Analysis of Twitter Data Based on Topic Models

Zhengtian Chu*

University of Nottingham
chuzhengtian99@gmail.com

September 5, 2020

Abstract

In this paper, we discuss several common topic models by introducing the process and analyzing its applicable situation, and apply them to the experiment of refining tweets topic. PLSA appeared in 1999. It has high efficiency in obtaining text topics, but with the increase of model parameters, it is easy to over fit. LDA makes a priori distribution of parameters, which solves this problem well. BTM makes a further improvement on LDA, which makes the topic model get higher accuracy in the analysis of short texts. These topic models are the core of this paper.

I. Introduction

When people write articles or give speeches, they often decide which topics to write first. For example, when an author wants to write a report on influenza, he may use 40% of the content to describe the virus and its harm, 30% to discuss medical measures, 20% to reflect the public's response and government's action, and the remaining 10% to discuss other topics.

When introducing virus, the following words may appear: virus, harm, damage, affect...; when referring to medical treatment, they often appear: hospital, doctor, mask, patient, cure...; the following words are often related to the people and the government: government, panic, appeal...

An article is usually composed of multiple topics, and the probability of these words appearing in such topics is very high. Therefore, the topic can be regarded as the probability distribution of words. Topic model is such a statistical model used to discover abstract topics in documents. Traditional text topic models such as PLSA and LDA can achieve better results when dealing with long texts with hundreds of words, but they are not dominant in dealing with short texts such as tweets, while BTM can achieve better results. This paper will introduce PLSA, LDA and BTM three text topic models, and use them to process tweets, and analyze the results.

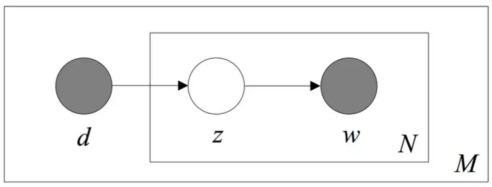

Figure 1: *PLSA Representation*

II. Methods

i. PLSA

Hoffmn[1] first proposed **PLSA**(Probabilistic Latent Semantic Analysis) Topic Model in 1999. He believes that every article is a mixture of multiple topics, each topic is a probability distribution on vocabulary, and every word in the article is generated by topics.

As Figure 1 shows, there are M documents: $D = \{d_1, d_2, ..., d_M\}$, N words: $W = \{w_1, w_2, ..., w_N\}$ and K topics: $\{z_1, z_2, ..., z_K\}$.

*Zhengtian Chu is with the University of Nottingham

Moreover, let the joint distribution *doc-topic* as $\vec{\theta_1}, \vec{\theta_2}..., \vec{\theta_m}$, the joint distribution *topic-word* as $\vec{\varphi_1}, \vec{\varphi_2}, ..., \vec{\varphi_k}$. The generation probability of word w_n in document d_m is:

$$p(w_n|d_m) = \sum_{k=1}^{K} p(w_n|z_k)p(z_k|d_m) = \sum_{k=1}^{K} \varphi_{z_k w_n} \theta_{m z_k} \tag{1}$$

The generation probability of the whole document d_n is as follows:

$$\begin{aligned} p(\vec{w}|d_m) &= \prod_{i=1}^{M} \sum_{k=1}^{K} p(w_i|z_k)p(z_k|d_m) \\ &= \prod_{i=1}^{M} \sum_{k=1}^{K} \varphi_{z_k w_i} \theta_{dk} \end{aligned} \tag{2}$$

We can see that in PLSA, topic distribution and word distribution are fixed. People always extract a topic in a document with a fixed probability, and then extract a word according to the word distribution of the topic. It can clearly distinguish the different meanings of words and topics, but with the increase of parameters in model, it will overfit.

ii. LDA

For PLSA model, D.Blei, A.Ng and M.Jordan[2] added prior distribution to $\vec{\theta_m}$ and $\vec{\varphi k}$, and carried out Bayesian transformation. Because $\vec{\theta_n}$ and $\vec{\varphi k}$ all correspond to Multinomial Distribution, a proper choice of prior distribution is Dirichlet Distribution ($\alpha_1, \alpha_2, ..., \alpha_k$ represents the hyper-parameters of θ):

$$Dir(\vec{p}|\vec{\alpha}) = \frac{\Gamma(\sum_{K=1}^{K} \alpha_k)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} \prod_{k=1}^{K} p_k^{\alpha_k - 1} \tag{3}$$

Thus, Latent Dirichlet Allocation(LDA) model is obtained. Assuming that there are M documents in a corpus, the set of words and topics can be expressed by: $\vec{w} = (\vec{w_1}, ..., \vec{w_M}), \vec{z} = (\vec{z_1}, ..., \vec{z_M})$. The joint distribution of *doc-topic* and *topic-word* are still θ and φ. There are two main process in LDA model as Figure 2 shows.

1. $\vec{\alpha} \rightarrow \vec{\theta_m} \rightarrow z_{m,n}$, this process means when generating the m-th document, select $\vec{\theta_m}$ and generate the n-th topic $z_{m,n}$ of d_m. Obviously, $\vec{\alpha} \rightarrow \vec{\theta_m}$ corresponds to Dirichlet distribution($\theta_m \sim Dir(\alpha)$) and $\vec{\theta_m} \rightarrow z_{m,n}$ corresponds to Multinomial Distribution($z \sim Multi(\theta_m)$), so the whole is a Dirichlet-Multinomial conjugate structure.
2. $\vec{\beta} \rightarrow \vec{\varphi_k} \rightarrow w_{m,n}|k = z_{m,n}$, similarly, this process means choosing $\vec{\varphi_k}$ from Dirichlet Distribution $\beta(\varphi_k \sim Dir(\beta))$to generate word $w_{m,n}$ when $k = z_{m,n}$ from Multinomial Distribution($w \sim Multi(\varphi_k)$). With the restrict $k = z_{m,n}$, we only consider words generated by the same topic, but no longer consider the concept of document. It is also a Dirichlet-Multinomial conjugate structure.

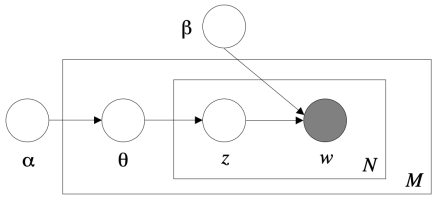

Figure 2: *Latent Dirichlet Allocation model representation[2]*

The probability of a document d containing n words can be expressed as[2]:

$$p(d) = \int_{\theta} (\prod_{m=1}^{M} \sum_{z_m=1}^{k} p(w_m|z_m;\beta)p(z_m|\theta))p(\theta;\alpha)d\theta \tag{4}$$

Different from PLSA, in LDA, topic distribution and word distribution are uncertain. The authors of LDA adopt the idea of Bayesian and use Dirichlet distribution as the conjugate prior distribution.

LDA model uses probability and statistics method to analyze documents, which is based on *bag of words* hypoyhesis[3]. It holds that a document can be expressed as a group of words in disorder, while ignoring the grammar and word order. Compared with the grammar rule-based linguistic text analysis method, LDA is more suitable for the modeling and analysis of network text with a large number of

grammatical irregularities by using the probability statistics of the occurrence times of words in the text.

iii. BTM

Short texts usually exist in webs, tweets, advertisements, comments, etc. The task of data mining is to extract topics from these massive short text data, and provide a good basis for subsequent analysis. Previous text models such as PLSA and LDA assume that an article is composed of multiple topics, and the theme of the article is determined according to the distribution probability of different topics. These methods have a high accuracy in learning long text, but when the length of the text is reduced, due to the lack of sufficient context, the identification ability of these techniques is greatly reduced. For those short text data, a Biterm Topic Model(BTM)[4] was born.
The biggest difference between BTM and LDA is that it introduces the concept of *biterm*.Biterm means a pair of words in a short text. For example, if there is a sentence "hello new world", LDA model breaks it down into three words from different topics: hello, new and world. However, in BTM, this sentenced can be considered as a mixture of three biterms(hello new, new world, hello world) belong to different topics. Because of this, BTM doesn't have sparsity problem as LDA does.
The process model of BTM is similar to LDA. As Figure 2 shows, Btm can be decomposed into following steps:

1. Generate a Dirichlet Distribution of topic on a document.($\theta \sim Dir(\alpha)$)
2. Generate a Dirichlet Distribution of word on a specific topic in all of the documents.($\varphi_k \sim Dir(\beta)$)
3. Draw a Multinomial Distribution of topics from θ.($z_m \sim Multi(\theta)$)
4. Draw a Multinomial Distribution of biterms on topic z_m.($w_i, w_j \sim Multi(\varphi_{z_m})$)

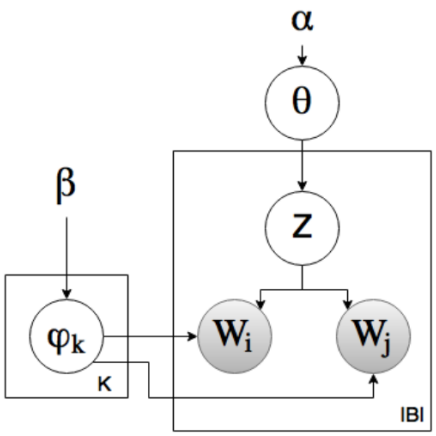

Figure 3: *Graphical representation of BTM*

[4] used Gibbs sampling to calculate θ and φ and here are the results after training:

$$\begin{aligned} \varphi_{w|z} &= \frac{n_{w|z} + \beta}{\sum_w n_{w|z} + M\beta} \\ \theta_z &= \frac{n_z + \alpha}{|B| + K\alpha} \end{aligned} \tag{5}$$

Where B is the set of biterms, K is the number of topics, M is the number of words, $n_{w|z}$ is the number of times word w assigned to the topic z, n_z is the number of times a biterm b is assigned to topic z.

iv. Programming Methods

III. Experiment

i. Dataset Preparation

Twitter is one of the most popular social platforms in the world, and the data on Twitter has good timeliness and practicability. The twitter data quoted in this experiment was downloaded from `https://github.com/qiang2100/STTM/blob/master/dataset/Tweet.txt`. These data were selected from 2011 and 2012 microblog tracks at Text REtrieval Conference (TREC) 2, which contains 2472 documents with an average length of 8.55 words, 89 topics and totaling 5098 words.

Table 1: *Five different topics from LDA training results with ten most probable words*

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
acai	yemen	king	superbowl	murray
berry	day	speech	commercial	djokovic
weight	government	top	doritos	open
loss	anti	award	ad	australian
diet	rage	sag	pepsi	blog
plan	protest	director	youtube	dinner
healthy	hit	british	best	final
amazon	protester	sanction	audi	unemployment
fruit	sanaa	tuesday	hate	novak
living	trailer	race	xoom	early

In order to obtain high-quality tweets, these steps are adopted in preprocessing:

(1) removing numbers and punctuations
(2) converting letters into lower case
(3) removing stopwords(such as *is*, *are*, *to*) by using *python nltk library*
(4) using *WordNetLemmatizer* in *python* to lemmatize the word, for example, change *ate* to *eat highest* to *high*, *cars* to *car*, etc.

i.1 LDA

This experiment use *gensim* in *python* to help programming LDA model. We set the number of topics $K = 89$ as the total topics metioned in the dataset description and hyperparameter $\alpha = 1/K$, $\beta = 0.01$.

i.2 BTM

This experiment set $K = 89$, $\alpha = 1/K$ and $\beta = 0.01$ for BTM, the same as LDA. For library, we selected *biterm* in *python*.

ii. Evalution

ii.1 LDA

Select five different topics from the LDA training results, as shown in Table 1. It is easy to interpret most of the topic. For instances, *Topic 1* describes a kind of berry named 'acai' growed in Amazon, which is benefit to lose weight; The second topic is obviously related to the anti-government armed movement in Sanaa, Yemen; The third topic is about the movie *The King's Speech*, which won many awards from the SAG Award. Topic four may be about businesses that put ads or commericials on the super bowl, such as Pepsi, Doritos, XOOM, Audi. What's more, the fifth theme is the men's singles final of the Australian Open between Andy Murray and Novak Djokovic, but we don't judge what are words 'dinner', 'blog', 'unemployment' mean.

ii.2 BTM

For BTM, in order to compare with the training effect of LDA model, we also selected the ten most likely words of similar topics to Table 1 in Table 2, in bold are different words from similar topics in LDA. By observing the results of BTM, we find that the results of the first, second and fourth topics are similar to LDA, and even if there are differences in words, they are easy to understand. However, in the third and fifth topics, the words obtained by BTM are obviously closer to the meaning of the theme than LDA. Therefore, we can draw the conclusion that BTM topic model is better than LDA in short text analysis.

ii.3 Coherence Score

In this experiment, we used coherence score[5] to evaluate the effect of topic models. This

Table 2: *Five different topics from BTM training results with ten most probable words*

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
acai	yemen	king	commercial	murray
berry	protest	speech	superbowl	djokovic
weight	**president**	award	doritos	open
diet	hu	**thousand**	pepsi	australian
loss	government	**best**	best	final
plan	**ali**	**nomination**	ad	**andy**
healthy	**yemeni**	**actor**	youtube	novak
reduction	anti	**win**	**max**	**win**
living	day	sag	**watch**	**men**
juice	rage	**guild**	**blitz**	**set**

score calculates the coherence of a topic by observing the correlation between different words under the same topic. Given a topic z and the 'top' N word set $S^z = \{w_1^z, w_2^z, ..., w_N^z\}$ belonging to z, $P_1(w)$ is the document frequency of the word w and $P_2(w_1, w_2)$ is the co-occurrence frequency of the word w_1, w_2. the consistency score of this topic is calculated as follows:

$$C(z; S^z) = \sum_{n=2}^{N} \sum_{l=1}^{n-1} \log \frac{P_2(w_n^z, w_l^z) + 1}{P_1(w_l^z)} \quad (6)$$

We counted the change of coherence score as the number of topics increased, as shown in the Figure 4. In the whole graph, the average value of LDA and BTM are -94.315 and -92.700 respectively. The higher the coherence score, the higher the relevance of each word in the topic, which can directly reflect the test effect of the model. The coherence score of BTM is higher than LDA, which shows that BTM is more advantageous in topic modeling of short text content such as tweet.

IV. Discussion

In this experiment, we analyzed a series of tweets data through LDA and BTM topic models, and the results are shown in Table 1 and Table 2, it can be found that the top words extracted by BTM is more close to the event hot spot than LDA. In the stage of evaluation model, we use the coherence score (Figure 4), and we can find that the effect of BTM is slightly better than LDA in the text modeling of the short text, but the difference between them is different. This may be due to the following reasons: (1) the quality of the data set is relatively high, because many tweets are obtained by searching for keywords or hashtags, and often have clear topics. (2) The average length of tweets is 8.55 words. Thus, for LDA model, word co-occurrence is not very sparse.

V. Conclusion and Future Work

In this paper, by analyzing the syntax features of tweets and combining LDA and BTM, we find that BTM can solve the sparsity problem faced by LDA in short text topic mining. This model has important research value and significance for short text mining. In the follow-up study, we hope to apply the BTM model to the analysis of long articles, which may lead to problems such as too much data and too slow modeling speed. We will consider how to improve the modeling speed and efficiency, so as to obtain better results with BTM than LDA.

References

[1] Hofmann, Thomas. "Probabilistic latent semantic indexing." Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval. 1999.

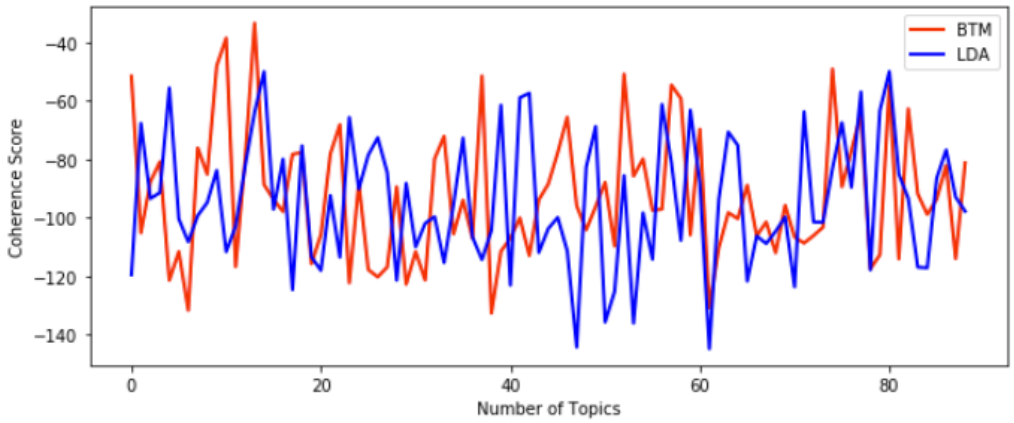

Figure 4: *The coherence score of LDA and BTM as the number of topics increased*

[2] Blei, David M., Andrew Y. Ng, and Michael I. Jordan. "Latent dirichlet allocation." Journal of machine Learning research 3.Jan (2003): 993-1022.

[3] Fei-Fei, Li, and Pietro Perona. "A bayesian hierarchical model for learning natural scene categories." 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). Vol. 2. IEEE, 2005.

[4] Yan, Xiaohui, et al. "A biterm topic model for short texts." Proceedings of the 22nd international conference on World Wide Web. 2013.

[5] David Mimno, Hanna M Wallach, Edmund Talley, Miriam Leenders, and Andrew McCallum. Optimizing semantic coherence in topic models. In Proceedings of the Conference on Empiri- cal Methods in Natural Language Processing, pages 262–272. Association for Computational Linguistics, 2011.