A Study of Latent Dirichlet Allocation using MapReduce

Luojie Xiang
Department of Computer Science
Purdue University
West Lafayette, Indiana, USA
Email: xiang7@purdue.edu

Junchao Yan
Department of Computer and Information Technology
Purdue University
West Lafayette, Indiana, USA
Email: yan114@purdue.edu

***Abstract*—Topic models, as a powerful tool to discover underlying topics, have been widely applied in many areas such as scientific texts, twitters, online reviews, blog posts, emails, and newspaper. One of the most popular topic models called Latent Dirichlet Allocation (LDA) has attracted tremendous interests. However, the large scale data might limit the use of LDA due to the expensive computations. Meanwhile, Hadoop is an open source software for processing large scale data on computer clusters. It provides a programming paradigm called MapReduce that allows researchers easily write applications to run on the clusters. In this project, we studied a parallelized LDA algorithm using Mapreduce Framework. In addition, we ran an experiment on the cluster using the LDA to find interesting topics of Stackoverflow, which is a website for technical Q&A.**

I. Introduction

With the exponential growth of data, it would be efficient for people to understand an area by extracting the topics from millions of documents. Latent Dirichlet Allocation (LDA) is a statistical model that discovers underlying topics from a collection of documents [5]. LDA assumes that the documents are generated from multiple topics, of which a topic is an distribution over a fixed size of words. For each document, it contains the topics with different proportions. Therefore, the words in the document are actually generated from several distributions of the topics. For example, a document might include topics of hadoop and machine learning, therefore it is not reasonable to treat the document as a single topic [18]. LDA has been used in various applications including scientific texts [5], [8], twitters [28], [9], online reviews [21], blog posts [26], emails [12], and newspaper [24]. Furthermore, many variants of LDA have been proposed including Hierarchical Topic Models (HLDA) [7], [1], Supervised Topic Models (sLDA) [4], Labeled LDA (LLDA) [15], Correlated Topic Models (CTM) [10], [3], Dynamic Topic Model [2] and so on. However, the large scale data might limit the use of LDA due to the expensive computations. Meanwhile, Hadoop is an open source software for processing large scale data on computer clusters. It provides a programming paradigm called MapReduce that allows researchers easily write applications to run on the clusters.

To improve the scalability of LDA, a parallelized LDA algorithm was in Mapreduce Framework, which used variational inference rather than Gibbs sampling for approximation [27]. Mahout, which is an open source software for scalable machine learning, implements Collapsed Variational Bayes (CVM) algorithm that takes the advantage of both Variational Bayes and Gibbs Sampling for LDA using the Mapreduve paradigm [6]. In this project, we extracted the topics from a corpus collected from Stack Overflow using the parallelized LDA on a hadoop cluster.

II. Latent Dirichlet Allocation

Denote K as the number of topics, V as the size of the vocabulary, $\overrightarrow{\alpha}$ as a positive vector, and η as a scalar. Therefore, for each topic, its distribution over the vocabulary is

$$\overrightarrow{\beta}_k \sim Dir_V(\eta) \tag{1}$$

For each document, its distribution is a mixture of topics, which can be given as

$$\overrightarrow{\theta}_d \sim Dir(\overrightarrow{\alpha}) \tag{2}$$

In addition, for each word,

$$Z_{d,n} \sim Mult(\overrightarrow{\theta}_d), Z_{d,n} \in \{1, \ldots, K\} \tag{3}$$

$$W_{d,n} \sim Mult(\overrightarrow{\beta}_{z_{d,n}}), W_{d,n} \in \{1, \ldots, V\} \tag{4}$$

A graphical model of LDA is shown in Figure 1. In addition, LDA is a generative model, which provides a joint distribution over observations and hidden variables. Given a collection of documents (D), the posterior distribution of the hidden variables is

$$p(\overrightarrow{\theta}_{1:D}, z_{1:D,1:N}, \overrightarrow{\beta}_{1:K} | w_{1:D,1:N}, \alpha, \eta) = \frac{p(\overrightarrow{\theta}_{1:D}, \overrightarrow{z}_{1:D}, \overrightarrow{\beta}_{1:K} | \overrightarrow{w}_{1:D}, \alpha, \eta)}{\int_{\overrightarrow{\beta}_{1:K}} \int_{\overrightarrow{\theta}_{1:D}} \sum_z p(\overrightarrow{\theta}_{1:D}, \overrightarrow{z}_{1:D}, \overrightarrow{\beta}_{1:K} | \overrightarrow{w}_{1:D}, \alpha, \eta)} \tag{5}$$

Given the posterior distribution, the probability of a word based on topics $\hat{\beta}_{k,v}$, the proportion of topics in a document $\hat{\theta}_{d,k}$, and the topic assignment of a word $\hat{z}_{d,n,k}$ can be calculated as below.

$$\begin{aligned} \hat{\beta}_{k,v} &= E[\beta_{k,v} | w_{1:D,1:N}] \\ \hat{\theta}_{d,k} &= E[\theta_{d,k} | w_{1:D,1:N}] \\ \hat{z}_{d,n,k} &= E[Z_{d,n,k} | w_{1:D,1:N}] \end{aligned} \tag{6}$$

However, the distribution can not be solved in a polynomial time because of the integrals. Therefore, approximation methods are used to address this computational problem. Several approximation methods have been proposed for LDA including

TABLE I. A COMPARISON OF DIFFERENT APPROACHES FOR LDA

Approach	Framework	Inference
Mallet [13]	Multi-thread	Gibbs
GPU-LDA [25]	GPU	Gibss & Variational Bayesian
Async-LDA [17]	Multi-thread	Gibbs
N.C.L. [11]	Master-Slave	Variational Bayesian
pLDA [23]	MPI & MapReduce	Gibbs
Y!LDA [16]	Hadoop	Gibbs
Mahout [6]	MapReduce	Variational Bayesian
Mr. LDA [27]	MapReduce	Variational Bayesian

Gibbs sampling [19], mean field variational inference [5], collapsed variational inference [20], and expectation propagation [14]. In this study, we are focusing on the variational inference approach for LDA.

The the idea of mean field variational inference is to fit the variational parameters so that the variational distribution q can approximate the true posterior distribution.

$$q(\overrightarrow{\theta}_{1:D}, z_{1:D,1:N}, \overrightarrow{\beta}_{1:K}|w_{1:D,1:N}) = \prod_{k=1}^{K} q(\overrightarrow{\beta}_k|\overrightarrow{\lambda}_k) \prod_{d=1}^{D} \left(q(\overrightarrow{\theta}_{dd}|\overrightarrow{\gamma}_d) \prod_{k=1}^{K} q(z_{d,n}|\overrightarrow{\phi}_{d,n}) \right) \quad (7)$$

The difference between mean field variational distribution and true posterior distribution is that the variables of the former are independent, which are determined by different variational parameters. By minimize the Kullback-Leibler (KL) between variational distribution and true posterior distribution, we can get the fitted variational parameters.

$$\arg \min_{\overrightarrow{\lambda}_{1:K}, \overrightarrow{\gamma}_{1:D}, \overrightarrow{\phi}_{1:D,1:N}} KL(q(\overrightarrow{\theta}_{1:D}, z_{1:D,1:N}, \overrightarrow{\beta}_{1:K}|w_{1:D,1:N}) ||p(\overrightarrow{\theta}_{1:D}, z_{1:D,1:N}, \overrightarrow{\beta}_{1:K}|w_{1:D,1:N}, \alpha, \eta)) \quad (8)$$

The objective function is

$$\begin{aligned} L = &\sum_{k=1}^{K} E[\log p(\overrightarrow{\beta}_k|\eta)] + \sum_{d=1}^{D} E[\log p(\overrightarrow{\theta}_d|\overrightarrow{\alpha})] \\ &+ \sum_{d=1}^{D}\sum_{n=1}^{N} E[\log p(Z_{d,n}|\overrightarrow{\theta}_d)] \\ &+ \sum_{d=1}^{D}\sum_{n=1}^{N} E[\log p(w_{d,n}|Z_{d,n}, \overrightarrow{\beta}_{1:K})] + H(q) \end{aligned} \quad (9)$$

Coordinate Ascent algorithm is used to iteratively optimize the variational parameters until the objective function converges. A high level view of Coordinate Ascent algorithm is summarized [22].

III. MAPREDUCE FOR LATENT DIRICHLET ALLOCATION

As shown in Table I, a comparison of different approaches for LDA was summarized [27]. From the table, we can see that pLDA, Mahout, and Mr.LDA were implemented in the MapReduce paradigm.

For the implementation of Mahout, it regards the variational inference as a generation of Expectation Maximization (EM) algorithm for hierarchical Bayesian models. Therefore, for the E step, it infers the posterior probability of each topic for each word for all the documents in the collection. Then, it emits the statistics for each word in each topic. For the M step, it sums and normalizes the statistics, therefore we can get the distribution of the corpus for each topic. These two steps were implemented in Mapper and Reducer, respectively [6].

Algorithm 1 Coordinate Ascent Algorithm

Require: Initialize global parameters λ
  repeat
    for each topic k and word v **do**
      $\lambda_{k,v}^{(t+1)} = \eta + \sum_{d=1}^{D}\sum_{n=1}^{N} 1(w_{d,n} = v)\phi_{n,k}^{(t)}$
    end for
    for each document $d \in \{1, \ldots, D\}$ **do**
      $\gamma_{d,k}^{(t+1)} = \alpha_k + \sum_{n=1}^{N} \phi_{d,n,k}^{(t)}$
      for each word n **do**
        $\phi_{d,n,k}^{(t+1)} \propto \left\{ \Phi(\gamma_{d,k}^{(t+1)}) + \Phi(\lambda_{k,w_n}^{(t+1)}) - \Phi(lambda_{k,v}^{(t+1)}) \right\}$
      end for
    end for
    Update the parameters
    $\hat{\beta}_{k,v} = \frac{\lambda}{\sum_{v'=1}^{V} \lambda_{k,v'}}$
    $\hat{\theta}_{d,k} = \frac{\gamma_{d,k}}{\sum_{k'=1}^{K} \gamma_{d,k'}}$
    $\hat{z}_{d,n,k} = \hat{\phi}_{d,n,k}$
  until Convergence

Algorithm 2 Mapper

  Key - document ID $d \in \{1, \ldots, D\}$
  Value - document content
Require: Initialize global parameters λ
  for each topic k and word v **do**
    $\lambda_{k,v}^{(t+1)} = \eta + \sum_{d=1}^{D}\sum_{n=1}^{N} 1(w_{d,n} = v)\phi_{n,k}^{(t)}$
    Emit $\lambda_{k,v}^{(t+1)}$
  end for
  for each document $d \in \{1, \ldots, D\}$ **do**
    $\gamma_{d,k}^{(t+1)} = \alpha_k + \sum_{n=1}^{N} \phi_{d,n,k}^{(t)}$
    for each word n **do**
      $\phi_{d,n,k}^{(t+1)} \propto \left\{ \Phi(\gamma_{d,k}^{(t+1)}) + \Phi(\lambda_{k,w_n}^{(t+1)}) - \Phi(lambda_{k,v}^{(t+1)}) \right\}$
      Emit $\phi_{d,n,k}^{(t+1)}$
    end for
    Emit $\gamma_{d,k}^{(t+1)}$
  end for

Algorithm 3 Reducer

  Calculate
  $\hat{\beta}_{k,v} = \frac{\lambda}{\sum_{v'=1}^{V} \lambda_{k,v'}}$
  $\hat{\theta}_{d,k} = \frac{\gamma_{d,k}}{\sum_{k'=1}^{K} \gamma_{d,k'}}$
  $\hat{z}_{d,n,k} = \hat{\phi}_{d,n,k}$

IV. EXPERIMENT

A. Dataset

The dataset of this project is obtained from Kaggle (www.kaggle.com), which is a platform for data analysis and

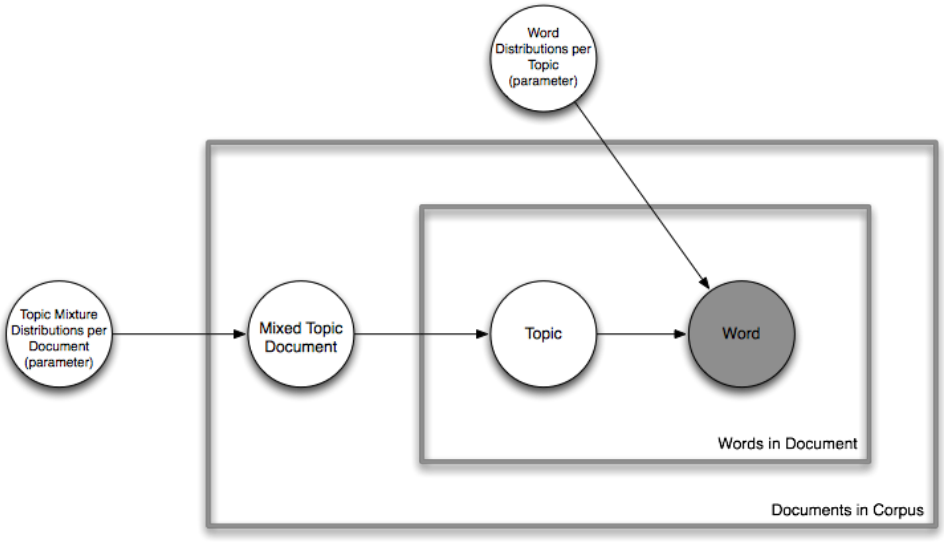

Fig. 1. A Graphical Model of LDA

prediction competitions. The data that we use are posted by Facebook for a keyword extraction competition. The dataset consists the files both for training and testing, of which the training file contains four attributes - id, title, body, and tags. In this project, only the title and body are used to extract the topics. A summary of the dataset is shown as below.

- Size: 7.3 GB
- Number of texts: 6,034,195
- Number of unique tags: 42048
- Top 10 tags:

```
c# 463526
java 412189
php 392451
javascript 365623
android 320622
jquery 305614
c++ 199280
python 184928
iphone 183573
asp.net 177334
```

Example:

id: 1
title: How to check if an uploaded file is an image without mime type?
content:
I'd like to check if an uploaded file is an image file (e.g png, jpg, jpeg, gif, bmp) or another file. The problem is that I'm using Uploadify to upload the files, which changes the mime type and gives a 'text/octal' or something as the mime type, no matter which file type you upload.
Is there a way to check if the uploaded file is an image apart from checking the file extension using PHP?

tags: php image-processing file-upload upload mime-types

B. Hadoop Setup

The hadoop cluster contains six nodes including one masternode and five slavenodes. To set up the hadoop cluster, we first configured the hosts file as shown below.

```
192.168.65.70 masternode
192.168.65.71 slavenode1
192.168.65.72 slavenode2
192.168.65.75 slavenode3
192.168.65.76 slavenode4
192.168.65.77 slavenode5
```

To configure the hadoop accordingly, we updated the `conf/masters` and `conf/slaves` files on the master node as shown below.

`conf/masters` on the master node:

```
masternode
```

`conf/slaves` one the master node:

```
slavenode1
slavenode2
slavenode3
slavenode4
slavenode5
```

In addition, configuration files `conf/core-site.xml`, `conf/mapred-site.xml`, and `conf/hdfs-site.xml` were modified on all the nodes as shown below.

`conf/core-site.xml` on all the nodes:

```
<configuration>
<property>
```

```
<name>fs.default.name</name>
<value>hdfs://masternode:9000</value>
<description>Enter your NameNode hostname
</description>
</property>
<property>
<name>fs.checkpoint.dir</name>
<value>/home/student/DAT500/fs/hdfs/snn
</value>
<description>A comma separated list of paths.
 Use the list of directories</description>
</property>
<property>
<name>hadoop.tmp.dir</name>
<value>/home/student/DAT500/fs/tmp</value>
<description>Comma separated list of paths
</description>
</property>
</configuration>
```

`conf/mapred-site.xml` on all the nodes:

```
<configuration>
<property>
<name>mapred.job.tracker</name>
<value>masternode:9001</value>
<description>Enter your JobTracker hostname
</description>
</property>
<property>
<name>mapred.local.dir</name>
<value>/home/student/DAT500/fs/tmp/mapred/
local</value>
<description>Comma separated list of paths
</description>
</property>
</configuration>
```

`conf/hdfs-site.xml` on all the nodes:

```
<configuration>
<property>
<name>dfs.name.dir</name>
<value>/home/student/DAT500/fs/hdfs/nn
</value>
<description>Comma separated list of paths
</description>
</property>
<property>
<name>dfs.data.dir</name>
<value>/home/student/DAT500/fs/hdfs/dn
</value>
<description>Comma separated list of paths
</description>
</property>
<property>
<name>dfs.replication</name>
<value>2</value>
</property>
</configuration>
```

C. Data Preprocessing

Due to the nature of the stack overflow platform, the posts come in various forms. Most posts has weird characters, huge amount of numbers and texts that's not natural language, representing mathematical formulations, program codes and also program or compiling outputs. For example, a lot of questions are asked about a certain compilation error or run time error, which very commonly contains a very long sequence of error report such as stack trace. These text do includes huge amount of text however, they're machine generated output and can be very confusing to the LDA training process. Therefore, we perform a certain steps of preprocessing before we actually run LDA. This not only prevent the unneccesary alements of the posts from confusing the training process but also reduce the data size tremendously.

- **Retrieve related fields.** The original data comes in a csv file with four fields for each record: id, title, body, tag. Since we want to find out the popular topics among the posts, we want to use the text-rich segments (title and body). Tag is essentially the topic in a sense (tags are mostly about what technology the post is asking like a certain programming language). This can be used as the gold standard later to evaluate the topics we found. Thus, we eliminate them for the training process.
- **Remove contents** in `<code></code>`. The `<code></code>` tag contains a lot of mathematical formulations, machine generated text etc. Since they're not helpful in finding topics, we eliminate them.
- **Remove tags.** Tags like `<p></p>` have a large frequency whereas they have no contribution to any topics. So they're all removed.
- **Remove punctuations.** Removing punctuations helps greatly with reducing the feature space. If punctuations are not removed, "happy", "happy." and "happy," would be regarded as different words whereas they should be the same.
- **Lowercase the text.** Lowercase the entire text also helps with reducing feature space since otherwise "happy" and "hapPy" would be regarded as different.
- **Remove newlines and excessive white spaces.**. Since the preprocessed text will be fed into mahout later and text is segmented into words by white space in mahout. Therefore, this would guarantee mahout segments words correctly.

D. Local Experiment

Local experiement is a sequential execution of all essential steps. There are three major steps:

- **Preprocessing.** Preprocessing the text with the steps described in the previous section. The output is a single file with individual documents taking one line in the file.
- **LDA input preparation.** Mahout's LDA takes a specific form of input. It comes with a utility to turn

TABLE II. RESULTS OF LDA FOR 10 TOPICS

Topic number	Top 10 words	Topic summary
1	time code run test start program process data memory thread	Program testing, measuring
2	page html php form http jquery javascript post url request	Web programing
3	file error code xml line python string script output files	N/A
4	windows server system version error directory install folder build linux	System programming
5	class object method function array call type return variable instance	OOP
6	number question set amp frac mathbb find int sum point	N/A
7	java server android apache eclipse http sun attrs hibernate jar	Java
8	data database table sql id query list row column mysql	Database
9	view image button text images display click background menu layout	HTML programming
10	user app application server api client google access facebook	App

TABLE III. RESULTS OF LDA FOR 20 TOPICS

Topic number	Topic words
1	server client network connection ip remote machine connect domain windows internet port access host set running servers address problem
2	application project xml web code android app library net build studio visual file system ui development source dll solution
3	class object method type function code instance objects call create called methods variable classes context properties variables reference created
4	image images star problem repository size pdf solution svn show git bar don add epsilon http photo png merge
5	button form user event click code list control tab item add selected menu text page change select input custom
6	frac mathbb sum amp attrs matrix function left times question show int vector points set space text order end
7	time process memory thread run application start running code device stop task mode app android threads long service play
8	page html php jquery javascript code http ajax content pages css js link post function url plugin working work
9	windows test install version run bit ve installed system running package cache dev rails os fine tests unit ubuntu
10	java org server apache sun impl core eclipse hibernate http jar invoke tomcat jersey postgresql rules loader lang method
11	table database sql data query id column mysql row tables db columns category order rows update insert server select
12	error code message service exception problem send call email fine works ve wrong string return mail case returns function
13	data time date store ve format save read customer correct location appreciated report json field sort point back convert
14	group node child number root set language add graph filter good book attributes list attribute xxxxxxxx tree lt english
15	question ve find make don good problem simple time lot found answer questions work solution bit idea thing edit
16	array string code function values number python variable loop key numbers simple characters output return find result input replace
17	file files command script directory error folder line program copy run output work path found python read doesn windows
18	user web app api request login users google site http facebook session access password application url create post browser
19	view method app controller model iphone set action remove length ios object null code failed mvc check nbsp problem
20	screen text layout code background left change color display problem top bar line content cell don width image size

a text file into sequence of entries and then process them further into numeric vectors which is the input form of LDA model training. Therefore, the output from the prepocessing step need to be turned into a form of one entry per physical file which is the input format of the utility function of mahout.

- **LDA model training.** After the input of LDA model training is prepared, it is inputed into mahout and an LDA model will be trained. A utility function is provided within mahout to provide transformation from numerical vectors into original words. We wrote a simple function to extract words with the highest scores in each topic and the result is shown in next section.

E. Hadoop Experiment

Hadoop experiment uses MapReduce to do all the three steps in local experiment.

For the preprocessing, since each document is preprocessed in exactly the same way and the preprocessing of each document is independent. Thus, this perfectly fits into MapReduce platform. We adapted the local preprocessing code to fit the MapReduce framework. A mapper takes in a chunck of text, splits it into entries and then preprocess each entry. A sampling is performed during emittion, the reason is explained in results session. We only emit the preprocessed entry by a predefined probability. The reducer putss each entry into a separate file.

For the LDA input preparation and LDA model training, MapReduce framework is already supported in mahout. The only thing difference is we use a different set of commands with hadoop.

F. Results

We ran the LDA on the preprocessed text. We do a sampling during preprocessing the text because using all the entries for training a LDA model would be too much both in terms of storage space and also time. Further, according to Blei etc. [5], LDA model performance plateaus with increasing training set size as all other machine learning models. They used 8000 data points and found out the model is trained sufficiently long before using the entire dataset. Therefore, to avoid over-training the model, we randomly sampled 4145 data entry for LDA model training. We present the result for 10 and 20 topics as shown in Table II and Table III, respectively.

V. CONCLUSION

In this project, we studied Latent Dirichlet Allocation using MapReduce. An experiment was conducted to discover the underlying topics of Stackoverflow using LDA. The results showed that LDA successfully extracted some interesting topics.

REFERENCES

[1] David M Blei, Thomas L Griffiths, and Michael I Jordan. The nested chinese restaurant process and bayesian nonparametric inference of topic hierarchies. *Journal of the ACM (JACM)*, 57(2):7, 2010.

[2] David M Blei and John D Lafferty. Dynamic topic models. In *Proceedings of the 23rd international conference on Machine learning*, pages 113–120. ACM, 2006.

[3] David M Blei and John D Lafferty. A correlated topic model of science. *The Annals of Applied Statistics*, pages 17–35, 2007.

[4] David M Blei and Jon D McAuliffe. Supervised topic models. *arXiv preprint arXiv:1003.0783*, 2010.

[5] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.

[6] Saikat Kanjilal David Hall. Latent dirichlet allocation, 2013.

[7] T Griffiths, M Jordan, J Tenenbaum, and David M Blei. Hierarchical topic models and the nested chinese restaurant process. *Advances in neural information processing systems*, 16:106–114, 2004.

[8] Thomas L Griffiths and Mark Steyvers. Finding scientific topics. *Proceedings of the National academy of Sciences of the United States of America*, 101(Suppl 1):5228–5235, 2004.

[9] Liangjie Hong and Brian D Davison. Empirical study of topic modeling in twitter. In *Proceedings of the First Workshop on Social Media Analytics*, pages 80–88. ACM, 2010.

[10] John D Lafferty and David M Blei. Correlated topic models. In *Advances in neural information processing systems*, pages 147–154, 2005.

[11] LE Mariote, CB Medeiros, and R da Torres. Parallelized variational em for latent dirichlet allocation: An experimental evaluation of speed and scalability. In *Data Mining Workshops, 2007. ICDM Workshops 2007. Seventh IEEE International Conference on*, pages 349–354. IEEE, 2007.

[12] Andrew McCallum, Andres Corrada-Emmanuel, and Xuerui Wang. Topic and role discovery in social networks. *Computer Science Department Faculty Publication Series*, page 3, 2005.

[13] Andrew Kachites McCallum. Mallet: A machine learning for language toolkit, 2002.

[14] Thomas Minka and John Lafferty. Expectation-propagation for the generative aspect model. In *Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence*, pages 352–359. Morgan Kaufmann Publishers Inc., 2002.

[15] Daniel Ramage, David Hall, Ramesh Nallapati, and Christopher D Manning. Labeled lda: A supervised topic model for credit attribution in multi-labeled corpora. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*, pages 248–256. Association for Computational Linguistics, 2009.

[16] Alexander Smola and Shravan Narayanamurthy. An architecture for parallel topic models. *Proceedings of the VLDB Endowment*, 3(1-2):703–710, 2010.

[17] Padhraic Smyth, Max Welling, and Arthur U Asuncion. Asynchronous distributed learning of topic models. In *Advances in Neural Information Processing Systems*, pages 81–88, 2008.

[18] Ashok Srivastava and Mehran Sahami. *Text mining: Classification, clustering, and applications*. CRC Press, 2010.

[19] Mark Steyvers and Tom Griffiths. Probabilistic topic models. *Handbook of latent semantic analysis*, 427(7):424–440, 2007.

[20] Yee W Teh, David Newman, and Max Welling. A collapsed variational bayesian inference algorithm for latent dirichlet allocation. In *Advances in neural information processing systems*, pages 1353–1360, 2006.

[21] Ivan Titov and Ryan McDonald. Modeling online reviews with multi-grain topic models. In *Proceedings of the 17th international conference on World Wide Web*, pages 111–120. ACM, 2008.

[22] Chong Wang. Variational inference: Mean field approximation, 2012.

[23] Yi Wang, Hongjie Bai, Matt Stanton, Wen-Yen Chen, and Edward Y Chang. Plda: Parallel latent dirichlet allocation for large-scale applications. In *Algorithmic Aspects in Information and Management*, pages 301–314. Springer, 2009.

[24] Xing Wei and W Bruce Croft. Lda-based document models for ad-hoc retrieval. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 178–185. ACM, 2006.

[25] Feng Yan, Ningyi Xu, and Yuan Qi. Parallel inference for latent dirichlet allocation on graphics processing units. In *Advances in Neural Information Processing Systems*, pages 2134–2142, 2009.

[26] Tae Yano, William W Cohen, and Noah A Smith. Predicting response to political blog posts with topic models. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 477–485. Association for Computational Linguistics, 2009.

[27] Ke Zhai, Jordan Boyd-Graber, Nima Asadi, and Mohamad L Alkhouja. Mr. lda: A flexible large scale topic modeling package using variational inference in mapreduce. In *Proceedings of the 21st international conference on World Wide Web*, pages 879–888. ACM, 2012.

[28] Wayne Xin Zhao, Jing Jiang, Jianshu Weng, Jing He, Ee-Peng Lim, Hongfei Yan, and Xiaoming Li. Comparing twitter and traditional media using topic models. In *Advances in Information Retrieval*, pages 338–349. Springer, 2011.