International Journal of Computational Intelligence Systems, Vol.2, No. 4 (December, 2009), 398-409

Text Categorization Based on Topic Model

Shibin Zhou [1,2] ,Kan Li [3] ,Yushu Liu [3]

[1] *School of Computer Science and Technology, China University of Mining and Technology Xuzhou, Jiangsu Province, 221116, P.R. China*

[2] *School of Computer Science and Technology, Beijing Institute of Technology Haidian District, Beijing, 100081, P.R. China*

E-mail: guoguos.zhou@gmail.com

[3] *School of Computer Science and Technology, Beijing Institute of Technology Haidian District, Beijing, 100081, P.R. China*

E-mail: {likan,liuyushu}@bit.edu.cn

Received: 29/12/08
Accepted: 19/08/09

Abstract

In the text literature, many topic models were proposed to represent documents and words as topics or latent topics in order to process text effectively and accurately. In this paper, we propose LDACLM or Latent Dirichlet Allocation Category Language Model for text categorization and estimate parameters of models by variational inference. As a variant of Latent Dirichlet Allocation Model, LDACLM regards documents of category as Language Model and uses variational parameters to estimate maximum a posteriori of terms. In general, experiments show LDACLM model is effective and outperform Naïve Bayes with Laplace smoothing and Rocchio algorithm but little inferior to SVM for text categorization.

Keywords: Topic model, Latent Dirichlet allocation, Variational Inference, Category Language Model.

1. Introduction

In the text analysis, standard algorithms are unsatisfactory because terms often were supposed independent, which was recognized as "bag of words" model. However, the "bag of words" model offers a rather impoverished representation of the data because it ignores any relationships between the terms.

In the recent past, a new class of generative models called Topic Model have quickly become more popular in some text-related tasks. Topic Model suppose documents and corpus composed of mixture topics and then documents can be thought of "bag of topics". Thus, these models can handle the problem effectively about terms dependency. Topics can be viewed as a probability distribution which implies semantic coherence about words. For example, a topic related to fruit would have high probabilities for the words "orange", "apple", and even "juicy". Wallach [13] demonstrated the "bag of topics" to surpass in performance to "bag of words" in unigram and bigram schemas.

There are many Topic Models proposed by researchers in the past such as Latent Semantic Analysis or LSA [4], the probabilistic Latent Semantic Indexing or pLSI [7], Latent Dirichlet allocation or LDA [1] and so on.

Latent Semantic Analysis (LSA) [4] is an approach that combines both term and document clustering.

LSA usually takes a term-document matrix in the vector space representation as input, and uses a singular value decomposition of the input matrix to identify a linear subspace in the vector space that captures most of the variance in the collection. Thus LSA can map text elements to a representation in the latent semantic space and can capture some aspects of basic linguistic notions such as synonymy and polysemy.

The probabilistic Latent Semantic Indexing (pLSI) model introduced by Hofmann [7], also known as the aspect model, was designed as a discrete counterpart of LSI or LSA to provide a better fit to text data and overcome deficiencies of Latent Semantic Indexing (LSI). pLSI is a latent variable model that models each document as a mixture of topics. Although there are some problems with the generative semantics of pLSI, Hoffmann has shown some encouraging results in Information Retrieval.

As one of these topic models, Latent Dirichlet Allocation (LDA) has quickly become the most popular probabilistic text modeling techniques. LDA has been shown to be effective in the text-related tasks. Processing fully generative semantics, LDA overcomes the drawbacks of previous topic models such as probabilistic Latent Semantic Indexing (pLSI) which is a MAP/ML estimated LDA model under a uniform Dirichlet distribution according to Girolami and Kaban discovery [5]. Latent Dirichlet allocation represents documents as mixtures over latent topics differentiated, but pLSI characterize each topic by a distribution over words. Wei and Croft[14] shown the LDA-based document model had good performance in Information Retrieval. Moveover, Griffiths and Steyvers [6] apply LDA model to find scientific document topics.

Our goal in this paper is to address a variant of LDA and an extension of Language Model [12], which is a novel model for text categorization as we known. This generative model represents words set of each category with a mixture of topics assumed independent as many state-of-the-art approaches did, and extends these approaches to estimate maximum a posteriori of category language model parameters by assuming that variance parameters would be multinomial and dirichlet parameters of category language model.

In Section 2, we briefly review some topic models proposed in the past. We demonstrate our approaches on how to estimate parameters of models and classify documents in section 3. In section 4, we evaluate correctness and efficiency of our model. We conclude the paper with a summary, and a brief discussion of future work in section 5.

2. Related Works

2.1. Probabilistic Latent Semantic Indexing

pLSI [7] was designed as a discrete counterpart of LSI to provide a better fit to text data. This model expresses each document as a convex combination of topics, and model co-occurrence data which associates an unobserved class variable $z \in \{z_1, \ldots, z_K\}$ with each observation. pLSI define a generative model for word by the following scheme:

- Pick a latent topic z_k with probability $p(z_k|d)$, where $p(z_k|d)$ denotes a document-specific probability of a latent variable z_k conditioned on the document d.
- Generate a word w_t with probability $p(w_t|z_k)$, where $p(w_t|z_k)$ denotes the class-conditional probability of a specific word w_t conditioned on the unobserved class variable z_k.

As a result, the probability of a word w_t generating by document d is $p(w_t|d) = \sum_{k=1}^{K} p(w_t|z_k)\, p(z_k|d)$, This amounts to a matrix decomposition with the constraint that both the vectors and mixture coefficients are normalized to make them probability distributions. Fitting the model involves determining the topic vectors which are common to all documents, determining the mixture coefficients which are specific for each document, and determining the model that gives high probability to the words that appear in the corpus $\mathscr{D}$.

Hofmann applied pLSI to retrieval tasks in the Vector Space Model framework on small collections. He exploited pLSI both as a unigram model to smoothen the empirical word distributions and as a latent space model to provide a low dimensional document representation. It significantly

overwhelms the standard term schema on retrieval performance.

2.2. Latent Dirichlet Allocation

In contrast to pLSA which is extended by sampling those weights from a Dirichlet distribution, LDA [1] treats the multinomial weight over topics as latent random variable. This extension allows the model to assign probabilities to data outside the training corpus and uses fewer parameters, thus reducing overfitting.

LDA represents each document as mixture of topics, where each topic is a multinomial distribution over words in a vocabulary. To generate a document, LDA first samples a per-document multinomial distribution over topics from a Dirichlet distribution. Then it repeatedly samples a topic from this multinomial and samples a word from the topic. The topic discovered by LDA capture correlations among words. LDA defines a generative model for word by the following scheme:

- Pick a latent topic z with probability $p(z|\theta)$, where $p(z|\theta)$ denotes probability of the topic z from a multinomial distribution with parameter vector θ.
- Generate a word with probability $p(w_t|z,\beta)$, where $p(w_t|z,\beta)$ denotes the topic-conditional probability of a specific word w_t conditioned on the unobserved topic variable z with a multinomial distribution parameter β.
- Pick a multinomial distribution β for each topic z from a Dirichlet distribution $p(\beta|\eta)$ with parameter η.
- Pick multinomial distribution θ_d for document d from a Dirichlet distribution $P(\theta_d|\alpha)$ with parameter α.

Thus, the likelihood of generating a corpus $\mathscr{D}$, whose vocabulary size is V, is

$$p(\mathscr{D}) = \prod_{d\in\mathscr{D}} \left\{ \int p(\theta_d|\alpha)p(\beta|\eta) \prod_{t=1}^{V}\sum_{k=1}^{K} p(z_t=k|\theta_d)p(w_t|z_t=k,\beta)\mathrm{d}\theta_d\mathrm{d}\eta \right\}$$

3. Latent Dirichlet Allocation Category Language Model

In this section we introduce our model that extends latent dirichlet allocation and Language Model called Latent Dirichlet Allocation Category Language Model. With the model defined, we turn to approximate posterior inference, parameter estimation. We develop a variational inference procedure for approximating the posterior. Moreover, we use this procedure in a variational expectation-maximization (EM) algorithm for parameter estimation. Finally, we show how a model whose parameters have been estimated can be used as a text categorization model.

3.1. Notation

We will describe LDACLM here using the notations similarly in the LDA. Suppose we have M categories or words sets, $\mathbf{w}_1,\mathbf{w}_2,\ldots,\mathbf{w}_M$, which can be viewed as category language models, containing words form corpus $\mathscr{D}$ who has a vocabulary of size V. In other words, one words set is "bag of words" of one category. The corpus of text documents is summarized in a M by V co-occurrence table, where $\mathrm{tf}_{t,\mathbf{w}}$ stores the number of occurrences of a word w_t in words set $\mathbf{w}$.

We would like to use $p(z|\theta_{\mathbf{w}})$ to denote probability of the topic $z \in \{1,2,\ldots,K\}$ from a multinomial distribution with parameter vector $\theta_{\mathbf{w}}$ specified to words set $\mathbf{w}$, $p(w_t|z,\beta)$ to denote the topic-conditional probability of a specific word w_t conditioned on the unobserved topic variable z with a multinomial distribution parameters β, $P(\theta_{\mathbf{w}}|\alpha)$ to denote the probability of vector $\theta_{\mathbf{w}}$ with Dirichlet distribution scalar parameter α.

3.2. Model Structure

As we know, LDA descripted in [1] used as dimension reducer in the discriminative framework of documents classification. But, as a variant of LDA, Latent Dirichelt Allocation Category Language Model or LDACLM regards the document in the same category generated by LDA distribution respectively. LDA distributions in LDACLM have the same Dirichlet prior parameters.

The prominent feature of LDACLM is that the model assume each word would be a independent topic that we called word topic and assume extra topics other than word topics would be model the correlation among the words. As we know, this distinguish to LDA and also tradeoff between effective and time consuming. The following process similar to LDA generates documents in the LDACLM model, represented as Fig.1.

- For each category language model or words set $\mathbf{w}$, pick multinomial distribution $\theta_{\mathbf{w}}$ from a symmetric Dirichlet distribution $p(\theta_{\mathbf{w}}|\alpha)$ with prior scalar parameter α which is identity to all category language models.
- Pick a topic z from a multinomial distribution $p(z|\theta_{\mathbf{w}})$ with parameter vector $\theta_{\mathbf{w}}$.
- Pick a word w_t from a multinomial distribution $p(w_t|z,\beta)$ with parameter vector β. Each parameter β_z in the vector β relates to specific z respectively.

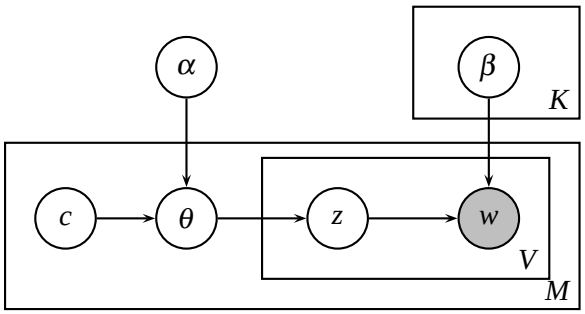

Fig. 1. Graphical model representation of LDACLM

3.3. *Inference*

The maximum likelihood of category language model $\mathbf{w}$ with model parameter vector β and model dirichlet parameter α may formulate as:

$$p(\mathbf{w}|\alpha,\beta) \propto \int \left(\prod_{k=1}^{K}\theta_k^{\alpha-1}\right)\left(\prod_{t=1}^{V}\left\{\sum_{k=1}^{K}(\theta_k\beta_{k,t})\right\}^{\mathrm{tf}_{t,\mathbf{w}}}\right)\mathrm{d}\theta$$

where word set $\mathbf{w}$ contains words from corpus $\mathscr{D}$ who has a vocabulary of size V and $\mathrm{tf}_{t,\mathbf{w}}$ stores the number of occurrences of a word w_t in word set $\mathbf{w}$.

Similar to LDA [1], We develop a variational approximation [9] for LDACLM by defining an approximating family distribution $q(\theta,z|\mathbf{w},\gamma,\phi)$, and choose the variational Dirichlet parameter vector γ and variational multinomial parameter vector ϕ which are different sets for each category language model to yield a tight approximation to the true posterior.The variational distribution of LDACLM is represented as Fig.2.

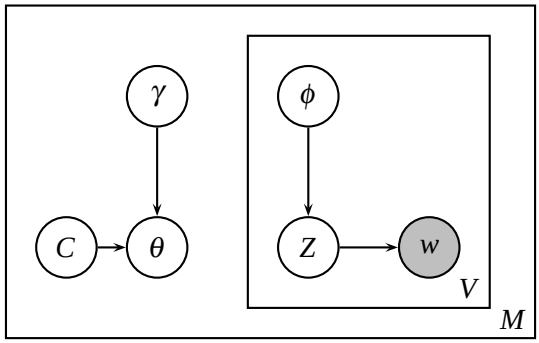

Fig. 2. Graphical model representation of the variational distribution to approximation the posterior in LDACLM

Suppose the factorized variational parameters distribution is

$$q(\theta,z|\mathbf{w},\gamma,\phi) = q(\theta|\mathbf{w},\gamma)\prod_{t=1}^{V}q(z_t|\mathbf{w},\phi_t)$$

with variational Dirichlet parameter vector γ and variational multinomial parameter vector ϕ. Especially, for each category language model, there is a different set of multinomial and Dirichlet variational parameter vectors. Thus, minimization of the KL divergence $D(q(\theta,z|\mathbf{w},\gamma,\phi)\,\|\,p(\theta,z|\mathbf{w},\alpha,\beta))$ we can derive approximation of $p(\theta,z|\mathbf{w},\alpha,\beta)$.

So, we can take decreasing steps in the KL divergence and converge to optimizing parameter by an iterative fixed-point method, bounding the marginal likelihood of a document using Jensen's inequality [9].

$$\begin{aligned}\log p(\mathbf{w}|\alpha,\beta) \geqslant & E_q\{\log p(\theta,z,\mathbf{w}|\alpha,\beta)\} \\ & - E_q\{\log q(\theta,z|\mathbf{w},\gamma,\phi)\}\end{aligned} \quad (1)$$

Letting $\mathscr{L}(\gamma,\phi|\mathbf{w},\alpha,\beta)$ denote the right-hand side of Eq.(1)

$$\begin{aligned}\mathscr{L}(\gamma,\phi|\mathbf{w},\alpha,\beta) =& E_q\{\log p(\theta,z,\mathbf{w}|\alpha,\beta)\} \\ &- E_q\{\log q(\theta,z|\mathbf{w},\gamma,\phi)\}\end{aligned} \tag{2}$$

Because we already have[11]

$$E_q\{\log(\theta_k)|\gamma\} = \Psi(\gamma_k) - \Psi\left(\sum_{k=1}^{K}\gamma_k\right)$$

and expand Eq.(2), we have

$$\begin{aligned}&\mathscr{L}(\gamma,\phi|\mathbf{w},\alpha,\beta) \\ =&\log\Gamma\left(\sum_{k=1}^{K}\alpha_k\right) - \sum_{k=1}^{K}\log\Gamma(\alpha_k) \\ &+ \sum_{k=1}^{K}(\alpha_k-1)\left(\Psi(\gamma_k) - \Psi\left(\sum_{j=1}^{K}\gamma_j\right)\right) \\ &+ \sum_{t=1}^{V}\sum_{k=1}^{K}\phi_{t,k}\left(\Psi(\gamma_k) - \Psi\left(\sum_{j=1}^{K}\gamma_j\right)\right) \\ &+ \sum_{t=1}^{V}\sum_{k=1}^{K}\mathrm{tf}_{t,\mathbf{w}}\phi_{t,k}\log\beta_{t,k} \\ &- \log\Gamma\left(\sum_{k=1}^{K}\gamma_k\right) + \sum_{k=1}^{K}\log\Gamma(\gamma_k) \\ &- \sum_{k=1}^{K}(\gamma_k-1)\left(\Psi(\gamma_k) - \Psi\left(\sum_{j=1}^{K}\gamma_j\right)\right) \\ &+ \sum_{t=1}^{V}\sum_{k=1}^{K}\phi_{t,k}\log\phi_{t,k}\end{aligned} \tag{3}$$

where $\Gamma(\cdot)$ is gamma function, $\Psi(\cdot)$ is digamma function.

Firstly, we maximize Eq.(3) with respect to $\phi_{t,k}$ which is the probability of the word t generated by latent topic z. This is a constrained maximization with constraint

$$\sum_{k=1}^{K}\phi_{t,k} = 1$$

We form the Lagrangian by isolating the terms which contain $\phi_{t,k}$ and add the appropriate Lagrange multipliers λ, so we have

$$\begin{aligned}\mathscr{L}^{\mathbf{w}}_{[\phi_{t,k}]} =& \phi_{t,k}\left(\Psi(\gamma_k) - \Psi\left(\sum_{j=1}^{K}\gamma_j\right)\right) \\ &+ \mathrm{tf}_{t,\mathbf{w}}\phi_{t,k}\log\beta_{t,k} + \phi_{t,k}\log\phi_{t,k} \\ &+ \lambda_t\left(\sum_{k=1}^{K}\phi_{t,k} - 1\right)\end{aligned} \tag{4}$$

Taking derivative with respect to $\phi_{t,k}$ and setting the derivative to zero yields the maximized , we have

$$\phi_{t,k} \propto (\beta_{t,k})^{\mathrm{tf}_{t,\mathbf{w}}}\exp\left(\Psi(\gamma_k) - \Psi\left(\sum_{j=1}^{K}\gamma_j\right)\right) \tag{5}$$

Secondly, we maximize Eq.(3) with respect to γ_k, the k^{th} component of the posterior Dirichlet parameter. Like Eq.(4), we also have

$$\begin{aligned}\mathscr{L}_{[\gamma]} =& \sum_{k=1}^{K}\left(\Psi(\gamma_k) - \Psi\left(\sum_{j=1}^{K}\gamma_j\right)\right)\left(\alpha_k + \sum_{t=1}^{V}\phi_{t,k} - \gamma_k\right) \\ &- \log\Gamma\left(\sum_{j=1}^{K}\gamma_j\right) + \sum_{k=1}^{K}\log\Gamma(\gamma_k)\end{aligned} \tag{6}$$

Take the derivative with respect to γ_k and setting to zero yields a maximum:

$$\gamma_k = \alpha_k + \sum_{t=1}^{V}\phi_{t,k} \tag{7}$$

3.4. Estimating

Given a corpus of $\mathscr{D} = \{\mathbf{w}_1,\ldots,\mathbf{w}_M\}$ that $\mathbf{w}$ is a category language model, we use a variational expectation-maximization (EM) algorithm (expectation-maximization algorithm with a variational expectation Step) [1] to find the parameters and which maximize a lower bound on the log marginal likelihood:

$$\ell(\alpha,\beta) = \sum_{\mathbf{w}\in\mathscr{D}}\log p(\mathbf{w}|\alpha,\beta)$$

As we have described above, we can bound the log likelihood using

$$\begin{aligned}\log p(\mathbf{w}|\alpha,\beta) =& \mathscr{L}(\gamma,\phi|\mathbf{w},\alpha,\beta) \\ &+ D(q(\theta,z|\mathbf{w},\gamma,\phi)\,\|\,p(\theta,z|\mathbf{w},\alpha,\beta))\end{aligned} \tag{8}$$

Which exhibits $\mathscr{L}(\gamma,\phi|\mathbf{w},\alpha,\beta)$ as a lower bound because the KL term is positive. We now obtain a variational EM algorithm that repeats the following two steps until Eq.(8) converges:

- (E step) Optimize values for the variational parameter vectors γ and ϕ for each category language model. The update rules are Eq.(5) and Eq.(7).
- (M step) Maximize the resulting lower bound on the log likelihood with respect to the model parameter α and parameter vector β. We can do this by finding the maximum likelihood estimates with expected sufficient statistics computed in the E-step.

Firstly, we maximize Eq.(3) with respect to $\beta_{t,k}$. This is a constrained maximization with constraint

$$\sum_{t=1}^{V}\beta_{t,k}=1\,.$$

We form the Lagrangian by isolating the terms which contain $\beta_{t,k}$ and add the appropriate Lagrange multipliers. So, we have

$$\begin{aligned}\mathscr{L}_{[\beta_{t,k}]}=&\sum_{\mathbf{w}\in\mathscr{D}}\sum_{t=1}^{V}\sum_{k=1}^{K}\mathrm{tf}_{t,\mathbf{w}}\phi_{t,k}\log\beta_{t,k}\\&+\sum_{k=1}^{K}\lambda_k\left(\sum_{t=1}^{V}\beta_{t,k}-1\right)\end{aligned}\tag{9}$$

Taking derivatives with respect to $\beta_{t,k}$ and setting the derivative to zero yields the maximized $\beta_{t,k}$, we have

$$\beta_{t,k}\propto\sum_{\mathbf{w}\in\mathscr{D}}\mathrm{tf}_{t,\mathbf{w}}\phi_{t,k}$$

Secondly, we maximize Eq.(3) with respect to α. Like Eq (9) and derive

$$\begin{aligned}\mathscr{L}_{[\alpha]}=\sum_{\mathbf{w}\in\mathscr{D}}\Bigg\{&\log\Gamma\left(\sum_{k=1}^{K}\alpha_k\right)-\sum_{k=1}^{K}\log\Gamma(\alpha_k)\\&+\sum_{k=1}^{K}(\alpha_k-1)\left(\Psi(\gamma_k)-\Psi\left(\sum_{j=1}^{K}\gamma_j\right)\right)\Bigg\}\end{aligned}$$

Then, take first derivative and second derivative with respective to α (α is a scalar dirichlet parameter). So according Newton-Raphson formula, we can find the maximal α by iteration as following:

$$\begin{aligned}\alpha^{\mathrm{new}}&=\alpha-\frac{\mathscr{A}+\mathscr{B}}{MK\left(K\Psi'(K\alpha)-\Psi'(\alpha)\right)}\\\mathscr{A}&=MK\left(\Psi(K\alpha)-\Psi(\alpha)\right)\\\mathscr{B}&=\sum_{\mathbf{w}\in\mathscr{D}}\sum_{k=1}^{K}\left\{\Psi(\gamma_{k,\mathbf{w}})-\Psi\left(\sum_{k=1}^{K}\gamma_{k,\mathbf{w}}\right)\right\}\end{aligned}$$

where Ψ' is trigamma function.

3.5. *Maximum a Posteriori of Multinomial Parameter*

After model parameter α, model parameter vector β and variational parameter vector ϕ converged, we can fit the variational parameter vector γ as Eq.(7) description.

As we have described in Subsection 3.2, the prominent feature of LDACLM is that the model assume each word would be a independent topic that we called word topic and assume extra topics other than word topics would be model the correlation among the words. Because LDACLM Variational inference of each category $\mathbf{w}$ need to calculate the different variational parameters ϕ, and each variational parameters ϕ are multiplied by the number of characteristics which necessarily limits the number of topics due to limited computer memory. So we present a compromise approach in our LDACLM model. We assume that each word is a independent topic, and derive semantic topic from reasoning. Actually, we derive a total of $\mathscr{K}=K+V$ topics, where V say that the number of characteristics and K is the number of semantic topics. Model parameters β is also need to expand. For the independent topic, $\beta_{t,k}$ associated with one independent topic is 1 , the others were 0; For the semantic topic, $\beta_{t,k}$ can be estimate by the above description method.

Hereafter, for specific category language model $\mathbf{w}$, the maximum a posteriori of multinomial parameter in vector $\theta^{\mathbf{w}}$ can be computed approximately ac-

cording to Frequentist approach to probability as

$$\theta_k^{\mathbf{w}} = \frac{\eta\gamma_k^{\mathbf{w}}}{\sum_{t=1}^{V} x_{t,\mathbf{w}} + \eta\sum_{j=1}^{K}\gamma_j^{\mathbf{w}}} \qquad k = \{1,2,\ldots K\}$$

$$\theta_{t+K}^{\mathbf{w}} = \frac{x_{t,\mathbf{w}}}{\sum_{t=1}^{V} x_{t,\mathbf{w}} + \eta\sum_{j=1}^{K}\gamma_j^{\mathbf{w}}} \qquad t = \{1,2,\ldots V\}$$

where η is a constant number related to specified corpus.

Next, based on our model, we can derive maximum likelihood of document d generating by category language model $\mathbf{w}$ as following formula:

$$p(d|\mathbf{w}) \propto \prod_{t\in d}\left\{\sum_{k=1}^{K}(\theta_k^{\mathbf{w}}\beta_{t,k})\right\}^{\mathrm{tf}_{t,d}} \qquad (10)$$

Eventually, we can classify new document d according to Eq.(10). The document d belong to category language model $\mathbf{w}$ who generate d with maximum probability.

4. Experiments and Results

We have conducted experiments on three real-world datasets, Reuters21578, WebKB and 20Newsgroups, to evaluate the effectiveness of our proposed model for text categorization.

4.1. Datasets

The Reuters21578[*]dataset contains documents collected from Reuters newswire articles which are assigned to 135 categories. However, there are only non-empty 118 categories, among which the 10 most frequent categories called R10 by Debole [3] contain about 75% of the documents as Table 1 show. There are several ways to split the documents into training and testing sets: 'ModLewis' split, 'ModApte' split, and 'ModHayes' split. The 'ModApte' train/test split is widely used in text classification research. We followed the 'ModApte' split in which the 10 most frequent categories called R10 represented in Table 1, and a large number of documents are used for training and testing.

Table 1. Number of Training and Test documents About R10

Category name	Num Train	Num test
earn	2877	1087
acq	1650	719
money-fx	538	179
grain	433	149
crude	389	189
trade	369	118
interest	347	131
wheat	212	71
ship	197	89
corn	182	56

The 20Newsgroups(20NG)[†]dataset is a collection of approximately 20,000 documents that were collected from 20 different newsgroups with about 1000 messages from each newsgroup. This collection consists of 19,974 non-empty documents distributed evenly across 20 newsgroups and we selected 19,946 non-empty documents after feature selection. We use the newsgroups to form categories, and randomly select 70% of the documents to be used for training and the remaining 30% for testing.

The WebKB[‡]dataset contains manually classified Web pages that were collected from the computer science departments of four university(Cornell, Texas, Washington and Wisconsin) and some other university. The pages are divided into seven categories: student, faculty, staff, course, project, department and other. In this paper, we use the four most populous entity-representing categories: student, faculty, course, and project, which all together contain 4199 pages. We called this selected WebKB dataset as WebKB top-4 dataset. Like handling 20Newsgroup dataset, We randomly select 70% of the documents to be used for training and the remaining 30% for testing.

4.2. Experiments

We employed free software MALLET [10] to implement the NaïveBaye(NB) with Laplace smoothing

[*]http://www.daviddlewis.com/resources/testcollections/reuters21578/reuters21578.tar.gz
[†]http://people.csail.mit.edu/jrennie/20Newsgroups
[‡]http://people.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data

and Rocchio methods for document classification tasks. Developed by Andrew McCallum, MALLET is a library of Java code for machine learning applied to text. It provides facilities for many natural language processing, such as document classification. Because the MALLET software does not handle the Reuters21578 dataset and WebKB dataset, we write extra Java code to read Reuters21578 and WebKB datasets in MALLET format. For these three datasets, we performed stop word removal, stemming, and case-conversion to lower case before feature selection was applied on the training set. Furthermore, We apply to information gain [15] feature selecting method to the documents by set information gain threshold 0.055 for 20NG dataset, 0.3 for Reuters dataset and -0.044 for WebKB dataset.

We deployed LIBSVM [2] implementation of SVM which uses the "one vs rest" method for multi-category classification because of its effectiveness and efficiency. Because there are a little difference between polynomial kernel and RBF kernel for LIBSVM in classifying documents, we use polynomial kernel for SVM in this paper. Most of LIBSVM parameters are set to default values and polynomial kernel parameter gamma and coef0 are set to 0.0003 and 1.0 for Reuters21578 datasets, 0.0005 and 1.1 for WebKB dataset, 0.0003 and 1.0 for 20Newsgroups dataset.

We have tried our proposed LDACLM with 100 topics modeling the relationship among words and extra topics set in that each topic belong to one word respectively. Moreover the parameter η of LDACLM has been set to 1.5 for Reuters dataset, 1.3 for WebKB dataset and 2.0 for 20Newsgroups dataset.

Table 2. The F1 experimental results on the Reuter R10 dataset

	NB	SVM	LDACLM	Rocchio
earn	0.982	0.976	0.981	0.973
grain	0.561	0.615	0.571	0.404
wheat	0.308	0.029	0.323	0.503
crude	0.783	0.790	0.798	0.780
acq	0.962	0.954	0.960	0.949
ship	0.673	0.651	0.656	0.725
interest	0.673	0.670	0.684	0.693
money-fx	0.723	0.725	0.757	0.719
corn	0.206	0.074	0.254	0.468
trade	0.817	0.824	0.817	0.831
Macro-aver F1	0.678	0.669	0.690	0.716

Table 3. The F1 experimental results on the WebKB top-4 dataset

	NB	SVM	LDACLM	Rocchio
course	0.946	0.933	0.95	0.880
student	0.885	0.871	0.894	0.850
project	0.813	0.724	0.795	0.741
faculty	0.825	0.830	0.837	0.765
Macro-aver F1	0.868	0.841	0.869	0.811

The results of NaïveBayes, LDACLM, Rocchio and SVM with polynomial kernel on three datasets described in Subsection 4.1 are shown in Figure 3, Figure 4 and Figure 5 respectively. In these figures, 'NB' means NaïveBayes, 'SVM' means LIBSVM with polynomial kernel, 'Macro-aver' means macro-averaging result. and 'Micro-aver' means micro-averaging results.

The F1 values of NaïveBayes, LDACLM, Rocchio and SVM with polynomial kernel on the three datasets are shown in Table 3, Table 2 and Table 4 respectively. The experimental results in tables has been derived by information gain feature selection. Especially, The number of features is 8000 for WebKB top-4 dataset, 8000 for Reuters R10 dataset, 13000 for 20NG dataset.

As Figure 4 shown, the experimental results on Reuters R10 show that support vector machine with polynomial kernel and Rocchio are more robust than Naïve Bayes and LDACLM, but their accuracy are very similar. As Table 2 shown, when the features number is 8000, the F1 value of SVM algorithm about 'wheat' and 'corn' categories are 0.029 and 0.074 . However, the F1 value of LDACLM model about the two categories are 0.323 and 0.254. Moreover the F1 value of LDACLM model is more higher than Naïve Bayes. On Reuters R10 dataset, the Rocchio algorithm do the best job. When the number of features is more than 8000, the the recall rate of LDACLM begin to decline , which also shows that independence topics increasing will result in the worse performance.

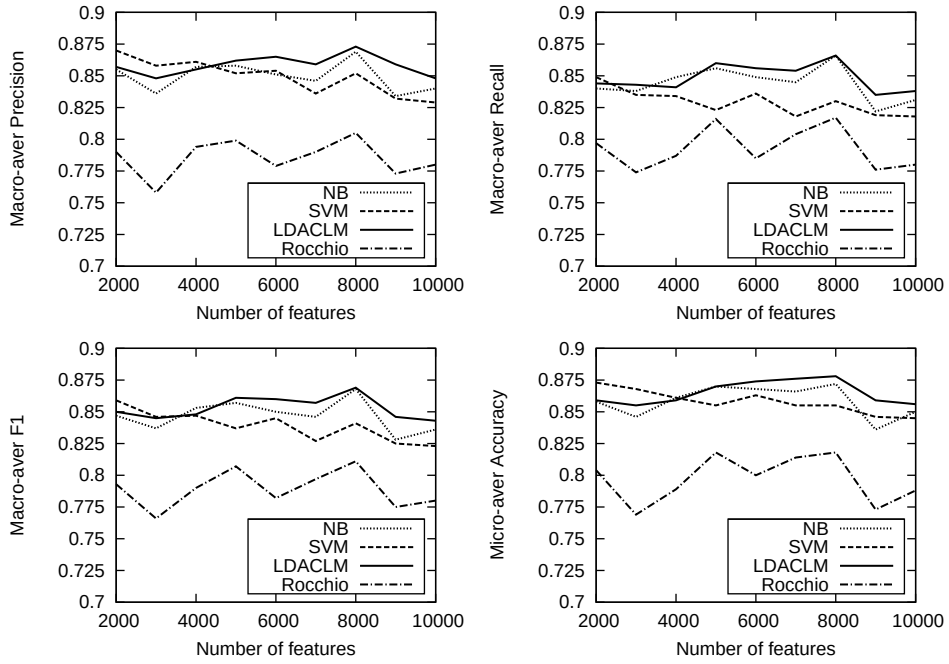

Figure 3: The Results of NB,SVM,LDACLM and Rocchio algorithms on WebKB top-4 Dataset

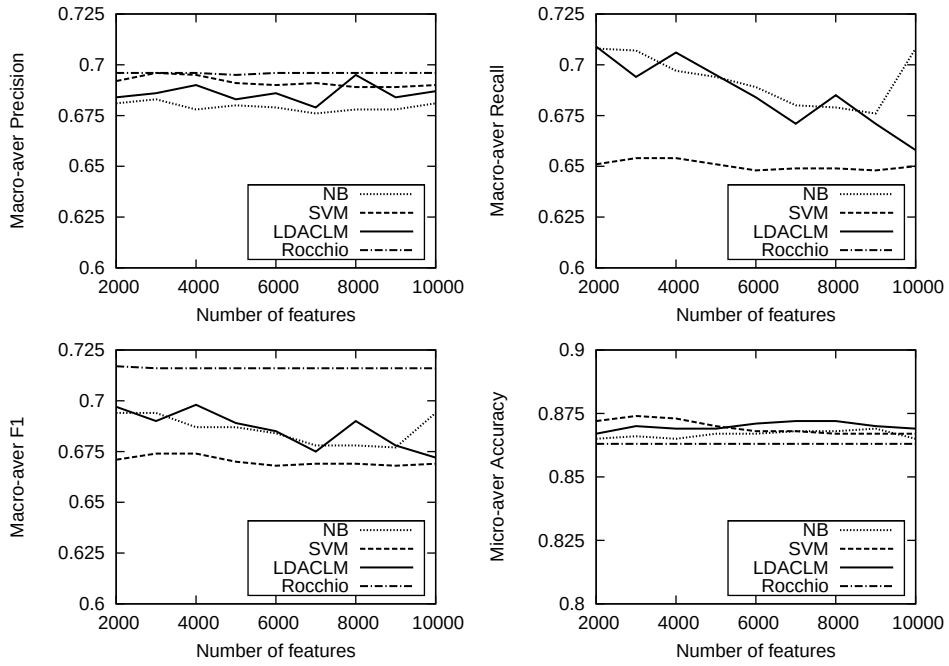

Figure 4: The Results of NB,SVM,LDACLM and Rocchio algorithms on R10 Dateset

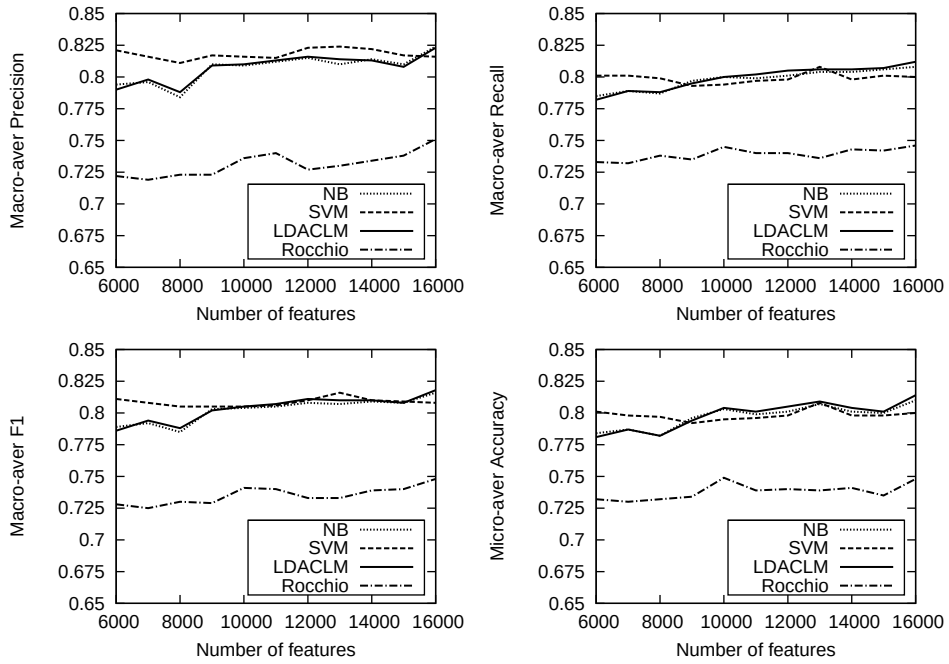

Figure 5: The Results of NB,SVM,LDACLM and Rocchio algorithms on 20NG Dataset

Table 4. The F1 experimental results on the 20NG dataset

	NB	SVM	LDACLM	Rocchio
alt.atheism	0.744	0.709	0.743	0.655
comp.graphics	0.747	0.762	0.749	0.699
comp.os.ms-windows.misc	0.087	0.789	0.189	0.021
comp.sys.ibm.pc.hardware	0.674	0.765	0.685	0.597
comp.sys.mac.hardware	0.759	0.825	0.788	0.675
comp.windows.x	0.768	0.851	0.759	0.639
misc.forsale	0.789	0.624	0.742	0.704
rec.autos	0.907	0.845	0.908	0.851
rec.motorcycles	0.930	0.910	0.934	0.910
rec.sport.baseball	0.955	0.932	0.947	0.922
rec.sport.hockey	0.954	0.932	0.944	0.938
sci.crypt	0.908	0.905	0.907	0.818
sci.electronics	0.831	0.781	0.806	0.712
sci.med	0.914	0.898	0.925	0.882
sci.space	0.925	0.910	0.908	0.861
soc.religion.Christian	0.887	0.884	0.878	0.816
talk.politics.guns	0.833	0.825	0.837	0.782
talk.politics.mideast	0.914	0.889	0.922	0.871
talk.politics.misc	0.745	0.718	0.764	0.635
talk.religion.misc	0.472	0.474	0.477	0.396
Macro-aver F1	0.807	0.816	0.810	0.733

As Figure 5 and Figure 3 shown, LDACLM outperform the Naïve Bayes and Rocchio on Reuters R10 and WebKB top-4 datasets. With less features, SVM is better than LDACLM in performance. On the 20NG dataset, when the number of featurs is 13000, the F1 values of SVM are larger than those of LDACLM as Table 4 shown. However, on the WebKB top-4 dataset, when the number of features is 8000, the F1 values of LDACLM are larger than those of SVM as Table 3 shown.

Specially, All results are averaged across 5 random runs for 20NG and WebKB datatsets. In result, LDACLM outperform NaïveBayes with Laplace smoothing and Rocchio algorithm, but SVM provide much better computational accuracy than LDACLM. Rocchio algorithm do the best on Reuters dataset, but present poor performance on WebKB and 20NG.

In Table 5, we show the test collection classification speed of LDACLM method compare with Naïve Bayes, Rocchio and SVM running on a Dell Optiplex 745 computer.

Table 5. The Naïve Bayes, LDACLM, Rocchio and SVM classification speed

	NB	LDACLM	Rocchio	SVM
20NG	0.234s	0.318s	15.157s	1138.234s
R10	0.062s	0.078s	1.734s	108.750s
WebKB	0.011s	0.018s	0.387s	28.324s

The data unit is second in Tabel 5. So we can conclude that the LDACLM is also a efficient classifier for the three real-world text collections.

5. Conclusion and Future Work

This paper proposed Latent Dirichlet Allocation Category Language Model, a novel model based on LDA model. We have presented variational inference approach, and parameters estimation method which is similar to LDA [1] in category language model. As Results on WebKB, 20NG and Reuters21578 datasets shown above, LDACLM cannot significantly improve performance. In our opinion, we think that it was because the semantic topics modeling the relationship among words is not abundant which constraint by computer memory. In the future work, we will try use topics by collection from Wordnet based on Gibbs sample, and this maybe create many topics which approximate words dependency than variational inference do.

Acknowledgement

We would like to thank the anonymous reviewers for their valuable comments and suggestions. We are grateful for Zhao Cao's helpful discussion and advice. Many thanks also to Shidong Feng, Yingfan Gao, Jian Cao, Jinghua Bai, and Xu Zhang for their suggestions regarding this paper.

References

1. D. Blei, A. Ng and M. Jordan, "Latent Dirichlet allocation,". *Journal of Machine Learning Research(JMLR)*, **3**, 993–1022 (2003).
2. C-C. Chang, C-J Lin, "LIBSVM: a library for support vector machines," Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`. (2001).
3. F. Debole and F. Sebastiani, "An Analysis of the Relative Difficulty of Reuters-21578 Subsets," *Journal of the American Society for Information Science and Technology*, **56(2)**, 584–596 (2004).
4. S. Deerwester, S. Dumais, G. Furnas, T. Landauer and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American Society for Information Science*, **41(6)**, 391–407 (1990).
5. M. Girolami and A. Kaban, "On an equivalence between PLSI and LDA," *Proceedings of the* 26^{th} *annual international ACM SIGIR conference on Research and development in informaion retrieval*, 433–434 (2003).
6. T. Griffiths and M. Steyvers, "Finding scientific topics," *Proceedings of the National Academy of Sciences*, **101** 5228–5235 (2004).
7. T. Hofmann, "Probabilistic Latent Semantic Indexing," *Proceedings of the* 22^{nd} *annual international ACM SIGIR conference on Research and development in information retrieval*, 50–57 (1999).
8. T. Joachims, "A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization," *Proceedings of the* 14^{th} *International Conference on Machine Learning*, (1997).
9. M. Jordan, Z. Ghahramani, T. Jaakkola and L. Saul, "An introduction to variational methods for graphical models," *Machine Learning*, **37** 183–233 (1999).
10. A. McCallum, "MALLET: A Machine Learning for Language Toolkit," `http://mallet.cs.umass.edu`. (2002).

11. T. Minka, "Estimating a dirichlet distribution," Unpublished paper available at `http://research.microsoft.com/~minka`. (2003).
12. J. Ponte and W. Croft, "A Language Modeling Approach to Information Retrieval," *Proceedings of the* 21^{st} *annual international ACM SIGIR conference on Research and development in information retrieval* 275–281 (1998).
13. H. Wallach, "Topic modeling: beyond bag-of-words," *Proceedings of the* 23^{rd} *International Conference on Machine Learning*, (2006).
14. X. Wei and W. Croft, "LDA-Based Document Models for Ad-hoc Retrieval," *Proceedings of the* 29^{th} *Annual International ACM SIGIR Conference on Research and Development on Information Retrieval*, 178–185 (2006).
15. Y. Yang and J. Pedersen, "A Comparative Study on Feature Selection in Text Categorization," *Proceedings of the* 14^{th} *International Conference on Machine Learning*, 412–420 (1997).