Topic Modeling of Large Scale Social Text
Jia-wen WANG and Qun YANG
College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China

Keywords: Topic modeling, Short texts, Large scale texts, Parallel modeling.

Abstract. In order to solve the problem of topic modeling for large-scale social short texts, this paper studied the parallel LDA modeling method and the dynamic topic model that can capture the dynamic characteristics of the topic. After that, a dynamic topic modeling method for large-scale text sets is proposed, which is based on the data decomposition and post-clustering method. It divided the whole corpus into independent fragments according to different features (e.g., time feature) and modeled the corpus in parallel. Then, it clustered the local topic in later stage. Experiments show that compared with DTM its execution time is less and it can capture the dynamic characteristics of the topic more effectively.

Introduction
The LDA model in the topic modeling has been widely used as an effective text modeling model [1]. One of the common parametric inference methods for LDA is based on the Markov Monte Carlo Gibbs sampling [2, 3]. The algorithm repeatedly samples the topic of each word, and then updates the topic until it converges to fit the document. But that method makes a hypothesis that the word and the document are disordered [4]. So all documents are generated at the same time, which making it difficult for the algorithm to be parallelized in real applications.

On the other hand, the web applications today, such as social media, generate billions text data, which limits the applications of LDA on large-scale text. Also, more topics will be covered, which further increases the computational complexity of model learning.

In streaming big data analysis, the application of the dynamic characteristics of the topic over time needs to be took into account when apply topic model. For example, the content of streaming short text on social media, e.g., Twitter, will change rapidly over time [5]. That makes the fixed generative model of LDA not applicable. Blei et al. proposed a Dynamic Topic Model (DTM) to weaken the impact of this assumption on all LDA models. In DTM, the corpus is divided into different fragments according to time, and each fragment has its own topic distribution. These topics represent some specific content within each time fragment, and the DTM can capture the topic's change over time. However, the update process of DTM is independent of the results of the previous time slice, and its convergence rate is not as good as LDA. In addition, due to the dependence of the sequence, the model cannot be well parallelized, so the application of the model is constrained by the size of the data set.

In general, data can be decomposed into partitions (e.g., by time segmentation) and combined to handle larger datasets. However, this method is hard to apply in DTM. Therefore, we propose a model called Decomposition Clustering-LDA (DC-LDA). The model can mine dynamic topic in parallel from large-scale text, which can be applied to large-scale streaming short text and data sets segmented by time, geographic location, etc.

The remainder of the paper is organized as follows. Section 2 introduces the characteristics of social text and some modeling methods. Section 3 elaborates the framework of DC-LDA model and the parallelization of LDA. Section 5 presents our experiment and the result. Section 6 summarizes our study.
Parallel Topic Model

LDA is deficient in terms of extensibility, and it takes a long time to do inference even for small-scale corpus. Therefore, Wang et al. proposed Parallel-LDA (PLDA) [6] to model the large-scale data.

PLDA is a distributed framework based on LDI, MPI, or MapReduce. PLDA stores the topic counts of each word rather than its probability distribution. It divides the document into different processors to achieve data parallelization and iterates over the entire corpus using Gibbs sampling. Each processor holds a copy of the word and the topic count, and if any of the counts on the processor changes, the topic of the word in the document will be updated by communication processing. In each iteration, the processor does not communicate with each other, so this method yields an approximate Gibbs sample, rather than the exact result of the overall range.

PLDA + [8], the expansion of PLDA, propose four strategies to improve scalability: data deployment, pipeline processing, word bundling, and priority-based scheduling. When a word is bundled with another, the data is deployed so that the pipeline can protect the delayed communication; the words are tied so that the computation time can be long enough to shield the communication. At the same time, scheduling is based on a planned round-robin rather than static processing, which is managed by some processors, so PLDA + implements model parallelization.

Dynamic Topic Model

One of the basic assumptions of LDA is that documents are equally important and are identically distributed. However, this assumption is obviously unreasonable for a long time span of a large corpus, the topic of documents will evolve in this duration. In this case, modeling for span topic must be considered.

DTM (Dynamic Topic Model), proposed by Blei et al. [4], can efficiently capture the overall topic over time. However, it cannot capture the generation and demise of the topic; 2) The model assumes that the evolution of the topic can be identified in both stages, but after evolution, the DTM cannot capture the topic's generation and extinction; Of the topics may not be recognized;

Parallel Topic Modeling Based on Data Decomposition

Aiming at the problems in modeling the topic of large scale social text, we proposed DC-LDA, a high parallel method based on data decomposition and clustering. Under this model, the whole data is divided into multiple fragments and estimate the local topic in each data fragment parallelly; then, merge and cluster local topics to get an overall distribution of the topic [9].

| DC-LDA |  
|---|---|
| 1: procedure begin: |  
| 2: Split D into S segments |  
| 3: set the number of local topics to \( L \) |  
| 4: for all segments \( s \in \{1,...,S\} \) in parallel do: |  
| 5: estimate local topic with LDA: \( \{t_i^{s}\}_{i=1}^{L} \) |  
| 6: end for |  
| 7: parallel procedure: \( U \leftarrow \text{Merge}(\{t_i^{s}\}_{i=1}^{L} \text{ for } s \in \{1,...,S\}) \) |  
| 8: set the number of overall topics to: \( K \) |  
| 9: cluster \( U \) into \( K \) overall topics |  
| 10:end procedure |  

As shown in Table 1, we first split the whole data into \( S \) segments: \( s_1, s_2, \ldots, s_S \), the number of local topics \( L \), which can be different with the overall number of topics [10], set to uniform \( L \) for simplification. Next, we parallelly estimate the local topic with LDA for each segment. In this step, the documents in each fragment can be used for inference based on the LDA in the schema space or using a highly parallelized LDA [11]. This step can be run on a different processor for parallel...
processing, or for multi-core single-use on the use of parallel PLDA inference. After this step, we can deduce the local topic \( \{t_{is}^L\}_{i=1}^L \) from each segmented text \( s \in \{1,...,S\} \), there are a total of \( S^*L \) local topics. For each \( U \). These topics are clustered at the next stage.

The third step in DC-LDA is to merge the topic \( \{t_{is}^L\}_{i=1}^L \) of each segment into an overall topic \( U \). When combine the topics, the distribution of each topic must be compatible mutually. We summarize all topics from all segments and organize a uniform feature representation. That is, map the original topic vector into a new feature space which contain all the words in vocabulary, keep the component which already exist, fill the value of no-exist words with zero.

After merging local topics, we cluster them with k-means algorithm [12] and Kullback-Leibler divergence. To initial independent with data, the cluster centers are obtained with LDA on entire corpus of the entire corpus with k-means, which can be done in parallel with other LDA steps. Since the quality of the initial topic does not affect subsequent clustering, we use fewer iterations to deduce these topics [13, 14].

After clustering, every cluster center is a global topic representation of topics in the cluster. The trends of topics can be obtained by analyzing global topics in a time span.

**Experiment**

We analyzed a subset of 4,673,560 tweets from SNAP of 7 months between June 2009 to December 2009, 533,560 articles from Science Abstract of 17 years between 1996 to 2012, which is shown in Table 2: The TWEETS dataset was divided by a monthly span of about 667,651, and the size of the non-repetitive vocabulary in the vocabulary was 78,774. The Science Abstract was divided by year span of 31,000 documents, and the non-repetitive vocabulary of the dataset is 22,410 and all of the stop words and words of frequency below 0.1% were removed. This section will use these two data sets to demonstrate the effectiveness and scalability of DC-LDA.

|              | Tweets     | Science Abstract |
|--------------|------------|------------------|
| Documents    | 4,673,560  | 533,560          |
| Segments     | 7          | 17               |
| Vocabulary   | 78,774     | 22,410           |
| Time Span    | 7 months   | 17 years         |

We evaluated the execution time performance, the confusion of the topic quality, and the dynamics of the local topic and the overall topic of DC-LDA and DTM on 2.30GHZ Linux system. The parallelization of DC-LDA is based on PLDA, and uses one of the implementation for DTM [16].

First we compared the execution time performance of DC-LDA and DTM on the Science dataset. The experiment uses a small vocabulary, which contains only 1,253 words, where each word appears in at least 1% of the document. The number of topics in the DTM model is set to 20, and the local topic \( L \) in the text of each fragment is set to 50 in DC-LDA, and the overall topic \( K \) is 20. Because DTM does not support parallel processing, it can only run on stand-alone. DC-LDA will run in parallel on multiple machines, where each machine's environment and processor is exactly the same with DTM's. The results in Table 3 show the execution time of the two models under different configurations. The results show that the DC-LDA which capable of parallel processing on multiple machines is significantly faster than the DTM. For the sequentiality of LDA, it takes longer to execute for DC-LDA, where PLDA is used for parallelization of each segmented text set. Since the size of the clustering input is much smaller than the text set itself, the subsequent k-means can be completed in a very short time, making the execution time much smaller than the execution time on the entire corpus.
Table 3. Run time on Science dataset.

| Model    | CPUs | Iteration | Run Time[min] | Run Time[h] |
|----------|------|-----------|---------------|-------------|
| DTM      | 1    | 100       | 1561          | 26          |
| DC-LDA   | 2    | 1000      | 12            | 0.2         |
| DC-LDA   | 4    | 1000      | 6             | 0.1         |
| DC-LDA   | 10   | 2000      | 9             | 0.15        |

In order to evaluate the effectiveness, we only compare similar topics obtained by DC-LDA and DTM. We use perplexity measure for measurement [18]. In the TWEETS data set, the confusion of the two models shown in Figure 1. It can be seen that the number of iterations increases, the perplexity of DC-LDA is always less than DTM. This indicates that our approach is slightly superior to DTM in terms of topic quality.

![Figure 1. Confusion on the Tweets dataset.](image)

DTM cannot capture the appearance and disappearance of any topic, which makes each fragment of the DTM use the same number of topics. And DTM fixes the number of topics that may change over time, and each local topic in the fragment has only one representation. DC-LDA relaxes this limitation: allows an overall topic to have any number of local representations in each fragment. In this way, in addition to allowing the separation of topic to better fit the local data, but also makes the overall topic can be arbitrary and disappear.

Figure 2 and Figure 3 show the changes in the three topics selected in the dataset TWEETS and Science, respectively. From the figure, DC-LDA can capture the trend of topic. Unlike the DTM, the overall topic of the DC-LDA does not need to be made up of an exact local topic, which more intuitively reflects the change in the topic over time.

![Figure 2. The trends of three topics on Tweets.](image)  
![Figure 3. The trends of three topics on Science.](image)
Summary

We first introduce the LDA topic modeling method on large-scale social text, and expounds the characteristics of LDA topic modeling that cannot capture the real-time attribute of social text. At the same time, in order to deal with the larger text data, we propose a method to decompose the data in order to achieve parallel processing. Then, we introduce a parallel LDA model based on approximate distributed topic and the dynamic topic which can capture the dynamic characteristics of the topic. Finally, based on the topic modeling on large-scale social text, a parallel topic model DC-LDA based on data decomposition and clustering is proposed. The model utilizes existing parallel techniques to increase training speed and captures the dynamic nature of the topic while digging the topic. The experimental results show that DC-LDA can capture the appearance and demise of the topic more effective than DTM, also, it shows that DC-LDA is appropriate for modeling dynamic attribute text, and the execution time is shorter, so it can be applied to large-scale text.

References

[1] D.M. Blei, A.Y. Ng, and M.I. Jordan, Latent dirichlet allocation, Journal of Machine Learning Research, vol. 3, 2003, pp. 993-1022.

[2] A. Ahmed, and E.P. Xing, Timeline: a dynamic hierarchical dirichlet process model for recovering birth/death and evolution of topics in text stream, uncertainty in artificial intelligence, 2010, pp. 20-29.

[3] W.R. Gilks, Markov Chain Monte Carlo, Springer New York, 1999.

[4] D.M. Blei, and J.D. Lafferty, Dynamic topic models, 2006, pp. 113-120.

[5] L. Hong, and B.D. Davison, Empirical study of topic modeling in Twitter, Book Empirical study of topic modeling in Twitter, Series Empirical study of topic modeling in Twitter, ed., Editor ed., 2010, pp. 80-88.

[6] D. Ramage, S.T. Dumais, and D.J. Liebling, Characterizing Microblogs with Topic Models, 2010.

[7] Y. Wang, H. Bai, M. Stanton, W. Chen, and E.Y. Chang, PLDA: Parallel Latent Dirichlet Allocation for Large-Scale Applications, 2009, pp. 301-314.

[8] Z. Liu, Y. Zhang, E.Y. Chang, and M. Sun, PLDA+: Parallel latent dirichlet allocation with data placement and pipeline processing, ACM Transactions on Intelligent Systems and Technology, vol. 2, no. 3, 2011, pp. 26.

[9] R. Mehrotra, S. Sanner, W.L. Buntine, and L. Xie, Improving LDA topic models for microblogs via tweet pooling and automatic labeling., 2013, pp. 889-892.

[10] H.M. Wallach, D.M. Mimno, and A. Mccallum, Rethinking LDA: Why priors matter, Advances in Neural Information Processing Systems, vol. 23, 2009, pp. 1973-1981.

[11] C.M. Bishop, Pattern recognition and machine learning, New York NY, vol. 16, no. 4, 2006, pp. 140-155.

[12] L. Hong, and B.D. Davison, Empirical study of topic modeling in Twitter, Book Empirical study of topic modeling in Twitter, Series Empirical study of topic modeling in Twitter, ed., 2010, pp. 80-88.

[13] I. Bordino, C. Castillo, D. Donato, and A. Gionis, Query similarity by projecting the query-flow graph, 2010, pp. 515-522.
[14] X. Yan, J. Guo, Y. Lan, and X. Cheng, A biterm topic model for short texts, international world wide web conferences, 2013, pp. 1445-1456.

[15] D.M. Blei, and S.M. Gerrish, Code for estimating dynamic topic models, Book Code for estimating dynamic topic models, Series Code for estimating dynamic topic models ed., 2011.

[16] H.M. Wallach, I. Murray, R. Salakhutdinov, and D. Mimno, Evaluation methods for topic models, Book Evaluation methods for topic models, Series Evaluation methods for topic models ed., 2009, pp. 1105-1112.