Inf Retrieval
DOI 10.1007/s10791-010-9141-9

Investigating task performance of probabilistic topic models: an empirical study of PLSA and LDA

Yue Lu · Qiaozhu Mei · ChengXiang Zhai

Received: 10 November 2009 / Accepted: 8 July 2010

Abstract Probabilistic topic models have recently attracted much attention because of their successful applications in many text mining tasks such as retrieval, summarization, categorization, and clustering. Although many existing studies have reported promising performance of these topic models, none of the work has systematically investigated the task performance of topic models; as a result, some critical questions that may affect the performance of all applications of topic models are mostly unanswered, particularly how to choose between competing models, how multiple local maxima affect task performance, and how to set parameters in topic models. In this paper, we address these questions by conducting a systematic investigation of two representative probabilistic topic models, probabilistic latent semantic analysis (PLSA) and Latent Dirichlet Allocation (LDA), using three representative text mining tasks, including document clustering, text categorization, and ad-hoc retrieval. The analysis of our experimental results provides deeper understanding of topic models and many useful insights about how to optimize the performance of topic models for these typical tasks. The task-based evaluation framework is generalizable to other topic models in the family of either PLSA or LDA.

Keywords Evaluation · Topic models · LDA · PLSA · Experimentation · Performance

Y. Lu (✉) · C. Zhai
Department of Computer Science, University of Illinois at Urbana-Champaign,
201 N Goodwin Ave, Urbana, IL 61801, USA
e-mail: yuelu2@illinois.edu

C. Zhai
e-mail: czhai@cs.uiuc.edu

Q. Mei
School of Information, University of Michigan, 1085 South University Ave,
Ann Arbor, MI 48109, USA
e-mail: qmei@umich.edu

Published online: 05 August 2010

1 Introduction

Recently, probabilistic topic models have received considerable attention in machine learning and text mining (Hofmann 1999a; Blei et al. 2003, 2004, Blei and Lafferty 2005; Wang and McCallum 2006, Nallapati et al. 2008). While many variants of topic models have been proposed, the common basic idea behind virtually all the models is to model text as a sample of words that are drawn from some mixture model involving multinomial component models. A topic is captured through a multinomial distribution of words (i.e., a unigram language model). A text document is generally assumed to contain potentially multiple topics. By inferring the parameter values of a topic model based on a set of observed text documents using either a Maximum Likelihood estimator (e.g,. Hofmann 1999a), or Bayesian inferences in the case of having a prior on parameters (e.g., Blei et al. 2003) , we would obtain a set of multinomial word distributions that can represent the major topics (or latent semantic themes) and a distribution over all the topics that can characterize the topic coverage in each document. Discovered topics and their distributions can then be used for many interesting text mining tasks as has been demonstrated in the previous work (see, e.g., Griffiths and Steyvers 2004; Mei et al. 2007; Wei and Bruce Croft 2006; Cai et al. 2008).

So far, the utilization of topic models can be categorized into three general ways. (1) Discovering topics from text (e.g. Hofmann 1999a; Blei et al. 2003; Griffiths and Steyvers 2004; Mei et al. 2007)): the discovered topic word distributions and topic coverage distributions can be directly useful (e.g., for revealing the major topics covered in a collection of text). (2) Obtaining a low-dimensional latent semantic representation of text (e.g. Hofmann 1999b; Blei et al. 2003; Cai et al. 2008): the topic coverage distribution for each document can serve as an alternative latent semantic representation of text, which is potentially better than the original word-based representation in supporting semantic matching of text. (3) Expanding text representation to accommodate semantic term matching (e.g. Wei and Bruce Croft 2006; Yi and Allan 2009). The latent semantic representation can also be expanded to recover a "smoothed" word-level representation of text, which includes not just the original words in the text, but also some semantically related words learned through the topic model; thus, such an expanded word-level representation can better achieve semantic term matching.

Despite the recent success of topic models in literature, there is still a considerable gap between the theoretical understanding of topic models and practical application for text mining tasks. That is, although many existing studies have reported promising performance of topic models, none of the work has systematically investigated the task performance of a topic model. As a result, many important practical questions related to using topic models for text mining tasks remain unanswered:

First, there has been no systematic comparison between alternative topic models in terms of their task performance. For example, most topic models are in the families of two basic models, i.e., probabilistic latent semantic analysis (PLSA) (Hofmann 1999a; Hofmann 1999b) and Latent Dirichlet Allocation (LDA) (Blei et al. 2003) and in general, a task can often be performed using either of the two families of models. However, the two models have not been systematically compared in existing work using typical tasks such as clustering, categorization, and retrieval. Indeed, the most popular comparison of the two models is based on perplexity of a held-out test set (Blei et al. 2003; Wallach et al. 2009), which measures the generalization performance of a model in document modeling. Theoretical analysis also reveals the relation between two models (Girolami and Kabán 2003), but it does not directly indicate the performance for any application task. As a result, it is unclear whether there is any difference between them in terms of task performance, although from modeling perspective, there are some known advantages of LDA over PLSA (Blei et al. 2003).

Second, all the topic models rely on some numerical algorithms to estimate parameters or make inferences. Since we generally have multiple local maxima, the task performance would presumably be affected by whether we can find a good local maximum. However, little work has been done on investigating the influence of this factor on task performance of topic models. LDA would theoretically suffer less from the problem of multiple local maxima; would this advantage lead to superior empirical performance for application tasks?

Third, there are often parameters that need to be set. For example, the estimation algorithms for all the topic models are iterative numerical algorithms, and insufficient iterations likely will lead to non-optimal parameter estimation, but how much influence do sub-optimal parameter values (due to insufficient iterations) actually have on task performance? Is there any difference between PLSA and LDA in this aspect? Also, we have to pre-specify the number of topics for many topic models. How does this parameter affect task performance? In many existing studies, parameters like this and some other parameters (e.g., hyper-parameter of LDA) are often set arbitrarily without much justification (Wei and Bruce Croft 2006; Cai et al. 2008).

Clearly, answers to these questions are necessary to guide us for optimizing the task performance of topic models in all kinds of applications. In this paper, we address these questions by conducting a systematic investigation of the two representative basic topic models, PLSA and LDA, using three representative text mining tasks, including document clustering, text categorization and ad-hoc retrieval. These tasks have been chosen because they are representative of the current text mining applications of topic models and also enable us to evaluate the three typical ways of using topic models mentioned earlier with standard test sets and quantitative measures. We systematically vary all the parameters of these two models, looking into the performance sensitivity of each task to these parameter values.

Our experimental results show that (1) when optimized, LDA has advantage over PLSA for classification task requiring a fine granularity latent semantic representation that is generalizable from training data to test data; in other tasks they tend to perform similarly. However, the performance of LDA on all tasks is quite sensitive to the setting of its hyper-parameter, and the optimal setting of hyper-parameters varies according to how the model is used in a task; (2) while the problem of local maxima affects the learned model (especially PLSA), whether it has a significant influence on the performance of tasks depends on the nature of the task; and (3) the latent semantic representation obtained through a topic model promotes semantic matching, but it may not perform well by itself in text mining tasks, especially in the case of tasks requiring fine granularity discrimination; in general, combining the low-dimensional latent semantic representation with high-dimensional original text representation tends to be most robust and effective. Our findings enable us to better understand the task performance of basic topic models and provide useful guidance for optimizing the performance of topic models for these typical tasks.

The rest of the paper is organized as follows. We first review the related work in Sect. 2. We then give a brief introduction to probabilistic topic models in Sect. 3, with a focus on PLSA and LDA. After that we discuss our design of experiments and evaluation results in Sects. 4, 5, and 6, for the three tasks respectively. Finally, we summarize the findings and conclude in Sect. 7.

2 Previous work

Hofmann (1999a) introduced the probabilistic latent semantic analysis (PLSA) in 1999, which models each word in a document as a sample from a mixture model with multinomial component models. While the PLSA model produced impressive results on a number of text document problems such as information retrieval (Hofmann 1999b), the parameterization of the model was susceptible to overfitting and did not provide a straightforward way to make inferences about new documents not seen in the training data. Blei et al., addressed these limitations by proposing a more general Bayesian probabilistic topic model called Latent Dirichlet Allocation (LDA) (Blei et al. 2003).

Later, many extensions of the two basic models have been proposed. For example, Griffiths and Steyvers (2004) showed how Gibbs sampling, a Markov chain Monte Carlo technique, could be applied to estimation and inferences in LDA, and experimented with this approach using 11 years of abstract data from the Proceedings of the National Academy of Sciences. In Zhai et al. (2004), PLSA was extended to include a background component to explain the non-informative background words and a cross-collection mixture model was proposed to support comparative text mining. The work by Steyvers et al. (2004) extends probabilistic topic models to include authorship information. Li and Mccallum (2006) extends LDA by capturing correlations between topics using a directed acyclic graph (DAG). Mei and Zhai (2006) propose a general contextual text mining model which is an extension of PLSA to incorporate context information. Recently, they further regularize PLSA with a harmonic regularizer based on a graph structure in the data (Mei et al. 2008). There are many other topic models proposed, such as Blei and Lafferty (2005), Wang and McCallum (2006), Nallapati et al. (2008), which demonstrated interesting results in various scenarios.

PLSA, LDA and their extensions have been successfully applied in many text mining tasks. The most popular application is to mine a large collection of document and summarize it using topics (Griffiths and Steyvers 2004; Mei et al. 2007). Promising results on text categorization were reported in the original LDA paper (Blei et al. 2003). Lacoste-Julien et al. (2008) proposes a supervised variation of LDA specialized for the classification task. Recently, Wei and Bruce Croft (2006) shows that LDA could improve the state-of-the-art information retrieval in the language modeling framework. Yi and Allan (2009) conduct in-depth retrieval studies and one of the conclusions states that PAM, an extension of LDA that captures topic dependencies, provides no additional gains over LDA. Cai et al. (2008) demonstrates promising clustering results of PLSA, LDA and their own extension called LapPLSI.

There has been some theoretical analysis studying the connection between PLSA and LDA (Girolami and Kabán 2003), but little comprehensive empirical comparison has been made, especially between most commonly used implementations of the two algorithms. Most comparison reported in the literature was based on probability of held-out data (Blei et al. 2003, Blei and Lafferty 2005, Steyvers and Griffiths 2007; Wallach et al. 2009). As a result, there is still little insight about how to optimize the performance of topic models for various text mining tasks and many questions related to the task performance of topic models have not been answered. We address these problems in our work. A recent paper (Chang et al. 2009) conducts user studies to quantitatively compare the semantic meaning in topics inferred by PLSA and LDA. While they focus to quantify the interpretability of topics with human effort, we study the task performance of topic models in three standard text mining applications, which can be quantified objectively using standard measures. Thus, our work is supplementary to theirs.

There is another line of closely related algorithms called the Non-negative Matrix Factorization (NMF) family. According to Gaussier and Goutte (2005), PLSA solves the problem of NMF with KL divergence. However, the NMF family are based on linear algebra, thus do not have probabilistic interpretations. We do not intent to cover them in this paper, but a comparative study of probabilistic topic models and NMF would be a very interesting future work.

3 Probabilistic topic models

In this section, we give a brief introduction to probabilistic topic models. Since most topic models are extended based on two representative ones, i.e. PLSA (Hofmann 1999a) and LDA (Blei et al. 2003), we focus our discussion on these two basic models.

All current probabilistic topic models are based on the same fundamental idea that documents are mixtures of topics, where a topic is represented by a multinomial distribution of words, i.e. a unigram language model. A topic model is a generative model for documents: it specifies a probabilistic process by which documents can be generated. To model the process of writing a new document, one usually first chooses a distribution over topics. Then, to generate each word in that document, one chooses a topic at random according to this distribution, and draws a word from that topic. Different topic models differ in how they vary this basic generation process and what statistical assumptions are made.

3.1 Notations

We assume that all the documents in the collection D fit into a finite set of K topics, each topic z is associated with a multinomial word distribution $P(w|z)$ over the vocabulary V. A document d is represented as a bag of words. We use $P(z|d)$ for the distribution over topics z in a particular document and $P(w|z)$ for the probability distribution over words w given topic z. To simplify the notations, let $\phi_w^{(j)} = P(w|z = j)$ refer to the multinomial distribution over words for topic j and $\theta_j^{(d)} = P(z = j|d)$ refer to the multinomial distribution over topics for document d. The parameters ϕ and θ indicate which words are important for which topic and which topics are important for a particular document, respectively.

3.2 Probabilistic latent semantic analysis

Model Probabilistic latent semantic analysis (PLSA) (Hofmann 1999a, 1999b) was introduced by Hoffman. A document d is regarded as a sample of the following mixture model.

$$P(w|d) = \sum_{j=1}^{K} \phi_w^{(j)} \theta_j^{(d)} \tag{1}$$

Estimation The word-topic distributions $\hat{\phi}$ and topic-document distributions $\hat{\theta}$ can be estimated using the Expectation-Maximization (EM) algorithm (Dempster et al. 1977) by maximizing the (log) likelihood that the collection D is generated by this model:

$$\log P(D|\phi,\theta) = \sum_{d\in D}\sum_{w\in V}\left\{c(w,d)\log\sum_{j=1}^{K}\phi_w^{(j)}\theta_j^{(d)}\right\} \tag{2}$$

where $c(w, d)$ is the number of times word w occurs in document d. In the E-step, we estimate the hidden variable z based on the model parameter at the previous iteration:

$$P(z_{d,w}=j) = \frac{\phi_w^{(j)}\theta_j^{(d)}}{\sum_{j'=1}^{K}\phi_w^{(j')}\theta_{j'}^{(d)}} \tag{3}$$

Then, in the M-step, the updating formulas for the model parameters are as follows:

$$\begin{aligned}\theta_j^{(d)} &= \frac{\sum_{w\in V}c(w,d)P(z_{d,w}=j)}{\sum_{j'}\sum_{w\in V}c(w,d)P(z_{d,w}=j')}\\ \phi_w^{(j)} &= \frac{\sum_{d\in D}c(w,d)P(z_{d,w}=j)}{\sum_{w'\in V}\sum_{d\in C}c(w',d)P(z_{d,w'}=j)}\end{aligned}$$

3.3 Latent Dirichlet Allocation

Model The PLSA model does not make any assumptions about how the mixture weights θ are generated, so its generative semantics are not well-defined (Blei et al. 2003); thus there is no natural way to predict a previously unseen document, and the number of parameters grows linearly with the number of training documents, which makes the model susceptible to overfitting. To solve these problems (Blei et al. 2003), introduced a new topic model, Latent Dirichlet Allocation (LDA). The basic generative process of LDA closely resembles PLSA. In PLSA, the topic mixture is conditioned on each document. In comparison, the topic mixture θ in LDA is drawn from a conjugate Dirichlet prior. As a conjugate prior for the multinomial, the Dirichlet distribution is a convenient choice as prior, which can simplify the statistical inference. The parameters of this Dirichlet distribution are specified by α_1,..., α_K. Each hyper-parameter α_j can be interpreted as a prior observation count for the number of times topic j is sampled in a document, before having observed any actual words from that document. It is mathematically convenient to use a symmetric Dirichlet distribution with a single hyper-parameter α such that $\alpha_1 = \alpha_2 = \cdots = \alpha_K = \alpha$. The probability density is given by:

$$P(\theta^{(d)}|\alpha) = Dir(\theta^{(d)}|\alpha) = \frac{\Gamma(K\alpha)}{\Gamma(\alpha)^K}\prod_{j=1}^{K}(\theta_j^{(d)})^{\alpha-1} \tag{4}$$

where $\Gamma(x) = \int_0^\infty t^{x-1}e^{-t}dt$ is an extension of the factorial function, with its argument shifted down by 1, to real and complex numbers. The Γ functions in the equation serve as normalization factors that make sure the probabilities sum up to 1. By placing a Dirichlet prior on the topic distribution θ, the result is a smoothed topic distribution, with the amount of smoothing determined by the α parameter. Similarly, a symmetric Dirichlet prior with hyper parameter β can be placed on ϕ as well. Given the parameters α and β, the probability of the collection of documents D can be calculated by integrating over θ and ϕ:

$$P(D|\alpha,\beta) = \int_{\phi^{(1)}} \cdots \int_{\phi^{(K)}} \prod_{i=1}^{K} P(\phi^{(i)}|\beta) \prod_{d\in D} \int P(\theta^{(d)}|\alpha) \prod_{w\in V} \left(\sum_{j=1}^{K} \theta_j^{(d)} \phi_w^{(j)} \right)^{c(w,d)} d\theta^{(d)} d\phi^{(1)} \cdots d\phi^{(K)}$$

Estimation The LDA model is complex and cannot be solved by exact inference. There are a few approximate inference techniques available in the literature: variational methods (Blei et al. 2003), expectation propagation (Minka and Lafferty 2002; Griffiths and Steyvers 2004), and Gibbs sampling (Geman and Geman 1984; Griffiths and Steyvers 2004). A recent work (Teh and Görür 2009) studies different estimation algorithms for LDA by evaluating on perplexity, and conclude that when hyperparameters are optimized, the performance differences among algorithms diminish significantly. In our study, we choose to use Gibbs sampling here, because of its advantage in finding the global optimum and its prevalent use in existing work (Geman and Geman 1984; Griffiths and Steyvers 2004; Wei and Bruce Croft 2006) . Nevertheless, it is interesting future work to also compare different estimation algorithms for LDA.

Given a text collection $\mathbf{w} = \{w_1, w_2, \ldots, w_N\}$, Gibbs sampling considers each word token w_i in document d_i in turn, and then samples a topic z_i for w_i: the probability of assigning the current word token to each topic is computed conditioned on the topic assignments to all other word tokens $\mathbf{z}_{-i}$.

$$P(z_i = j|\mathbf{z}_{-i}, \mathbf{w}) \propto \frac{n_{-i,j}^{(w_i)} + \beta}{n_{-i,j}^{(\cdot)} + N\beta} \times \frac{n_{-i,j}^{(d_i)} + \alpha}{n_{-i}^{(d_i)} + K\alpha} \tag{5}$$

where $n_{-i,j}^{(w_i)}$ is the number of word instances w_i being assigned to topic j, not including the current word; $n_{-i,j}^{(\cdot)}$ is the total number of words being assigned to topic j, not including the current word; $n_{-i,j}^{(d_i)}$ is the number of words from document d_i assigned to topic j, not including the current one; and $n_{-i}^{(d_i)}$ is the total number of words in document d_i not including the current one. From this conditional distribution $P(z_i = j|\mathbf{z}_{-i}, \mathbf{w})$, a topic is sampled and stored as the new topic assignment for this word token. After the estimation is done, the word-topic distributions $\hat{\phi}$ and topic-document distributions $\hat{\theta}$ can be estimated from one complete Gibbs sample of the whole collection as:

$$\hat{\phi}_j^{(w)} = P(w|z = j) = \frac{n_j^{(w)} + \beta}{\sum_w n_j^{(w)} + N\beta} \tag{6}$$

$$\hat{\theta}_j^{(d)} = P(z = j|d) = \frac{n_j^{(d)} + \alpha}{\sum_j n_j^{(d)} + K\alpha} \tag{7}$$

where $n_j^{(w)}$ is the number of instances of word w assigned to topic $z = j$ and $n_j^{(d)}$ is the number of words in document d assigned to topic $z = j$. So $\sum_w n_j^{(w)}$ is the total number of words assigned to topic $z = j$; and $\sum_j n_j^{(d)}$ is the total number of words in document d.

The Dirichlet prior on the topic distributions can be interpreted as forcing on the topic combinations with higher α moving the topics away from the corners of the simplex, leading to more smoothing effect (Steyvers and Griffiths 2007). For $\alpha < 1$, the modes of the Dirichlet distribution are located at the corners of the simplex. In this case, there is a bias towards sparsity, and the pressure is to pick topic distributions favoring just a few topics. A commonly used setting recommended by (Steyvers and Griffiths 2007) is $\alpha = 50/K$ and $\beta = 0.01$.

Many other topic models have been proposed in recent literature (Zhai et al. 2004; Steyvers et al. 2004; Blei and Lafferty 2005; Li and Mccallum 2006; Mei and Zhai 2006, Mei et al. 2008; Wang and McCallum 2006; Nallapati et al. 2008). These models are generally extended from the basic form of either PLSA or LDA. In this paper, we select PLSA and LDA as the representative for the two family of models.

3.4 Un-answered questions

Many extensions of the two basic topic models (i.e., PLSA and LDA) have been proposed, no existing work, however, has systematically investigated the issue of how to optimize the task performance of a topic model. Moreover, the comparison of the PLSA and LDA in existing work only include (1) Blei et al. (2003) shows that LDA outperforms PLSA in perplexity of predicting new document and (2) Girolami and Kabán (2003) theoretically proves that PLSA is equivalent to MAP estimated LDA under a uniform prior. Nevertheless, perplexity does not directly imply task performance in applications and different estimation methods can also lead to different task performance. Since there is no systematic emperical study of the two topic models, the following important questions about how to optimally apply topic models to text mining tasks remain mostly unanswered:

- Many tasks can be performed with either PLSA or LDA, but these two basic representative models have not been systematically compared directly in terms of their performances in real mining tasks. Is one significantly better than the other, or do they perform similarly? The only comparison between the two models seems to be based on perplexity (Blei et al. 2003), and PLSA is shown to be more susceptible to overfitting than LDA. However, to what extent does the overfitting affect the performance in mining tasks? In text mining applications, how can we choose one of these competing models?
- Existing work has shown that topic models can potentially improve the performance of text mining tasks such as classification and retrieval through providing a low-dimensional latent semantic representation. But how robust is the improvement? Can they always improve the performance over the original word-based high-dimensional representation? If not, under what conditions they can and under what conditions they cannot?
- It is well known that iterative parameter estimation algorithms such as the EM algorithm can only find a local maximum; if there are multiple local maxima, it is not guaranteed to reach the global maximum. Similar stability issue also exists in Gibbs sampling. However, how does this affect task performance?
- We know that the Dirichlet parameter α in LDA has the effect of regularizing topic distributions, but how should we set this parameter in practice? Is the recommended

setting in literature (e.g., as in Steyvers and Griffiths 2007) reasonable in different tasks?

In the following three sections, we address these questions by investigating the performance of the two representative basic topic models (PLSA and LDA) in three representative text mining tasks. These three tasks correspond to three typical ways of using topic models in applications described in Sect. 1. Although in this paper we study these questions with the two basic representative models, these questions can be asked for almost any topic model. Thus our findings can also offer insights about how to optimize the task performance of other extensions of these basic models.

We acknowledge that there are numerous techniques proposed for each of the three tasks. To avoid distraction of our main focus, we intentionally do not compare with them. Rather, we start with reasonably good baseline; the purpose is to provide a fair basis to compare the two topic modeling algorithms, and to understand their commonalities and differences, instead of proposing the best method for the specific tasks.

4 Document clustering

The first task we have chosen is document clustering. Generally, there are two ways of using topic models for document clustering. The first approach uses a topic model to map the original high-dimensional representation of documents (word features) to a low-dimensional representation (topic features) and then applies a standard clustering algorithm like *k*-means in the new feature space. The other approach uses topic models more directly. The basic assumption is that each topic corresponds to a cluster, so the number of topics in topic models matches the number of clusters. After estimating the parameters, the documents are clustered into the topic with the highest probability:

$$\arg\max_{j=1\ldots K}\theta_j^{(d)}.$$

Here we focus on the second approach of clustering, because it allows us to examine how well the topic distribution θ could recover the most significant topic, i.e. cluster. We leave the study of the effectiveness of dimension reduction using topic models to the next task.

4.1 Experimental setup

Data sets We use two standard data sets: the TDT2[1] and the Reuters-21578[2] (or simply, Reuters) document corpora. Every document in the corpora has been manually assigned one or more labels indicating which topic(s) it belongs to. The TDT2 corpus consists of 11,201 on-topic documents which are classified into 96 semantic categories. In this experiment, since we want to study applications of topic models in the standard clustering setting, where each document belongs to exactly one cluster or class, documents appearing in two or more categories are removed. We further select the largest 10 categories so as to eliminate outlier categories, leaving us with 7,456 documents in total. The Reuters corpus contains 21,578 documents which are grouped into 135 categories. It is much more unbalanced than TDT2, with some large clusters more than 30 times larger than some small ones. In our experiments, we discarded documents with multiple category labels, and

[1] http://projects.ldc.upenn.edu/TDT2/.

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

Table 1 Statistics of TDT2 and Reuters corpora

Statistics	TDT2	Reuters
Number of documents used	7,456	8,246
Number of unique words	47,047	26,269
Number of categories used	10	10
Average document length	401	117
Max category size	1,844	3,945
Min category size	167	116
Mean category size	746	825

only selected the largest 10 categories, which left us with 8,246 documents in total. Table 1 provides some statistics of the two document corpora.

Evaluation metric The clustering result is evaluated by comparing the Normalized mutual information (Xu et al. 2003; Cai et al. 2008) of the obtained label of each document and that provided by the document corpus (the ground truth). Let C denote the set of clusters obtained from the ground truth and C' obtained from our algorithm. Their mutual information metric $MI(C, C')$ is defined as:

$$MI(C,C') = \sum_{c_i \in C, c_j' \in C'} p(c_i, c_j') log_2 \frac{p(c_i, c_j')}{p(c_i)p(c_j')}$$

where $p(c_i)$ and $p(c_j{}')$ are the probabilities that a document arbitrarily selected from the corpus belongs to the clusters c_i, and is assigned to $c_j{}'$, respectively; $p(c_i, c_j{}')$ is the joint probability that the arbitrarily selected document belongs to the cluster c_i and is assigned to $c_j{}'$ at the same time. We use the normalized mutual information $\overline{MI}$:

$$\overline{MI} = \frac{MI(C,C')}{\max(H(C), H(C'))}$$

where $H(C)$ and $H(C')$ are the entropies of C and C', respectively. It is easy to see that $\overline{MI}$ ranges from 0 to 1. $\overline{MI} = 1$ if the cluster assignments and the ground truth are identical, and $\overline{MI} = 0$ if they are independent.

Default setting Unless specified otherwise, our default setting for document clustering uses $K = 10$ clusters, $\alpha = 50/K$ for LDA, and runs each topic model for 200 iterations.

4.2 Performance

We summarize the observations made from our experiments according to the questions to be answered. The first a few questions provide us with guidance on how to optimize the performance of topic models in clustering. The last question compares the two topic models under their optimal setting.

Does number of iterations matter? The inference and parameter estimation of all topic models are based on some iterative numerical algorithms. A practical parameter that has to be set for such an algorithm is the number of iterations. Existing work (e.g., Cai et al. 2008) often leaves out the discussion of this. A commonly adopted practice is to run the estimation until some convergence condition is met (for PLSA) or until a pre-set large number of iterations (2000 iterations are usually used in Gibbs sampling estimation of

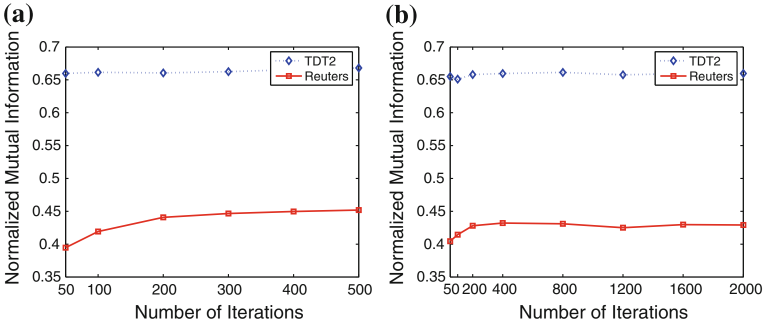


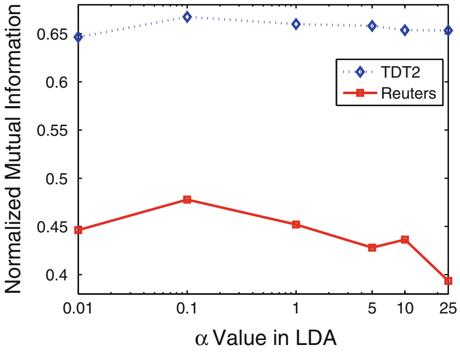
Fig. 1 Clustering performance versus number of iterations: performance converges after 200 iterations ($K = 10$). **a** PLSA: number of iterations, **b** LDA: number of iterations

LDA). This is often a time consuming process, which limits the use of topic models in a real world application. However, to achieve a good performance in a task such as clustering, it may not be necessary to reach the exact convergence of a topic model. Here we vary the number of iterations, and investigate the performance of PLSA and LDA on both data sets. The results in Fig. 1a, b show that after 200 iterations, performance becomes quite stable for both PLSA and LDA. Although the minimum number of iterations to achieve the near-optimum performance should vary across different data sets, we can tell from the performance curves that the clustering task performance is not very sensitive to "insufficient" iterations. For example, LDA with 200 iterations can achieve similar performance as with 2000 iterations which is commonly used.

How do topic models compare with k-means baseline? To compare topic models for clustering with a traditional clustering method, we have aslo tested a baseline of k-means. It gives a normalized mutual information of 0.4901 on Reuters and 0.8136 on TDT2, which are significantly higher than the best performance using PLSA or LDA. This suggests that using the most likely topic in topic models as the cluster is not as accurate as traditional clustering baseline such as k-means. This may be because topic models assume that all documents can potentially cover multiple topics, and such a topic may not necessarily aligned well to the topic covered in a cluster of documents.

How does α affect LDA? α is the hyper parameter in LDA which controls the smoothing effect of θ (see Eq. 6), where larger α results in more smoothed topics while $\alpha < 1$ would cause the modes of the Dirichlet distribution to be located at the corners of the simplex, thus favoring more sparse topics. A recommended setting is $\alpha = 50/K$ (Griffiths and Steyvers 2004; Steyvers and Griffiths 2007), keeping constant the sum of the Dirichlet hyper-parameters, which can be interpreted as the number of virtual samples contributing to the smoothing of θ. However, since in our definition of the clustering task, each document belongs to only one cluster, we would naturally expect topic models to assign a skewed topic distribution to a document, where there is a clear preference for some topic j. So we test this hypothesis with different values of α in LDA in Fig. 2. The default α of $50/K = 5$ clearly is not optimal. Using a smaller α, instead of the default, gives better clustering performance. It is interesting to notice that LDA with the Reuters collection is more sensitive when α value is changed compared with the TDT2 collection. One explanation is that Reuters has shorter documents and more diverse category distribution (as in Table 1).

Fig. 2 Clustering performance versus α: α affects clustering performance; $\alpha = 0.1$ significantly outperforms default setting ($K = 10$)



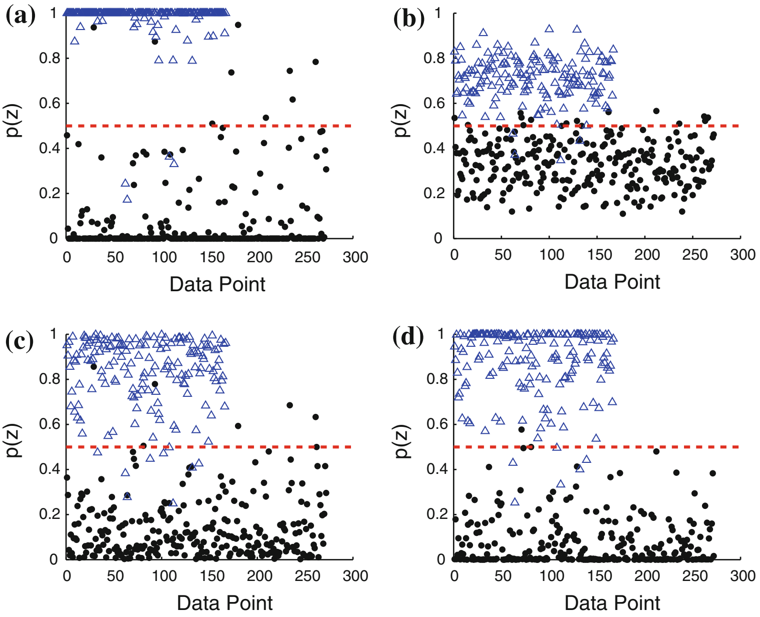


Fig. 3 Structure of two clusters using different α, compared with PLSA. **a** PLSA, **b** LDA with default $\alpha = 5$, **c** LDA with $\alpha = 1$, **d** LDA with $\alpha = 0.1$

To provide a more intuitive illustration of the smoothing effect of α, we show, in Fig. 3, a plot of $P(z)$ to visualize $P(z)$ together with the true class labels in the setting of two clusters. The data points represent documents while the two shapes of data points correspond to their true class labels. Since there are only two clusters, the y value of each data point represent the probability of the corresponding document assigned to one cluster; Thus 0.5 is the cut-off for assigning documents to the two clusters. We color the data points based on the gold standard. It could be observed that using a default setting $\alpha = 5$, the data points tend to locate in the middle area, so the clustering decision boundary is crowded, resulting in lower performance. As we use smaller α, the data pointers tend to move to the two extremes, which helps breaking close-tie in the boundary cases. For comparison, we also show the scatter plot for PLSA, which generates more skewed distribution of topics because it overfits the data by calculating a maximum likelihood estimate.

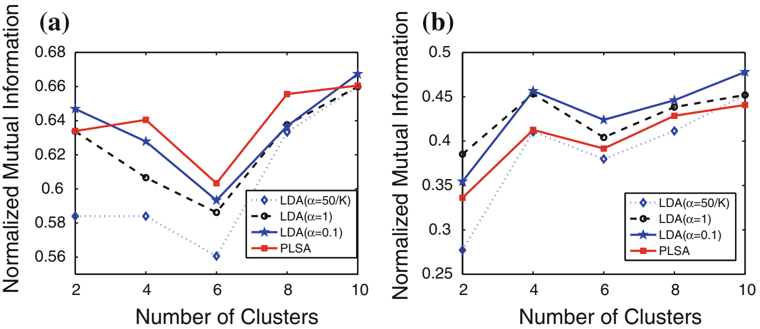


Fig. 4 Clustering performance versus number of clusters. **a** TDT2, **b** Reuters

Table 2 Stability of topic models in clustering

Data	$\overline{MI}$	LDA ($\alpha = 0.1$)	LDA ($\alpha = 1$)	PLSA
Reuters	Max	0.4803	0.4787	0.4837
	Min	0.4619	0.4537	0.4317
	Mean	0.4680	0.4646	0.4620
	SD	0.0110	0.0113	0.0195
TDT2	Max	0.6686	0.6762	0.6691
	Min	0.6300	0.6305	0.6266
	Mean	0.6492	0.6498	0.6441
	SD	0.0147	0.0146	0.0218

Does number of topics matter? To answer this question, we vary the number of clusters in Fig. 4. For each given cluster number K (except for $K = 10$, when we only have one run), 10 test runs were conducted on different randomly chosen clusters, and the final performance scores were calculated by averaging the scores from the 10 tests. Again, LDA using a default α value does not perform as good as PLSA especially when the number of clusters is smaller (i.e. α value is big). If we set a smaller $\alpha = 0.1$ LDA performance could be greatly improved over the default setting. But there is no clear sign that one topic model is better than the other, because even with the optimal parameter LDA outperforms PLSA on one data set but not the other.

Do topic models provide stable performance? Since in clustering task, we use the topic distribution θ directly to obtain the cluster label, the performance could be sensitive to different trials of topic model where different initial values of parameters are used. Indeed, PLSA has the problem of multiple local maxima and Gibbs LDA has the inconsistency issue between samples, because θ and ϕ are computed from only one sample. In Table 2, we show the MAX, MEAN, MIN, and standard deviation of the clustering performance of each topic model by running it for 10 independent trials. We can see that both models suffer from a non-trivial variance, which is larger when using PLSA. The variation of alpha that we tried does not seem to affect this stability. Thus, if we use topic models for clustering, especially PLSA, we do want to run multiple trials of the estimation to avoid the local maxima problem.

How do the two models compare? In our experiments, PLSA and LDA achieved comparable performance in terms of the normalized mutual information. When optimized, LDA generally outperforms PLSA on the Reuters dataset, but TDT2 presents a mixed pattern, where LDA outperforms PLSA when dealing with 2 clusters and 10 clusters, while PLSA outperforms LDA when dealing with 4, 6, and 8 clusters. Nevertheless, the difference is not statistically significant according to Wilcoxon signed rank test. Note that there is one more parameter to tune in LDA, which significantly affects the clustering performance. When a suboptimal parameter α is used (e.g., the default α), the performance of LDA quickly drops below PLSA.

5 Text categorization

Next we would like to examine the effectiveness of the low-dimensional latent semantic representation obtained from PLSA and LDA. For this purpose, we use text categorization as a test task, where we want to classify a document into one of the defined categories. We focus on the discriminative framework in (Blei et al. 2003). A challenging aspect in the discriminative framework is the choice of features. Unigram features are known as a reasonable representation and a strong baseline, where there are as many features as the number of words in the vocabulary and the value of a feature is the normalized frequency of the corresponding word, i.e. $\frac{c(w,d)}{|d|}$. As an alternative, we can use a topic-based representation with each $P(z = j)$ as an element in the vector; we thus have K features, corresponding to K topics. More specifically, topic models can be applied on all the documents without reference to their true class label. After that the dimensionality of the feature space is reduced to the number of topics, and the new feature set becomes $\theta^{(d)} = P(z|d)$. Compared with document clustering, where only the most likely topic in $\theta^{(d)}$ affects the performance, the categorization performance is expected to be more sensitive to the whole $\theta^{(d)}$.

5.1 Experimental setup

Data sets and evaluation metric The same two data sets are used as those in document clustering. We utilize the $SVM^{multiclass}$ toolkit.[3] In order to compare unigram representation with topic representation, we feed them as features to the classifier and evaluate the error rate of the multi-class classification. A lower error rate indicates a better feature representation.

Default setting In order to examine each parameter separately, we fix other parameters to some default settings. Unless specified otherwise, we use 10 categories, a random subset of 0.01 (1%) as training data, $\alpha = 50/10 = 5$ for LDA, run each topic model for 200 iterations to estimate 100 topics, and average the scores of each topic models from three independent trials. The regularization parameter in $SVM^{multiclass}$ that trades off margin size and training error is tuned to be optimal for each feature representation and for each training size.

[3] http://svmlight.joachims.org/svm_multiclass.html.

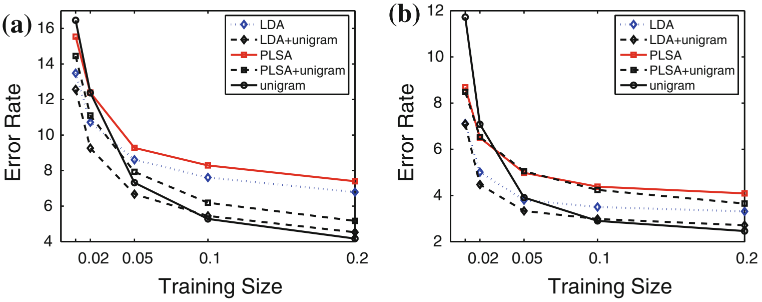


Fig. 5 Classification error rate versus training size. **a** Error rate versus training size on Reuters, **b** error rate versus training size on TDT2

5.2 Performance

The observations made from our experiments can be summarized with answers to the following questions:

Do topic models help? We would like to see whether/when the low-dimensional representation by topic model could improve over the unigram baseline. Figure 5a, b show the classification error rates of unigram, PLSA and LDA with different percentage of training data on two data sets. It can be seen that, low-dimensional representation of topic models outperforms the unigram baseline when the training data is small, i.e. 1 or 2%. But as the size of training data increases, the richness of unigram features demonstrates its superiority. What is more important, if we further combine the unigram feature with the topic model features, it usually helps even more, showing that combining low-dimension latent semantic representation with high-dimension original text representation is more robust and effective. However, in order to study the property of topic models more directly, we only use topic features in low-dimension in our following experiments. Besides, we observe that LDA consistently outperforms PLSA on both data sets, indicating that (1) PLSA may suffer from the over-fitting problem since it does not make any assumption about how θ is generated; and (2) LDA is more effective when used as latent semantic structure.

Do topic models provide stable performance? First, in order to see (1) how multiple local maxima affect categorization and (2) how many iterations are appropriate to achieve stable performance, we run topic models with different number of iterations in three trials. The results on both data sets are shown in Fig. 6. First of all, the variance among different trials of PLSA can be large. Especially, one trial on Reuters data is significantly worse than the other two, but all trials outperform the baseline after 100 iterations. Second, the error rate does not necessarily decrease as we run more interations of topic model estimation: In the case of LDA, sampling method is used, so it is not surprising that the performance can be fluctuating across different number of iterations as in Fig. 6a, c; in the case of PLSA, error rate may even go up as we run more iterations as in Fig. 6d. In order to examine whether this is caused by the over-fitting effect of PLSA, we also look into the training errors in Fig. 7: although the EM algorithm of PLSA is guaranteed to achieve higher likelihood values after more iterations, the training errors are not always decreasing. This observation rules out the possibility of overfitting. An intuitive explanation is that since topic models

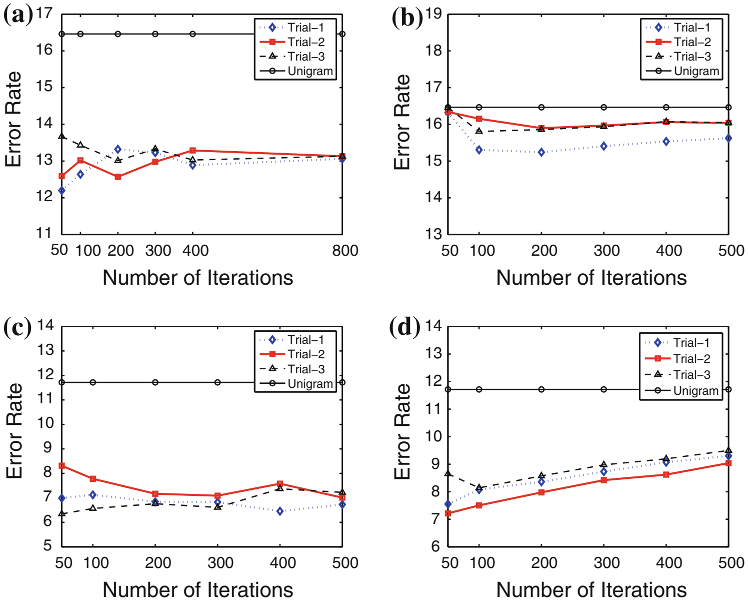


Fig. 6 Classification error rate versus number of iterations. **a** LDA: number of iterations on Reuters, **b** PLSA: number of iterations on Reuters, **c** LDA: number of iterations on TDT2, **d** PLSA: number of iterations on TDT2

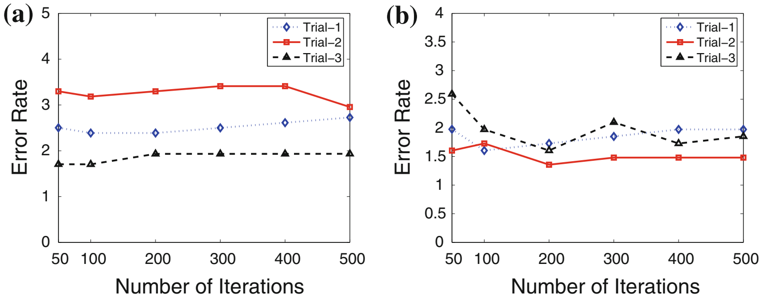


Fig. 7 Classification training error rate versus number of iterations. **a** PLSA: training error on Reuters, **b** PLSA: training error on TDT2

are used for dimension reduction in the classification task, larger likelihood values do not mean that data in the reduced dimensions are easier to separate.

As a result, for fair comparison, in the rest of the experiments, we average the performance of three trials for each topic models, and each trial is run for 200 iterations.

Does number of topics matter? How does the number of topics affect classification? As shown in Fig. 8, as long as we use 20–100 topics, topic models outperform the unigram baseline in both data sets. But PLSA is more sensitive to the number of topics used: 50 topics performs best, and using more topics hurts the performance. In contrast, the error rate of using LDA is more stable if at least 50 topics are used.

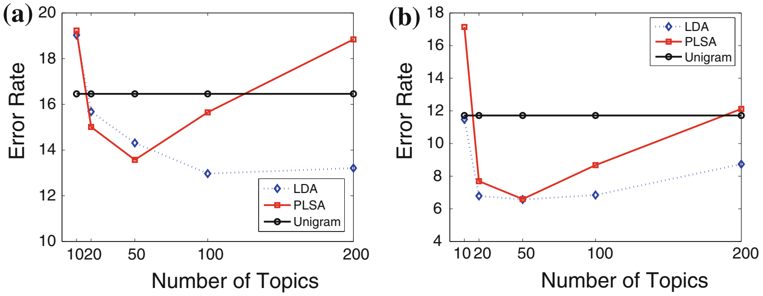


Fig. 8 Classification error rate versus number of topics. **a** Error rate versus number of topics on Reuters, **b** error rate versus number of topics on TDT2

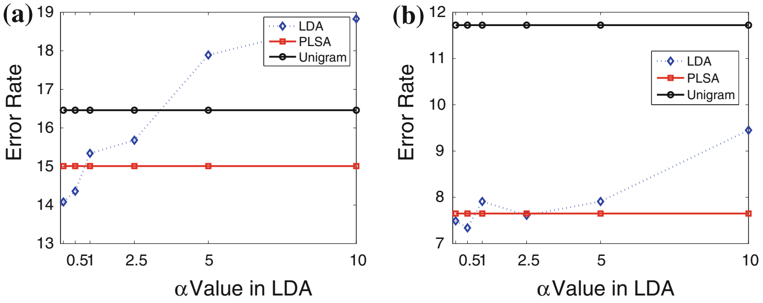


Fig. 9 Classification error rate versus α. **a** Error rate versus α on Reuters, **b** error rate versus α on TDT2

How does α affect LDA? Is the default α in LDA optimal in categorization task? Figure 9a, b show the error rate of LDA with different values of the α parameter on Reuters data; for comparison, we also plot the error rate of unigram and PLSA with the same setting. The plot for TDT2 data has a similar trend. Using 20 topics, LDA with the default α setting 2.5 does not perform as well as PLSA. But a smaller α, e.g. 0.1 or 0.5 tends to produce better performance than PLSA. One possible explanation is that using a large α does not work well in classification, because too much smoothing decreases the discrimination among topics.

How do the two models compare? In general, LDA with the optimal setup works better in text categorization than PLSA, although the difference is only statistically significant on the Reuters data at 95% level based on Wilcoxon test. Note that in a text classification task, the topic-document distribution θ is used as low dimension representation of documents. We can see that when training data is small, the problem of over-fitting has limited the performance of PLSA. On the other hand, when the training set is large enough, simple unigram features outperform the low dimension representation of documents when either of the topic models is used. Despite its better optimal performance, LDA has one more parameter to tune than PLSA. When a suboptimal α is used, the performance of LDA could become worse than PLSA and even worse than the unigram baseline.

6 Ad-hoc information retrieval

In the previous two tasks, we test the document-topic distribution θ through clustering and categorization. What we have not explored is the topic-word distribution ϕ, which could be used to manipulate the representation of text documents. In this section, we investigate the topic-word distribution of the two topic models in the context of ad hoc document retrieval.

We adopt KL-Divergence (Zhai and Lafferty 2001) as our retrieval framework, which is one of the most effective retrieval models derived under the language modeling framework. In this framework, queries and documents are all represented by unigram language models. And it scores a document d with respect to a query q by computing the Kullback-Leibler divergence between the query language model θ_q and the document language model θ_d as follows:

$$-D_{KL}(\theta_d||\theta_d) = -\sum_{w \in V} P(w|\theta_q) log \frac{P(w|\theta_q)}{P(w|\theta_d)}$$

where V is the set of all words in the vocabulary.

What remains to be solved is how to appropriately estimate the θ_q and θ_d. The query language model θ_q can be estimated using maximum likelihood which equals the empirical distribution.

$$P(w|\theta_q) = P(w|q) = \frac{c(w,q)}{|q|}$$

where $c(w, q)$ is the number of occurrences of word w in query q, and $|q|$ is the length of query. A state-of-the-art baseline of document language model θ_d is estimated with Dirichlet prior smoothing method (Zhai and Lafferty 2001) as shown below:

$$P(w|d) = \frac{c(w,d) + \mu \cdot P(w|\mathcal{C})}{|d| + \mu}$$

where $c(w, d)$ is the number of occurrences of word w in document d, $P(w|\mathcal{C})$ is the empirical word distribution in the whole collection $\mathcal{C}$, and μ is a parameter that controls the degree of smoothing.

The performance of a retrieval model essentially depends on both the query language model and the document language model. Topic models can be applied to improve the estimation of θ_d, as the information in a document tends to be too sparse to estimate an accurate θ_d. And topic models can be applied to further smooth the document representation. Following Wei and Bruce Croft (2006), we use the topic model representation of a document to further smooth the document language model.

$$P(w|\theta_d) = (1-\lambda)\frac{c(w,d) + \mu P(w|\mathcal{C})}{|d| + \mu} + \lambda P_{TM}(w|\theta_d) \tag{8}$$

where

$$P_{TM}(w|\theta_d) = \sum_{j=1}^{K} \phi_w^{(j)} \theta_j^{(d)},$$

and $\hat{\phi}$, $\hat{\theta}$ are the estimated the parameters of a topic model.

Table 3 Statistics of SJMN and LA corpora

Statistics	SJMN	LA	TREC78
Number of documents	90,257	131,896	528,155
Number of unique terms	334,913	326,609	730,270
Average document length	266	290	481
Collection size (Gb)	0.29	0.48	2.15

Note the two extreme cases: (1) if $\lambda = 0$, we simply use the document language model with Dirichlet smoothing, which is our baseline; (2) if $\lambda = 1$, we simply use the document representation computed from the topic models.

Fold-in heuristic Note that to compute $P_{TM}(w|\theta_d)$, one doesn't have to estimate the topic models from the entire collection. Instead, a "fold-in" scheme can be used. As stated in Hofmann (1999b), folding-in is a heuristic way to solve the problem of computing a representation for a document that was not contained in the original training collection where the topic models are estimated. Instead of estimating topic model parameters from all the documents, we estimate topics $P(w|\hat{\phi}, z)$ from a randomly sampled subset, then we use "fold-in" heuristic to estimate the topic distributions θ for all documents. Basically, it can be computed using an EM algorithm, similar to that used in PLSA estimation, except that the $P(w|\hat{\phi}, z)$ is fixed so that only θs are updated in the M step. Essentially, PLSA and LDA produce different $P(w|\hat{\phi}, z)$ on the sampled subset; after that, the same "fold-in" heuristic is applied to estimate θ. Thus any performance difference between LDA and PLSA would be entirely due to the different estimate of $P(w|\hat{\phi}, z)$. There are two advantages of using a "fold-in" scheme: (1) the fold-in scheme investigates the adaptiveness of topic models across different document collections; (2) folding-in with a sampled document set is much more efficient than estimating topic models based on the whole collection.

6.1 Experimental setup

Data sets and evaluation metric Our goal is study the empirical difference between PLSA and LDA for the retrieval task; thus, to ensure a thorough and fair comparison, we need to run many experiment using different settings for each algorithm. However, due to the high computational complexity of PLSA and LDA, we mainly employ two modest-size test collections from TREC: San Jose Mercury News (SJMN) 1991 with queries 51–150, and LA Times (LA) with queries 301–400, which follows the practice of previous works on using topic models for information retrieval (Wei and Bruce Croft 2006, Yi and Allan 2009). Moreover, we also conduct experiments on TREC7/TREC8 which is a relatively large heterogeneous collection with queries 351–400 (for TREC7) and 401–450 (for TREC8). However, because of the high computational complexity of PLSA and LDA, it is infeasible to vary all the parameters to run many experiments. We thus focused on using this large data set to test whether the major conclusions we draw on the two modest-size data sets can be generalized to this larger data set. Specifically, we focused on the comparison of PLSA and LDA and varied the most important parameter α of LDA. Even for such focused experiments, it is still expensive to estimate PLSA and LDA on the whole

[4] For example, estimating PLSA on the whole collection of LA would take about 40 h for a single run of 50 iterations on a Linux server four 2.2 GHz AMD Opteron 848 processors and 32GB memory.

collection, we thus opted to estimate PLSA and LDA on a random subset of 10K documents.[4] Although the topic models are estimated using a subset, the application of the model for retrieval is done for the *entire* set of documents using the fold-in heuristic. Queries are taken from the "title" field of TREC topics. Relevance judgments are taken from the judged pool of top retrieved documents by various participating retrieval systems from previous TREC conferences.

Statistics of the collections are given in Table 3. Mean average precision (MAP) is used as the basis of evaluation for ad hoc retrieval, because it reflects the quality of the overall ranking accuracy.

Default setting Our default setting for ad hoc retrieval uses 5% of the collection to estimate topic models, $\alpha = 50/K$ for LDA, and runs each topic model for 50 iterations on 400 topics (which is the recommended setting in Wei and Bruce Croft (2006). Unless specified otherwise, the performance we report is the best MAP performance by tuning the μ and λ parameter in the document language model.

6.2 Performance

As in the previous sections, we analyze the results on retrieval evaluation according to the questions to be answered. The first a few questions provide guidance on how to optimize the task performance of topic models in the context of ad hoc retrieval. The last question compares the two topic models with their optimal setting.

Do topic models help? Our general observation on two data set is that smoothing with topic models improves over the KL-divergence baseline if $\lambda \in [0.5, 0.7]$ and μ is set to a reasonable range, e.g. [250,1000]. Recall that λ is the parameter which controls the effect of topic models in smoothing. This means smoothing using the topic model based representation indeed improves retrieval performance.

Can we use topic models alone? We have also tried to set $\lambda = 1$ deliberately, which is essentially using only the topic model representation for a document based on topics. The MAP performance is consistently below 0.1 for both data sets, which is significantly worse than the baseline. This indicates that in ad hoc retrieval task, which requires a fine granularity matching of query words, low dimension representation of topics is not fine enough for distinguishing relevant documents from irrelevant ones. So an interpolation of document representation is important.

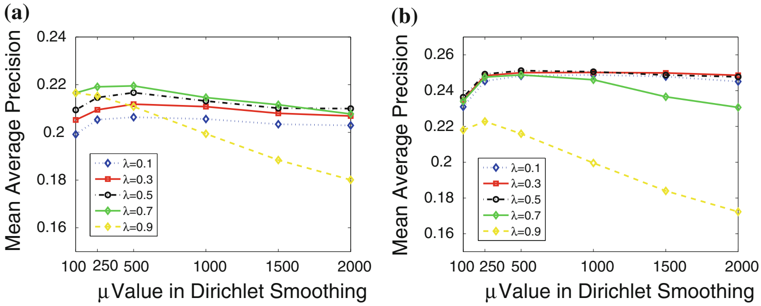


Fig. 10 MAP versus μ. **a** SJMN: MAP versus μ, **b** LA: MAP versus μ

Are two smoothing methods interfering? There is some correlation of setting the optimal Dirichlet prior smoothing parameter μ and topic model smoothing parameter λ. In Fig. 10, we plot the retrieval performance of PLSA on SJMN and LA data for different combinations of μ and λ. (The trend is similar for LDA) We can observe that: when λ is small, e.g. 0.1 or 0.3, the retrieval performance is not so sensitive to the value of Dirichlet smoothing parameter μ; however, as λ goes larger, we need to set a smaller μ in order to get better retrieval performance. In Eq. 8, we can see that both μ and λ provide smoothing effect of the original document model. So the observation from Fig. 10 shows that we need to reserve enough weights for the original document representation and "over-smoothing" the document model (e.g. by setting $\lambda = 0.9$ and $\mu = 2000$) would deteriorate performance of information retrieval where fine granularity matching of query words is important. In our experiments, $\mu = 500$ and $\lambda \in [0.5,0.7]$ usually gives the near-optimal performance.

Does number of iterations matter? We test the number of iterations of topic models on SJMN and LA data in Fig. 11. Performance of both models becomes relatively stable when iteration number reaches 50, which is similar to the findings of LDA-based retrieval (Wei and Bruce Croft 2006). Again, due to the sampling procedure, LDA is not guaranteed to achieve better performance if we run it for more iterations (e.g. Fig. 11a).

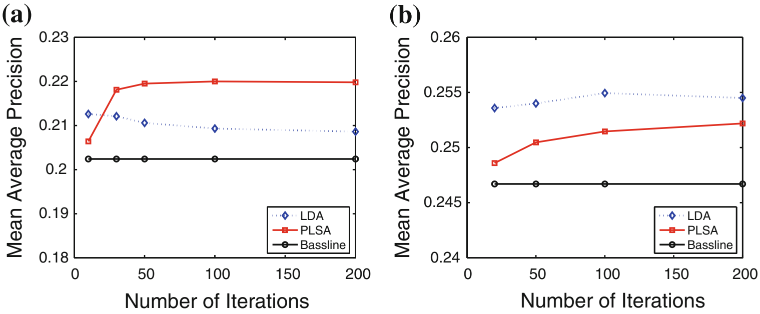


Fig. 11 MAP versus number of iterations. **a** SJMN: number of iterations, **b** LA: number of iterations.

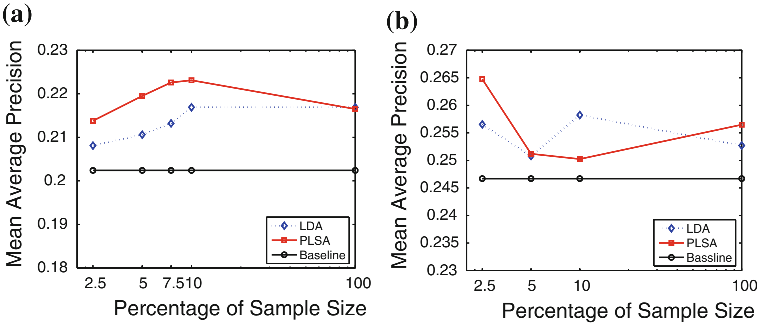


Fig. 12 MAP versus sample size (in log scale). **a** SJMN: sample size, **b** LA: sample size

Does fold-in work? It would be interesting to see how our "fold-in" scheme work with different number of documents to use as the subset. In Fig. 12, we show the MAP performance when using different size of subset to learn the topic models. Interestingly, it is not necessarily true that the larger subset we use the better retrieval performance we can achieve. This could be explained by the fact that: topic models are estimated by maximizing the collection likelihood, essentially

$$\prod_d \prod_w P_{TM}(w|d) = \prod_d \prod_w \sum_{j=1}^{K} \phi_w^{(j)} \theta_j^{(d)}$$

Each $P_{TM}(w|d)$ is then used to smooth the document language model. If the estimation is "perfect," $P_{TM}(w|d)$ would be the same as $P_{ML}(w|d)$, where there is no effect of topic model smoothing at all; instead smoothing with λ would essentially discount the Dirichlet smoothing effect of μ. However, if we estimate word-topic distribution $\phi_w^{(j)}$ on a subset, and use fold-in to get topic-document distribution $\theta_j^{(d)}$, the new $P_{TM}(w|d)$ can obtain more smoothing power.

Does number of topics matter? Selecting the right number of topics is also an important problem in topic modeling. As we can see in Fig. 13a, b, although the topic model performance is sensitive to the number of topics used, it consistently outperforms the baseline for a wide range of numbers of topics.

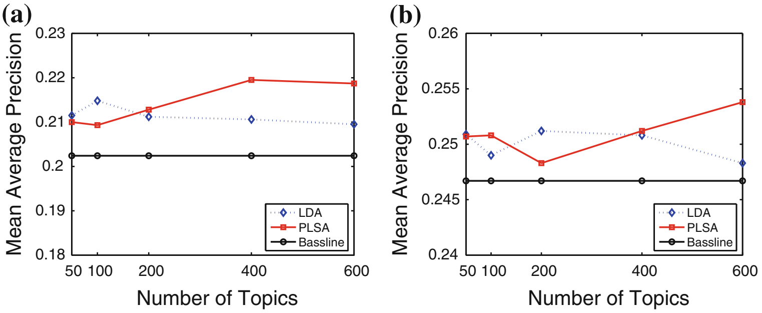


Fig. 13 MAP versus number of topics. **a** SJMN: number of topics, **b** LA: number of topics

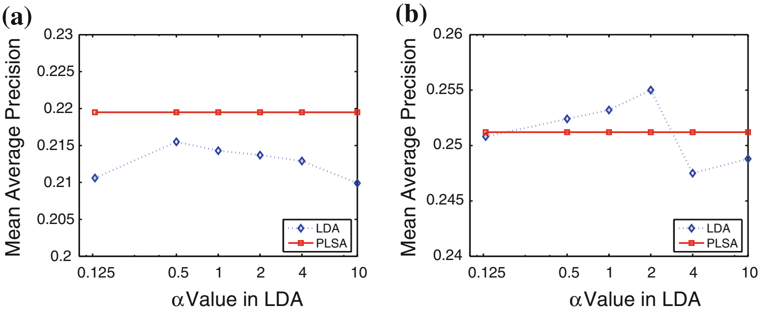


Fig. 14 MAP versus α in LDA: retrieval performance is sensitive to α; optimal α lies between 0.5 and 2. **a** SJMN: α value in LDA, **b** LA: α value in LDA.

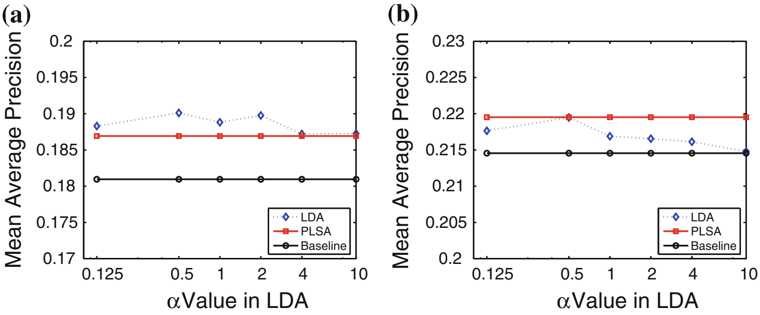


Fig. 15 MAP versus α in LDA: retrieval performance is sensitive to α; optimal α lies between 0.5 and 2. **a** TREC7: α value in LDA, **b** TREC8: α value in LDA

Table 4 Retrieval trial variance

Data set	SJMN		LA	
Topic model	LDA	PLSA	LDA	PLSA
Trial-1	0.2138	0.2179	0.2508	0.2512
Trial-2	0.2115	0.2185	0.2542	0.2527
Trial-3	0.2106	0.2195	0.2540	0.2525
Mean	0.2120	0.2186	0.2530	0.2521
SD	0.0016	0.0008	0.0019	0.0008

How does α affect LDA? In the previous experiments, we used the default α in LDA, i.e. $\alpha = 50/K$. Since we use a large number of topics in retrieval (K has a large value), resulting in small value of α, the estimated topic distribution is less smooth with topics more closely fitting the sampled subset. So during the fold-in process, those topics may not generalize well on the unseen documents. Indeed, in Fig. 14a, b, we see that setting a larger α, e.g. between 0.5 and 2, gives better retrieval performance, which is to make the topics learned from the subset more smooth and general.

To further test this important observation, we conduct this experiment on TREC7/TREC8 data, where a subset of 10K documents are used to estimate the topic models. Although the topic models are estimated using a subset, the application of the model for retrieval is done for the entire set of documents. In Fig. 15, we plot the MAP performance of the KL-Divergence baseline, smoothing with PLSA, and smoothing with LDA using different α parameters. Consistent with previous finding, the optimum α is 0.5 in both TREC7 and TREC8, which is larger than the optimum setting in the classification and clustering tasks.

Do topic models provide stable performance? Finally, we run each topic model with the same setting for three times with different random starting points, and it turns out that the retrieval performance is not very sensitive to different trials (see Table 4). This is because we interpolate the topic models with the original representation of documents, which is a more indirect way of using the learned topic models than in clustering or categorization.

How do the two models compare? Based on the observations from the experiments, PLSA and LDA achieve comparable performance when used appropriately in retrieval.

PLSA appears superior than LDA on the SJMN collection, and LDA outperforms PLSA on the LA collection with its optimal setup; but neither of the differences is statistically significant based on Wilcoxon test. However, unlike PLSA, the performance of LDA is contingent on the optimal setting of parameter α. When a suboptimal α is used, the performance of LDA drops considerably and thus performs worse than PLSA.

7 Conclusions

In this paper, we conduct a systematic and thorough investigation of two representative probabilistic topic models, probabilistic latent semantic analysis (PLSA) and Latent Dirichlet Allocation (LDA), using three representative text mining tasks, including document clustering, text categorization and ad-hoc retrieval. To summarize the findings in our experiments with PLSA and LDA in the three tasks, a few points could be made:

How to use topic models appropriately to improve task performance? When used inappropriately, topic models could hurt the task performance. Our emperical findings suggest that when used as a latent semantic representation, topic models are more effective than the original high dimension representation when both (1) the task only requires coarse category matching, such as text categorization and (2) the data is sparse (training data is small) so that the latent semantic representation helps bridging the vocabulary gap. In tasks like retrieval which needs fine granularity matching of query words, the latent semantic representation alone is not sufficient due to the lack of discrimination. As a result, combining the low dimension representation of topic models with the original high dimension representation tends to be most robust and effective, such as combining unigram features with topic features for text categorization and smoothing document language models with topic distributions for ad hoc retrieval.

How does hyper-parameter α affect task performance? The performance of LDA on all tasks is quite sensitive to the setting of its hyper-parameter α, and the optimal setting varies according to how the model is used in a task. In tasks where the latent semantic representation of topics is used alone, such as classification and clustering, optimal performance is achieved when α is set to be small, e.g. between 0.1 and 0.5. In tasks like retrieval, where we interpolate the original high-dimensional representation with the topics, a relatively larger α between 0.5 and 2 would give better performance. Our experiments also show that there is no stable single optimal value of α. The choice of this parameter also depends on the collection.

How does local maxima affect task performance? Although there is no guarantee of finding the global maximum for topic models, the problem of getting to only local maxima does not necessarily affect task performance much. If we use the topics directly as in classification and clustering, we do want to run topic models multiple times, especially PLSA. But if we use topics indirectly, a finely tuned topic model does not necessarily lead to good task performance. For example, the retrieval performance is not sensitive to different trials of topic models, since only aggregated topic model parameters are used to smooth the document language model in retrieval, which does not need a finely tuned topic model to achieve optimal performance. This observation can help saving some concerns of local maxima in applications where the learned topics are not used directly by themselves.

Which topic model is better? From our emprical observations, PLSA and LDA (when optimized) perform similarly on two out of the three application tasks, i.e. clustering where the most significant topic is used as cluster label and retrieval where only the word-topic distribution is different for the two models. However, LDA works better in text categorization where the topic-document distribution θ is used as low dimension representation. Although it has been shown in existing work (Blei et al. 2003) that LDA is better than PLSA by measuring perplexity of predicting new documents, evidences from emperical experiments indicate that LDA has a competitive edge over PLSA at computing a *generalizable* fine-granularity low-dimension semantic representation by eliminating the overfitting problem; while in other cases there is no clear distinction of task performance difference. This provides more insight for practitioners who would like to choose one of the two models.

A major limitation of our study is that due to the high computational complexity of current topic model implementations, the size of collections used in evaluation (especially in the case of retrieval experiments) is relatively small. Thus an important future work is to use more efficient implementations (e.g., based on parallel algorithms) to further confirm our observations on larger-scale experiments. Our study has exclusively focused on PLSA and LDA, the two representative basic topic models. In the future, it would be interesting to further evaluate other topic models as well as other commonly used estimation algorithms for solving topic models. We are also interested in investigating some other tasks, such as summarization, which is the most popular application of topic models and also the most difficult one to quantitatively evaluate.Another future direction is to conduct a comparison study between topic models and the non-negative matrix factorization family.

References

Blei, D. M., Griffiths, T. L., Jordan, M. I., & Tenenbaum, J. B. (2004). Hierarchical topic models and the nested chinese restaurant process. In *Advances in neural information processing systems* (p. 2003). MIT Press.

Blei, D. M., & Lafferty, J. D. (2005). Correlated topic models. In *NIPS*. MIT Press

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research, 3*, 993–1022.

Cai, D., Mei, Q., Han, J., & Zhai, C. (2008). Modeling hidden topics on document manifold. In J. G. Shanahan, S. Amer-Yahia, I. Manolescu, Y. Zhang, D. A. Evans, A. Kolcz, K.-S. Choi, & A. Chowdhury (Eds.), *CIKM* (pp. 911–920). ACM.

Chang, J., Boyd-Graber, J., Wang, C., Gerrish, S., & Blei, D. M. (2009). Reading tea leaves: How humans interpret topic models. In *Neural information processing systems*. MIT Press.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of Royal Statistical Society, Series B, 39*, 1–38.

Gaussier, E., & Goutte, C. (2005). Relation between plsa and nmf and implications. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 601–602). New York, NY, USA: ACM.

Geman, S., & Geman, D. (1984). Stochastic relaxation, gibbs distributions and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 6*(6), 721–741.

Girolami, M., & Kabán, A. (2003). On an equivalence between plsi and lda. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on research and development in informaion retrieval* (pp. 433–434). New York, NY, USA: ACM.

Griffiths, T. L., & Steyvers, M. (2004). Finding scientific topics. *Proceedings of the National Academy of Sciences of the United States of America, 101*(Suppl. 1), 5228–5235.

Hofmann, T. (1999a). Probabilistic latent semantic analysis. In K. B. Laskey & H. Prade (Eds.), *UAI* (pp. 289–296). Morgan Kaufmann.

Hofmann, T. (1999b). Probabilistic latent semantic indexing. In *SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on research and development in information retrieval* (pp. 50–57). New York, NY, USA. ACM.

Lacoste-Julien, S., Sha, F., & Jordan, M. I. (2008). Disclda: Discriminative learning for dimensionality reduction and classification. In D. Koller, D. Schuurmans, Y. Bengio, & L. Bottou (Eds.), *NIPS* (pp. 897–904). MIT Press.

Li, W., & Mccallum, A. (2006). Pachinko allocation: Dag-structured mixture models of topic correlations. In *ICML '06* (pp. 577–584). ACM.

Mei, Q., Cai, D., Zhang, D., & Zhai, C. (2008). Topic modeling with network regularization. In *WWW '08: Proceeding of the 17th international conference on World Wide Web* (pp. 101–110). New York, NY, USA: ACM.

Mei, Q., Ling, X., Wondra, M., Su, H., & Zhai, C. (2007). Topic sentiment mixture: modeling facets and opinions in weblogs. In *WWW '07: Proceedings of the 16th international conference on World Wide Web* (pp. 171–180). New York, NY, USA: ACM.

Mei, Q., & Zhai, C. (2006). A mixture model for contextual text mining. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 649–655). New York, NY, USA: ACM.

Minka, T. P., & Lafferty, J. D. (2002). Expectation-propogation for the generative aspect model. In A. Darwiche, & N. Friedman (Eds.), *UAI* (pp. 352–359). Morgan Kaufmann.

Nallapati, R. M., Ahmed, A., Xing, E. P., & Cohen, W. W. (2008). Joint latent topic models for text and citations. In *Proceeding of the 14th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 542–550). New York, NY, USA: ACM.

Steyvers, M., & Griffiths, T. (2007). *Probabilistic topic models*. Lawrence Erlbaum Associates.

Steyvers, M., Smyth, P., Rosen-Zvi, M., & Griffiths, T. (2004). Probabilistic author-topic models for information discovery. In *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 306–315). New York, NY, USA: ACM.

Teh, Y. W., & Görür, D. (2009). Indian buffet processes with power-law behavior. In *Advances in neural information processing systems*. MIT Press.

Wallach, H. M., Murray, I., Salakhutdinov, R., & Mimno, D. (2009). Evaluation methods for topic models. In *ICML '09: Proceedings of the 26th annual international conference on machine learning* (pp. 1105–1112). New York, NY, USA: ACM.

Wang, X., & McCallum, A. (2006). Topics over time: A non-markov continuous-time model of topical trends. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 424–433). New York, NY, USA: ACM.

Wei, X., & Bruce Croft, W. (2006). Lda-based document models for ad-hoc retrieval. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 178–185). New York, NY, USA: ACM.

Xu, W., Liu, X., & Gong, Y. (2003). Document clustering based on non-negative matrix factorization. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on research and development in informaion retrieval* (pp. 267–273). New York, NY, USA: ACM.

Yi, X., & Allan, J. (2009). A comparative study of utilizing topic models for information retrieval. In *ECIR '09: Proceedings of the 31th European conference on IR research on advances in information retrieval* (pp. 29–41). Berlin, Heidelberg: Springer.

Zhai, C., & Lafferty, J. (2001). Model-based feedback in the language modeling approach to information retrieval. In *CIKM '01: Proceedings of the tenth international conference on information and knowledge management* (pp. 403–410). New York, NY, USA: ACM.

Zhai, C., & Lafferty, J. (2001). A study of smoothing methods for language models applied to ad hoc information retrieval. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 334–342). New York, NY, USA: ACM.

Zhai, C., Velivelli, A., & Yu, B. (2004). A cross-collection mixture model for comparative text mining. In *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 743–748). New York, NY, USA: ACM.