a small number of labeled examples are available. This is because:

1. Most previous methods are based on supervised learning framework. For example, the PEBL algorithm proposed in [9] uses SVM as the basic classifier and weighted logistic regression [10] is based on logistic regression. However, the underlying assumption of these methods is that the class distribution can be estimated precisely from labeled data. The distribution of unlabeled data is ignored in these methods. Because our method is based on unsupervised learning, the distribution of unlabeled data is properly used. When there are only a few labeled examples, the distribution of the unlabeled data becomes very important to achieve a good performance.

2. Although Spy-EM performs Expectation Maximization (EM) iterations to estimate the distribution of unlabeled data, this method is based on Naive Bayes classifier, which assumes that all the negative documents are generated from the same distribution. However, in real-world applications, the negative documents may be quite diverse and are related to many different topics. In this case, Spy-EM is not able to estimate the distribution of negative class accurately. In our method, the negative class is modeled as a mixture of several different distributions. As a result, the distribution of the negative class can be estimated more accurately. As shown in the experiments (Section 4.3), fitting the distribution of negative class can enhance the performance, especially when the number of positive examples is extremely small.

We first review the previous work with probabilistic latent semantic analysis proposed in [11], which is the basis of our work. Then the objective function of our method is proposed. This objective function trades off the likelihood of observed data and the enforcement of the constraints generated from the supervision of users. Finally, we propose an algorithm to obtain the local optimum of this objective function.

3.1 Probabilistic Latent Semantic Analysis

pLSA [11], [22] is originally developed for providing a probabilistic method for the analysis of co-occurrence data. pLSA has long been used in text document retrieval, clustering, and related areas. Let $\mathcal{D}$ denote the set of documents and $\mathcal{W}$ denote the vocabulary set. Then the corpus of documents is represented by a co-occurrence matrix n of size $|\mathcal{D}| \times |\mathcal{W}|$, with each entry $n(d, w)$ is the number of times that the word w occurs in document d. In pLSA, an unobservable class variable $z \in \mathcal{Z} = \{z_1, \ldots, z_K\}$ is associated with each observation (d, w), the joint probability distribution over $\mathcal{D} \times \mathcal{W}$ can be expressed as follows:

$$\mathrm{P}(d, w) = \mathrm{P}(d)\mathrm{P}(w \mid d), \tag{1}$$

where

$$\mathrm{P}(w \mid d) = \sum_z \mathrm{P}(w \mid z)\mathrm{P}(z \mid d). \tag{2}$$

Equation (2) means that the distribution of words conditioned on documents $\mathrm{P}(w \mid d)$ can be expressed by a convex combination of the topic-specific word distributions $\mathrm{P}(w \mid z)$.

Given the document-term matrix $n(d, w)$, the log-likelihood of observed data can be expressed as

$$\mathcal{L} = \sum_d \sum_w n(d, w) \log \mathrm{P}(d, w), \tag{3}$$

$$= \sum_d \sum_w n(d, w) \log \sum_z \mathrm{P}(w \mid z)\mathrm{P}(z \mid d)\mathrm{P}(d). \tag{4}$$

Following the maximum-likelihood principle, the parameters of the model $\mathrm{P}(d)$, $\mathrm{P}(z \mid d)$, and $\mathrm{P}(w \mid z)$ are determined by maximizing the log-likelihood function.

Since the log-likelihood defined by (4) is nonconvex, the global optimality of the function is difficult to obtain. Thus, we have to seek locally optimal solutions for the log-likelihood function $\mathcal{L}$. To this end, the EM algorithm [23] can be applied. The EM algorithm alternates between two steps: the E-step computes the posterior of the latent variable z based on the current estimation of the parameters and the M-step updates the parameters once the latent variables are known using the posterior estimated in the previous E-step.

For pLSA model, the E-step computes the posterior of the latent variable $\mathrm{P}(z \mid d, w)$ as follows:

$$\mathrm{P}(z \mid d, w) = \frac{\mathrm{P}^{(t)}(d)\mathrm{P}^{(t)}(z \mid d)\mathrm{P}^{(t)}(w \mid z)}{\sum_{z'} \mathrm{P}^{(t)}(d)\mathrm{P}^{(t)}(z' \mid d)\mathrm{P}^{(t)}(w \mid z')}, \tag{5}$$

where $\mathrm{P}^{(t)}(d)$, $\mathrm{P}^{(t)}(z \mid d)$, and $\mathrm{P}^{(t)}(w \mid z)$ are parameters computed in the last round of iteration. Then the M-step updates $\mathrm{P}(d)$, $\mathrm{P}(z \mid d)$, and $\mathrm{P}(w \mid z)$ according to the following equations:

$$\mathrm{P}(w \mid z) \propto \sum_d n(d, w)\mathrm{P}(z \mid d, w), \tag{6}$$

$$\mathrm{P}(z \mid d) \propto \sum_w n(d, w)\mathrm{P}(z \mid d, w), \tag{7}$$

$$\mathrm{P}(d) \propto \sum_w n(d, w). \tag{8}$$

The value of the log-likelihood function defined in (4) is expected to increase and converges to a local maximum by alternating the E-step and M-step defined above.

3.2 Incorporating Topic Information

The proposed topic-sensitive pLSA model enables a user to select an interesting topic by clustering all documents related to the topic together. The topic of interest is specified by a number of documents belonging to this topic. Assume that the user selects a set of documents $\mathcal{T} = \{d_1, d_2, \ldots, d_n\} \subset \mathcal{D}$. It means that the documents in $\mathcal{T}$ all belong to the target topic. Let $\mathcal{U} = \mathcal{D} - \mathcal{T}$ be the unlabeled documents in $\mathcal{D}$. Following the work of Lee and Liu [10], we transform the problem of learning from positive and unlabeled data into a problem of learning with noisy labels. Specifically, we assume that all unlabeled documents belong to the negative class. Then, we generate constraints according to the labeled positive data and the noisy negative data. Inspired by the idea of must-link constraints and cannot-link constraints as they are used in semisupervised clustering, we encode the supervision information by the above-mentioned constraints as follows:

- For all $d_i, d_j \in \mathcal{T}$, d_i, d_j are in the same topic. In other words, there is a must-link constraint for all pairs (d_i, d_j). Similarly, for all $d_i \in \mathcal{T}$ and $d_j \in \mathcal{U}$, there is a cannot-link constraint.
- To enforce the must-link constraint for (d_i, d_j), we add the following penalty term $f_{\mathcal{T}}(d_i, d_j)$ to the log-likelihood function:

$$f_{\mathcal{T}}(d_i, d_j) = \log \sum_z \mathrm{P}(z \mid d_i)\mathrm{P}(z \mid d_j). \tag{9}$$

Note that $\mathrm{P}(z \mid d_i)\mathrm{P}(z \mid d_j)$ represents the probability that d_i and d_j contain the given topic z. The penalty

term $f_{\mathcal{T}}(d_i, d_j)$ denotes the log-probability that two documents d_i and d_j are about the same topic. If both documents d_i and d_j belong to the same topic z with a probability of one, i.e., $\mathrm{P}(z \mid d_j) = \mathrm{P}(z \mid d_j) = 1$ for some z, the value of the penalty term is zero: $f_{\mathcal{T}}(d_i, d_j) = 0$.

Similarly, the cannot-link constraint can be enforced by the penalty term $f_{\mathcal{U}}(d_i, d_j)$ defined as follows:

$$\begin{aligned} f_{\mathcal{U}}(d_i, d_j) &= \log\left(1 - \sum_z \mathrm{P}(z \mid d_i)\mathrm{P}(z \mid d_j)\right) \\ &= \log \sum_{z_1 \neq z_2} \mathrm{P}(z_1 \mid d_i)\mathrm{P}(z_2 \mid d_j), \end{aligned} \tag{10}$$

where $f_{\mathcal{U}}(d_i, d_j)$ represents the log-probability that two documents d_i and d_j do not belong to the same topic. Considering the penalty terms of $f_{\mathcal{T}}(d_i, d_j)$ and $f_{\mathcal{U}}(d_i, d_j)$ together with the log-likelihood function $\mathcal{L}$, the objective function can be expressed as follows:

$$\mathcal{L}_1 = \mathcal{L} + \beta_1 \sum_{d_i, d_j \in \mathcal{T}} f_{\mathcal{T}}(d_i, d_j) + \beta_2 \sum_{d_i \in \mathcal{T}} \sum_{d_j \in \mathcal{U}} f_{\mathcal{U}}(d_i, d_j). \tag{11}$$

Recall that $\mathcal{L}$ is the log-likelihood of the observed data. The second term of (11) denotes that all documents in $\mathcal{T}$ should be grouped into the same cluster. The third term denotes that the documents in the topic $\mathcal{T}$ are not likely to be in the same cluster with the documents in unlabeled set $\mathcal{U}$. The reason behind this term is that we only want to find the documents that are reliably related to the given topic $\mathcal{T}$, and thus, we assume that all unlabeled data are irrelevant. Since the penalty terms applied in (11) are soft constraints, documents in unlabeled set $\mathcal{U}$ that are relevant to the topic are grouped into the same cluster with the documents in $\mathcal{T}$. The two parameters β_1 and β_2 determine the enforcement of the must-link and cannot-link constraints. In general, we would prefer a large value β_1 since the documents in topic $\mathcal{T}$ are labeled by the user manually, where such labels are assumed to be of high confidence. However, we will set a smaller value β_2 because there are some documents in the unlabeled data set $\mathcal{U}$ that are actually relevant to the given topic $\mathcal{T}$.

3.3 Learning

The parameters $\Theta = \{\mathrm{P}(w \mid z), \mathrm{P}(z \mid d), \mathrm{P}(d)\}$ can be determined by maximizing of the objective function $\mathcal{L}_1$ defined by (11). Similar to the case of the original pLSA, the objective function $\mathcal{L}_1$ is nonconvex, and thus, difficult for us to obtain a global optimal solution. Therefore, we need to seek a locally optimal solution. In the case of the original pLSA, the local optimal solution can be found by the EM algorithm as discussed in Section 3.1. However, in the case of topic-sensitive pLSA, the EM algorithm is no longer applicable since two extra penalty terms are added. Thus, we apply the Minorize-Maximization (MM) [24], [25] algorithm to optimize the objective function. Since its ability of optimizing complex objective functions, the MM algorithm has been applied to solve several machine learning tasks such as variable selection [26] and classification [27].

In order to maximize a difficult function, the MM algorithm solves a sequence of relatively easier optimization problems. The value of the objective function keeps growing during this iterative process. For example, we need to maximize a function $\mathcal{L}_1(\Theta)$ that is nonconvex, and thus, difficult to optimize. Instead of maximizing $\mathcal{L}_1(\Theta)$ directly, we maximize another function $\mathcal{M}(\Theta, \Theta^{(t)})$ satisfying the following two conditions:

$$\mathcal{L}_1(\Theta) \geq \mathcal{M}\big(\Theta, \Theta^{(t)}\big), \tag{12}$$

$$\mathcal{L}_1(\Theta^{(t)}) = \mathcal{M}\big(\Theta^{(t)}, \Theta^{(t)}\big), \tag{13}$$

where $\Theta^{(t)}$ denotes the value of parameters obtained in last iteration. The function $\mathcal{M}(\Theta, \Theta^{(t)})$ is called the minorizer of the function $\mathcal{L}_1(\Theta)$ at $\Theta^{(t)}$. Let $\Theta^{(t+1)}$ denote the maximal point of $\mathcal{M}(\Theta, \Theta^{(t)})$, then we have $\mathcal{M}(\Theta^{(t+1)}, \Theta^{(t)}) \geq \mathcal{M}(\Theta, \Theta^{(t)})$. Considering the value of the objective function $\mathcal{L}_1$ at $\Theta^{(t+1)}$, we have

$$\begin{aligned} \mathcal{L}_1(\Theta^{(t+1)}) &\geq \mathcal{M}\big(\Theta^{(t+1)}, \Theta^{(t)}\big) \\ &\geq \mathcal{M}\big(\Theta^{(t)}, \Theta^{(t)}\big) \\ &= \mathcal{L}_1\big(\Theta^{(t)}\big). \end{aligned} \tag{14}$$

Equation (14) implies that the value of the objective function keeps increasing, and thus, will finally converge to a local maximum during the MM iterations. In order to apply the MM algorithm to the objective function $\mathcal{L}_1$, we need to find the minorizer of the objective function, i.e., a function $\mathcal{M}(\Theta, \Theta^{(t)})$ that satisfies Conditions (12) and (13). According to (11), the objective function $\mathcal{L}_1$ can be written as

$$\begin{aligned} \mathcal{L}_1 = \sum_{d,w} n(d, w) \log \mathrm{P}(d, w) + \beta_1 \sum_{d_i, d_j \in \mathcal{T}} f_{\mathcal{T}}(d_i, d_j) \\ + \beta_2 \sum_{d_i \in \mathcal{T}} \sum_{d_j \in \mathcal{U}} f_{\mathcal{U}}(d_i, d_j). \end{aligned} \tag{15}$$

The objective function $\mathcal{L}_1$ is the linear combination of three terms $\log \mathrm{P}(d, w)$, $f_{\mathcal{T}}(d_i, d_j)$, and $f_{\mathcal{U}}(d_i, d_j)$. In order to minorize the object function, we only need to obtain minorizer for the terms $\log \mathrm{P}(d, w)$, $f_{\mathcal{T}}$, and $f_{\mathcal{U}}$, respectively.

Focusing on the term $\log \mathrm{P}(d, w)$, we have the following theorem:

Fact 1. *Let*

$$\mathcal{M}(d, w) = \sum_z C(d, w, z) \log \frac{\mathrm{P}(w \mid z)\mathrm{P}(z \mid d)\mathrm{P}(d)}{C(d, w, z)}, \tag{16}$$

where

$$C(d, w, z) = \frac{\mathrm{P}^{(t)}(w \mid z)\mathrm{P}^{(t)}(z \mid d)\mathrm{P}^{(t)}(d)}{\sum_{z'} \mathrm{P}^{(t)}(w \mid z')\mathrm{P}^{(t)}(z' \mid d)\mathrm{P}^{(t)}(d)}, \tag{17}$$

then $\mathcal{M}(d, w)$ is a minorizer of function $\log \mathrm{P}(d, w)$.

Proof. By noticing the fact that $\sum_z C(d, w, z) = 1$, it can be derived from the convexity of the log function that

$$\begin{aligned}\log \mathrm{P}(d,w) &= \log \sum_z \mathrm{P}(w \mid z)\mathrm{P}(z \mid d)\mathrm{P}(d) \\ &\geq \sum_z C(d,w,z) \log \frac{\mathrm{P}(w \mid z)\mathrm{P}(z \mid d)\mathrm{P}(d)}{C(d,w,z)} \\ &= \mathcal{M}(d,w.).\end{aligned} \tag{18}$$

And if we substitute $\mathrm{P}(w \mid z)$, $\mathrm{P}(z \mid d)$, and $\mathrm{P}(d)$ for $\mathrm{P}^{(t)}(w \mid z)$, $\mathrm{P}^{(t)}(z \mid d)$, and $\mathrm{P}(d)^{(t)}$, respectively, we have

$$\begin{aligned}\mathcal{M}(d,w) &= \sum_z C(d,w,z) \log \sum_{z'} \mathrm{P}^{(t)}(w \mid z')\mathrm{P}^{(t)}(z' \mid d)\mathrm{P}^{(t)}(d) \\ &= \sum_z C(d,w,z) \log \mathrm{P}^{(t)}(d,w) \\ &= \log \mathrm{P}^{(t)}(d,w).\end{aligned} \tag{19}$$

It can be derived from (18) and (19) that $\mathcal{M}(d,w)$ is a minorizer of P(d, w). □

Analogously, the minorizer of $f_{\mathcal{T}}(d_i,d_j)$ and $f_{\mathcal{U}}(d_i,d_j)$ can be constructed as follows:

$$\mathcal{M}_{\mathcal{T}}(d_i,d_j) = \sum_z C_{\mathcal{T}}(d_i,d_j,z) \log \frac{\mathrm{P}(z \mid d_i)\mathrm{P}(z \mid d_j)}{C_{\mathcal{T}}(d_i,d_j,z)}, \tag{20}$$

$$\mathcal{M}_{\mathcal{U}}(d_i,d_j) = \sum_{z1 \neq z2} C_{\mathcal{U}}(d_i,d_j,z_1,z_2) \log \frac{\mathrm{P}(z_1 \mid d_i)\mathrm{P}(z_2 \mid d_j)}{C_{\mathcal{U}}(d_i,d_j,z_1,z_2)}, \tag{21}$$

where $C_{\mathcal{T}}(d_i,d_j,z)$ and $C_{\mathcal{U}}(d_i,d_j,z_1,z_2)$ are defined by the parameters of the last iteration $\Theta^{(t)}$:

$$C_{\mathcal{T}}(d_i,d_j,z) = \frac{\mathrm{P}^{(t)}(z \mid d_i)\mathrm{P}^{(t)}(z \mid d_j)}{\sum_{z'} \mathrm{P}^{(t)}(z' \mid d_i)\mathrm{P}^{(t)}(z' \mid d_j)}, \tag{22}$$

$$C_{\mathcal{U}}(d_i,d_j,z_1,z_2) = \frac{\mathrm{P}^{(t)}(z_1 \mid d_i)\mathrm{P}^{(t)}(z_2 \mid d_j)}{1 - \sum_z \mathrm{P}^{(t)}(z \mid d_i)\mathrm{P}^{(t)}(z \mid d_j)}. \tag{23}$$

The expression for $C(d,w,z)$ is the same as $\mathrm{P}(z \mid d,w)$ in original pLSA. For all $d_i,d_j \in \mathcal{T}$ and $z \in \mathcal{Z}, C_{\mathcal{T}}(d_i,d_j,z)$ can be interpreted as the probability of a topic z on the condition that the two documents d_i and d_j belong to the same topic. Similarly, for all $d_i \in \mathcal{U}, d_j \in \mathcal{T}$, and $z_1,z_2 \in \mathcal{Z}, C_{\mathcal{U}}(d_i,d_j,z_1,z_2)$ is the probability that two documents d_i and d_j belong to the topics z_1 and z_2, respectively, on the condition that they are in different topics. The terms $C_{\mathcal{T}}$ and $C_{\mathcal{U}}$ correspond to the must-link and cannot-link constraints, respectively. These terms can be analogous to the posterior of the latent variable in the first step of the EM algorithm.

Taking all minorizers $\mathcal{M}(d,w)$, $\mathcal{M}_{\mathcal{T}}(d_i,d_j)$, and $\mathcal{M}_{\mathcal{U}}(d_i,d_j)$ into account, the minorizer of the objective function $\mathcal{L}_1(\Theta)$ can be expressed as follows:

$$\begin{aligned}\mathcal{M}(\Theta,\Theta^{(t)}) = &\sum_{d,w} n(d,w)\mathcal{M}(d,w) \\ &+ \beta_1 \sum_{d_i,d_j \in \mathcal{T}} \mathcal{M}_{\mathcal{T}}(d_i,d_j) \\ &+ \beta_2 \sum_{d_i \in \mathcal{T}} \sum_{d_j \in \mathcal{U}} \mathcal{M}_{\mathcal{U}}(d_i,d_j).\end{aligned} \tag{24}$$

Similar to the EM algorithm, the MM algorithm performs the two steps alternatively. During the first step, $C(d,w,z)$, $C_{\mathcal{T}}(d_i,d_j,z)$, and $C_{\mathcal{U}}(d_i,d_j,z_1,z_2)$ are obtained according to (17), (22), and (23) using parameters of the last iteration.

After $C(d,w,z)$, $C_{\mathcal{T}}(d_i,d_j,z)$, and $C_{\mathcal{U}}(d_i,d_j,z_1,z_2)$ are determined in the first step, the new values of parameters $\mathrm{P}(w \mid z), \mathrm{P}(z \mid d), \mathrm{P}(d)$ are obtained by maximizing the minorizer $\mathcal{M}(\Theta,\Theta^{(t)})$. Taking partial derivatives with respect to each parameter, the maximal point of $\mathcal{M}(\Theta,\Theta^{(t)})$ can be obtained.

For all $d \in \mathcal{T}$, the equation to update the value of $\mathrm{P}(z \mid d)$ can be expressed as follows:

$$\begin{aligned}\mathrm{P}(z \mid d) \propto &\sum_w n(d,w)C(d,w,z) + \beta_1 \sum_{d_i \in \mathcal{T}} C_{\mathcal{T}}(d_i,d,z) \\ &+ \beta_2 \sum_{d_i \in \mathcal{U}} \sum_{z_1 : z_1 \neq z} C_{\mathcal{U}}(d_i,d,z_1,z),\end{aligned} \tag{25}$$

and for all $d \in \mathcal{U}$,

$$\begin{aligned}\mathrm{P}(z \mid d) \propto &\sum_w n(d,w)C(d,w,z), \\ &+ \beta_2 \sum_{d_i \in \mathcal{T}} \sum_{z_1 : z_1 \neq z} C_{\mathcal{U}}(d,d_i,z,z_1).\end{aligned} \tag{26}$$

The equations to update the value of $\mathrm{P}(w \mid z)$ and $\mathrm{P}(d)$ are similar to those of the original pLSA:

$$\mathrm{P}(w \mid z) \propto \sum_d n(d,w)C(d,w,z), \tag{27}$$

$$\mathrm{P}(d) \propto \sum_w n(d,w). \tag{28}$$

The principle of the MM algorithm ensures that the value of the objective function $\mathcal{L}_1$, as defined in (11), increases during each iteration and converges to a local optimum. For a detailed derivation of the MM algorithm, the reader is referred to [24]. When the MM algorithm converges, the values of parameters $\mathrm{P}(z \mid d)$ are used to assign each document to a cluster. The cluster that contains the most labeled positive examples is considered to be the positive cluster.

A summary of our topic-sensitive pLSA algorithm is shown in Algorithm 1.

Algorithm 1. Topics Sensitive pLSA

Input: Document-term matrix n of size $\mathcal{D} \times \mathcal{W}$ and a set of document $\mathcal{T} \subset \mathcal{D}$ specifying the topic of interest.

Output: A set $\mathcal{R}$ of documents that are relevant to $\mathcal{T}$

1: Initialize $\mathrm{P}(z \mid d), \mathrm{P}(w \mid z), \mathrm{P}(d)$ randomly, and then normalize them.
2: **while** not converge **do**
3: Compute the values of $C(d,w,z)$, $C_{\mathcal{T}}(d_i,d_j,z)$, and $C_{\mathcal{U}}(d_i,d_j,z_1,z_2)$ according to (17), (22), and (23), respectively.
4: Update the values of $\mathrm{P}(z \mid d)$, $\mathrm{P}(w \mid z)$, and $\mathrm{P}(d)$ according to (25), (26), (27), and (28), respectively.
5: **end while**
6: $c(d) \leftarrow \arg\max_z \mathrm{P}(z \mid d)$
7: $z^\star \leftarrow \arg\max_z |\{d \in \mathcal{T} \mid c(d) = z\}|$
8: $\mathcal{R} \leftarrow \{d : c(d) = z^\star\}$

4 Performance Evaluations

In this section, we evaluate the performance of the proposed topic-sensitive pLSA algorithm. The performance of our algorithm is compared to that of five previously proposed algorithms: Spy-EM [7], Weighted Logistic Regression [10], PEBL [28], Roc-SVM [6], and MPC-Kmeans [21]. Our experiments are conducted on three document data sets: 20 Newsgroups, ODP, and Reuters-21578, followed by detailed discussions. Our goal is to evaluate whether the proposed topic-sensitive pLSA model is able to achieve better performance for the learning task with only positive labeled data. Another goal is to reveal the impact of parameters on the performance.

4.1 Data Corpora

We conduct the performance evaluation using three text corpora, 20 Newsgroups, Open Directory Project, and Reuters-21578, as follows:

- **20 Newsgroups:** The 20 Newsgroups is a popular data set for document classification and related tasks. It contains approximately 20,000 documents collected from 20 electronic newsgroups. The number of documents from each newsgroup is about 1,000. These documents cover different topics from computer graphics, hardware to politics, sports, etc.
- **ODP:** The ODP data set contains 10,000 Web pages crawled from the Open Directory Project.[1] All pages have been manually classified into hierarchical directories. The top level contains 17 categories. In our experimental studies, we use a subset of all categories that contains 21 most frequent second-level categories.
- **Reuters:** The Reuters-21578 is one of the most frequently used data sets in document classification and clustering. This document corpus contains 21,578 documents that are manually clustered into 135 clusters based on their topics. Each document may belong to multiple topics. All documents with multiple labels are excluded from our experiments. Clusters with less than 100 documents are removed, leaving 6,896 documents in the data set. The final data set is still unbalanced, with the largest cluster about 30 times larger than the smallest ones. This data set is more difficult than the 20 Newsgroups and ODP data because the data set is imbalanced where documents in each cluster contain a wider range of topics.

The statistics of all three document corpora are shown in Table 1. Each document is tokenized into a bag of words and represented by term-frequency vector. Then, we preprocess each document by applying downcasting, stop words removal, and stemming to the words. All the words with less than 5 occurrences are removed from the data sets.

4.2 Evaluation Measurements

The performance is compared using the F_1 score on positive class. The F_1 measure is a popular measure used in information retrieval and other areas. The F_1 measure trades off precision p and recall r. The precision p and recall r are defined as follows:

1. The Web site of the Open Directory Project is http://dmoz.org/.

TABLE 1
Statistics of Document Corpora

	20Newgroups	ODP	Reuters
No. documents	19997	10000	21578
No. docs. used	19997	8879	6869
No. clusters	20	27	135
No. clusters used	20	21	10
Max. cluster size	1000	472	3148
Min. cluster size	997	110	105
Avg. cluster size	999	393	687

$$p = \frac{\text{No. true positive documents}}{\text{No. the documents classified as positive}}, \tag{29}$$

$$r = \frac{\text{No. true positive documents}}{\text{No. positive documents in fact}}. \tag{30}$$

Then the F_1 measure is defined to be the harmonic mean of precision p and recall r:

$$F_1 = \frac{2pr}{p+r}. \tag{31}$$

To obtain a high F_1 score, both the precision and recall should be high.

4.3 Overall Performance

We have evaluated the proposed topic-sensitive pLSA model on three text corpora described above and compared the results with five different methods as follows:

- **Weighted logistic regression (WLR) [10]:** This method transforms the problem of learning with positive and unlabeled data to learning with noisily labeled data. Different weights are assigned to examples to deal with the noisy labels.
- **Spy-EM [7]:** This method is a revision of Naive Bayes algorithm. A subset of labeled positive documents is mixed with the unlabeled documents. The unknown positive data in the unlabeled set can be inferred by these *spy documents*. Then it runs EM iterations in order to build the final classifier. In previous study [6], it has been shown that this method is stable on a wide range of conditions.
- **PEBL [28], [9]:** This method extracts reliably negative examples from unlabeled data. Then SVM is applied iteratively to identify more and more negative documents.
- **Roc-SVM [6], [8]:** A Rocchio classifier is built from the labeled positive data and unlabeled data, assuming that all the unlabeled data are negative. Then it is applied to the unlabeled data in order to extract the reliably negative data. The final classifier is built using SVM iteratively. It has been reported in [8] that this method is robust with different numbers of labeled positive examples.
- **MPC-Kmeans [21]:** The MPC-Kmeans algorithm extends the unsupervised Kmeans algorithm to incorporate the user supervision. Must-link and cannot-link constraints are integrated with distance learning. This algorithm has not been applied to the task of learning from positive and unlabeled data

TABLE 2
Results on 20-Newsgroups Data Set

%	1%	2%	3%	4%	5%
Topic Sensitive PLSA	0.7229	0.7462	0.7678	0.7867	0.8306
WLR	0.6591	0.6716	0.7120	0.7222	0.7406
p-value with WLR	0.0037	0.0109	0.0323	0.0235	0.0027
Spy-EM	0.3203	0.5719	0.7279	0.7815	0.8266
p-value with Spy-EM	<0.0001	0.0036	0.0124	0.5257	0.3905
PEBL	0.0104	0.0201	0.0311	0.0573	0.0792
p-value with PEBL	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
Roc-SVM	0.3694	0.6149	0.7239	0.7747	0.8224
p-value with Roc-SVM	<0.0001	0.0002	0.0081	0.2953	0.3524
MPC-Kmeans	0.7098	0.7393	0.7449	0.7735	0.8008
p-value with MPC-Kmeans	0.0011	0.0091	0.0302	0.0199	0.0021

TABLE 3
Results on ODP Data Set

%	1%	2%	3%	4%	5%
Topic Sensitive PLSA	0.7458	0.7527	0.7959	0.7616	0.8000
WLR	0.6591	0.6834	0.6957	0.6920	0.7417
p-value with WLR	0.0061	0.0039	0.0027	0.0135	0.0205
Spy-EM	0.3929	0.5171	0.5975	0.6463	0.7059
p-value with Spy-EM	0.0001	0.0009	0.0005	0.0366	0.0438
PEBL	0.0137	0.0244	0.0403	0.0570	0.0709
p-value with PEBL	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
Roc-SVM	0.1803	0.5040	0.5990	0.6679	0.7060
p-value with Roc-SVM	<0.0001	0.0001	0.0002	0.0257	0.0276
MPC-Kmeans	0.7195	0.7210	0.7570	0.7712	0.7837
p-value with MPC-Kmeans	0.0041	0.0037	0.0015	0.0124	0.0201

before. We include this method in our experiments in order to present a comprehensive comparison to semisupervised clustering methods. We do not use COP-Kmeans [16] since it encodes the supervision into hard constraints, while soft constraints are required in our setting.

In order to compare the method proposed in this paper with the methods described above, we conducted experiments using different proportion of the labeled positive examples in all positive data, ranging from 1 to 5 percent. This range of the positive data is a typical range in semisupervised learning settings. For each of the three corpora, 20 test sets were generated, each test set was created by mixing six topics randomly selected from the document corpus. One topic is randomly selected from each test set and used as the positive class. For each test, some proportion λ of positive data are assumed to be labeled. To compare the performance, we vary λ from 1 to 5 percent and apply these algorithms to each of the test set using the same set of labeled data. The final performance score is obtained by averaging the scores from all 20 test runs.

For the Spy-EM algorithm and Roc-SVM algorithm, we use an implementation of LPU System[2] and simply use the default parameters. Since the implementation of PEBL and Weighted Logistic Regression is not publicly available, we implemented them based on the descriptions of [28] and [6], respectively. For the MPC-Kmeans algorithm, the penalty weights for the must-link constraints are set to 1, while the penalty weights for the cannot-link constraints are set to 0.2. For our method, we choose parameters as $\beta_1 = 2$ and $\beta_2 = 0.5$ and set the number of topics K to be the actual number of topics in the data set unless otherwise stated. The impact of these parameters on performance will be discussed in parameters tuning section. In our experiments, we first extract the constraints from the training set as described in Section 3.2. Thus, the same set of constraints is used to train topic-sensitive pLSA and MPC-Kmeans models.

The performance on 20 Newsgroups, ODP, and Reuters data sets with respect to the percent of the labeled positive data λ is reported in Tables 2, 3, and 4. We also calculated the p-values of the paired-sample Wilcoxon signed-rank test between each the of the five baseline algorithms and the topic-sensitive pLSA. The results with a p-value that is smaller than 0.05 are considered statistically significant. The corresponding p-values are also reported in Tables 2, 3, and 4.

We observe that the topic-sensitive pLSA model outperforms all of other five baseline algorithms in all cases. This indicates that the topic-sensitive pLSA is able to find related documents of specified topics from a set of unlabeled documents. When the proportion of the labeled

2. The software package can be downloaded from the following URL: http://www.cs.uic.edu/~liub/LPU/LPU-download.html.

TABLE 4
Results on Reuters Data Set

%	1%	2%	3%	4%	5%
Topic Sensitive PLSA	0.5845	0.5915	0.6260	0.6445	0.6525
WLR	0.5400	0.5512	0.5699	0.5725	0.5849
p-value with WLR	0.0188	0.0346	0.0086	0.0103	0.0182
Spy-EM	0.2568	0.2761	0.3575	0.4369	0.5736
p-value with Spy-EM	<0.0001	<0.0001	0.0001	0.0002	0.0013
PEBL	0.0164	0.0205	0.0322	0.0435	0.0561
p-value with PEBL	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
Roc-SVM	0.1803	0.4680	0.5473	0.5930	0.6138
p-value with Roc-SVM	<0.0001	0.0072	0.0056	0.0391	0.0330
MPC-Kmeans	0.5756	0.5615	0.6181	0.6258	0.6478
p-value with MPC-Kmeans	0.0179	0.0344	0.0081	0.0093	0.0180

positive data grows, the performance of all algorithms generally increases. Therefore, all these methods can make use of the labeled information and improve their performance. The performance of the topic-sensitive pLSA and MPC-Kmeans is superior to other methods in general. Considering the fact that the topic-sensitive pLSA and MPC-Kmeans are based on unsupervised learning methods, we conclude from this observation that algorithms based on unsupervised learning models are more superior when the labeled positive data are rare. Topic-sensitive pLSA outperforms MPC-Kmeans in general. We think this is because that the objective function of MPC-Kmeans is difficult to optimize. Thus, MPC-Kmeans cannot obtain precise solutions in some cases.

The p-values of the significance tests reported in Tables 2, 3, and 4 show that in most cases, the topic-sensitive pLSA outperforms other algorithms significantly. For the case that the proportion of labeled data in 20 Newsgroups data set is greater than 4 percent, the p-values of Spy-EM and Roc-SVM are greater than 0.05. Considering the fact that 20 Newsgroups is a balanced, and thus, relatively easy data set to work with, we conclude that for balanced and relatively easy-to-classify data set such as 20 Newsgroups, the performance of Spy-EM and Roc-SVM is comparable to topic-sensitive pLSA given the number of labeled data is not too small.

Another observation is that when the labeled data are extremely rare (i.e., $\lambda < 3$ percent), our method outperforms Spy-EM by a large margin. This is because in our algorithm, the similarity between documents is properly exploited. Another important reason is that Spy-EM assumes that all negative data are generated according to the same distribution, while our method treats negative data in the same way as they are generated from a different distribution. Our method can thus capture the diversity of the negative data better. This is especially important when the number of labeled positive data is small, since in this situation, we cannot know precisely the distribution of the positive data. Therefore, the performance of our method depends largely on fitness of the distribution of the negative data estimated. If we have more knowledge about the negative data, it will certainly help us know better about the positive data as well.

Roc-SVM generally works better than Spy-EM when the number of positive labeled examples is not too small ($\lambda > 2$ percent) except for a few cases. This is mainly because the classifier based on Rocchio's method is able to identify reliable negative data from the unlabeled data set more accurately. However, when the number of positive labeled examples is extremely small, negative data cannot be identified with high confidence. As a result, the performance of both algorithms decreases.

The performance of PEBL is not as good as other methods. The reason is that the number of labeled positive examples is small in our experience. Without enough positive examples, the iterative process of PEBL will reduce the performance. This property of PEBL is also reported in previous study [8]. We will explore this point a bit more. To this end, we plot the ROC curves of topic-sensitive pLSA together with PEBL and Roc-SVM. These ROC curves show the true positive rates with respect to the false positive rates. An ROC curve reveals the *ranking quality* of the model. As represented in Fig. 2, the ROC curves of PEBL and Roc-SVM are generally below that of the topic-sensitive pLSA, indicating that our topic-sensitive pLSA algorithm can generate better ranking. Also, we can observe that the performance difference measured by ROC curves is much smaller than that measured by the F values. We believe the reason is that ranking is usually considered to be easier than classification since we do not need to decide the cutoff point in ranking settings. Thus, even there are not sufficient number of labeled positive examples for learning, we can still obtain acceptable ranking results.

It is interesting to observe that the performance of WLR algorithm decreases slowly when the number of the labeled positive examples reduces (Row 3 of Tables 2, 3, and 4). This is because the weighted logistic regression algorithm transforms the problem of learning with positive and unlabeled data into the one of learning with noise, and thus, does not include a step to extract an initial set of negative data based on labeled positive data. Consequently, it is more stable when there are only a few labeled positive examples.

4.4 The Enforcement of Constraints

In order to enforce the must-link and cannot-link constraints generated from users' supervision, two types of penalty

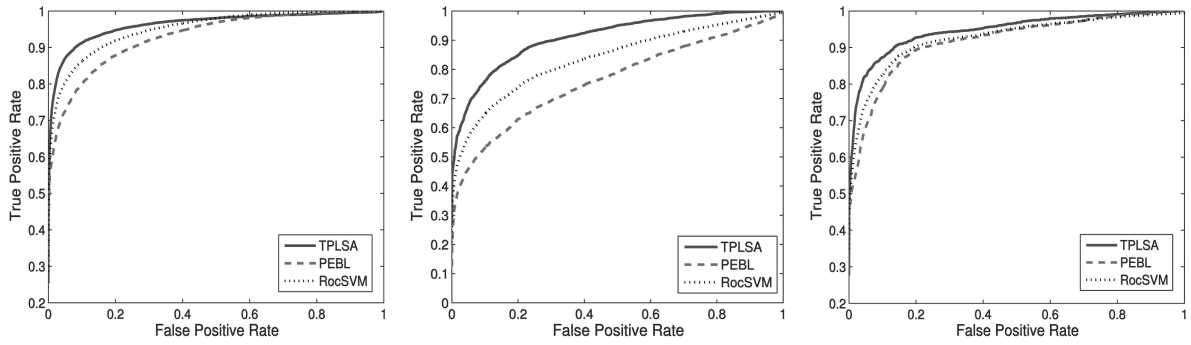


Fig. 2. ROC analysis over three data sets: (a) 20 Newsgroups, (b) ODP, and (c) Reuters.

TABLE 5
Percentage of Satisfied Must-Link Constraints

β_1	20 Newsgroups	ODP	Reuters
2	0.9833	0.9217	0.9296
4	0.9941	0.9551	0.9546
6	0.9980	0.9916	0.9738
8	0.9970	0.9890	0.9927
10	0.9990	0.9937	1.0000

TABLE 6
Percentage of Satisfied Cannot-Link Constraints

β_2	20 Newsgroups	ODP	Reuters
0.2	0.7983	0.8037	0.8222
0.4	0.8022	0.8147	0.8238
0.6	0.8095	0.8236	0.8248
0.8	0.8016	0.8316	0.8285
1.0	0.8142	0.8428	0.8375

terms, $f_{\mathcal{T}}$ and $f_{\mathcal{U}}$, are introduced in Section 3. The enforcement of the must-link and cannot-link constraints is represented by β_1 and β_2, respectively. With the increase of β_1 and β_2, the enforcement of the must-link and cannot-link constraints becomes stronger and stronger. In order to reveal the effectiveness of these penalty terms, we conduct experiments by varying the coefficients β_1 and β_2. The percentage of satisfied constraints is reported in Tables 5 and 6.

Table 5 shows the percentage values of satisfied must-link constraints with respect to β_1 on 20 Newsgroups (Column 1), ODP (Column 2), and Reuters (Column 3) data sets. We observe that the proportion of the satisfied must-link constraints grows when the coefficient β_1 increases, which indicates that the more we take the penalty terms $f_{\mathcal{T}}$ into account, the more must-link constraints are satisfied. Hence, the penalty terms $f_{\mathcal{T}}$ are effective to enforce the must-link constraints. The effectiveness of penalty terms $f_{\mathcal{U}}$ corresponding the cannot-link constraints is shown in Table 6. Similar to Table 5, the percents of satisfied cannot-link constraints grow with the increase of coefficient β_2.

4.5 Parameters Tuning

The method that we apply to find the optimum of objective function $\mathcal{L}_1$ is based on the MM algorithm, which is an iterative process that converges to a local optimum. Fig. 3 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 120 iterations are performed. This suggests that our algorithm will converge in about 120 iterations.

Our model has three tunable parameters: the number of topics K, and coefficients β_1 and β_2 for the penalty terms.

It is important to select the right number of latent variables K in our method. Small values of K will lead to poor performance while large K values will make the algorithm costly. We set the value of K to range from 4 to 15 and report the performance of our method in Fig. 4. An interesting observation is that the performance of our method peaks when the number of latent variables K is set to be a greater value than the actual number of topics in the data set. This is mainly because there are a small number of documents that are unlikely to belong to any of the topics. We found that these noisy documents are usually mixed with positive data so that the performance will reduce in this case. The effect of these noisy documents on performance can be reduced by setting the number of latent variables K in the model higher than the actual number of topics. In our experiments, we set K to be the actual number of topics, without loss of generality.

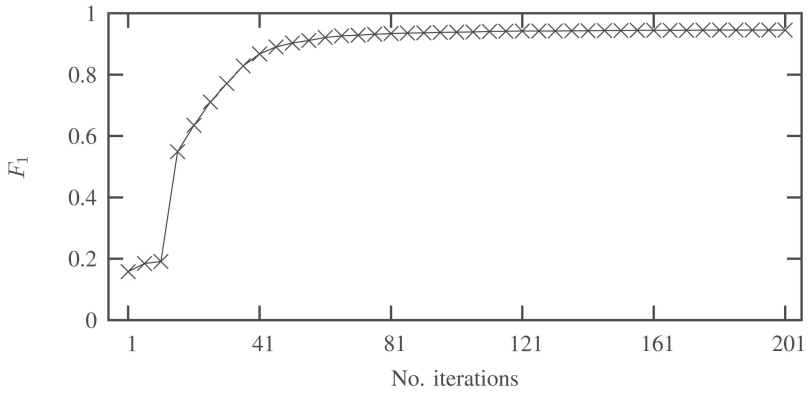


Fig. 3. Performance during iteration.

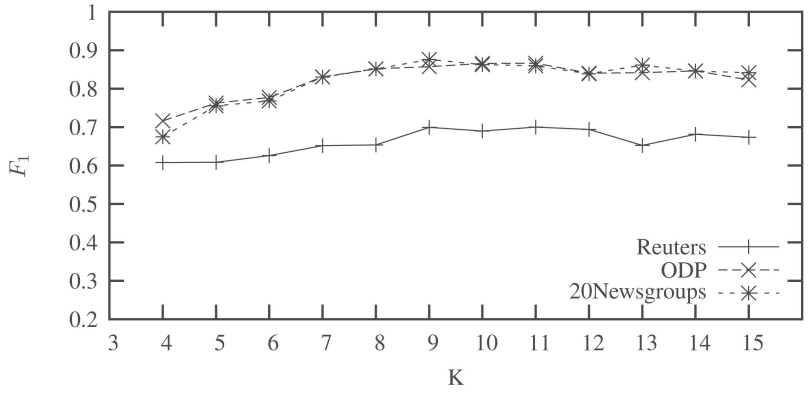


Fig. 4. Performance on different K.

TABLE 7
Results on 20 Newsgroups Data Set with Respect to β_1 and β_2

β_1	$\beta_2 = 0$	$\beta_2 = 0.2$	$\beta_2 = 0.4$	$\beta_2 = 0.6$	$\beta_2 = 0.8$	$\beta_2 = 1.0$
0	0.7247	0.7245	0.7250	0.7251	0.7249	0.7246
2	0.7542	0.7641	0.7640	0.7741	0.7939	0.7886
4	0.8054	0.8055	0.8058	0.8048	0.8041	0.8040
6	0.8075	0.8074	0.8074	0.8088	0.8074	0.8020
8	0.8043	0.8041	0.8048	0.8119	0.8114	0.8058

TABLE 8
Results on ODP Data Set with Respect to β_1 and β_2

β_1	$\beta_2 = 0$	$\beta_2 = 0.2$	$\beta_2 = 0.4$	$\beta_2 = 0.6$	$\beta_2 = 0.8$	$\beta_2 = 1.0$
0	0.7554	0.7552	0.7600	0.7651	0.7549	0.7546
2	0.7842	0.7941	0.8040	0.7941	0.7939	0.7886
4	0.7958	0.8054	0.8055	0.8118	0.8041	0.8040
6	0.7988	0.8075	0.8124	0.8074	0.8074	0.8020
8	0.8019	0.8114	0.8058	0.8048	0.8043	0.8041

The other two parameters of our model are: the coefficients β_1 and β_2 for the penalty terms. The coefficient β_1 represents the degree of enforcement that labeled positive data should be in the same cluster, while the coefficient β_2 regulates the degree of enforcement that unlabeled data should be negative. Intuitively, we set a larger β_1 value and a smaller β_2 value because we know that some of the unlabeled data are actually positive. 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 investigate the change in performance.

Tables 7, 8, and 9 show the impact of β_1 and β_2 on the performance of topics-sensitive pLSA on 20 Newsgroups, ODP, and Reuters corpus, respectively. The impacts of these two parameters on the performance over the three document corpora are similar. The common character of the three tables is that when β_2 increases from 0 to 1, the performance first increases and then decreases. This indicates that setting β_2 properly will give a better performance. It can also be seen from the rows of Tables 7, 8, and 9 that changing the values of β_2 does not have much impact on the performance.

TABLE 9
Results on Reuters Data Set with Respect to β_1 and β_2

β_1	$\beta_2 = 0$	$\beta_2 = 0.2$	$\beta_2 = 0.4$	$\beta_2 = 0.6$	$\beta_2 = 0.8$	$\beta_2 = 1.0$
0	0.5178	0.5180	0.5180	0.5180	0.5182	0.5182
2	0.6132	0.6203	0.6296	0.6317	0.6268	0.6156
4	0.6170	0.6237	0.6323	0.6303	0.6196	0.6181
6	0.6235	0.6321	0.6399	0.6279	0.6266	0.6256
8	0.6244	0.6335	0.6303	0.6398	0.6290	0.6277

When the value of β_1 decreases to close to 0, the impacts of the penalty terms for the must-links $f_{\mathcal{T}}$ are removed, which will lead to a decrease in performance (Row 1 of Tables 7, 8, and 9), indicating that the must-link constraints have a strong impact on the performance. And when $\beta_1 > 0$, the performance on different values of β_1 does not vary a lot. The performance is not very sensitive to the values of β_1 and β_2 in general. In our experiments, $\beta_1 = 2$ and $\beta_2 = 0.5$ seem a good choice in most cases.

The results of the experimental evaluation can be summarized as follows:

1. The proposed topic-sensitive pLSA algorithm outperforms many existing methods for learning with positive and unlabeled data, especially when the number of the labeled positive examples is very small.
2. Better performance can be obtained by setting the number of latent variables K slightly bigger than the actual number of topics in data corpus.
3. Although properly setting of parameters can achieve better performance, the performance of the algorithm is generally not sensitive to the change of the parameters.

5 Conclusions

In this paper, we propose a novel model called topic-sensitive pLSA for the task of learning with positive and unlabeled data. The proposed model is able to find the documents of a specified topic even when only several positive examples are given. We revise the basic pLSA to incorporate the supervision from user. In our method, this supervision is encoded by two types of constraints: the must-link constraints and cannot-link constraints. In order to enforce these two types of constraints, we introduce two penalty terms $f_{\mathcal{T}}$ and $f_{\mathcal{U}}$. An objective function is proposed integrating the likelihood of the data and the constraints generated from the user input. The local optimum of the defined model can be found through an iterative process. We evaluate the performance of this method on three document corpora: 20 Newsgroups, ODP, and Reuters-21578. Experimental results show that our method outperforms the existing methods especially with very small number of labeled examples.

The approach of formulating the problem of learning with positive and unlabeled data by semisupervised clustering with two types of constraints can be extended to other learning methods, such as k-means and spectral-based clustering algorithms. In our future work, we will study these clustering algorithms using different types of constraints. Furthermore, we will continue this research in collaborative filtering, image clustering, and other document analysis tasks.

Acknowledgments

The authors thank the anonymous reviewers for their valuable and constructive comments. Gui-Rong Xue thanks the National Natural Science Foundation of China (No. 60873211) for such generous support. Qiang Yang thanks the support of CERG Grant 621307.

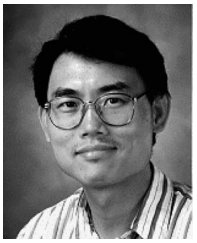
References

[1] T. Joachims, "Transductive Inference for Text Classification Using Support Vector Machines," *Proc. 16th Int'l Conf. Machine Learning (ICML '99)*, I. Bratko and S. Dzeroski, eds., pp. 200-209, 1999.
[2] T. Joachims, "Transductive Learning via Spectral Graph Partitioning," *Proc. 20th Int'l Conf. Machine Learning (ICML '03)*, 2003.
[3] K.P. Bennett and A. Demiriz, "Semi-Supervised Support Vector Machines," *Proc. 1998 Conf. Advances in Neural Information Processing Systems II*, pp. 368-374, 1999.
[4] R. Ghani, "Combining Labeled and Unlabeled Data for Multiclass Text Categorization," *Proc. 19th Int'l Conf. Machine Learning (ICML '02)*, pp. 187-194, 2002.
[5] K. Nigam, A.K. McCallum, S. Thrun, and T.M. Mitchell, "Text Classification from Labeled and Unlabeled Documents Using EM," *Machine Learning*, vol. 39, nos. 2/3, pp. 103-134, 2000.
[6] B. Liu, Y. Dai, X. Li, W.S. Lee, and P.S. Yu, "Building Text Classifiers Using Positive and Unlabeled Examples," *Proc. Third IEEE Int'l Conf. Data Mining (ICDM '03)*, pp. 179-188, 2003.
[7] B. Liu, W.S. Lee, P.S. Yu, and X. Li, "Partially Supervised Classification of Text Documents," *Proc. 19th Int'l Conf. Machine Learning (ICML '02)*, pp. 387-394, 2002.
[8] X. Li and B. Liu, "Learning to Classify Texts Using Positive and Unlabeled Data," *Proc. 18th Int'l Joint Conf. Artificial Intelligence (IJCAI '03)*, pp. 587-594, Aug. 2003.
[9] H. Yu, J. Han, and K.C.-C. Chang, "PEBL: Positive Example Based Learning for Web Page Classification Using SVM," *Proc. Eighth ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD '02)*, pp. 239-248, 2002.
[10] W.S. Lee and B. Liu, "Learning with Positive and Unlabeled Examples Using Weighted Logistic Regression," *Proc. 20th Int'l Conf. Machine Learning (ICML '03)*, pp. 448-455, 2003.
[11] T. Hofmann, "Probabilistic Latent Semantic Analysis," *Proc. Conf. Uncertainty in Artificial Intelligence (UAI '99)*, 1999.
[12] B. Ribeiro-Neto and R. Baeza-Yates, *Modern Information Retrieval*. Addison-Wesley, 1999.
[13] F. Denis, "PAC Learning from Positive Statistical Queries," *Proc. Ninth Int'l Conf. Algorithmic Learning Theory (ALT '98)*, vol. 1501, 1998.
[14] G.P.C. Fung and H. Lu, "Text Classification without Negative Examples Revisit," *IEEE Trans. Knowledge and Data Eng.*, vol. 18, no. 1, pp. 6-20, Jan. 2006.
[15] B. Scholkopf, J.C. Platt, J. Shawe-Taylor, A.J. Smola, and R.C. Williamson, "Estimating the Support of a High-Dimensional Distribution," *Neural Computation*, vol. 13, no. 7, pp. 1443-1471, 2001.

[16] K. Wagstaff, C. Cardie, S. Rogers, and S. Schroedl, "Constrained k-Means Clustering with Background Knowledge," *Proc. 18th Int'l Conf. Machine Learning (ICML '01)*, pp. 577-584, 2001.
[17] J. Shi and J. Malik, "Normalized Cuts and Image Segmentation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888-905, Aug. 2000.
[18] I. Davidson and S.S. Ravi, "Clustering with Constraints: Feasibility Issues and the k-Means Algorithm," *Proc. SIAM Int'l Conf. Data Mining*, 2005.
[19] D. Pelleg and D. Baras, "K-Means with Large and Noisy Constraint Sets," *Proc. European Conf. Machine Learning (ECML '07)*, pp. 674-682, 2007.
[20] X. Ji and W. Xu, "Document Clustering with Prior Knowledge," *Proc. 29th Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR '06)*, 2006.
[21] S. Basu, M. Bilenko, and R.J. Mooney, "A Probabilistic Framework for Semi-Supervised Clustering," *Proc. 10th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD '04)*, pp. 59-68, 2004.
[22] T. Hofmann, "Probabilistic Latent Semantic Indexing," *Proc. 22nd Ann. ACM Conf. Research and Development in Information Retrieval*, 1999.
[23] A. Dempster, N. Laird, and D. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *J. Royal Statistical Soc., Series B*, vol. 39, no. 1, 1977.
[24] D.R. Hunter and K. Lange, "A Tutorial on MM Algorithms," *The Am. Statistician*, vol. 58, no. 1, pp. 30-37, 2004.
[25] J.D. Leeuw and W.J. Heiser, "Convergence of Correction-Matrix Algorithms for Multidimensional Scaling," *Geometric Representations of Relational Data*, Mathesis Press, 1977.
[26] D.R. Hunter and R. Li, "Variable Selection Using MM Algorithms," *Annals of Statistics*, vol. 33, pp. 1617-1642, 2005.
[27] Z. Zhang, J.T. Kwok, and D.-Y. Yeung, "Surrogate Maximization/Minimization Algorithms for AdaBoost and the Logistic Regression Model," *Proc. 21st Int'l Conf. Machine Learning (ICML '04)*, p. 117, 2004.
[28] H. Yu, J. Han, and K.C.-C. Chang, "PEBL: Web Page Classification without Negative Examples," *IEEE Trans. Knowledge and Data Eng.*, vol. 16, no. 1, Jan. 2004.
[29] V.N. Vapnik, *The Nature of Statistical Learning Theory*. Springer-Verlag, 1995.
[30] G.P.C. Fung, J.X. Yu, H. Lu, and P.S. Yu, "Text Classification without Labeled Negative Documents," *Proc. 21st Int'l Conf. Data Eng. (ICDE '05)*, pp. 594-605, Apr. 2005.
[31] T. Joachims, *Learning to Classify Text Using Support Vector Machines: Methods, Theory and Algorithms*. Kluwer Academic Publishers, 2002.
[32] F. Letouzey, F. Denis, and R. Gilleron, "Learning from Positive and Unlabeled Examples," *Proc. 11th Int'l Conf. Algorithmic Learning Theory (ALT '00)*, pp. 71-85, 2000.
[33] F.D. Comité, F. Denis, R. Gilleron, and F. Letouzey, "Positive and Unlabeled Examples Help Learning," *Proc. 10th Int'l Conf. Algorithmic Learning Theory (ALT '99)*, pp. 219-230, 1999.

Ke Zhou received the BS degree in computer science and engineering from Shanghai Jiao-Tong University in 2007. His research interests include information retrieval, machine learning, and Web image mining.

Gui-Rong Xue received the PhD degree from Shanghai Jiaotong University in 2006. He is a faculty member in the Department of Computer Science and Engineering at Shanghai Jiao-Tong University. His research interests are machine learning, data mining, and information retrieval.



Qiang Yang received the PhD degree from the University of Maryland at College Park. He is a professor in the Department of Computer Science and Engineering at the Hong Kong University of Science and Technology. His research interests are AI planning, machine learning, and data mining. He is an associate editor of the *IEEE Transactions on Knowledge and Data Engineering* and the *IEEE Intelligent Systems*. He is a fellow of the IEEE.

Yong Yu received the master's degree from the Computer Science Department at East China Normal University. He is currently a professor in the Department of Computer Science and Engineering at Shanghai Jiao-Tong University. His research interests include semantic Web, Web mining, and information retrieval.