Final project

Carlos Zanini, Mengjie Wang, Mingzhang Yin

1 Introduction

In this project, we implemented a stochastic variational inference (SVI) algorithm to the latent Dirichlet allocation model (LDA). This model is mainly used in the area of topic modeling. Topic models are generative models for collections of text documents based on underlying unknown topics. LDA is arguably one of the most commonly used models in this area. Usually, the main goals in topic modeling include identifying such underline (latent) topics as well as classifying the documents according to those topics.

According to LDA model, each topic is represented by a probability distribution over the entire aggregated vocabulary and each document has its own topic proportions, being strongly related to some of the topics, while less related to other ones.

We will illustrate the infenrence method to the NIPS set of documents publicaly available in

https://archive.ics.uci.edu/ml/datasets/Bag+of+Words .

The website also includes New York Times articles, PubMed abstracts and other datasets. The amount of documents in these collections vary from few thousands (e.g., NIPS dataset) to millions depending on the collection. These data sets consist of document index, unique word index and frequency of unique words in each document.

We also compare our SVI implementation for LDA with MCMC, highlighting the advantages on both.

2 Model

In latent Dirichlet allocation models, documents are generated in the following way: for each blank space in the document, we sample a topic from the list of available topics, and then we randomly pick a word using the probability distribution over words corresponding to the selected topic. The generative model is described bellow.

1. Draw K topics: $\boldsymbol{\beta}_k = (\beta_{k1}, ..., \beta_{kN}) \sim \text{Dirichlet}(\eta_1, ..., \eta_N)$, for $k \in \{1, ..., K\}$, where N is the number of unique words in the vocabulary;

2. For each document $d \in \{1, ..., D\}$,

   - Draw topic proportions: $\boldsymbol{\theta}_d = (\theta_{d1}, ..., \theta_{dK}) \sim \text{Dirichlet}(\alpha_1, ..., \alpha_K)$

- For each word blank space $n \in \{1, ..., N_d\}$ in document d,
  (a) Draw a topic assignment: $z_{dn} \sim \text{Multinomial}(\boldsymbol{\theta}_d)$

  (b) Draw a word from that topic: $w_{dn} \sim \text{Multinomial}(\beta_{z_{dn},1}, ..., \beta_{z_{dn},N})$

3 Inference

The joint distribution can be factorized as

$$p(\boldsymbol{\theta}, \boldsymbol{z}, \boldsymbol{w}, \boldsymbol{\beta}|\boldsymbol{\alpha}, \boldsymbol{\eta}) = p(\boldsymbol{\theta}|\boldsymbol{\alpha})p(\boldsymbol{z}|\boldsymbol{\theta})p(\boldsymbol{w}|\boldsymbol{z}, \boldsymbol{\beta})p(\boldsymbol{\beta}|\boldsymbol{\eta})$$

where $p(\boldsymbol{\theta}|\boldsymbol{\alpha})$ is Dirichlet distribution, $p(\boldsymbol{z}|\boldsymbol{\theta})$ follows Multinomial, $p(\boldsymbol{w}_n|\boldsymbol{z}_n, \boldsymbol{\beta})$ is Multinomial conditional on topic $\boldsymbol{z}_n$, and $p(\boldsymbol{\beta}|\boldsymbol{\eta})$ is exchangeable Dirichlet distribution. From the joint distribution, the full conditionals can be explicitly written as:

- Topic allocation per word:

$$p(z_{dn} = k \mid \boldsymbol{\theta}_d, \boldsymbol{\beta}_{1:K}, w_{dn}) \propto \exp\{\log \theta_{dk} + \log \beta_{k,w_{dn}}\}; \tag{1}$$

- Topic proportions per document:

$$p(\boldsymbol{\theta}_d \mid \boldsymbol{z}_d) = \text{Dirichlet}\left(\alpha_1 + \sum_{n=1}^{N_d} \mathbb{1}(z_{dn} = 1), ..., \alpha_K + \sum_{n=1}^{N_d} \mathbb{1}(z_{dn} = K)\right); \tag{2}$$

- Word distribution per topic:

$$p(\boldsymbol{\beta}_k \mid \boldsymbol{z}, \boldsymbol{w}) = \text{Dirichlet}\left(\eta_1 + \sum_{d=1}^{D}\sum_{n=1}^{N_d} \mathbb{1}(z_{dn} = k, w_{dn} = 1), ..., \eta_K + \sum_{d=1}^{D}\sum_{n=1}^{N_d} \mathbb{1}(z_{dn} = k, w_{dn} = N)\right) \tag{3}$$

3.1 MCMC

One way of doing the statistical inference for LDA model is using Gibbs sampling. MCMC is expected to give a better result in terms of accuracy in comparison with SVI, since MCMC converges to the true posterior distribution, while variational inference approximates it by a simpler class of distributions. However, the problem is if the sample space is huge, consequently, the Markov Chain will take too long to converge, since before each update of the global parameters $\boldsymbol{\lambda}$, we need to go through the entire corpse of documents (see the full conditional distributions in equations (1) (2) and (3)). One way to make the sampler faster is to use Collapsed Gibbs Sampler, which integrates out the θ_d, making the z_{dn} dependent. The *lda* package use this algorithm.

3.2 Variational Inference

The approximating distribution corresponding to the posterior is chosen from mean-field family which assumes q fully factorizes as

$$\begin{aligned} q(\boldsymbol{\beta}, \boldsymbol{z}, \boldsymbol{\theta} | \boldsymbol{\lambda}, \boldsymbol{\phi}, \boldsymbol{\gamma}) &= q(\boldsymbol{z}|\boldsymbol{\phi}) q(\boldsymbol{\theta}|\boldsymbol{\gamma}) q(\boldsymbol{\beta}|\boldsymbol{\lambda}) \\ &= \left(\prod_{d=1}^{D} \prod_{n=1}^{N_d} q(z_{dn} \mid \phi_{dn}) \right) \left(\prod_{d=1}^{D} q(\boldsymbol{\theta}_d \mid \boldsymbol{\gamma}_d) \right) \left(\prod_{k=1}^{K} q(\boldsymbol{\beta}_k \mid \boldsymbol{\lambda}_k) \right). \end{aligned}$$

Here the variational distribution is chosen to be in the same family as their corresponding complete conditional posterior. $q(\boldsymbol{z}|\boldsymbol{\phi})$ is Multinomial, $q(\boldsymbol{\theta}|\boldsymbol{\gamma})$ is Dirichlet and $q(\boldsymbol{\beta}|\boldsymbol{\lambda})$ is Dirichlet.

- $q(z_{dn}|\phi_{dn}) = \text{Multinomial}(\phi_{dn})$
- $q(\boldsymbol{\theta}_d|\boldsymbol{\gamma}_d) = \text{Dirichlet}(\boldsymbol{\gamma}_d)$
- $q(\boldsymbol{\beta}_k|\boldsymbol{\lambda}_k) = \text{Dirichlet}(\boldsymbol{\lambda}_k)$

The optimization objective is to minimize the Kullback–Leibler divergence between posterior distribution $q(\boldsymbol{\beta}, \boldsymbol{z}, \boldsymbol{\theta}|\boldsymbol{\lambda}, \boldsymbol{\phi}, \boldsymbol{\gamma})$and variational distribution $p(\boldsymbol{\theta}, \boldsymbol{z}, \boldsymbol{\beta}|\boldsymbol{w})$. The variable in optimization problem is the variational parameters $\boldsymbol{\lambda}, \boldsymbol{\phi}, \boldsymbol{\gamma}$. According to Hoffman et al. [2013], the minimizing problem is equivalent to maximizing the evidence lower bound(ELBO)

$$\mathcal{L}(\boldsymbol{\lambda}, \boldsymbol{\phi}, \boldsymbol{\gamma}) = \mathbf{E}_q[log(p(\boldsymbol{\theta}, \boldsymbol{z}, \boldsymbol{w}, \boldsymbol{\beta}))] - \mathbf{E}_q[log(q(\boldsymbol{\beta}, \boldsymbol{z}, \boldsymbol{\theta}))].$$

The vanilla variational inference algorithm updates the variational parameter $\boldsymbol{\phi}, \boldsymbol{\gamma}, \boldsymbol{\lambda}$ by using the whole corpus.

$$\phi_{ndk} \propto \exp\left\{ \psi\left(\gamma_{dk}\right) - \psi\left(\sum_k \gamma_{dk} \right) + \psi\left(\lambda_{k,w_{dn}}\right) - \psi\left(\sum_{n=1}^{N_d} \lambda_{kn} \right) \right\} \tag{4}$$

$$\boldsymbol{\gamma}_d = \boldsymbol{\alpha} + \sum_{n=1}^{N_d} \boldsymbol{\phi}_{nd} \tag{5}$$

$$\lambda_{kv} = \eta_{kv} + \sum_{d} \sum_{i:w_{di}=v} \phi_{dik} \tag{6}$$

The main problem with the updates of the vanilla variational inference scheme (see the set of equations above) is that we update the global parameters $\boldsymbol{\lambda}$ after running through the entire corpus of documents to update each $\boldsymbol{\phi}_{nd}$ and $\boldsymbol{\gamma}$. And what could be worse is that we have to wait a long time between the updates of $\boldsymbol{\lambda}$ thus providing low quality $\boldsymbol{\lambda}$ to the updates of local parameters for a long initial time. These behaviors make the uniformly sampled coordinate descent methods very slow and call for applying stochastic methods. This is particularly critical in the context of big data applications, where one usually have a huge amount of documents, since the algorithm will require many multiple passes through the entire dataset so that $\boldsymbol{\lambda}$ can converge.

4 Implementation of the mean field SVI algorithm

In this section, we summarize the steps of the variational inference algorithm for LDA and also discuss some useful details concerning its implementation.

The mean field SVI algorithm for LDA can be summarized by the following steps:

1. Initialize λ_{kn} and $\gamma_{dk},$ for all documents $d,$ topics k and words in the vocabulary $n.$

2. Sample one minibatch $\mathcal{M}$ of documents.

3. For all documents d in minibatch $\mathcal{M}$, update the local parameters until convergence:

$$\phi_{dnk} \propto \exp\left\{\psi\left(\gamma_{dk}\right) - \psi\left(\sum_{k} \gamma_{dk}\right) + \psi\left(\lambda_{k,w_{dn}}\right) - \psi\left(\sum_{n=1}^{N_d} \lambda_{kn}\right)\right\}, \tag{7}$$

   where the proportionality is over the topic index k;

$$\gamma_{dk} = \alpha_{dk} + \sum_{n=1}^{N_d} \phi_{dnk}. \tag{8}$$

4. Update the global parameters

$$\hat{\lambda}_{kn} = \eta_{kn} + \frac{D}{|\mathcal{M}|} \sum_{d \in \mathcal{M}} \sum_{i:w_{di}=n} \phi_{dik}. \tag{9}$$

$$\lambda_{kn}^{(new)} = (1-\rho_t)\lambda_{kn}^{(old)} + \rho_t \hat{\lambda}_{kn}. \tag{10}$$

5. Update the stepsize ρ_{t+1}.

6. Sample another minibatch of documents and iterate steps 3, 4 and 5 again.

First of all, notice that the local updates described in equations (7) and (8) are done multiple times for that same document. This is important because, otherwise, ϕ_{dnk} would be based solely on the initial values of the other parameters for all d, n and k during the first loop through the entire dataset. Although it is ideal to repeat these local updates until convergence, in practice we observed that repeating it 5 to 10 times is enough to update the local parameters reasonably well.

By making use of profile tools, we observed that the evaluations of digamma function ψ consist of a bottleneck in the algorithm. Therefore, it is important to precompute them to avoid unnecessary calculations of the same quantities.

Notice also that the update for ϕ_{dn_1k} and ϕ_{dn_2k} described in equation (7) are the same whenever the words w_{dn_1} and w_{dn_2} are equal, so it is more efficient to aggregate the local parameters ϕ_{dnk} so that n accounts for unique words in the document and keeping track of their observed frequency within the document.

It is fairly well known that SVI for LDA is an optimization problem that presents multimodality, so initialization becomes very important. For example, notice by equations (4), (5) and (6) that $\lambda_{kv} = \lambda^{(0)}$, $\phi_{dnk} = \phi_d^{(0)}$ and $\gamma_{dk} = \gamma_d^{(0)}$ for all $k \in \{1, ..., K\}$ is a local mode (the updates do not change the values of the variational parameters). To try to mitigate this issue, we run the algorithm with different initial values for the global parameters, following the suggestion in Hoffman et al. [2013], i. e. randomly sampling λ_{kn} from

$$\lambda_{kn} - \eta_{kn} \sim \text{Exponential}(D \times 100/(KN)).$$

Finally, it is worth mentioning that the second term in equation (7) could be dropped to the proportionality constant, since it does not depend on k. However, keeping it makes the algorithm more stable since it helps preventing large values inside of the exponential function.

5 Results

We applied the stochastic variational inference algorithm described previously to the NIPS dataset, which is publicly available at https://archive.ics.uci.edu/ml/datasets/Bag+of+Words . The NIPS data set contains papers from the NIPS conferences between 1987 and 1999 which includes 1500 papers already preprocessed. After tokenization and removal of stopwords, the vocabulary of unique words was truncated by only keeping words that occurred more than ten times. After preprocessing, the documents have, on average, about 500 unique words, consisting of about 12000 unique words in the whole vocabulary.

The code for implementation of svi was written in C++ and integrated to R through Rcpp [Eddelbuettel et al., 2011]. For the NIPS dataset, we chose to work with minibatches of ten documents. Within each minibatch, we update the local parameters $\boldsymbol{\phi}$ and $\boldsymbol{\gamma}$ five times and then update the global parameter $\boldsymbol{\lambda}$ after each local update in the minibatch. To access convergence, the full log likelihood is calculated once at each 50 documents. Under these conditions, the program could analyze about 100 documents per second, which would scale to 360,000 documents per hour. Such computational performance is not very far from what was obtained by Hoffman et al. [2013] (see for example figure 7, p.1328).

In the optimization process, after incorporating a minibatch of documents we obtain an updated version of variational parameters $\boldsymbol{\phi}$, $\boldsymbol{\gamma}$ and $\boldsymbol{\lambda}$. Plugging them into the variational distribution $q(z_{dn}|\boldsymbol{\phi}_{dn})$,$q(\boldsymbol{\theta}_d|\boldsymbol{\gamma}_d)$ and $q(\boldsymbol{\beta}_k|\boldsymbol{\lambda}_k)$, we can calculate the expected $\hat{\boldsymbol{z}}$, $\hat{\boldsymbol{\theta}}$ and $\hat{\boldsymbol{\beta}}$. Therefore, we get an approximated log likelihood for all observations w_{dn}

$$\text{Likelihood} = \sum_{d=1}^{D} \sum_{n=1}^{N_d} \log p(w_{dn}|\hat{\beta}_{z_{dn}})$$

To access convergence, we evaluate the log likelihood at every 50 documents, passing through the whole dataset 10 times, randomly picking three different initial values of the global parameters λ_{kn}. Figure 1 shows that the algorithm is sensitive to initialization, which is fairly often cited in variational inference literature. Notice that there is a drastic improvement of the log likelihood in the first loop through the data, which is due to the fact that the first loop contains variables that were not yet updated, which have lower likelihood.

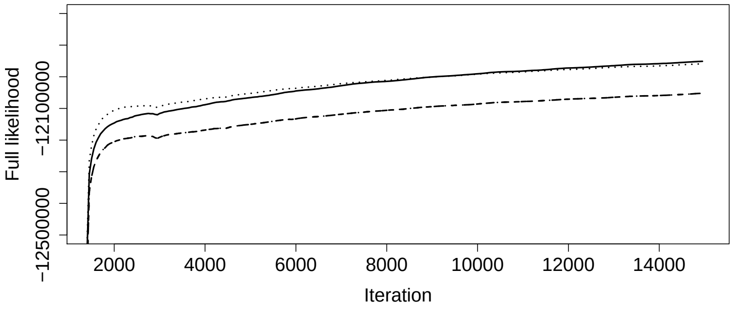

Figure 1: Full likelihood updating trace for 3 different initial points.

To see the improvements of the global parameter $\boldsymbol{\beta}$ during updating, we can calculate the likelihood of the first minibatch documents. Because its local parameter will not be changed by reading in other documents, its likelihood is influenced by the global parameter $\boldsymbol{\beta}$ alone thus reflecting the quality of estimated $\boldsymbol{\beta}$.

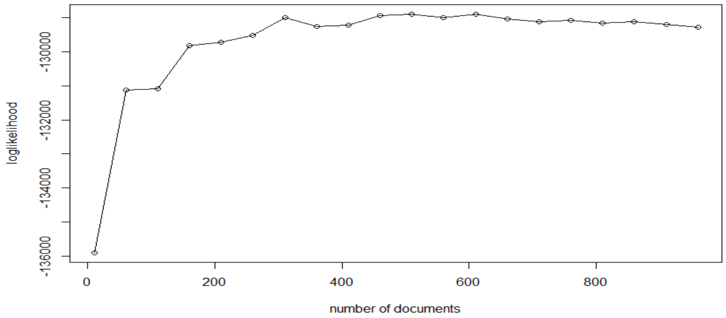

Figure 2: First minibatch likelihood updating trace

Ideally we want to calculate the optimization object ELBO. However to calculate ELBO explicitly requires calculate expectation with respect to $q(\boldsymbol{z}, \boldsymbol{\theta}, \boldsymbol{\beta})$ which is a very high dimensional joint distribution. This makes calculating the complete ELBO very difficult without referring to some sampling techniques. Instead we project the ELBO $\mathcal{L}(\boldsymbol{\lambda}, \boldsymbol{\phi}, \boldsymbol{\gamma})$ to $\mathcal{L}(\boldsymbol{\lambda}_k)$ which is the part of ELBO associated with global parameter $\boldsymbol{\lambda}_k$ for a certain topic k. A typical $\mathcal{L}(\boldsymbol{\lambda}_k)$ is shown below though there exist some fluctuation pattern for some other special topics.

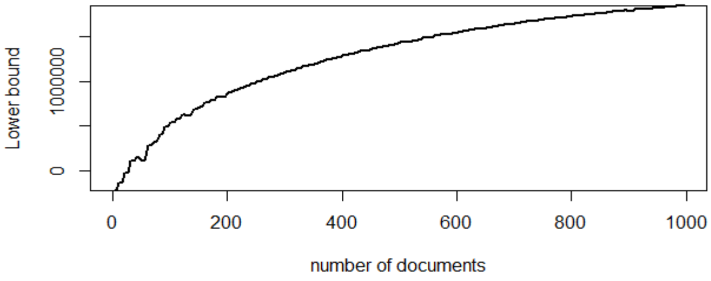

Figure 3: ELBO with respect to $\boldsymbol{\beta}_k$

Table 1 shows the estimation via SVI to the top ten words of each topic. It is possible to identify the subjects for most of the estimated topics. For example, topic 2 seems to refer to neural networks, topic 3 may be related to general keywords in math theory (e.g. theorem, algorithm, examples, proof), topic 7 contains different methods for dimension reduction (such as partial component analysis (PCA) and manifold, which are also related to support vector machines (SVM) - PCA, for example, is sometimes used as a way to initialize SVMs), Markov decision process (MDP) in topic 8, cellular motion in topic 14, brain activity in topic 16 and electrical circuits in topic 20.

To build table 1, we select the most frequent words in each topic by computing the term-score measure Blei and Lafferty [2009]

$$\text{score}(k, v) = \hat{\beta}_{k,v} \log \left(\frac{\hat{\beta}_{k,v}}{\left(\prod_{j=1}^{K} \hat{\beta}_{j,v} \right)^{1/K}} \right).$$

This measure selects the frequent words in the topic, but also penalize words that exhibit high frequency in all the topics.

Table 2 shows the result from *lda* package. The top ten words of each topic is given. It is clear enough to identify the subject of each individual topic, and they seems easy to be differentiated compared to each other. For example, topic 2 is image recognition; topic 5 is mixture model; topic 6 is pattern recognition; topic 7 is neuron network; topic 12 classification; topic 17 dimension deduction; topic 20 speech recognition.

Table 1: Ten most frequent words for each one of the 20 topics estimated by SVI.

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6	Topic 7	Topic 8	Topic 9	Topic 10
learning	neuron	bound	unit	dynamic	activity	data	model	word	object
action	neural	function	states	attractor	eye	mixture	solution	expert	system
control	network	distribution	network	equation	spatial	prediction	matrix	data	human
robot	system	theorem	task	connection	sound	svm	mdp	test	image
reinforcement	analog	algorithm	rules	field	response	space	unit	ica	images
trajectory	recurrent	number	adaboost	energy	visual	neural	markov	recognition	model
controller	simulation	examples	hidden	region	cue	pca	mean	set	module
dynamic	circuit	proof	validation	correlation	ocular	approximation	parameter	training	speech
agent	delay	probability	http	phase	critic	manifold	committee	task	face
path	hopfield	loss	www	cell	auditory	regression	term	performance	recognition

Topic 11	Topic 12	Topic 13	Topic 14	Topic 15	Topic 16	Topic 17	Topic 18	Topic 19	Topic 20
noise	representation	policy	cell	classifier	function	point	training	orientation	circuit
gaussian	memory	distribution	motion	classification	algorithm	image	network	spike	channel
density	entropy	optimal	visual	kernel	method	method	unit	fig	voltage
signal	probability	gaussian	motor	hmm	learning	model	error	cortex	chip
belief	weight	reward	direction	mlp	weight	local	hidden	cluster	frequency
posterior	probabilities	error	stimulus	nodes	blind	error	weight	routing	signal
component	binary	latent	movement	search	regression	map	character	cortical	current
detection	generative	policies	field	class	nonlinear	variance	recognition	feature	synapse
filter	bit	linear	firing	decision	optimization	vector	performance	preference	analog
variational	likelihood	sampling	velocity	set	gradient	dimensional	digit	map	vlsi

Table 2: Ten most frequent words for each one of the 20 topics estimated by MCMC.

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6	Topic 7	Topic 8	Topic 9	Topic 10
rules	image	function	set	model	character	unit	learning	neuron	signal
word	object	bound	network	data	recognition	network	error	spike	filter
string	images	theorem	algorithm	distribution	digit	input	weight	firing	information
language	pixel	network	function	gaussian	tangent	hidden	gradient	synaptic	noise
symbol	features	threshold	learning	likelihood	image	weight	equation	cell	channel
representation	recognition	proof	problem	parameter	pattern	training	noise	input	frequency
grammar	view	number	number	probability	handwritten	output	generalization	potential	auditory
structure	face	polynomial	neural	mixture	pixel	layer	order	synapses	source
rule	visual	result	result	bayesian	distance	learning	function	model	component
connectionist	feature	weight	model	prior	images	net	stochastic	membrane	sound

Topic 11	Topic 12	Topic 13	Topic 14	Topic 15	Topic 16	Topic 17	Topic 18	Topic 19	Topic 20
control	classifier	cell	tree	learning	chip	vector	network	hint	speech
model	training	model	node	action	circuit	space	system	schedule	word
system	data	visual	graph	policy	analog	data	neuron	song	recognition
motor	error	unit	nodes	reinforcement	input	dimensional	dynamic	instruction	network
movement	classification	direction	trees	function	neural	map	pattern	return	hmm
controller	class	stimulus	probability	optimal	vlsi	pca	memory	expert	training
arm	set	cortex	code	reward	current	clustering	attractor	market	system
trajectory	examples	response	bit	step	voltage	component	neural	stock	speaker
dynamic	test	field	genetic	algorithm	network	cluster	oscillator	play	context
position	decision	motion	matching	problem	implementation	projection	phase	ranking	acoustic

6 Discussion and future work

We implemented stochastic variational inference to latent Dirichlet allocation for analyzing topic models. The implementation follows the guidelines in [Hoffman et al., 2013], comparing the results with MCMC(takes about 352 seconds for 1000 iterations). We applied the three methods to the NIPs dataset, which contains 1500 documents and compare the results. We showed that the SVI algorithm was considerably faster than MCMC due to the fact that SVI does not need to go through all the documents before each update of the global parameters.

The main advantage of using SVI to make inference on LDA is its scalability to big datasets. Therefore, an essential future step for this work is to apply the algorithm to datasets containing much more documents. We tried to implement SVI for other documents in

https://archive.ics.uci.edu/ml/datasets/Bag+of+Words

but we could not get results due to numerical instability of the code, or need for further preprocessing of the documents. However, by extrapolating the execution time we observed for the NIPS dataset, we can have an idea of how fast the algorithm would run on bigger datasets, as long as the documents' size are similar to NIPS.

Since SVI it is an optimization method that uses gradient descent/ascent updates (by minimizing the KL divergence or, equivalently, maximizing the evidence lower bound), we can experiment using adaGrad for picking different stepsize ρ_t for different entries of λ. We only tested Robins-Monroe stepsize. We can also modify the Robins-Monroe step size by using Nesterov's accelerated gradient methods. Other more recent methods such as Nesterov's accelerated coordinate gradient descent method (ACDM) is potentially useful for LDA.

Another aspect that can be improved is to introduce the variance reduction tricks into the stochastic variational inference by using methods such as control variates, reparameterization, Rao-Blackwellization and others methods that are suited to problems where we cannot analytically compute the expectations that appear in the ELBO (Kingma and Welling [2013], [Mnih and Gregor, 2014], [Ranganath et al., 2014]). The variance reduction can improve the convergence rate. Also using variational families with more general dependence structures, for example neural networks [Mnih and Gregor, 2014], can achieve better approximation of the posterior distribution.

References

David M Blei and John D Lafferty. Topic models. *Text mining: classification, clustering, and applications*, 10(71):34, 2009.

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

Dirk Eddelbuettel, Romain François, J Allaire, John Chambers, Douglas Bates, and Kevin Ushey. Rcpp: Seamless r and c++ integration. *Journal of Statistical Software*, 40(8):1–18, 2011.

Matthew D Hoffman, David M Blei, Chong Wang, and John William Paisley. Stochastic variational inference. *Journal of Machine Learning Research*, 14(1):1303–1347, 2013.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Andriy Mnih and Karol Gregor. Neural variational inference and learning in belief networks. *arXiv preprint arXiv:1402.0030*, 2014.

Rajesh Ranganath, Sean Gerrish, and David M Blei. Black box variational inference. In *AISTATS*, pages 814–822, 2014.