Frontiers of Information Technology & Electronic Engineering
www.zju.edu.cn/jzus; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Topic modeling for large-scale text data*

Xi-ming LI†1,2, Ji-hong OUYANG†‡1,2, You LU1,2

(1College of Computer Science and Technology, Jilin University, Changchun 130012, China)

(2MOE Key Laboratory of Symbolic Computation and Knowledge Engineering, Jilin University, Changchun 130012, China)

†E-mail: liximing86@gmail.com; ouyj@jlu.edu.cn

Received Oct. 15, 2014; Revision accepted Mar. 12, 2015; Crosschecked May 7, 2015

Abstract: This paper develops a novel online algorithm, namely moving average stochastic variational inference (MASVI), which applies the results obtained by previous iterations to smooth out noisy natural gradients. We analyze the convergence property of the proposed algorithm and conduct a set of experiments on two large-scale collections that contain millions of documents. Experimental results indicate that in contrast to algorithms named 'stochastic variational inference' and 'SGRLD', our algorithm achieves a faster convergence rate and better performance.

Key words: Latent Dirichlet allocation (LDA), Topic modeling, Online learning, Moving average
doi:10.1631/FITEE.1400352 **Document code:** A **CLC number:** TP391.1

1 Introduction

Hundreds of thousands of text documents are now readily available online. For example, the public article archive PubMed has more than 23 million records going back to 1966, and the popular web-based encyclopedia Wikipedia contains over 30 million articles in 285 languages, and thousands of new articles are created every day. Modeling and analyzing such large-scale data are significantly challenging.

Probabilistic topic modeling approaches (Blei, 2012) are mainstays for modern data analysis. Such models provide us with a visual language for expressing the hidden semantics of data (Hoffman *et al.*, 2013). During the past decade, latent Dirichlet allocation (LDA) (Blei *et al.*, 2003) has been paid more and more attention, and it has been acknowledged as one of the most successful topic modeling approaches.

The LDA model can analyze and explore documents by inferring the posterior distribution of hidden variables. To the best of our knowledge, standard inference methods include the Markov chain Monte Carlo (MCMC) algorithm (Andrieu *et al.*, 2003) and variational learning. The former is a kind of sampling approach, and the latter transforms the posterior inference problem into an optimization problem. For LDA, the representative MCMC algorithm is collapsed Gibbs sampling (Griffiths and Steyvers, 2004), and the representative variational learning algorithms are variational inference (VI) (Blei *et al.*, 2003) and collapsed VI (Teh *et al.*, 2007). However, these algorithms are under a batch learning framework. It means that, to update global parameters, the entire corpus must be accessed for each iteration. Intuitively, this framework limits LDA for analyzing very large data. For example, it is impractical to use these batch algorithms to analyze the two corpora of PubMed and Wikipedia.

We are interested in how to efficiently model large-scale text data, consisting of millions of documents. Most recently, some algorithms have been developed for the same goal. In Newman *et al.* (2009), Wang *et al.* (2009), Yan *et al.* (2009), and Liu *et al.* (2011), researchers used parallel

‡ Corresponding author

* Project supported by the National Natural Science Foundation of China (Nos. 61170092, 61133011, and 61103091)

ORCID: Xi-ming LI, http://orcid.org/0000-0001-8190-5087

computing to speed up the inference procedure of LDA. However, these methods require parallel hardware, which can be complicated and expensive. For example, for individual parallel hardware, e.g., GPU, this is time-consuming for very large data; for cloud infrastructure, the data communication might be expensive. Instead of parallel methods, some authors have tried to extend batch methods, e.g., MCMC and VI, to online methods (Song *et al.*, 2005; Canini *et al.*, 2009; Hoffman *et al.*, 2010; 2013; Patterson and Teh, 2013; Ranganath *et al.*, 2013; Wang *et al.*, 2013; Ye *et al.*, 2013; Ouyang *et al.*, 2014), which can be implemented on personal computers. For online MCMC approaches, Song *et al.* (2005) and Canini *et al.* (2009) used previously analyzed words to simplify the sampling of topic assignments; Patterson and Teh (2013) developed a scalable MCMC algorithm, namely stochastic gradient Riemannian Langevin dynamics (SGRLD), which asymptotically produces samples from the posterior distribution. For online VI approaches, Hoffman *et al.* (2010; 2013) proposed an online Bayesian variational inference algorithm, i.e., stochastic variational inference (SVI), which updates the parameters of interest using the stochastic natural gradients; Ranganath *et al.* (2013) studied how to adaptively set the learning rate of SVI; Wang *et al.* (2013) constructed control variates to reduce the variance of stochastic natural gradients of SVI.

Empirically, online VI approaches provide a tiny advantage compared to online MCMC approaches (Hoffman *et al.*, 2010), and SVI is a representative online VI approach. As mentioned above, SVI uses stochastic natural gradients to update parameters. Unfortunately, due to the high dimensionality of text document data, the noise of stochastic natural gradients is commonly large. Following Tadić (2009) and Wang *et al.* (2013), this leads to slower convergence and worse performance. To improve SVI, we propose a moving average SVI (MASVI) algorithm, and apply it to the LDA model. MASVI takes full advantage of the results obtained by previous iterations to smooth out the noisy natural gradients. In this work, we analyze the convergence property of MASVI, and conduct extensive experiments to evaluate MASVI. Experimental results on two large corpora PubMed and Wikipedia indicate that MASVI outperforms the state-of-the-art online methods such as SVI and SGRLD.

2 Stochastic variational inference for latent Dirichlet allocation

In this section, we review SVI (Hoffman *et al.*, 2010; 2013) for LDA (Blei *et al.*, 2003). We begin with the introduction of LDA. Note that we do not differentiate scalars and vectors when dealing with the formalization of variables in this study.

2.1 Latent Dirichlet allocation

LDA is acknowledged as one of the most popular topic models, used to analyze discrete data, such as text document collections. It assumes that each document is described by a set of latent topics, where each topic is a multinomial distribution over words. For example, suppose that corpus $\boldsymbol{W}$ has two topics (i.e., t_1 and t_2) and three word types (i.e., w_1, w_2, and w_3). Topic t_1 is parameterized by $[w_1\,(0.1)\,, w_2\,(0.3)\,, w_3\,(0.6)]$ and topic t_2 is parameterized by $[w_1\,(0.8)\,, w_2\,(0.1)\,, w_3\,(0.1)]$. A document from $\boldsymbol{W}$ might be described by $[t_1\,(0.7)\,, t_2\,(0.3)]$ or any other topic proportions.

Suppose that there are in total K topics. The generative process of LDA (Fig. 1a) is as follows:

1. For each topic k, sample a distribution over words: $\phi_k \sim \text{Dirichlet}(\beta)$.

2. For each document d with N_d words:

(1) Sample a distribution over topics: $\theta_d \sim \text{Dirichlet}(\alpha)$;

(2) For each word $w_{d,n}$: (a) sample a topic $z_{d,n} \sim \text{Multinomial}(\theta_d)$, and (b) sample a word $w_{d,n} \sim \text{Multinomial}\big(\phi_{z_{d,n}}\big)$.

Its model parameters are defined as $\boldsymbol{U} = \{\alpha, \beta\}$, where α and β are Dirichlet priors. Given a D-size collection with a V-size vocabulary and observed words $w \triangleq w_{1:D}$, the latent variables are summarized as $\boldsymbol{H} = \Big\{\phi \triangleq \phi_{1:K},\ \theta \triangleq \theta_{1:D},\ z \triangleq z_{1:D}\Big\}$, where ϕ, θ, and z are topic-word distributions, document-topic distributions, and topic assignments, respectively. We want to estimate the posterior distribution of latent variables:

$$p\,(\boldsymbol{H}|w,\,\boldsymbol{U}) \propto \prod_{k=1}^{K} p\,(\phi_k|\beta) \prod_{d=1}^{D} p\,(\theta_d|\alpha) \prod_{n=1}^{N_d} p\,(z_{d,n}|\theta_d)\, p\left(w_{d,n}|\phi_{z_{d,n}}\right).$$

2.2 Batch variational inference

This posterior is intractable to compute in general. Batch VI (Blei *et al.*, 2003) is a popular

algorithm to compute its approximation. The basic idea behind batch VI is that it introduces a variational distribution parameterized by free variational parameters and then minimizes the Kullback-Leibler (KL) divergence between the variational distribution and true posterior: $\mathrm{KL}\left(q(\boldsymbol{H}|\boldsymbol{\Omega})||p\left(\boldsymbol{H}|w,\boldsymbol{U}\right)\right)$. For LDA, batch VI defines the variational distribution by removing the coupling edges and nodes (Fig. 1b):

$$q(\boldsymbol{H}|\boldsymbol{\Omega})=\prod_{k=1}^{K}q(\phi_k|\tilde{\beta}_k)\prod_{d=1}^{D}q(\theta_d|\tilde{\alpha}_d)\prod_{n=1}^{N_d}q(z_{d,n}|\tilde{\theta}_{d,n}),$$

where $\boldsymbol{\Omega}$ (i.e., $\left\{\tilde{\alpha}\triangleq\tilde{\alpha}_{1:D},\ \tilde{\beta}\triangleq\tilde{\beta}_{1:K},\ \tilde{\theta}\triangleq\tilde{\theta}_{1:D}\right\}$) is the variational parameter, $\tilde{\alpha}$ and $\tilde{\beta}$ are Dirichlet parameters, and $\tilde{\theta}$ is the multinomial distribution parameter. The task of minimizing the KL divergence $\mathrm{KL}\left(q(\boldsymbol{H}|\boldsymbol{\Omega})||p\left(\boldsymbol{H}|w,\boldsymbol{U}\right)\right)$ can be transformed into the problem of maximizing the following function:

$$\mathcal{L}\left(\boldsymbol{\Omega}\right)\triangleq E_q\left[\log p\left(\boldsymbol{H},w|\boldsymbol{U}\right)\right]-E_q\left[\log q\left(\boldsymbol{H}|\boldsymbol{\Omega}\right)\right],\tag{1}$$

where $E_q\left[\cdot\right]$ is the expectation with respect to the variational distribution $q(\boldsymbol{H}|\boldsymbol{\Omega})$.

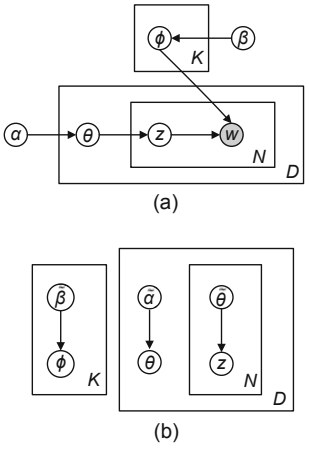

Fig. 1 Graphical model representations: (a) LDA; (b) variational distribution $q(\boldsymbol{H}|\boldsymbol{\Omega})$

Batch VI uses the expectation maximization (EM) framework to optimize the function $\mathcal{L}\left(\boldsymbol{\Omega}\right)$ with respect to the variational parameters $\boldsymbol{\Omega}$. At each EM iteration, it first fixes the global variational parameter $\tilde{\beta}$, and updates the two local variational parameters $\tilde{\alpha}$ and $\tilde{\theta}$ for each document. By setting the derivatives of $\tilde{\alpha}$ and $\tilde{\theta}$ to zero, the update rules are given as follows:

$$\tilde{\theta}_{d,n,k}\propto\exp\left(\Psi(\tilde{\alpha}_{d,k})+\Psi(\tilde{\beta}_{k,w_{d,n}})-\Psi\left(\sum_{v=1}^{V}\tilde{\beta}_{k,v}\right)\right),\tag{2}$$

$$\tilde{\alpha}_{d,k}=\alpha+\sum_{n=1}^{N_d}\tilde{\theta}_{d,n,k},\tag{3}$$

where $\Psi\left(\cdot\right)$ is the digamma function. After obtaining the optimal $\tilde{\alpha}$ and $\tilde{\theta}$ for all documents, batch VI updates the global variational parameter $\tilde{\beta}$ for each topic k by setting the corresponding derivative to zero. The update rule is as follows:

$$\tilde{\beta}_{k,v}=\beta+\sum_{d=1}^{D}\sum_{n=1}^{N_d}\tilde{\theta}_{d,n,k}w_{d,n}^{v},\ v\in\{1,2,\ldots,V\},\tag{4}$$

where

$$w_{d,n}^{v}=\begin{cases}1, & w_{d,n}=v,\\ 0, & \text{otherwise.}\end{cases}$$

Iterating this EM process until convergence, the approximate posterior of LDA can be obtained. Unfortunately, Eq. (4) shows that we have to compute the local parameter $\tilde{\theta}$ for all documents before updating the global variational parameter $\tilde{\beta}$. This drawback of batch VI leads to expensive computations for large-scale collections.

2.3 Stochastic variational inference

SVI (Hoffman *et al.*, 2010; 2013) uses the stochastic gradient descent algorithm to speed up the inference procedure of LDA. For each EM iteration, SVI uses stochastic gradients to update the global variational parameter $\tilde{\beta}$, instead of Eq. (4). Since stochastic gradients are formed by only a small subset of the corpus (i.e., mini-batch), SVI is efficient for large-scale data. Because the variational parameters are sufficient statistics, a better type of gradient is the natural gradient (Amari, 1998), which accounts for the information geometry of its parameter space using a Riemannian metric. If we draw only a single document d at iteration t, the corresponding stochastic natural gradient with respect to the global parameter $\tilde{\beta}$ is as follows:

$$\nabla^{(t)}\left(\tilde{\beta}_{k,v}^{(t-1)}\right)\triangleq-\tilde{\beta}_{k,v}^{(t-1)}+\beta+D\sum_{n=1}^{N_d}\tilde{\theta}_{d,n,k}^{(t-1)}w_{d,n}^{v}.\tag{5}$$

Because LDA assumes that documents are independent from each other, SVI still computes the

two local parameters $\tilde{\alpha}$ and $\tilde{\theta}$ using Eqs. (2) and (3). Under the framework of the stochastic gradient descent algorithm, given a decreasing learning rate ρ_t, SVI rewrites the update rule of $\tilde{\beta}$ as follows:

$$\tilde{\beta}_{k,v}^{(t)} = \tilde{\beta}_{k,v}^{(t-1)} + \rho_t \nabla^{(t)} \left(\tilde{\beta}_{k,v}^{(t-1)} \right). \tag{6}$$

3 The proposed algorithm

3.1 Moving average stochastic variational inference

SVI uses stochastic natural gradients to update global variational parameters $\tilde{\boldsymbol{\beta}}$. Unfortunately, due to the high dimensionality of $\tilde{\boldsymbol{\beta}}$ (i.e., $K \times V$), the noise of stochastic natural gradients will be significantly large. For example, suppose that corpus $\boldsymbol{W}$ has 2 MB documents and 7700 unique words (i.e., V=7700). If we fit a 100-topic LDA model, the dimension of $\tilde{\boldsymbol{\beta}}$ is 100×7700. If the mini-batch size is M=100, we have to use only 100 documents to form a 100×7700 stochastic natural gradient. The noise of this stochastic natural gradient is very large compared to the true 100×7700 natural gradient, formed by all 2 MB documents. More importantly, if too many word types are absent in these 100 documents, the noise will be larger. Such large noise might lead to slower convergence and worse performance (Tadić, 2009; Wang *et al.*, 2013). Taking a large mini-batch size M could reduce this noise. However, it can weaken the advantage in efficiency of SVI.

We argue that the random documents from different iterations commonly provide a lot of different words. To support this argument, we conduct some experiments on the corpora PubMed and Wikipedia (corpora details can be seen in Section 4). We randomly sample an M-size mini-batch of documents from the entire corpora 10 times, and then pairwise compare their recurrence rates of words. Table 1 shows the average recurrence rates for different M values. It is observed that the recurrence rates are commonly small, even for relatively large M values. Intuitively, averaging previous stochastic natural gradients formed by distinct documents at each iteration can take much more words in vocabulary, i.e., much more components of $\tilde{\boldsymbol{\beta}}$, into account. Based on this analysis, we propose MASVI for LDA. Reviewing the update process of SVI, we find that the noise is generated by the third term on the right side of Eq. (5). Thus, we average this term from different iterations. Define a parameter moving frequency R (in each iteration, MASVI constructs the stochastic gradient via moving the sufficient statistic obtained by the new mini-batch and R defines the frequency with respect to this movement). The MASVI algorithm can be described as follows: Given arbitrary R initial documents $d_0^1, d_0^2, \cdots, d_0^R$, an initial moving average $f^{(0)}$ is computed as $\sum_{r=1}^{R} \sum_{n=1}^{N_d} \tilde{\theta}_{d_0^r,n,k}^{(0)} w_{d_0^r,n}^v$. For any iteration $t \geq 1$, the moving average $f^{(t)}$ is defined as follows (a graphic illustration is shown in Fig. 2):

$$\begin{aligned} f^{(t)} = f^{(t-1)} &+ \sum_{n=1}^{N_d} \tilde{\theta}_{d_{t-1},n,k}^{(t-1)} w_{d_{t-1},n}^v \\ &- \sum_{n=1}^{N_d} \tilde{\theta}_{d_{\lfloor t-R \rfloor},n,k}^{(\lfloor t-R \rfloor)} w_{d_{\lfloor t-R \rfloor},n}^v, \end{aligned} \tag{7}$$

where $\lfloor t-R \rfloor$ equals 0 or $t-R$ depending on whether $t-R<0$, and $d_{\lfloor t-R \rfloor}$ is any remaining initial document when $t-R<0$. We use the following equation to replace the true stochastic natural gradient:

$$\widehat{\nabla}^{(t)} \triangleq -\tilde{\beta}_{k,v}^{(t-1)} + \beta + \frac{D}{R} f^{(t)}. \tag{8}$$

The update rule of $\tilde{\beta}$ in MASVI is as follows:

$$\tilde{\beta}_{k,v}^{(t)} = \tilde{\beta}_{k,v}^{(t-1)} + \rho_t \widehat{\nabla}^{(t)}. \tag{9}$$

Table 1 Average recurrence rates for different M values

M	Average recurrence rate (%)	
	PubMed	Wikipedia
20	5	6
50	7	9
100	18	23
150	23	27
200	28	29

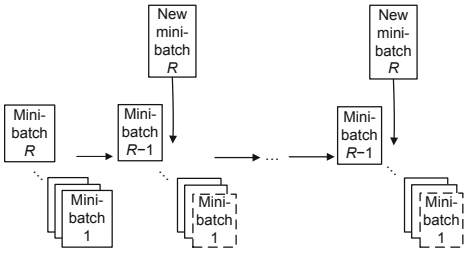

Fig. 2 A graphic illustration for the moving average scheme (at each iteration, we sample a new mini-batch R and discard old mini-batch 1)

Another benefit of MASVI is that it shares the same computational complexity with SVI. This advantage makes MASVI very practical for the true online data. We have formally described MASVI with a single document at each iteration. It easily generalizes to mini-batches, where we sample M documents from the corpus at each iteration. MASVI for LDA is summarized in Algorithm 1.

Algorithm 1 MASVI for LDA

Initialize parameters, including α, β, ρ, M, and R
Generate $\tilde{\beta}^{(0)}$, and then initialize $f^{(0)}$
For $t = 1, 2, \cdots, \infty$ **do**
  Sample M documents
  For d=1 to M **do**
    Compute $\tilde{\alpha}_d$ and $\tilde{\theta}_d$ using Eqs. (2) and (3)
  End for
  Update the moving average $f^{(t)}$ using Eq. (7)
  Update $\tilde{\beta}^{(t)}$ using Eq. (9)
End for

3.2 Analysis of convergence

From the perspective of stochastic optimization, the objective function of LDA, defined by Eq. (1), is non-convex. Proving convergence for a non-convex optimization problem is still an open problem.

As discussed in Section 3.1, we know that at each iteration the stochastic natural gradient involves only a few components of $\tilde{\boldsymbol{\beta}}$ that correspond to the words occurring in the sampled documents. Ideally, if in MASVI any R neighboring iterations can sample entirely disjoint documents (i.e., documents that contain distinct word types), our moving average scheme is very similar to using a larger size mini-batch to form the stochastic natural gradient. Although this assumption is too ideal for practical implementation, moving averages can be used to approximate the expectations within the stochastic optimization algorithm (Schaul *et al.*, 2013). For MASVI, we can obtain

$$
\begin{aligned}
&E\left[\widehat{\nabla}^{(t)}\right] = E\left[-\tilde{\beta}_{k,v}^{(t-1)} + \beta + \frac{D}{R} f^{(t)}\right] \\
&= -\tilde{\beta}_{k,v}^{(t-1)} + \beta + E\left[\frac{D}{R} f^{(t)}\right] \\
&\approx -\tilde{\beta}_{k,v}^{(t-1)} + \beta + E\left[D \sum_{n=1}^{N_d} \tilde{\theta}_{d_{t-1},n,k}^{(t-1)} w_{d_{t-1},n}^{v}\right] \\
&= E\left[\nabla^{(t)}\left(\tilde{\beta}_{k,v}^{(t-1)}\right)\right].
\end{aligned}
$$

The approximation (the third row) uses $f^{(t)} \approx R\sum_{n=1}^{N_d} \tilde{\theta}_{d_{t-1},n,k}^{(t-1)} w_{d_{t-1},n}^{v}$.

Following discussions in Hoffman *et al.* (2010), given a decreasing learning rate ρ_t, MASVI will almost certainly converge to a local optimum.

3.3 Related work

There exist some other modifications of SVI (Ranganath *et al.*, 2013; Wang *et al.*, 2013; Ouyang *et al.*, 2014). Ranganath *et al.* (2013) suggested an adaptive learning rate method for SVI. Wang *et al.* (2013) proposed a general framework (VRSGO) to deal with the ubiquitous noise to stochastic gradient optimization algorithms, and then applied it to LDA. VRSGO uses the low-order information to construct control variates, and then uses these control variates to reduce the variance of stochastic gradients. Although VRSGO is effective, control variates are sometimes difficult to compute, resulting in much more expensive computation for each iteration. MASVI also attempts to smooth out the noise generated by stochastic natural gradients. Different from VRSGO, it uses the existing results from previous iterations to average stochastic natural gradients. MASVI is more straightforward and efficient.

From the perspective of stochastic optimization, MASVI is in similar spirit to the incremental aggregated gradient (IAG) algorithm (Blatt *et al.*, 2007). For the purpose of smoothing out the noise generated by stochastic gradients, they integrated the stochastic knowledge from different iterations. However, there are some differences between the two algorithms: first, MASVI uses the moving averages with respect to the noisy term, i.e., the third term on the right side of Eq. (5), instead of the full stochastic gradient. This behavior makes the algorithm more straightforward to implement. Second, the traditional IAG algorithm assumes that the size of the corpus is known and the true gradient is computable. In contrast, MASVI focuses on the large-scale data, as well as online data, whose true gradient is almost unprocurable. This leads to the difficulty of convergence analysis. In Section 4, the experimental results show that MASVI converges to a local minimum in practice. Under this infinite assumption, proving convergence is an open problem, especially for non-convex problems such as LDA. To the best of our knowledge, there is little research on this aspect.

4 Evaluation

In this section, we evaluate the performance of MASVI for LDA. We first investigate the moving frequency R, and then compare MASVI with SVI and SGRLD.

4.1 Experimental setting

1. Dataset

Two large corpora were used in our experiments. For the PubMed corpus, we randomly selected 2 MB documents from the original PubMed collection (http://archive.ics.uci.edu/ml/datasets/Bag+of+Words). We removed the stop words and rare words, resulting in a vocabulary of 7000 words (i.e., the total occupancy of these 7000 words is above 95%). For the Wikipedia corpus, we randomly downloaded 2 MB documents from Wikipedia using the public implementation (http://www.cs.princeton.edu/∼mdhoffma/). Then we processed these documents using a vocabulary of 7700 words, which is an acknowledged standard vocabulary suggested in Hoffman *et al.* (2013). For both corpora, 2000 documents were randomly sampled from the entire collections for testing.

2. Evaluation method

Because topic models are acknowledged as Bayesian hierarchical models, the predictive performance can be evaluated by the probability that they assign to the test data. A popular metric is per-word likelihood (likelihood$_{\mathrm{pw}}$), which has been frequently used in Ranganath *et al.* (2013) and Wang *et al.* (2013). Unlike previous metrics, e.g., perplexity, evaluating the predictive distribution avoids comparing bounds or forming approximations of the evaluation metric (Hoffman *et al.*, 2013). Given a test document w_d, we partitioned w_d into two disjoint splits (in this paper, we provide a split of 50/50 observed/held-out): a set of observed words w_d^{obs} and held-out words w_d^{ho}. We then used the approximate topic-word distributions ϕ^* implied by the training data to estimate the predictive distribution $p\left(w_d^{\mathrm{ho}}|w_d^{\mathrm{obs}},\alpha,\phi^*\right)$. We computed the log probability of the words in w_d^{ho}. Averaging these log probabilities in the test data, we can obtain likelihood$_{\mathrm{pw}}$ as follows:

$$\text{likelihood}_{\mathrm{pw}} \triangleq \frac{\sum\limits_{d\in D_{\mathrm{test}}} \log p\left(w_d^{\mathrm{ho}}|w_d^{\mathrm{obs}},\alpha,\phi\right)}{\sum\limits_{d\in D_{\mathrm{test}}} \left|w_d^{\mathrm{ho}}\right|},$$

where $|\cdot|$ denotes the number of words. The distribution $p\left(w_d^{\mathrm{ho}}|w_d^{\mathrm{obs}},\alpha,\phi^*\right)$ for a K-topic LDA model can be approximately estimated as follows:

$$\begin{aligned}
&p\left(w_d^{\mathrm{ho}}|w_d^{\mathrm{obs}},\alpha,\phi^*\right)\\
&=\sum_{n=1}^{\left|w_d^{\mathrm{ho}}\right|}\int\left(\sum_{k=1}^{K}\theta_{d,k}\phi^*_{k,w_{d,n}^{\mathrm{ho}}}\right)p\left(\theta_d|w_d^{\mathrm{obs}},\alpha,\phi^*\right)\mathrm{d}\theta\\
&\approx\sum_{n=1}^{\left|w_d^{\mathrm{ho}}\right|}\int\left(\sum_{k=1}^{K}\theta_{d,k}\phi^*_{k,w_{d,n}^{\mathrm{ho}}}\right)q\left(\theta_d\right)\mathrm{d}\theta\\
&=\sum_{n=1}^{\left|w_d^{\mathrm{ho}}\right|}\sum_{k=1}^{K}E_q\left[\theta_{d,k}\right]\phi^*_{k,w_{d,n}^{\mathrm{ho}}}.
\end{aligned}$$

A higher value of likelihood$_{\mathrm{pw}}$ implies better performance.

4.2 Moving frequency R

The moving frequency R controls the number of documents used in each update process for global parameters. At the extreme, if we set R as D/M, MASVI approaches the batch VI algorithm in some degree. Theoretically, a large value of R can effectively reduce the noise of stochastic gradients. However, if R is too large, MASVI will be time-consuming. In this section, we empirically investigate the moving frequency R. Our goal is to suggest a reasonable setting for R in practice.

We fitted a 100-topic (i.e., K=100) LDA model, where $\alpha = 50/K$ and $\beta = 0.01$. According to discussions in Hoffman *et al.* (2013), we used the following learning rate, where the delay τ and forgetting rate κ were set as 1024 and 1, respectively:

$$\rho_t = (t+\tau)^{-\kappa}. \tag{10}$$

We fixed the mini-batch size M=100 or 500, and evaluated the performance with different R values over the set $\{2,4,8,16,32,64\}$. Fig. 3 shows the results. It is observed that larger R values perform better than smaller values. When M=100, for PubMed, the highest scores were obtained by R=32 and 64; for Wikipedia, the performances of R=16, 32, and 64 were almost the same, and they all significantly outperformed other values. When M=500, very similar trends can be observed. There was not an obvious difference between 32 and 64. Consequently, we suggest R=32 as the default setting.

In addition, we attempted to explain why R=32 is enough to obtain acceptable performance. At each

iteration, the stochastic natural gradient of MASVI is in fact generated by $R \times M$ documents. The quality of the stochastic natural gradient is dependent mainly on the 'difference' between these $R \times M$ documents and the entire corpus. The smaller 'difference' corresponds to smaller noise. We defined a concept of word frequency to roughly measure this 'difference', where for each word v, its word frequency is equal to its count divided by the total number of occurring words. Table 2 shows the word frequencies of five random words for different R values across Wikipedia under M=500 in a random iteration. Under this measurement, on the one hand, we observed that the 'differences' between R=2, 4, 8, 16 and the entire corpus are obviously larger than the 'differences' with respect to the settings R=32, 64; on the other hand, we found that the word frequencies under R=32 approach the true values. In our experiments, value 32 seems a critical point that provides acceptable performance.

Table 2 Word frequencies of five random words for different R values across Wikipedia under $M = 500$ in a random iteration

R	Frequency (%)				
	Cooking	Speedy	Partner	Like	Guard
2	0	0.57	0.28	0	0
4	0.06	1.80	0	0.02	0.12
8	0.73	0.04	0.58	0.22	0
16	0.53	0.04	0.01	1.47	0.35
32	0.11	0.01	0.08	0.45	0.05
64	0.03	0.03	0.11	0.69	0.09
True data	0.06	0.01	0.13	0.58	0.02

4.3 Topic modeling performance

We compared MASVI with two state-of-the-art algorithms, including an online MCMC algorithm, i.e., SGRLD (http://www.stats.ox.ac.uk/~teh/sgrld.html) (Patterson and Teh, 2013), and an online VI algorithm, i.e., SVI (Hoffman *et al.*, 2013). For the two baseline algorithms, we used their public codes, and set their parameters following the suggestions in the original papers. For MASVI, the moving frequency R was fixed at 32. We fitted a 100-topic LDA model, and independently ran all online LDA algorithms 10 times. The average results are shown in Fig. 4.

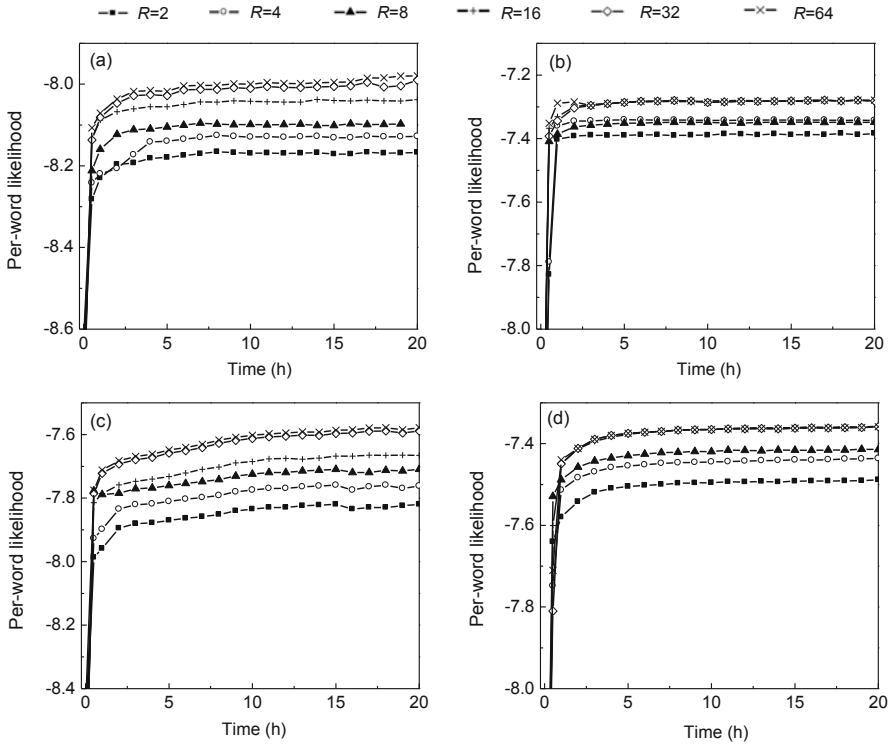

Fig. 3 Experiments on moving frequency R: (a) PubMed ($M = 100$); (b) Wikipedia ($M = 100$); (c) PubMed ($M = 500$); (d) Wikipedia ($M = 500$)

Obviously, MASVI performs better than the two baseline approaches. For SGRLD, MASVI is the winner in all cases, e.g., about 0.18 gain across PubMed and about 0.06 gain across Wikipedia. For SVI, MASVI also achieves better performance. This proves that our moving average scheme achieves a positive influence on smoothing out the large noise of stochastic natural gradients in SVI. The performance of MASVI is an improvement on both the online MCMC algorithm and the online VI algorithm. For a more convincing evaluation, we additionally conducted pairwise t-tests between MASVI and the baselines at the 0.05 significance level. As shown in Table 3, MASVI is statistically superior to baseline algorithms.

It is observed that MASVI converges faster than the two baseline algorithms in most cases. For example, for the PubMed corpus with M=100, MASVI converges after about 2 h, but SGRLD converges after about 5 h, and for the PubMed corpus with M=500, MASVI converges after about 3 h, but SVI converges after about 4 h. This further proves that using the moving average scheme effectively reduces the noise of stochastic gradient and leads to faster convergence speed.

5 Discussion and conclusions

In this paper, we have investigated how to efficiently and effectively infer the LDA model for large-scale text data. To approach this goal, we suggested an extension of the SVI algorithm, namely MASVI.

Table 3 The P-values from pairwise t-tests between MASVI and baseline algorithms

Dataset	P-value	
	SGRLD	SVI
PubMed ($M = 100$)	0.0012	0.0017
PubMed ($M = 500$)	0.0025	0.0120
Wikipedia ($M = 100$)	0.0028	0.0057
Wikipedia ($M = 500$)	0.0096	0.0025

All P-values are less than 0.05

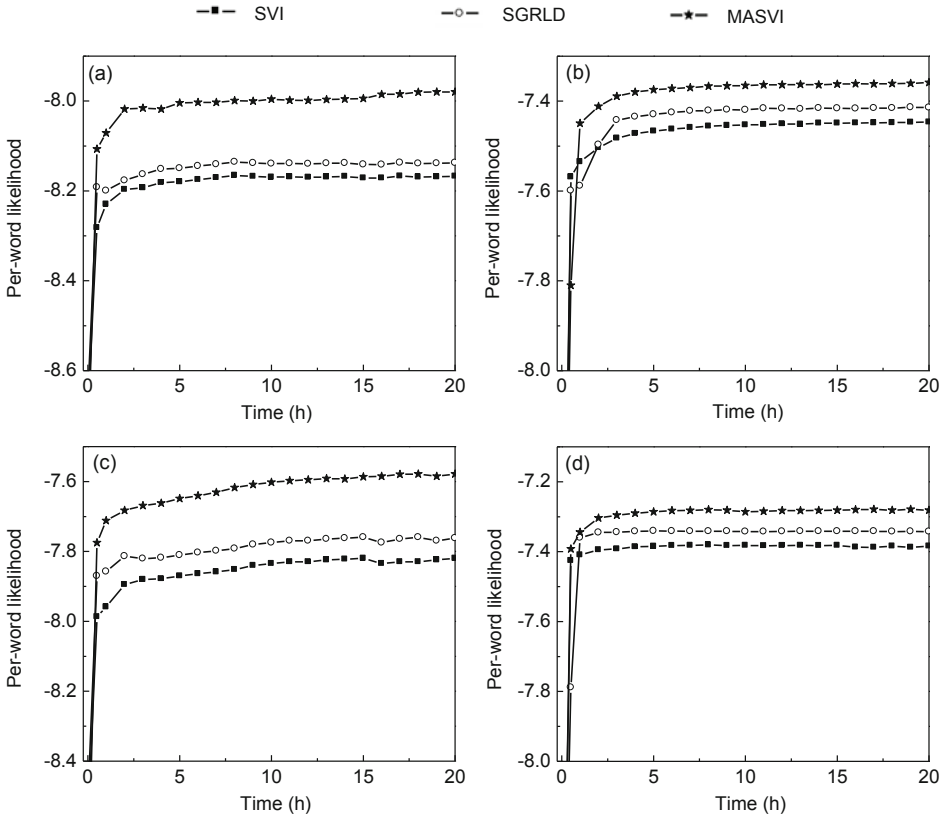

Fig. 4 A 100-topic LDA inference: (a) PubMed ($M = 100$); (b) Wikipedia ($M = 100$); (c) PubMed ($M = 500$); (d) Wikipedia ($M = 500$)

In this work, we used the results obtained by previous iterations to smooth out the noisy gradients. We defined a moving average variable to average the current stochastic natural gradient. We analyzed the convergence of MASVI and conducted extensive experiments on two large-scale collections, which contain millions of documents. The experimental results showed that MASVI achieves competitive performance with SVI and SGRLD.

Although the empirical results on corpora PubMed and Wikipedia indicate that MASVI performs well, we need further experiments to evaluate MASVI on other large corpora, especially for the parameter R. We found that R=32 was optimal for the used corpora, which follow relatively small vocabularies. However, whether this setting is suitable for a corpus with a large vocabulary is still unknown. In the future, we want to evaluate this problem and attempt to discuss the setting of R theoretically. In addition, we will attempt to apply MASVI to basic text analysis tasks, such as sentiment analysis. We are also interested in modeling the true online data with an infinite vocabulary.

References

Amari, S., 1998. Natural gradient works efficiently in learning. *Neur. Comput.*, **10**(2):251-276. [doi:10.1162/089976698300017746]

Andrieu, C., de Freitas, N., Doucet, A., *et al.*, 2003. An introduction to MCMC for machine learning. *Mach. Learn.*, **50**(1-2):5-43. [doi:10.1023/A:1020281327116]

Blatt, D., Hero, A.O., Gauchman, H., 2007. A convergent incremental gradient method with a constant step size. *SIAM J. Optim.*, **18**(1):29-51. [doi:10.1137/040615961]

Blei, D.M., 2012. Probabilistic topic models. *Commun. ACM*, **55**(4):77-84. [doi:10.1145/2133806.2133826]

Blei, D.M., Ng, A.Y., Jordan, M.I., 2003. Latent Dirichlet allocation. *J. Mach. Learn. Res.*, **3**:993-1022.

Canini, K.R., Shi, L., Griffiths, T.L., 2009. Online inference of topics with latent Dirichlet allocation. *J. Mach. Learn. Res.*, **5**(2):65-72.

Griffiths, T.L., Steyvers, M., 2004. Finding scientific topics. *PNAS*, **101**(suppl 1):5228-5235. [doi:10.1073/pnas.0307752101]

Hoffman, M., Bach, F.R., Blei, D.M., 2010. Online learning for latent Dirichlet allocation. Advances in Neural Information Processing Systems, p.856-864.

Hoffman, M., Blei, D.M., Wang, C., *et al.*, 2013. Stochastic variational inference. *J. Mach. Learn. Res.*, **14**(1):1303-1347.

Liu, Z., Zhang, Y., Chang, E.Y., *et al.*, 2011. PLDA+: parallel latent Dirichlet allocation with data placement and pipeline processing. *ACM Trans. Intell. Syst. Technol.*, **2**(3), Article 26.

Newman, D., Asuncion, A., Smyth, P., *et al.*, 2009. Distributed algorithms for topic models. *J. Mach. Learn. Res.*, **10**:1801-1828.

Ouyang, J., Lu, Y., Li, X., 2014. Momentum online LDA for large-scale datasets. Proc. 21st European Conf. on Artificial Intelligence, p.1075-1076.

Patterson, S., Teh, Y.W., 2013. Stochastic gradient Riemannian Langevin dynamics on the probability simplex. Advances in Neural Information Processing Systems, p.3102-3110.

Ranganath, R., Wang, C., Blei, D.M., *et al.*, 2013. An adaptive learning rate for stochastic variational inferencen. *J. Mach. Learn. Res.*, **28**(2):298-306.

Schaul, T., Zhang, S., LeCun, Y., 2013. No more pesky learning rates. arXiv preprint, arXiv:1206:1106v2.

Song, X., Lin, C.Y., Tseng, B.L., *et al.*, 2005. Modeling and predicting personal information dissemination behavior. Proc. 11th ACM SIGKDD Int. Conf. on Knowledge Discovery in Data Mining, p.479-488. [doi:10.1145/1081870.1081925]

Tadić, V.B., 2009. Convergence rate of stochastic gradient search in the case of multiple and non-isolated minima. arXiv preprint, arXiv:0904.4229v2.

Teh, Y.W., Newman, D., Welling, M., 2007. A collapsed variational Bayesian inference algorithm for latent Dirichlet allocation. Advances in Neural Information Processing Systems, p.1353-1360.

Wang, C., Chen, X., Smola, A.J., *et al.*, 2013. Variance reduction for stochastic gradient optimization. Advances in Neural Information Processing Systems, p.181-189.

Wang, Y., Bai, H., Stanton, M., *et al.*, 2009. PLDA: parallel latent Dirichlet allocation for large-scale applications. Proc. 5th Int. Conf. on Algorithmic Aspects in Information and Management, p.301-314. [doi:10.1007/978-3-642-02158-9_26]

Yan, F., Xu, N., Qi, Y., 2009. Parallel inference for latent Dirichlet allocation on graphics processing units. Advances in Neural Information Processing Systems, p.2134-2142.

Ye, Y., Gong, S., Liu, C., *et al.*, 2013. Online belief propagation algorithm for probabilistic latent semantic analysis. *Front. Comput. Sci.*, **7**(5):526-535. [doi:10.1007/s11704-013-2360-7]