An empirical study of text classification using Latent Dirichlet Allocation

Lei Li
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
leili@cs.cmu.edu

Yimeng Zhang
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
yimengz@cs.cmu.edu

1 Introduction

Text classification has been intensively studied during the last 15 years. A number of classifiers have been developed to obtain better classification result [6]. Besides, different models of document have been proposed to better representing documents. In this project, we study latent dirichlet allocation (LDA) [1] and use it to do text classification.

The most popular methodology proposed for document is to present each document as a weight vector of real numbers, each of which is a *tf-idf* weight [5] of each words in the document. This weight is based on the frequency of the word in a document, and tend to give a measure which shows the power of the word to present the document in a corpus. Language models are also used to represent documents, such as bigram [4], instead of unigram which is assumed in *tf-idf* method. Moreover, some more sophisticated ways for document modeling are emerging these years, such as laten dirichlet allocation (LDA). The central idea in LDA is to view each document as a mixture of topics, and try to learn these topics and words generated by each topic for each document. This model also provides a more compressed format to represent documents, which is very useful when handling large corpus.

In this project, we will mainly focus on studying the LDA model and compare it with word feature models, such as tf-idf and bigram.

The remaining part of this paper is organized as follows. Section 2 gives definition of the problem. Section 3 presents the LDA model and its underlying assumptions. Section 4 describes the experiments and discusses the results. Finally, section 5 concludes this paper.

2 Problem Definition

Given a set of documents $D = \{d_1, d_2, \cdots, d_{|D|}\}$, and a set of classes $C = \{C_1, C_2, \cdots, C_M\}$. The classification of a document d is a function to assign it a binary vector of length M, $f(d) = b \in \{0,1\}^M$, such that the j-th bit of b equals 1 if and only if $d \in C_j$. To simplify the case, we consider a typical single class binary classification problem, i.e. for each document d, to tell whether it belongs to a class C or not.

The learning problem of document classification is that, given a training document set D_{train}, and its class labels C_{train}. To output an optimal classification function f. The performance of the classification function is usually measured by classification accuracy on a separate test document set D_{test} and true classification C_{test}.

$$Accuracy = \frac{\sum\limits_{d \in D_{test}} \delta(f(d) - C_{test}(d))}{|D_{test}|} \tag{1}$$

where $\delta(0) = 1$ and $\delta(x) = 0$ if $x \neq 0$.

3 Proposed Method

To treat documents mathematically, a document d is represented as a vector $d = (w_1, w_2, \cdots, w_N)$, where w_i is i-th term in the document (normally after eliminating stop words). With the corpus vocabulary V[1], each term w_i is a $V \times 1$ vector in 1-of-V scheme. It follows normally that w_i forms some multinomial distribution. While it may not be sufficient to express this distribution with explicit parameters, it is more plausible to consider some kind of latent variables. A natural idea is to think these words generated by some topics or mixture of topics. The LDA model [1] lies in this direction.

3.1 Underlying Probability Distribution

A k-dimensional random vector z taking 1-of-k scheme could be viewed as a special case of multinomial distribution Multi(π), while it probability density function specified as:

$$p(z|\pi) = \prod_{i=1}^{k} \pi_i^{z_i} \tag{2}$$

where $\sum_{i=1}^{k} \pi_i = 1$ and $\pi_i \geq 0$.

A k-dimensional random variable θ follows the Dirichlet distribution Dir(α) if its probability density function takes the following form:

$$p(\theta|\alpha) = \frac{\Gamma(\sum_{i=1}^{k} \alpha_i)}{\prod_{i=1}^{k} \Gamma(\alpha_i)} \prod_{i=1}^{k} \theta_i^{\alpha_i - 1} \tag{3}$$

where $\alpha (> 0)$ is a k-dimensional vector, $\Gamma(x)$ is the Gamma function defined as $\Gamma(x) = \int_0^\infty t^{x-1} e^{-t} dt$, and θ satisfies

$$\sum_{i=1}^{k} \theta_i = 1, \theta_i \geq 0 \tag{4}$$

3.2 The LDA Model

The LDA model of a document $d = (w_1, w_2, \cdots, w_N)$ is as follows:

1. Model a k dimensional random vector θ as a Dirichlet distribution Dir(α). This step generates θ as parameters for the probability distribution of latent variable z_n.

$$p(\theta|\alpha) = \frac{\Gamma(\sum_{i=1}^{k} \alpha_i)}{\prod_{i=1}^{k} \Gamma(\alpha_i)} \prod_{i=1}^{k} \theta_i^{\alpha_i - 1} \tag{5}$$

2. For n-th term,

   (a) Model n-th latent variable z_n as a multinomial distribution Multi(θ). z_n is k dimensional vector with k-of-1 schema.

$$p(z_n|\theta) = \prod_{i=1}^{k} \theta_i^{z_{n,i}} \tag{6}$$

   (b) Model w_n as a multinomial distribution Multi$(z_n \beta)$, where β is a $k \times V$ parameter matrix. Indeed, this step is using z_n to choose one row of β as the parameter for the probability distribution of z_n. Together with the whole space of z_n, it indeed makes a mixture multinomial distribution for w_n.

$$p(w_n|z_n, \beta) = \prod_{j=1}^{V} (\sum_{i=1}^{k} z_{n,i} \beta_{i,j})^{w_{n,j}} \tag{7}$$

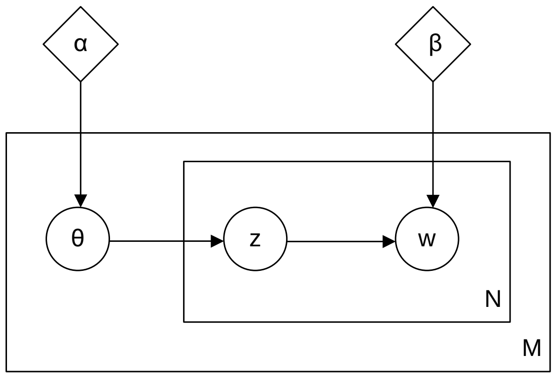

Figure 1: The graphical model of LDA

The graphical illustration of LDA model is shown in Figure 1. Once given the parameters α and β, the joint distribution of θ, $z = \{z_1, z_2, \dots z_N\}$ and $d = \{w_1, w_2, \dots, w_N\}$ is given by:

$$p(\theta, z, d|\alpha, \beta) = p(\theta|\alpha) \prod_{n=1}^{N} p(z_n|\theta) p(w_n|z_n, \beta) \tag{8}$$

The marginal distribution of a document d is given by:

$$\begin{aligned} p(d|\alpha, \beta) &= \int_\theta \sum_z p(\theta, z, w|\alpha, \beta) \\ &= \int_\theta p(\theta|\alpha) \Big(\prod_{n=1}^{N} \sum_{z_n} p(z_n|\theta) p(w_n|z_n, \beta) \Big) d\theta \\ &= \frac{\Gamma(\sum_{i=1}^k \alpha_i)}{\prod_{i=1}^k \Gamma(\alpha_i)} \int_\theta \Big(\prod_{i=1}^{k} \theta_i^{\alpha_i - 1} \Big) \Big(\prod_{n=1}^{N} \sum_{i=1}^{k} (\theta_i \prod_{j=1}^{V} \beta_{i,j}^{w_{n,j}}) \Big) d\theta \\ &= \frac{\Gamma(\sum_{i=1}^k \alpha_i)}{\prod_{i=1}^k \Gamma(\alpha_i)} \int_\theta \Big(\prod_{i=1}^{k} \theta_i^{\alpha_i - 1} \Big) \Big(\prod_{n=1}^{N} \sum_{i=1}^{k} \prod_{j=1}^{V} (\theta_i \beta_{i,j})^{w_{n,j}} \Big) d\theta \end{aligned} \tag{9}$$

For a corpus D of M documents, the data log-likelihood is given by:

$$L(D; \alpha, \beta) = \log p(D|\alpha, \beta) = \sum_{d \in D} \log p(d|\alpha, \beta) \tag{10}$$

3.3 Variational Inference

The task of model documents is to find optimal α and β that maximize the data log-likelihood $L(D; \alpha, \beta)$. While directly using MLE on $L(D; \alpha, \beta)$ proves to be rather intractable, it is tractable to use variational method to approximately calculate a lower bound of the log-likelihood. The essential idea is using Jensen's inequality[2] to obtain a lower bound. Consider the log-likelihood

[1]Not intended to cause confusion, we also refer V to the total number of terms in the vocabulary.

[2]Jensen's inequality states that for a concave function φ, $\varphi(\mathbf{E}[X]) \geq \mathbf{E}[\varphi(X)]$.

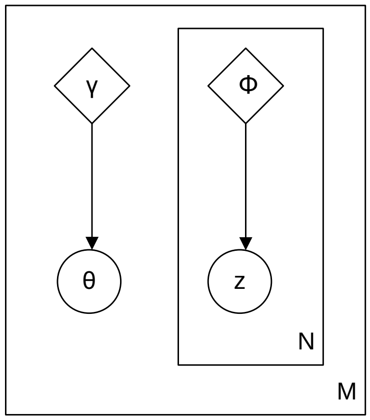

Figure 2: The graphical model of variational distribution

for a document d

$$\begin{aligned}\log p(d|\alpha,\beta) &= \log \int_\theta \sum_z p(\theta,z,d|\alpha,\beta)d\theta \\ &= \log \int_\theta \sum_z \frac{p(\theta,z,d|\alpha,\beta)}{q(\theta,z)} q(\theta,z)d\theta \\ &= \log \mathbf{E}_q[\frac{p(\theta,z,d|\alpha,\beta)}{q(\theta,z)}] \\ &\geq \mathbf{E}_q[\log p(\theta,z,d|\alpha,\beta)] - \mathbf{E}_q[\log q(\theta,z)] \triangleq L\end{aligned} \tag{11}$$

Thus $\log p(d|\alpha,\beta)$ is lower-bounded by L, and the maximization of $\log p(d|\alpha,\beta)$ could be (approximately) reduced to maximize L. The idea is to using an approximate version of probability distribution on latent variables θ and z, denoted as $q(\theta,z|\gamma,\phi)$, where γ and ϕ are variational parameters. To rather simplify the maximization of L, we assume the factorizations of q.

$$q(\theta,z|\gamma,\phi) = q(\theta|\gamma)\prod_{n=1}^{N} q(z_n|\phi_n) \tag{12}$$

where $q(\theta|\gamma)$ follows Dir(γ), and $q(z_n|\phi_n)$ follows Multi(ϕ_n)(Figure 2).

Using the factorization in Eq. (8) and Eq. (12), L can be expanded as:

$$\begin{aligned}L(\gamma,\phi;\alpha,\beta) &= \mathbf{E}_q[\log p(\theta|\alpha)] + \sum_{n=1}^{N}\mathbf{E}_q[p(z_n|\theta)] + \sum_{n=1}^{N}\mathbf{E}_q[p(w_n|z_n,\beta)] \\ &\quad - \mathbf{E}_q[q(\theta|\gamma)] - \sum_{n=1}^{N}\mathbf{E}_q[p(z_n|\phi_n)]\end{aligned} \tag{13}$$

To maximize L, we could take derivative of L with respect to γ, ϕ, α and β, which yield the following update rule:

$$\phi_{n,i} \propto \beta_{i,w_n} \exp(\Psi(\gamma_i)) \tag{14}$$

$$\gamma = \alpha + \sum_{n=1}^{N} \phi_n \tag{15}$$

$$\beta_{i,j} \propto \sum_{d=1}^{M}\sum_{n=1} N_d \phi_{d,n,i} w_{d,n,j} \tag{16}$$

α is estimated using Newton method.

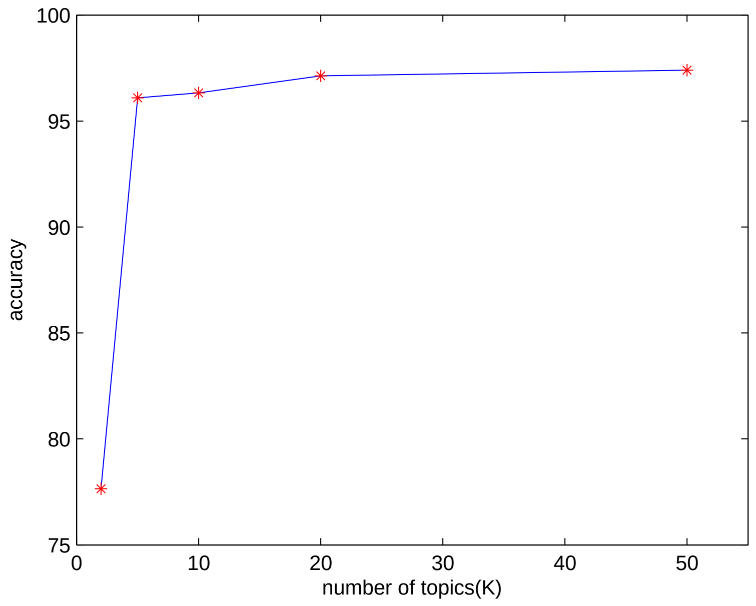

Figure 3: Classification accuracy with LDA features

3.4 Classification with LDA

The LDA model could be used to classify document in a discriminative framework. For each document d, using LDA to learn γ_d for that document, treat γ as the features, and use any off-the-shelf classifier (e.g. SVM [3]) to attach class labels. Normally the number of topics is rather smaller than the number of terms, thus LDA can effectively reduce the feature dimension.

4 Experiments

We carried out a binary classification experiment using the Reuters-21578, Distribution 1.0 dataset [3]. We split the dataset into training set (8,681 documents) and test set (2,966 documents) using ModApte split. The task is to classify document into two categories: EARN and NON-EARN. We preprocess the data using standard elimination of stop words and numbers.

The experiment environment is Pentium 3.2 GHz dual CPU, 2G memory, 160G hard disk, with WINDOWS XP and JAVA SDK 6.0 beta2.

We compare the classification accuracy with the different number of LDA topics(Figure 3) on the data set using ModApte split. Figure 4 shows the classification time. It shows that with a small number of topics(5), the SVM could perform reasonably well, while the increase of topics seems not much helpful to improve the performance.

We compare performance of LDA features, tf-idf, and bigram, using SVM as the classifier. The tf-idf weight [5] is a measure of word importance in document, as well as in corpus. The term frequency (*tf*) is simply the number of times a given term appears in that document. The inverse document frequency (*idf*) is the logarithm of the number of all documents divided by the number of documents containing the term, which is the general importance of the term in corpus. Then $tfidf = tf \cdot idf$. In tf-idf weighting scheme, we assume that in a document, each word is exchangeable with each other. Conversely, bigram model assumes that the distribution of the current word is dependent on and only on the previous word, thus bigram model simple counts the co-occurrences of two consecutive

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

[4]The classification time includes both training SVM and classification on test set.

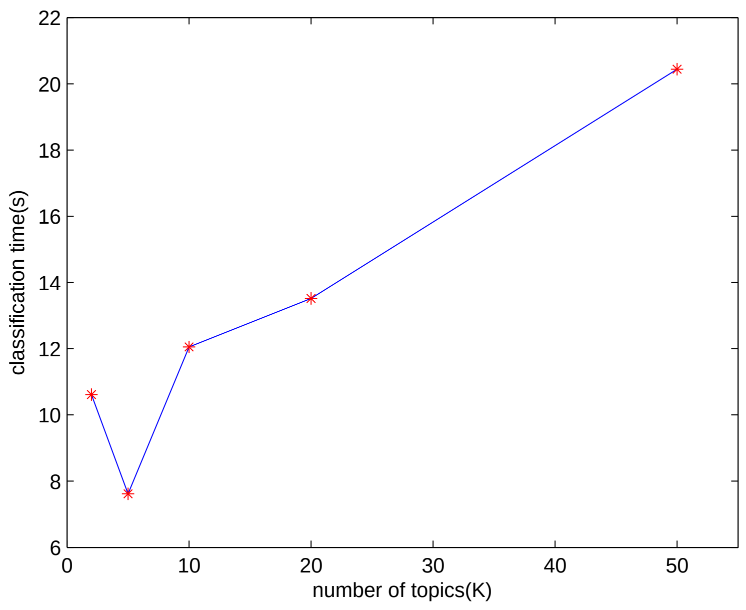

Figure 4: Classification time[4]with LDA features

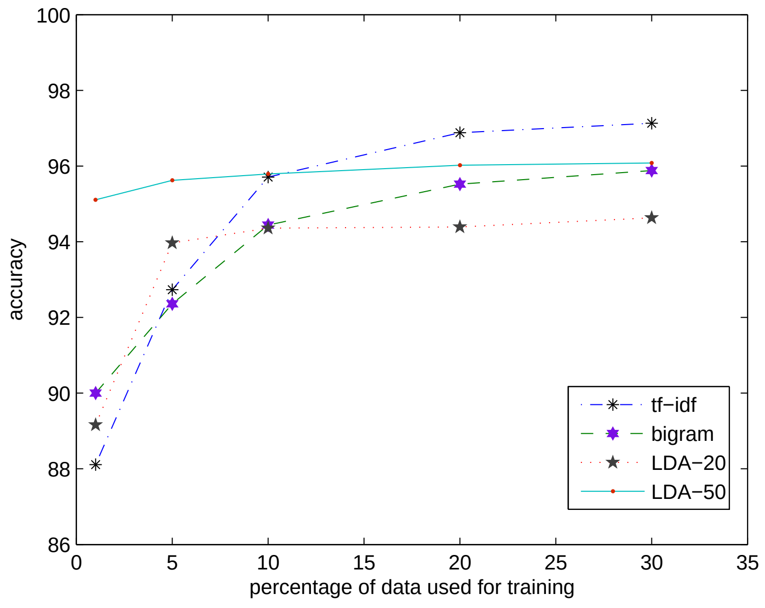

Figure 5: Classification Results: comparing accuracy of tf-idf, bigram, LDA with 20 topics, and LDA with 50 topics

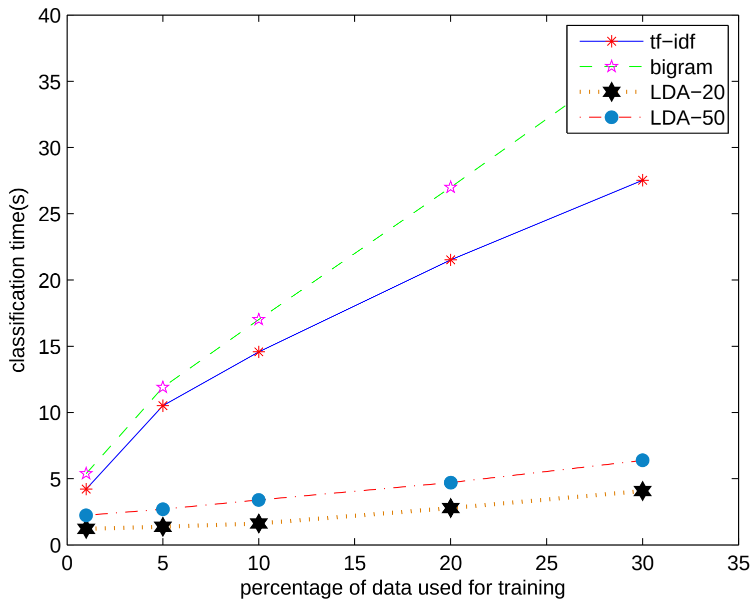

Figure 6: Classification Results: comparing speed of tf-idf, bigram, LDA with 20 topics, and LDA with 50 topics

words. All weights are normalized for each document using cosine normalization. For the baseline implementation, we use the Minorthird toolkit [2].

We compare the classification accuracy using different proportion of training set to train SVM, the rest as test data. Figure 5 shows an interesting property of LDA: it performs better than others with very few training data. Another obvious conclusion is that LDA works faster than others (Figure 6). It is reasonable that LDA works well with only a small set of training text, because LDA uses the global information of the whole corpus to extract the latent features for each document. Thus these features reflects, more or less, the distribution of the whole corpus.

5 Conclusions

We have implemented the LDA algorithm to model documents and extract features for classification. We have compared the performance of classification with different number of LDA topics, and the performance with tf-idf and bigram. The LDA can be used as an effective dimension reduction method for text modelling. It works relatively well, especially with small set of training data.

References

[1] D. Blei, A. Ng, and M. Jordan. Latent dirichlet allocation, 2003.

[2] William W Cohen. Minorthird: Methods for identifying names and ontological relations in text using heuristics for inducing regularities from data. http://minorthird.sourceforge.net, 2004.

[3] Thorsten Joachims. A statistical learning learning model of text classification for support vector machines. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 128–136, New York, NY, USA, 2001. ACM Press.

[4] Jay M. Ponte and W. Bruce Croft. A language modeling approach to information retrieval. In *Research and Development in Information Retrieval*, pages 275–281, 1998.

[5] Gerard Salton and Chris Buckley. Term weighting approaches in automatic text retrieval. Technical report, Ithaca, NY, USA, 1987.

[6] Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002.