ELSEVIER

PATTERN RECOGNITION

Locally discriminative topic modeling

Hao Wu, Jiajun Bu, Chun Chen, Jianke Zhu *, Lijun Zhang, Haifeng Liu, Can Wang, Deng Cai

College of Computer Science, Zhejiang University, Hangzhou 310027, China

ARTICLE INFO

Article history:
Received 25 June 2010
Received in revised form
25 January 2011
Accepted 29 April 2011
Available online 20 May 2011

Keywords:
Topic modeling
Generative
Discriminative
Local learning

ABSTRACT

Topic modeling is a powerful tool for discovering the underlying or hidden structure in text corpora. Typical algorithms for topic modeling include probabilistic latent semantic analysis (PLSA) and latent Dirichlet allocation (LDA). Despite their different inspirations, both approaches are instances of generative model, whereas the discriminative structure of the documents is ignored. In this paper, we propose *locally discriminative topic model* (LDTM), a novel topic modeling approach which considers both *generative* and *discriminative* structures of the data space. Different from PLSA and LDA in which the topic distribution of a document is dependent on all the other documents, LDTM takes a local perspective that the topic distribution of each document is strongly dependent on its neighbors. By modeling the local relationships of documents within each neighborhood via a local linear model, we learn topic distributions that vary smoothly along the geodesics of the data manifold, and can better capture the discriminative structure in the data. The experimental results on text clustering and web page categorization demonstrate the effectiveness of our proposed approach.

1. Introduction

With the increasing growth of digital data on the web, the automatic tools for exploratory data analysis are in great demand in various elds, including data mining, machine learning, pattern recognition and information retrieval. As one of the representative exploratory data analysis tools, probabilistic topic modeling [1] has received considerable attentions in recent years [2 9]. Topic modeling approaches can provide concise topical descriptions of document corpora that are semantically interpretable by human [10]. At the same time, they can also preserve the underlying statistical relationships that are helpful for document indexing, organization and common discriminative tasks such as clustering and classi cation [11].

Two of the most popular topic modeling algorithms are probabilistic latent semantic analysis (PLSA) [12] and latent Dirichlet allocation (LDA) [11]. Both methods are generative models that model each document as a mixture over a xed set of underlying topics, where each topic is characterized as a distribution over words. Speci cally, each word w in d is assumed to be generated from a distribution over words speci c to a latent topic z, where z is sampled from a distribution corresponding to d with a probability $P(z|d)$. The topic probabilities can be indirectly inferred by maximizing the log-likelihood of the data to be generated. One limitation of these two approaches is that they fail to consider the intrinsic geometrical structure of the data space [13].

In contrast to *generative* approaches, *discriminative* approaches

It is worthwhile to notice that LDTM in spirit is closely related to locally consistent topic modeling (LTM) [22,13]. In particular, LTM puts the smoothness constraint on the topic distributions using the Laplacian or manifold regularizer [23] to emphasize the pairwise similarities. It de nes a cost function in the form of the summation of $D(P(z|d_i)||P(z|d_j))W_{ij}$ over all document pairs, where $D(\cdot\|\cdot)$ denotes the distance between two topic distributions according to some distance metric such as KL-divergence, and W_{ij} is the edge weight of d_i and d_j in the nearest neighbor graph. The performance of LTM, therefore, largely relies on the weight assignment. Different from LTM, LDTM has its distinct features:

1. LDTM learns the graph Laplacian automatically using a local learning approach to model the topic relation between a document and its neighbors. This attains the robustness of model by avoiding the explicit assignment of the edge weights in the graph, to which the model (e.g., LTM) is very sensitive.
2. LDTM provides a complementary *discriminative* learning scheme to infer the topic distributions via a learning machine, i.e., regressions, in hope of boosting the *generative* scheme for topic modeling. This is beyond the focus of LTM. Moreover, LDTM is exible to incorporate any other generative scheme (e.g., LDA) or discriminative learning scheme (e.g., SVM) as an alternative.

2. Background

Two of the most popular probabilistic topic modeling approaches are probabilistic latent semantic analysis (PLSA) [12] and latent Dirichlet allocation (LDA) [11]. Both of these two models assume documents are generated by the activation of a xed set of latent topics, where each topic are modeled as a distribution over words.

Speci cally, PLSA, which is also known as *aspect model*, is indeed a latent variable model for general co-occurrence data which associates with an unobserved topical class variable $z \in = \{z_1 \ldots z_K\}$ with each observation, i.e., with each occurrence of a word $w \in = \{w_1 \ldots w_M\}$ in a document $d \in = \{d_1 \ldots d_N\}$. As a generative model, PLSA simulates the data generation process by de ning a joint probability model:

$$P(d\ w) = P(d)P(w|d)$$

$$P(w|d) = \sum_{z \in} P(w|z)P(z|d). \tag{1}$$

The parameters are estimated by maximizing the log-likelihood of the whole collection to be generated:

$$= \sum_{d \in} \sum_{w \in} n(d\ w)\log P(d\ w) \propto \sum_{d \in} \sum_{w \in} n(d\ w)\log \sum_{z \in} P(w|z)P(z|d) \tag{2}$$

where $n(d,w)$ denotes the number of times w occurred in d. The standard EM algorithm [24] is applied to estimate the parameters $\{P(w|z)\ P(z|d)\}_{w\ z\ d}$.

Note that PLSA estimates the topic distributions $P(z|d)$ independently for different d, therefore the number of parameters, which is $KM+KN$, grows linearly with the number of training documents N. This indicates that PLSA is susceptible to over tting [11]. To overcome this problem, latent Dirichlet allocation (LDA) [11] treats the topic distribution as a K-dimensional Dirichlet random variable. Thus the $KM+K$ parameters in a K-topic LDA model do not grow with the size of the training corpus and LDA does not suffer from the same over tting issue as PLSA.

3. Locally discriminative topic modeling

Recent studies [25,26] have shown that naturally occurring data, such as texts and images, cannot possibly " ll up" the ambient Euclidean space, rather it must concentrate around lower-dimensional manifold structures which plays an essential role in developing various kinds of algorithms including dimensionality reduction, supervised learning and semi-supervised learning algorithms [27,23,28,29]. To model this manifold structure, recent work on topic modeling [22,13] proposed to incorporate the so-called manifold regularizer [23] in the maximum likelihood estimation.

The manifold regularizer emphasizes the pairwise similarities of the data and de nes the cost function based on the weight matrix of a nearest neighbor graph. However, these approaches are very sensitive to the weighting scheme. In this section, we will introduce how to learn a locally discriminative regularizer automatically with local learning approaches for topic modeling.

3.1. Locally discriminative regularizer

The goal of PLSA is to estimate the parameters $\{P(w|z)\ P(z|d)\}_{w\ z\ d}$. The topic distribution $P(z|d)$ gives an explicit representation of a document in aspects. A *discriminative* interpretation of $P(z|d)$ would be the probabilities with which a speci c document d is clustered into each topical class z. Hence, the inference of the class posteriors $P(z|d)$ P

documents into a set of local neighborhoods, and appropriately models a mapping function to approximate the topic distributions within each neighborhood.

Given a document d_i, let (d_i) denote the set of its neighbors including itself, with the size $n_i = |(d_i)|$. In this paper, we assume $n_1 = n_2 = \cdots = n_N = k$ for simplicity. To construct the neighborhood for each d_i, we nd the *k-nearest neighbors* according to *cosine similarity* which is de ned as

$$\cos(\mathbf{x}_i\ \mathbf{x}_{i'}) = \frac{\mathbf{x}_i^T \mathbf{x}_{i'}}{\|\mathbf{x}_i\| \|\mathbf{x}_{i'}\|} \tag{7}$$

for two arbitrary document vectors $\mathbf{x}_i$ and $\mathbf{x}_{i'}$.

Let $_i$ denote the set containing the indices of all the documents in the neighborhood (d_i), that is $_i = \{j | d_j \in (d_i)\}$. Let $\mathbf{X}_i = [\mathbf{x}_j] \in \mathbb{R}^{M \times n_i}$ for $j \in _i$ be the local data matrix of (d_i). Let $\mathbf{Y}_i = [\mathbf{y}_j] \in \mathbb{R}^{K \times n_i}$ for $j \in _i$ be the local representation matrix of (d_i) in the latent topic space.

Following the idea of local learning [18], we try to t a local model $f_i(\mathbf{X}_i) = \mathbf{A}_i^T \mathbf{X}_i + \mathbf{b}_i \mathbf{1}_{n_i}^T$ for each (d_i) to best approximate $\mathbf{Y}_i$. Note that the subscript i for f_i means that it is trained within the neighborhood (d_i). In this model, $\mathbf{A}_i \in \mathbb{R}^{M \times K}$ is the transformation matrix speci c to (d_i), $\mathbf{1}_{n_i}$ is the n_i-dimensional vector of all ones and $\mathbf{b}_i \in \mathbb{R}^K$ is the intercept. For simplicity, we append a new element "1" to each $\mathbf{x}$. Thus, the intercept $\mathbf{b}_i$ can be absorbed into $\mathbf{A}_i$ and we have $f_i(\mathbf{X}_i) = \mathbf{A}_i^T \mathbf{X}_i$. Fitting this model can be mathematically formulated as

$$\min_{\mathbf{A}_i\ \mathbf{Y}_i} \frac{1}{n_i} \|\mathbf{Y}_i - \mathbf{A}_i^T \mathbf{X}_i\|_F^2 + \mu \|\mathbf{A}_i\|_F^2 \tag{8}$$

where $\|\cdot\|_F$ is the Frobenius norm for matrices, and the penalty term $\mu\|\mathbf{A}_i\|_F^2$ with $\mu > 0$ is introduced to avoid over tting [30]. This linear model nds a mapping from the word space $\mathbf{X}_i$ to the topic space $\mathbf{Y}_i$ *locally*.

Taking the rst-order partial derivative of Eq. (8) with respective to $\mathbf{A}_i$ and requiring it to be zero, we get the optimal solution for $\mathbf{A}_i$:

$$\mathbf{A}_i^* = (\mathbf{X}_i \mathbf{X}_i^T + n_i \mu \mathbf{I})^{-1} \mathbf{X}_i \mathbf{Y}_i^T \tag{9}$$

where $\mathbf{I}$ is an identity matrix. Substituting $\mathbf{A}_i$ in Eq. (8) with Eq. (9), we get the following minimization problem:

$$\min_{\mathbf{Y}_i} \frac{1}{n_i} \|\mathbf{Y}_i (\mathbf{I} - \mathbf{X}_i^T (\mathbf{X}_i \mathbf{X}_i^T + n_i \mu \mathbf{I})^{-1} \mathbf{X}_i)\|_F^2 + \mu \|(\mathbf{X}_i \mathbf{X}_i^T + n_i \mu \mathbf{I})^{-1} \mathbf{X}_i \mathbf{Y}_i^T\|_F^2. \tag{10}$$

Following some simple algebraic steps, Eq. (10) can be reduced to

$$\min_{\mathbf{Y}_i} \mathrm{Tr}(\mathbf{Y}_i \mathbf{\Delta}_i \mathbf{Y}_i^T) \tag{11}$$

where $\mathrm{Tr}(\cdot)$ denotes the trace operator and $\mathbf{\Delta}_i$ is given by

$$\mathbf{\Delta}_i = \frac{1}{n_i} (\mathbf{I} - \mathbf{X}_i^T (\mathbf{X}_i \mathbf{X}_i^T + n_i \mu \mathbf{I})^{-1} \mathbf{X}_i). \tag{12}$$

By applying the Woodbury Morrison formula [31], the above equation can be simpli ed as

$$\mathbf{\Delta}_i = \mu (\mathbf{X}_i^T \mathbf{X}_i + n_i \mu \mathbf{I})^{-1}. \tag{13}$$

For each (d_i), we can nd the best local model by optimizing Eq. (11). By summing the costs of all local models, we get

$$\min_{\mathbf{Y}_1 \ \cdots\ \mathbf{Y}_N} \sum_{i=1}^{N} \mathrm{Tr}(\mathbf{Y}_i \mathbf{\Delta}_i \mathbf{Y}_i^T). \tag{14}$$

It is clear $\mathbf{Y}_i$ is a sub-matrix of $\mathbf{Y}$, we can construct a selection matrix $\mathbf{S}_i$ such that $\mathbf{Y}_i = \mathbf{Y}\mathbf{S}_i$. $\mathbf{S}_i$ is constructed as follows: $\mathbf{S}_i = [\mathbf{e}_j] \in \mathbb{R}^{N \times n_i}$ for $j \in _i$, where $\mathbf{e}_j$ is the j-th unit vector whose j-th element is one and all other elements are zero. Substituting $\mathbf{Y}_i$ in Eq. (14) with $\mathbf{Y}\mathbf{S}_i$, we have

$$\min_{\mathbf{Y}} \mathrm{Tr}(\mathbf{Y} \mathbf{\Delta} \mathbf{Y}^T) \tag{15}$$

where $\mathbf{\Delta}$ is computed as

$$\mathbf{\Delta} = \sum_{i=1}^{N} (\mathbf{S}_i \mathbf{\Delta}_i \mathbf{S}_i^T). \tag{16}$$

By taking into account the local geometric structure, the optimal $\mathbf{Y}$, which are the topic distributions $\{P(z|d)\}_{z\,d}$, should minimize

$$= \mathrm{Tr}(\mathbf{Y} \mathbf{\Delta} \mathbf{Y}^T). \tag{17}$$

We call $\mathrm{Tr}(\mathbf{Y}\mathbf{\Delta}\mathbf{Y}^T)$ the *locally discriminative regularizer*. By incorporating this regularizer into traditional topic modeling approaches, we can obtain probabilistic topic distributions which are concentrated around the data manifold.

3.2. Locally discriminative topic modeling

Incorporating the locally discriminative regularizer into the generative scheme of PLSA, we obtain our locally discriminative topic modeling (LDTM) approach. Following [13], we de ne the log-likelihood of LDTM as a linear combination of Eqs. (2) and (17):

$$\max_{\Theta} \quad -\lambda \tag{18}$$

where $\lambda > 0$ is the regularization parameter and $\Theta = \{P(w|z)\ P(z|d)\}_{w\,z\,d}$ is the set of parameters to be estimated.

In Eq. (18), represents how likely the collection of documents are generated via the generative scheme. By maximizing , we seek a set of parameters $\{P(w|z)\}_{w\,z}$ and $\{P(z|d)\}_{z\,d}$ which t the data best. measures the smoothness of the topic distributions on the local manifold structure of data. By maximizing $-\lambda$, we nd $\{P(z|d)\}_{z\,d}$ that best ts the local geometrical structure of document space.

The standard procedure for maximum likelihood estimation in latent variable models is the expectation maximization (EM) algorithm [24]. The EM algorithm starts with some initial guess of the parameters, Θ_0, then generate successive estimates, Θ_t for $t = 1,2,\ldots$ until convergence by repeatedly alternating the following two steps: (i) an expectation (E) step where posterior probabilities are computed for the latent variables, based on the current estimates of the parameters, (ii) a maximization (M) step, where parameters are updated based on maximizing the so-called expected complete data log-likelihood which depends on the posterior probabilities computed in the E-step. In the following, we describe the two steps in our algorithm for parameter estimation at each t-th iteration for $t = 1,2,\ldots$.

E-step: Compute the posterior probabilities for the latent variables:

$$P(z|d\ w)_t = \frac{P(w|z)_{t-1} P(z|d)_{t-1}}{\sum_{z' \in} P(w|z')_{t-1} P(z'|d)_{t-1}}. \tag{19}$$

M-step: Maximize the expected complete data log-likelihood:

$$(\Theta) = \ (\Theta) - \lambda = \sum_{d \in} \sum_{w \in} n(d\ w) \sum_{z \in} P(z|d\ w) \log[P(w|z) P(z|d)] - \lambda \quad . \tag{20}$$

Since the regularizer does not involves $P(w|z)$, we have the closed form re-estimation equation for $P(w|z)$, which is the same as that of PLSA [32]:

$$P(w|z)_t = \frac{\sum_{d \in} n(d\ w) P(z|d\ w)_t}{\sum_{w' \in} \sum_{d \in} n(d\ w') P(z|d\ w')_t}. \tag{21}$$

For $P(z|d)$, we cannot obtain a closed form re-estimation equation since nding the global optimum of Eq. (20) is hard [22]. Therefore, instead of applying the traditional EM, we use the

generalized EM (GEM) algorithm [33]. In each M-step, GEM only nds a "better" Θ that increases (Θ). To achieve this, in t-th iteration, we rst apply Eq. (21) and the following equation:

$$P(z|d)_t = \frac{\sum_{w\in} n(d\ w)P(z|d\ w)_t}{n(d)} \tag{22}$$

which is also the same as in PLSA, in order to obtain Θ_t^0 which maximizes (Θ). Obviously, it is not guaranteed that $(\Theta_t^0) \geq (\Theta_{t-1})$. Then we apply the Newton Raphson method [34] to decrease iteratively from (Θ_t^0) to successive (Θ_t^m), for $m=1, 2,\ldots$, in hope of nding Θ_t^m which satis es $(\Theta_t^m) \geq (\Theta_{t-1})$. Given a function $f(x)$, we adopt the updating formula of Newton Raphson in the general form:

$$x^m = x^{m-1} - \gamma \frac{f'(x)}{f''(x)} \tag{23}$$

where x^m is the new estimate of parameters based on the previous estimate x^{m-1}. It is easy to verify ≥ 0, thus will decrease at each updating step of the Newton Raphson method. By taking the rst and second derivatives of with respect to $P(z|d_i)$, we obtain the speci c Newton Raphson updating equation as follows:

$$P(z|d_i)_t^m = P(z|d_i)_t^{m-1} - \gamma \frac{\sum_{j=1}^{N} \delta_{ij} P(z|d_j)_t^{m-1}}{\delta_{ii}} \tag{24}$$

where δ_{ij} is the (i,j)-th element of $\mathbf{\Delta}$, and $0 < \gamma < 1$ is the step size. It is clear that $\sum_{z\in} P(z|d)_t^m = 1$ always holds, and a relative small value of γ ensures $P(z|d) \geq 0$. Once we obtain a Θ_t^m which satis es $(\Theta_t^m) \geq (\Theta_{t-1})$, we stop iterating Eq. (24). This completes the M-step.

The E-step and M-step are alternated repeatedly until convergence is reached. We summarized the procedure of our LDTM in Algorithm 1.

Algorithm 1. Generalized EM for LDTM

Input:
- $\{P(w|d)\}$ for all w and d
- the parameters K, k, λ, γ, and convergence condition threshold ε

Output:
- $\Theta = \{P(w|z)\ P(z|d)\}$ for all w z and d

Procedure:

compute the matrix $\mathbf{\Delta}$ by Eq. (16)
initialize Θ_0: $P(w|z)_0 = 1/M$, $P(z|d)_0 = 1/K$ for all w, z and d
$t \leftarrow 0$
repeat
  $t \leftarrow t+1$
  E-step: compute $P(z|w\ d)$ by Eq. (19) for all w, z and d
  M-step:
  compute $P(w|z)_t$ by Eq. (21) for all w and z
  compute $P(z|d)_t$ by Eq. (22) for all z and d
  $P(z|d)_t^0 \leftarrow P(z|d)_t$ for all z and d
  $m \leftarrow 0$
  repeat
    $m \leftarrow m+1$
    compute $P(z|d)_t^m$ by Eq. (24) for all z and d
  until $(\Theta_t^m) > (\Theta_t)$
until $[(\Theta_t) - (\Theta_{t-1})] \leq \varepsilon$
return Θ_t

3.3. Computational complexity

In this subsection, we provide a computational cost analysis of LDTM in comparison to PLSA. We present operation counts measured by *flam* [35], which is a compound operation consisting of one addition and one multiplication. The document vector $\mathbf{x}$ is usually sparse, and we use S to denote the sparseness of $\mathbf{x}$, i.e., the average number of non-zero features per document. The major computational cost in LDTM include two parts:

1. Computation of the matrix $\mathbf{\Delta}$ given by Eq. (16). This part rst requires k-nearest neighbors construction for all the N documents, which costs about $O(SN^2+kN^2)$ am. $O(SN^2)$ is used to calculate the pairwise cosine similarity given by Eq. (7) and $O(kN^2)$ is used to sort the pairwise similarity for nding k-nearest neighbors for all documents. Secondly, given $n_i=k$, around $O(k^2S+k^3)$ am is required to compute each $\mathbf{\Delta}_i$ given by Eq. (12) which mainly involves one matrix matrix product using $O(k^2S)$ am and one matrix inversion using $O(k^3)$ am. Since every $k \times k$ matrix $\mathbf{\Delta}_i$ contributes each element only once to form the matrix $\mathbf{\Delta}$ given by Eq. (16), there is at most k^2N non-zero elements in $\mathbf{\Delta}$. Hence, to obtain the nal $\mathbf{\Delta}$ as in Eq. (16), the matrix matrix products need trivial $O(k^2)$ am since $\mathbf{S}_i$ has only k non-zero elements and the summation need $O(k^2N)$ am. Therefore, this part in total requires around $O(SN^2+kN^2+k^2SN+k^3N)$ am. We also need k^2N memory to store the sparse matrix $\mathbf{\Delta}$.
2. Parameter estimation using generalized EM algorithm. In each iteration, KSN posterior probabilities $P(z|d\ w)$ have to be computed in the E-step as in Eq. (19) since there are SN distinct observation pairs (d,w), each of which has K posterior probabilities. We can easily verify that the E-step requires $O(KSN)$ am for all $P(z|d\ w)$. In M-step, each $P(z|d\ w)$ contributes to exactly one re-estimation both in Eqs. (21) and (22). Therefore, these two equations cost $O(KSN)$ am. The work load of each Newton Raphson updating is around $O(k^2KN)$ am since each row of $\mathbf{\Delta}$ has approximately k^2 non-zero elements. If each M-step repeats an average of m iterations at Newton Raphson updating, then the cost is $O(mk^2KN)$ am. Assuming that LDTM converges after t iterations of the EM, this part costs $O(tKSN+tmk^2KN)$ in total. We also have to use $SN+SKN+MK+KN$ memory to store $P(w|d)$, $P(z|d\ w)$, $P(w|z)$ and $P(z|d)$.

In conclusion, LDTM costs $O[(S+k)N^2+(k^2S+k^3+tKS+tmk^2K)N]$ in total to nd the optimum of the parameters. Since k is usually set as a small value such as 5 or 10 (see our experiments in Section 4), it is clear $k \ll S$ and $k^3 \ll tmk^2K$ in usual cases. We thus can rewrite the computational cost of LDTM as $O[SN^2+(k^2S+tKS+tmk^2K)N]$. We also require about $(k^2+S+SK+K)N+MK$ memory to store all non-zero elements of the matrix $\mathbf{\Delta}$ and all parameters. For PLSA, it requires $O(tKSN)$ am if the EM algorithm converges after t iterations and need $(S+SK+K)N+MK$ memory to store all the parameters. Table 1 summarizes our complexity analysis of LDTM, together with PLSA.

4. Experiments

In this section, we rst present the task of document modeling to evaluate how well our LDTM algorithm gives topical representations of documents. We then investigate discriminative tasks, i.e., text clustering and web page classi cation to evaluate how much discriminative power LDTM can provide, in order to compare with PLSA [12], LDA [11] and LTM [13] in an objective and quantitative way.

Throughout the experiments, we x $\mu = 1$, $\gamma = 0.1$, and empirically set the number of nearest neighbors $k=5$ and the value of the regularization parameter $\lambda = 1000$.

Table 1
Computational cost of LDTM and PLSA.

Algorithm	Operations (am)	Memory
LDTM	$O[SN^2+(k^2S+tKS+tmk^2K)N]$	$(k^2+S+SK+K)N+MK$
PLSA	$O(tKSN)$	$(S+SK+K)N+MK$

N: the number of documents.
M: the number of distinct words.
S: the average number of non-zero features per document.
K: the number of latent topics.
k: the number of nearest neighbors.
m: the average number of iterations in Newton Raphson updating.
t: the number of iterations in EM.

4.1. Document modeling

Let us brie y discuss an illustrative example of hidden topic modeling using our LDTM approach. We use a subset of the TREC AP corpus[1] consisting of 2246 news articles with 10,473 distinct words. For this dataset, a 100-topic LDTM model is trained using GEM algorithm described in Algorithm 1.

We select an example article which is about *tax payment of farmers* and illustrate the top words from the most probable topics generating the document in Fig. 1 (top). Each color codes a latent topic (*Tax*, *Time*, *Farm* or *Office*) that we named using a representative word. As we have hoped, each listed topic-speci c word distribution can capture the semantics of the corresponding topic to some extent. In the article text in Fig. 1 (bottom), each word is coded as the same color as a topic if it is both among the top 100 words of the topic and have the largest $P(w|z)$ over the four listed topics. With such illustration, one can easily identify how the different topics are mixing in this article.

4.2. Text clustering

Clustering is one of the key tasks of text organization in the unsupervised setting. The topic modeling methods reduce the word feature of documents into lower-dimensional topic distributions. Each hidden topic can be regarded as a cluster. The estimated topic distribution $P(z|d)$ can be used to infer which cluster a document d belongs to. We conduct this experiment on two datasets, the 20 Newsgroups corpus[2] and Yahoo! News K-series.[3] The 20 Newsgroups contains 18,846 documents with 26,214 distinct words. The data are organized into 20 different newsgroups (clusters), each of which corresponds to a distinct topic. These clusters have varying sizes from 628 to 999. Yahoo! News K-series has 2340 documents from 20 different categories, with 8104 distinct words in total. The sizes of these categories in this dataset are highly skewed, ranging from 9 to 494. The skewness imposes challenges to the clustering task. The statistics of the two datasets we investigate is summarized in Table 2.

We evaluate the clustering performance of our locally discriminative topic model (LDTM) by comparing against all the following methods:

- K-means clustering algorithm based on word feature (Word);
- probabilistic latent semantic analysis (PLSA) [12];
- latent Dirichlet allocation (LDA) [11];
- locally consistent topic model (LTM) [13];
- spectral clustering based on normalized cut (NCut) [36,37];
- nonnegative matrix factorization based clustering (NMF) [38].

"Tax" "Time" "Farm" "Office"
tax la

The standard clustering metric *accuracy* (*AC*) is used to measure the clustering performance [38]. Given a data point $\mathbf{x}_i$, let r_i and s_i be the cluster label and the label provided by the data set, respectively. The AC is de ned as follows:

$$AC=\frac{\sum_{i=1}^{n}\delta(s_i\ map(r_i))}{n} \tag{25}$$

where n is the total number of samples and $\delta(x\ y)$ is the delta function that equals one if $x=y$ and equals zero otherwise, and $map(r_i)$ is the permutation mapping function that maps each cluster label r_i to the equivalent label from the data set. The best mapping can be found by using the Kuhn Munkres algorithm [39].

Tables 3 and 4 show the clustering accuracy on 20 Newsgroups and Yahoo! News K-series, respectively. The evaluations were conducted with the cluster numbers ranging from two to ten. For each given cluster number p (from 2 to 10), 20 test runs were conducted on different randomly chosen clusters and the average performance as well as the standard deviation are reported. As we can see, the PLSA, LDA models fail to achieve good performance since they do not consider the geometric structure of the data space. Among the four topic modeling algorithms (i.e., PLSA, LDA, LTM and LDTM), our LDTM consistently outperforms its competitors. This indicates LDTM is more capable of giving semantic representations of documents and provide more discriminating power. LDTM also demonstrates its advantage over other clustering methods NCut and NMF.

Table 3
Clustering accuracy (mean ± std-dev%) on 20 Newsgroups.

p	Word	PLSA	LDA	LTM	LDTM	NCut	NMF
2	80.5 ± 14.2	79.1 ± 12.7	94.0 ± 5.1	95.9 ± 2.5	**96.7** ± 2.8	95.3 ± 2.0	94.7 ± 2.7
3	63.6 ± 12.6	73.5 ± 14.0	87.9 ± 9.6	90.4 ± 8.9	**91.4** ± 9.2	87.9 ± 12.9	85.8 ± 11.0
4	53.9 ± 7.3	69.2 ± 11.7	85.5 ± 7.8	88.5 ± 6.4	**90.6** ± 5.7	87.6 ± 6.6	83.1 ± 7.4
5	51.3 ± 5.6	69.3 ± 9.1	81.9 ± 8.2	85.6 ± 8.7	**88.4** ± 7.4	86.0 ± 6.8	84.3 ± 8.4
6	45.9 ± 5.8	67.8 ± 8.9	80.2 ± 8.3	81.4 ± 7.3	**86.7** ± 6.9	82.4 ± 8.1	80.6 ± 7.9
7	43.6 ± 4.0	68.6 ± 5.5	77.4 ± 7.7	79.5 ± 8.4	**83.0** ± 7.3	80.7 ± 8.2	78.8 ± 7.7
8	42.3 ± 2.6	64.5 ± 6.2	73.0 ± 8.3	77.3 ± 9.4	**80.6** ± 7.3	75.0 ± 6.3	76.3 ± 6.4
9	41.9 ± 4.1	66.9 ± 8.4	73.1 ± 5.0	73.3 ± 4.4	**79.4** ± 5.1	71.9 ± 6.3	73.0 ± 6.7
10	37.5 ± 3.9	65.1 ± 7.1	67.9 ± 8.0	70.3 ± 8.3	**74.6** ± 7.8	68.3 ± 8.2	68.2 ± 7.2
Avg.	51.1	69.3	80.1	82.4	**85.7**	81.7	80.5

Table 4
Clustering accuracy (mean ± std-dev%) on Yahoo! News K-series.

p	Word	PLSA	LDA	LTM	LDTM	NCut	NMF
2	68.7 ± 14.4	58.0 ± 4.8	68.8 ± 14.3	81.5 ± 15.7	**84.1** ± 16.0	78.5 ± 11.8	71.0 ± 16.0
3	57.7 ± 15.0	50.5 ± 10.2	59.3 ± 13.0	69.0 ± 14.5	**73.8** ± 14.1	67.1 ± 11.9	67.2 ± 14.1
4	55.2 ± 11.6	42.6 ± 7.7	64.5 ± 13.6	67.4 ± 14.3	**73.9** ± 12.6	56.5 ± 12.6	69.6 ± 16.1
5	55.9 ± 10.0	36.1 ± 5.5	63.0 ± 12.4	59.4 ± 11.8	**66.0** ± 15.6	55.8 ± 8.5	63.8 ± 12.8
6	50.5 ± 6.9	33.8 ± 6.0	51.7 ± 9.5	62.6 ± 13.2	**65.7** ± 12.5	54.6 ± 9.0	54.8 ± 10.5
7	49.0 ± 8.2	30.3 ± 4.2	52.3 ± 7.6	59.2 ± 14.5	**61.4** ± 14.0	49.5 ± 7.6	53.2 ± 9.1
8	48.2 ± 5.9	28.3 ± 3.1	51.9 ± 8.7	57.9 ± 12.0	**62.8** ± 11.2	49.0 ± 6.8	54.9 ± 6.4
9	43.6 ± 4.7	27.5 ± 3.2	47.8 ± 4.9	56.9 ± 10.9	**59.9** ± 11.0	49.0 ± 4.6	49.6 ± 5.4
10	43.6 ± 4.8	25.8 ± 3.3	49.3 ± 6.6	53.2 ± 9.0	**56.5** ± 10.4	48.0 ± 6.3	50.2 ± 6.0
Avg.	52.5	37.0	56.5	63.0	**67.1**	56.4	59.4

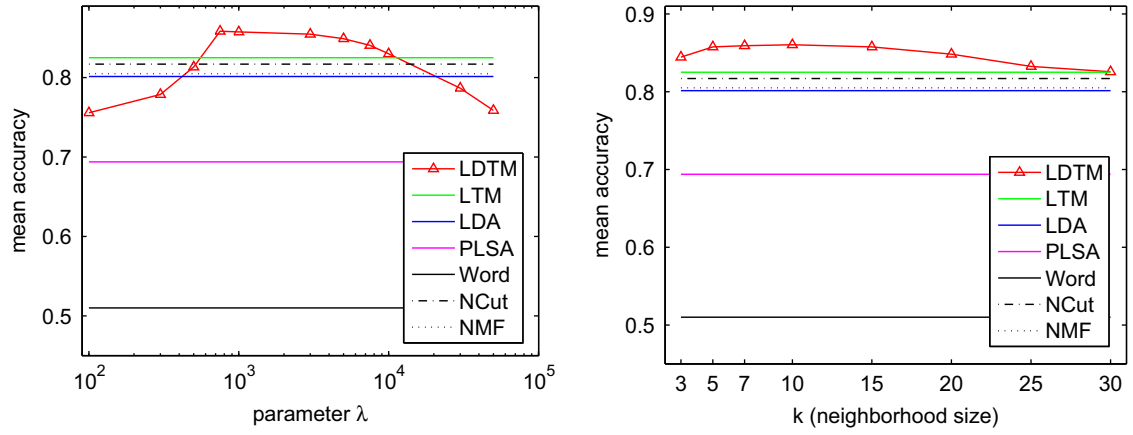

Fig. 2. The performance of LDTM vs. λ and k on 20 Newsgroups.

We also study the in uence of different choices for the regularization parameter λ and the neighborhood size k in LDTM. Fig. 2 shows the curves for 20 Newsgroups, where accuracy values are averaged over all given numbers of sampled clusters. It is illustrated that LDTM achieves good performance as λ varies from 500 to 10,000. The performance is stable with respect to k when k is relative small (such as between 5 and 15), and the performance drops as k continues to increase. This con rms the assumption that the local learning method with a small neighborhood size rather than the global learning (where $k=+\infty$), is capable to capture the geometric structure of the data in each distinct cluster.

4.3. Web categorization on WebKB

A challenging aspect of the document classi cation problem is the choice of features. Treating individual words as features yields a rich but very large feature set. One way to reduce this feature set is to use topic modeling approaches for dimensionality reduction. For example, PLSA reduces the document to a xed set of real-valued features $P(z|d)$. It is of interest to see how much discriminating information we may lose in reducing the document description to these parameters [11].

In this experiment, we investigate the web page categorization task on the WebKB dataset.[4] We address a subset consisting of pages from the four universities: *Cornell*, *Texas*, *Washington* and *Wisconsin*. After removing empty web pages, *html* tags and words that occur fewer than ve documents, we obtain 4128 documents with 9933 instinct words in total. These pages were manually classi ed into the following categories: *student*, *faculty*, *staff*,

[4] http://www-2.cs.cmu.edu/~webkb/

department, *course*, *project* and *other*. The statistics are listed in Table 5.

We address two tasks: (i) predicting which university pages belongs to; (ii) predicting the category label of the pages. For task (ii), we only address a subset consisting of the four populous entity-representing categories: *student*, *faculty*, *course*, *project*, which has more balanced categories of data samples and contains 1039 documents in total. Note here we are only interested in how much we can obtain from textual content of documents and no link structure is considered. In both tasks, we first apply the topic models for dimensionality reduction on the original word features. Then support vector machine (SVM) trained on the resulting low-dimensional features is used for page classification. SVM trained on the original word features is served as the baseline.

Table 5
The statistics of the subset of WebKB.

School	*Student*	*Faculty*	*Staff*	*Dept.*	*Course*	*Project*	*Other*	Total
Cornell	128	34	21	1	44	20	612	860
Texas	147	46	3	1	38	20	566	821
Washington	126	31	10	1	77	21	931	1197
Wisconsin	155	42	12	1	85	25	930	1250
Total	556	153	46	4	244	86	3039	4128

Table 6
Classification error rate (mean ± std-dev%) on WebKB for task (i).

# Train	Word	PLSA	LDA	LTM	LDTM
5	66.9 ± 5.6	59.1 ± 3.9	57.8 ± 5.1	54.0 ± 6.3	**53.9** ± 7.8
10	57.0 ± 5.2	52.0 ± 2.5	48.4 ± 3.0	42.1 ± 3.2	**39.5** ± 4.5
15	53.4 ± 4.3	49.5 ± 2.5	46.1 ± 2.2	40.5 ± 2.9	**37.0** ± 3.5
20	49.8 ± 4.3	47.1 ± 2.1	43.4 ± 2.2	37.0 ± 2.0	**33.8** ± 2.0
25	47.2 ± 4.2	45.4 ± 2.1	41.9 ± 2.1	35.3 ± 2.3	**31.6** ± 2.2
30	45.5 ± 4.7	44.0 ± 1.8	40.4 ± 2.0	33.7 ± 2.3	**30.1** ± 2.6
35	43.3 ± 2.5	42.7 ± 2.0	39.1 ± 1.9	32.4 ± 1.7	**28.9** ± 2.0
40	43.2 ± 3.3	41.7 ± 2.1	38.5 ± 1.9	31.5 ± 1.8	**28.4** ± 1.7
45	41.6 ± 3.0	40.6 ± 1.5	37.2 ± 1.6	30.7 ± 1.3	**27.5** ± 1.4
50	41.0 ± 3.1	40.0 ± 1.6	36.3 ± 1.1	29.1 ± 1.5	**26.1** ± 1.3
Avg.	48.0	46.2	42.9	36.6	**33.7**

Table 7
Classification error rate (mean ± std-dev%) on WebKB for task (ii).

# Train	Word	PLSA	LDA	LTM	LDTM
4	52.1 ± 11.6	41.2 ± 9.5	36.9 ± 8.5	36.1 ± 8.3	**32.9** ± 7.2
8	40.4 ± 8.2	33.8 ± 4.9	30.8 ± 3.7	28.7 ± 4.5	**28.5** ± 4.3
12	36.3 ± 5.6	31.8 ± 2.9	28.5 ± 3.2	27.6 ± 2.5	**25.7** ± 2.7
16	33.6 ± 5.5	31.5 ± 3.3	28.3 ± 3.2	27.3 ± 2.5	**24.8** ± 2.1
20	31.1 ± 4.2	30.4 ± 3.3	27.6 ± 2.5	26.4 ± 2.2	**24.4** ± 2.6
24	29.9 ± 4.1	28.8 ± 2.7	26.9 ± 1.7	26.3 ± 2.4	**23.8** ± 2.0
28	29.2 ± 3.5	28.9 ± 1.8	26.5 ± 2.4	25.6 ± 2.3	**23.5** ± 2.0
32	27.6 ± 3.3	27.7 ± 1.9	25.7 ± 1.9	24.7 ± 1.8	**22.5** ± 1.7
36	25.8 ± 0.2	27.4 ± 1.8	25.9 ± 2.0	24.3 ± 1.9	**22.6** ± 1.6
40	24.5 ± 2.7	26.8 ± 1.9	25.5 ± 1.9	23.7 ± 1.7	**22.1** ± 1.5
Avg.	33.1	30.8	28.3	27.1	**25.1**

Tables 6 and 7 show the classification results for the two tasks, respectively. Given a value of training size per university or category, we randomly select the training data and this process is repeated 50 times. The average performance with standard deviations is recorded. As we can see, all the topic modeling methods, i.e., PLSA, LDA, LTM, and LDTM, achieve better performance than using the word features when the numbers of training samples are small. Among the four compared topic modeling approaches, LDTM is obviously the best. Especially, LDTM yields substantial and consistent improvements of performance over pure generative topic modeling approaches (i.e., PLSA and LDA), which shows the effectiveness of our discriminative learning scheme.

A key problem for all the topic modeling approaches is how to estimate the number of hidden topics. Fig. 3 shows how the performance of the four topic models varies with different numbers of topics. In task (i), the performance of all the topic models increases as the number of topics increases. In task (ii), LDTM, LTM and LDA are less sensitive to the topic number in comparison to PLSA. The performance of PLSA degrades with larger numbers of topics, which may suggest the over fitting issue of PLSA [11].

5. Conclusions and future work

We have introduced a novel probabilistic topic modeling approach for semantic analysis of documents, called locally discriminative topic model (LDTM), which takes into account both generative and discriminative structures. Specifically, LDTM uses local learning to explore the intrinsic geometric structure in the data. As a result, LDTM can provide more discriminating power than traditional topic modeling approaches, e.g., PLSA and LDA. Comparing to LTM [13], LDTM automatically learns a locally discriminative regularizer which avoids hand-crafting weight setting. Experimental results on TREC AP, WebKB, Yahoo! K-series and 20 Newsgroups data sets have demonstrated that our algorithm can better capture the hidden topics of the documents and therefore enhance the learning performance.

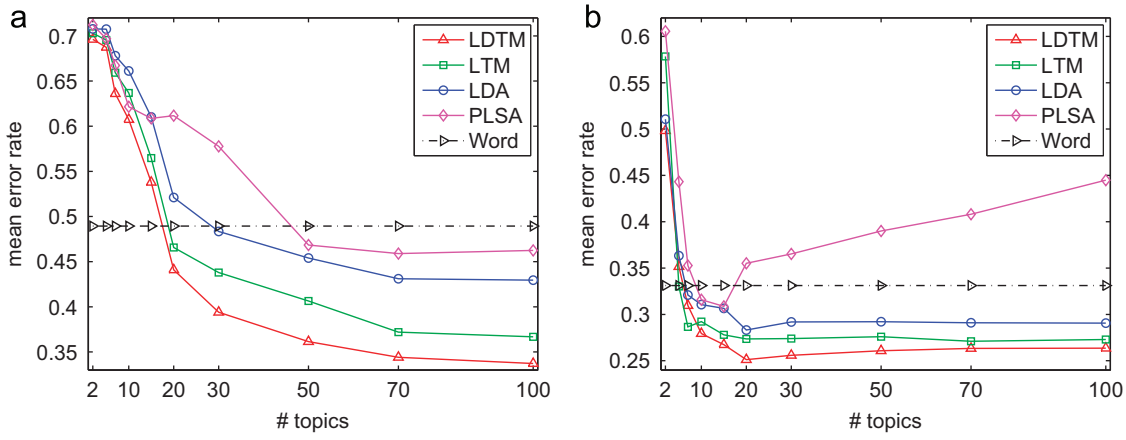

Fig. 3. The classification performance of the four topic models vs. number of topics: (a) task (i) and (b) task (ii) on WebKB.

For our approach, constructing the document neighborhood is not limited to exploring the intrinsic word features of documents. For example, we can use alternatives such as authorship, citation and hyperlink information of documents, which is common in real-world data and attracts much renewed interests in recent years [40,5,8,9]. We will investigate this in the future work.

Lijun Zhang received the B.E. degree in Software Engineering from Zhejiang University, China, in 2007. He is currently a Ph.D. candidate in Computer Science at Zhejiang University. His research interests include machine learning, information retrieval, and data mining.

Haifeng Liu received her Ph.D. degree in Department of Computer Science, University of Toronto under the supervision of Professor H-Arno Jacobsen in 2009. She got her B.S. degree in Computer Science and Technology from the Special Class for the Gifted Young, University of Science and Technology of China. Now She works as an assistant professor in the College of Computer Science, Zhejiang University.

Can Wang received the B.S. degree in Economics and M.S. degree in Computer Science from Zhejiang University, China, in 1995 and 2003, respectively. He then received his Ph.D. degree in Computer Science from Zhejiang University, China, in 2009. He is currently a faculty member in the College of Computer Science at Zhejiang University. His research interests include information retrieval, data mining and machine learning.

Deng Cai is currently an associate professor in the State Key Lab of CAD&CG, College of Computer Science at Zhejiang University, China. He received the Ph.D. degree in Computer Science from University of Illinois at Urbana Champaign in 2009. Before that, he received his Bachelor's degree and a Master's degree from Tsinghua University, China in 2000 and 2003, respectively, both in automation. His research interests include machine learning, data mining and information retrieval.