Abstract

The field of statistical machine learning has seen a rapid progress in complex hierarchical Bayesian models. In Stochastic Variational Inference (SVI), the inference problem is mapped to an optimization problem involving stochastic gradients. While this scheme was shown to scale up to massive data sets, the intrinsic noise of the stochastic gradients impedes a fast convergence. Inspired by gradient averaging methods from stochastic optimization, we propose a variance reduction scheme tailored to SVI by averaging successively over the sufficient statistics of the local variational parameters. Its simplicity comes at the cost of biased stochastic gradients. We show that we can eliminate large parts of the bias while obtaining the same variance reduction as in simple gradient averaging schemes. We explore the tradeoff between variance and bias based on the example of Latent Dirichlet Allocation.

1 Introduction

Stochastic variational inference (SVI) lets us scale up Bayesian computation to massive data \[1\]. SVI has been applied to many types of models, including topic models \[1\], probabilistic factorization \[3\], statistical network analysis \[4, 5\], and Gaussian processes \[2\].

SVI uses stochastic optimization \[6\] to fit a variational distribution, following easy-to-compute noisy natural gradients that come from repeatedly subsampling from the large data set. As with most traditional stochastic optimization methods, SVI takes precautions to use unbiased, noisy gradients whose expectations are equal to the true gradients. This is necessary for the conditions of \[6\] to apply, and guarantees that SVI climbs to a local optimum of the variational objective. Innovations on SVI, such as subsampling from data non-uniformly \[3\] or using control variates \[18\], have maintained the unbiasedness of the noisy gradient.

In this paper, we explore the idea of following a biased stochastic gradients in SVI. We are inspired by the recent work in stochastic optimization that uses biased gradients. For example, stochastic averaged gradients (SAG) iteratively updates only a subset of terms in the full gradient \[15\]; averaged gradients (AG) follows the average of the sequence of stochastic gradients \[9\]. These methods lead to faster convergence on many problems.

However, SAG and AG are not immediately applicable to SVI. First, SAG requires storing all of the terms of the gradient. In most applications of SVI there is a term for each data point, and avoiding such storage is one of the motivations for using the algorithm. Second, the SVI update has a nice form where we update the variational parameter with a convex combination of the previous parameter and a new noisy version of it. This property falls out of the special structure of the gradient of the variational objective, and has the significant advantage of keeping the parameter in its feasible space. (E.g., the parameter may be constrained to be positive or even on the simplex.) Averaged
To understand our method, we must briefly explain the special structure of the SVI stochastic natural gradient. At any iteration of SVI, we have a current estimate of the variational parameter \( \lambda_i \), i.e., the parameter governing an approximate posterior that we are trying to estimate. First, we sample a data point \( w_i \). Then, we use the current estimate of variational parameters to compute expected sufficient statistics \( \hat{S}_i \) about that data point. (The sufficient statistics \( \hat{S}_i \) is a vector of the same dimension as \( \lambda_i \).) Finally, we form the stochastic natural gradient of the variational objective \( L \) with this simple expression:

\[
\nabla_{\lambda} L = \eta + N \hat{S}_i - \lambda_i,
\]

where \( \eta \) is a prior from the model and \( N \) is an appropriate scaling. This is an unbiased noisy gradient \([12, 11]\), and we follow it with a step size \( \rho_t \) that decreases across iterations \([6]\). Because of its algebraic structure, each step amounts to taking a weighted average,

\[
\lambda_{i+1} = (1 - \rho_i) \lambda_i + \rho_i (\eta + N \hat{S}_i).
\]

Note that this keeps \( \lambda_i \) in its feasible set.

With these details in mind, we can now describe our method. Our method replaces the natural gradient in Eq. (1) with a similarly constructed vector that uses a fixed-window moving average of the previous sufficient statistics. That is, we replace the sufficient statistics with an appropriate scaled sum, \( \sum_{j=0}^{L-1} \hat{S}_{i-j} \). Note this is different from averaging the gradients, which also involves the current iteration’s estimate.

We will demonstrate the many advantages of this technique. First, its computational cost is the same as for SVI and storage requirements only multiply by a constant factor (the window length \( L \)). Second, it enjoys significant variance reduction over the unbiased estimates, smaller bias than averaged gradients, and leads to smaller mean-squared error against the full gradient. Finally, we tested our method on latent Dirichlet allocation with two large corpora. We found it leads to faster convergence and better local optima.

**Related work** We first discuss the related work from the SVI literature. Both Ref. [19] and Ref. [13] introduce control variates to reduce the gradient’s variance. The method leads to unbiased gradient estimates. On the other hand, every few hundred iterations, an entire pass through the data set is necessary, which makes the performance and expenses of the method depend on the size of the data set. Ref. [10] deals with a method to pre-select documents according to their influence on the global update. For large data sets, however, it also suffers from high storage requirements. In the stochastic optimization literature, we have already discussed SAG [15] and AG [9]. Similarly, Ref. [8] introduces an exponentially fading momentum term. As mentioned before, all gradient averaging schemes require modification when applied to SVI, because they violate positivity constraints on the variational parameters.

## 2 Smoothed stochastic gradients for SVI

**Latent Dirichlet Allocation and Variational Inference** We start by reviewing stochastic variational inference for LDA \([11, 14]\), a topic model that will be our running example for SVI. Given
a corpus of \( D \) documents, we want to infer the hidden topics, defined as multinomial distributions over a vocabulary of size \( V \). The parameter \( \beta_{ij} \) describes the weight of word \( i \in \{1, \ldots, V\} \) in topic \( j \in \{1, \ldots, K\} \). Furthermore, each document \( d \) is associated with a normalized vector of topic weights \( \Theta_d \). A binary variable \( z_{dn} \in \{0, 1\} \) assigns the \( n \)th word in the \( d \)th document to topic \( k \) if it is 1. In the assumed generative process, each of the \( K \) topics is drawn from a Dirichlet distribution, \( \beta_k \sim \text{Dirichlet}(\alpha) \). Finally, for each word in the document, an assignment \( z_{dn} \sim \text{Multinomial}(\Theta_d) \) is drawn, and then the word is drawn from the assigned topic, \( w_{dn} \sim \text{Multinomial}(\beta_{z_{dn}}) \). The model has the following joint probability distribution:

\[
p = \prod_{i=1}^{K} p(\beta_i|\eta) \prod_{d=1}^{D} p(\Theta_d|\alpha) \prod_{n=1}^{N} p(z_{dn}|\Theta_d)p(w_{dn}|\beta_{1:K}, z_{dn})
\]  

(3)

Notably, the topics \( \beta \) will be referred to as global parameters, as they are shared among all the documents. The assignments \( z \) and topic proportions \( \Theta \) are local, as they characterize a single document. In variational inference, we approximate the posterior distribution,

\[
p(\beta, \Theta, z|w) = \frac{p(\beta, \Theta, z, w)}{\int d\beta d\Theta \sum_z p(\beta, \Theta, z, w)},
\]  

(4)

which is intractable to compute. We will use variational methods \[7\]. The full distribution (3) is approximated by a factored distribution,

\[
q(\beta, z) = q(\beta|\lambda) \prod_{d=1}^{D} \prod_{n=1}^{N} q(z_{dn}|\phi_{dn})
\]  

(5)

Here, \( q(\beta|\lambda) \) is a Dirichlet distribution and the \( q(z_{dn}|\phi_{dn}) \) are multinomials. The parameters \( \lambda \) and \( \phi \) of this variational distribution are fit so that the variational distribution is as close as possible to the true posterior; closeness being measured in terms of the Kullback-Leibler (KL) divergence \[1\]. As shown in Refs. \[11\]-\[13\], the objective to maximize is the evidence lower bound (ELBO),

\[
\mathcal{L}(q) = \mathbb{E}_q[p(x, z, \beta)] - \mathbb{E}_q[q(\beta, z)].
\]  

(6)

This is a lower bound on the marginal probability of the observations. It is a sensible objective function because, up to a constant, it is equal to the KL divergence between \( q \) and the posterior. Thus optimizing the ELBO with respect to \( q \) is equivalent to minimizing its KL divergence to the posterior.

In traditional variational methods, we iteratively update the local and global parameters. The local parameters are updated as described in \[11\]-\[13\]. They are a function of the global parameters, so at iteration \( i \) the local parameter is \( \phi_{dn}(\lambda_i) \). We are interested in the global parameters. They are updated based on the sufficient statistics \( S(\lambda_i) \),

\[
S(\lambda_i) = \sum_{d \in \{1, \ldots, D\}} \sum_{n=1}^{N} \phi_{dn}(\lambda_i) W_{dn}
\]  

(7)

\[
\lambda_{i+1} = \eta + S(\lambda_i)
\]

where \( W_{nd} \) is a binary \( K \)–vector such that \( W_{dn} = 1 \) implies that the \( n \)th word in document \( d \) is \( \nu \). Alternating updates finally lead to the convergence of the algorithm.

**Stochastic variational inference for LDA** The computation of the sufficient statistics is inefficient because it involves a pass through the entire data set. In Stochastic Variational Inference for LDA \[11\]-\[13\], it is therefore approximated by stochastically sampling a “minibatch” \( B_i \subset \{1, \ldots, D\} \) of \( |B_i| \) documents, computing \( S \) on the basis of the minibatch, and scaling the result appropriately,

\[
\hat{S}(\lambda_i, B_i) := \frac{D}{|B_i|} \sum_{d \in B_i} \sum_{n=1}^{N} \phi_{dn}(\lambda_i) W_{dn}
\]

Because it depends on the minibatch, \( \hat{S}_i = \hat{S}(\lambda_i, B_i) \) is now a random variable, and in the following we will be denoting variables that explicitly depend on the random minibatch \( B_i \) at the current time \( i \) by circumflexes, such as \( \hat{g} \) and \( \hat{S} \).
In SVI, the update on \( \lambda \) is done by carefully admixing the random estimate of the sufficient statistics to the current value of \( \lambda \). This involves a learning rate \( \rho_t < 1 \),

\[
\lambda_{t+1} = (1 - \rho_t) \lambda_t + \rho_t (\eta + \hat{S}(\lambda_t, B_t))
\]  

(8)

The case of \( \rho = 1 \) and \( |B_t| = D \) corresponds to batch variational inference (when sampling without replacement). Interestingly, this update can be associated to stochastic gradient descent, as a stochastic estimate of the natural gradient of the ELBO was shown \([11]\) to be

\[
\hat{g}(\lambda_t, B_t) = (\eta - \lambda_t) + \hat{S}(\lambda_t, B_t),
\]  

(9)

This interpretation opens the world of gradient smoothing techniques. Note that the above stochastic gradient is unbiased: its expectation value is the full gradient. However, it has a variance. The goal of this paper will be to reduce this variance at the expense of introducing a bias.

Algorithm 1: Smoothed stochastic gradients for Latent Dirichlet Allocation

Input: \( D \) documents, minibatch size \( B \), number of stored sufficient statistics \( L \), step width \( \rho_t \), hyperparameters \( \alpha, \eta \).

Output: Hidden variational parameters \( \lambda, \phi, \gamma \).

1. Initialize \( \lambda \) randomly and \( \hat{g}^t = 0 \).
2. Initialize empty queue \( Q = \{\} \).
3. for \( i = 0 \) to \( \infty \) do
4.     Sample minibatch \( B_i \subset \{1, \ldots, D\} \) uniformly.
5.     initialize \( \gamma \)
6.     repeat
7.         For \( d \in B_i \) and \( n \in \{1, \ldots, N\} \) set
8.             \( \phi_{dn}^k \propto \exp(\mathbb{E}[\log \Theta_{dk}] + \mathbb{E}[\log \beta_{k,w_d}]), \) \( k \in \{1, \ldots, K\} \)
9.         \( \gamma_d = \alpha + \sum_n \phi_{dn} \)
10. until \( \phi_{dn} \) and \( \gamma_d \) converge.
11. For each topic \( k \), calculate sufficient statistics for minibatch \( B_i \):
12. \( \hat{S}_t = \frac{D}{|B_i|} \sum_{d \in B_i} \sum_{n=1}^{N} \phi_{dn} W_{dn} \)
13. Add new sufficient statistic in front of queue \( Q \):
14. \( Q \leftarrow \{\hat{S}_t\} + Q \)
15. Remove last element when length \( L \) has been reached:
16. if \( \text{length}(Q) > L \) then
17.     \( Q \leftarrow Q - \{\hat{S}_{i-L}\} \)
18. end
19. Update \( \lambda \), using stored sufficient statistics:
20. \( \hat{S}_t^L \leftarrow \hat{S}_t^{L-1} + (\hat{S}_t - \hat{S}_{i-L}) / L \)
21. \( \hat{g}_t = (\eta - \lambda_t) + \hat{S}_t^L \)
22. \( \lambda_{t+1} = \lambda_t + \rho_t \hat{g}_t \).
23. end

Smoothed stochastic gradients for SVI  Noisy stochastic gradients can slow down the convergence of SVI or lead to convergence to sub-optimal local minima. Hence, we propose a smoothing scheme to reduce the variance of the noisy natural gradient. To this end, we average the sufficient statistics over the past \( L \) iterations. Here is a sketch:

1. Uniformly sample a minibatch \( B_i \subset \{1, \ldots, D\} \) of documents. Compute the local variational parameters \( \phi \) from a given \( \lambda_t \).
2. Compute the sufficient statistics \( \hat{S}_t = \hat{S}(\phi(\lambda_t), B_t) \).
3. Store \( \hat{S}_t \), along with the \( L \) most recent sufficient statistics. Compute \( \hat{S}_t^L = \frac{1}{L} \sum_{j=0}^{L-1} \hat{S}_{t-j} \) as their mean.
4. Compute the smoothed stochastic gradient according to

\[
\hat{g}_t^L = (\eta - \lambda_t) + \hat{S}_t^L
\]  

(10)
5. Use the smoothed stochastic gradient to calculate \( \lambda_{i+1} \). Repeat.

Details are given in algorithm 1. We now explore its properties. First, note that smoothing the sufficient statistics comes at almost no extra computational costs. In fact, the mean of the stored sufficient statistics does not explicitly have to be computed, but rather amounts to the update

\[
\hat{S}_i^L \leftarrow \hat{S}_{i-1}^L + (\hat{S}_i - \hat{S}_{i-L})/L
\]

(11)
after which \( \hat{S}_{i-L} \) is deleted. Storing the sufficient statistics can be expensive for large values of \( L \): In the context of LDA involving the typical parameters \( K = 10^2 \) and \( V = 10^4 \), using \( L = 10^2 \) amounts to storing \( 10^5 \) 64-bit floats which is in the Gigabyte range.

Note that when \( L = 1 \) we obtain stochastic variational inference (SVI) in its basic form. This includes deterministic variational inference for \( L = 1, B = D \) in the case of sampling without replacement within the minibatch.

**Biased gradients** Let us now investigate the algorithm theoretically. Note that the only noisy part in the stochastic gradient in Eq. (9) is the sufficient statistics. Averaging over \( L \) stochastic sufficient statistics thus promises to reduce the noise in the gradient. We are interested in the effect of the additional parameter \( L \).

Averaging comes at the cost of biased gradients. As the variational parameters change during each iteration, the mean of the \( L \) most recent estimates of the sufficient statistics deviates in expectation from its current value. In a nutshell, large values of \( L \) will reduce the variance but increase the bias.

To better understand this tradeoff, we need to introduce some notation. We defined the stochastic gradient \( g_i = \hat{g}(\lambda_i, B_i) \) in Eq. (9) and refer to \( g_i = \mathbb{E}_{B_i}[\hat{g}(\lambda_i, B_i)] \) as the full gradient (FG). We also defined the smoothed stochastic gradient \( \hat{g}_i^L \) in Eq. (10). Now, we need to introduce an auxiliary variable, \( g_i^L := (\eta - \lambda_i) + \frac{1}{L} \sum_{j=0}^{L-1} S_{i-j} \). This is the time-averaged full gradient. It involves the full sufficient statistics \( S(\lambda_i) \) evaluated along the sequence \( \lambda_1, \lambda_2, \ldots \) generated by our algorithm. The smoothed stochastic gradient can thus be written as follows:

\[
\hat{g}_i^L = \frac{g_i}{\text{FG}} + \frac{(g_i^L - g_i)}{\text{bias}} + \frac{(\hat{g}_i^L - g_i^L)}{\text{noise}}
\]

It decomposes into the full gradient, a bias term and a stochastic noise term. We will show that the optimal choice of the parameter \( L \) is determined by a tradeoff between variance and bias.

For the following analysis, we need to compute expectation values with respect to realizations of our algorithm, which is a stochastic process. Those expectation values shall in the following be denoted by \( \mathbb{E}[\cdot] \). Notably, not only the minibatches \( B_i \) are random variables under this expectation, but also the entire sequences \( \lambda_1, \lambda_2, \ldots \). Therefore, one needs to keep in mind that even the full gradients \( g_i(\lambda_i) \) are random variables and need to be studied under this expectation.

We find that the mean squared error of the smoothed stochastic gradient dominantly decomposes into a mean squared bias and a noise term:

\[
\mathbb{E}[(\hat{g}_i^L - g_i)^2] \approx \mathbb{E}[(\hat{g}_i^L - g_i)^2] + \mathbb{E}[(g_i^L - g_i)^2]
\]

(13)

---

1 To see this, consider the mean squared error of the smoothed stochastic gradient w.r.t the full gradient, \( \mathbb{E}[(\hat{g}_i^L - g_i)^2] \), adding and subtracting \( g_i^L \):

\[
\mathbb{E}[(\hat{g}_i^L - g_i^L + g_i^L - g_i)^2] = \mathbb{E}[(\hat{g}_i^L - g_i^L)^2] + 2\mathbb{E}[(\hat{g}_i^L - g_i^L)(g_i^L - g_i)] + \mathbb{E}[(g_i^L - g_i)^2].
\]

A priori, we encounter a cross-term, which we argue to be negligible. In defining \( \Delta \hat{S}_i = (\hat{S}_i - S_i) \) we find that

\[
(\hat{g}_i^L - g_i^L) = \frac{1}{L} \sum_{j=0}^{L-1} \Delta S_{i-j},
\]

and hence 2 \( \mathbb{E}[(\hat{g}_i^L - g_i^L)(g_i^L - g_i)] = \frac{2}{L} \sum_{j=0}^{L-1} \mathbb{E}[\Delta S_{i-j}(g_i^L - g_i)] \). As \( \Delta \hat{S}_i \) (the fluctuations of the sufficient statistics) is a random variable with mean zero, and the randomness of \( (g_i^L - g_i) \) enters only via \( \lambda_i \), one can assume a very small statistical correlation between those two terms, and hence the cross-term can be expected to be small. We confirmed this fact empirically in our numerical experiments, where the cross-term was at least two orders of magnitude smaller than bias and variance.
Figure 2: Empirical test of the variance-bias tradeoff on 2,000 abstracts from the Arxiv repository.
We show mean squared bias (left), variance (middle) and mean squared error (right) of the smoothed stochastic gradients, defined in Eq. (17), as a function of the iteration steps. Intermediate values of \( L \) lead to the smallest mean squared error and hence to the best tradeoff between small variance and small bias.

By construction, all bias comes from the sufficient statistics:
\[
E[(g^L_i - g_i)^2] = E \left[ \left( \frac{1}{L} \sum_{j=0}^{L-1} (S_{i-j} - S_i) \right)^2 \right].
\]
At this point, little can be said in general about the bias term, apart from the fact that it should shrink with the learning rate. We will explore it empirically in the next section. Concerning the variance term, averaging over the sufficient statistics yields
\[
E[(\hat{g}^L_i - g^L_i)^2] = E \left[ \left( \frac{1}{L} \sum_{j=0}^{L-1} \Delta \hat{S}_{i-j} \right)^2 \right] = \frac{1}{L^2} \sum_{j=0}^{L-1} E[(\hat{g}_{i-j} - g_{i-j})^2],
\]
which can be reformulated as \( \text{var}(\hat{g}^L_i) = \frac{1}{L^2} \sum_{j=0}^{L-1} \text{var}(\hat{g}_{i-j}) \). Assuming that the variance changes little during those \( L \) successive updates, we can approximate \( \text{var}(\hat{g}_{i-j}) \approx \text{var}(\hat{g}_i) \), which yields
\[
\text{var}(\hat{g}^L_i) \approx \frac{1}{L} \text{var}(\hat{g}_i),
\]
and arrive at a variance reduction of a factor \( L \). Hence, larger values of \( L \) will reduce the variance linearly in \( L \).

Bias-variance tradeoff To understand and illustrate the effect of \( L \) in our optimization problem, we used a small data set of 2000 abstracts from the Arxiv repository that allowed us to compute the full sufficient statistics for reference. More details on the data set and the corresponding parameters will be given below.

We computed mean squared bias (MSB), variance (VAR) and mean squared error (MSE) according to Eq. (17) for a single stochastic optimization run. More explicitly,
\[
\text{MSB}_i = \sum_{k=1}^{K} \sum_{v=1}^{V} (g^L_i - g_i)_{kv}^2, \quad \text{VAR}_i = \sum_{k=1}^{K} \sum_{v=1}^{V} (\hat{g}^L_i - g^L_i)_{kv}^2, \quad \text{MSE}_i = \sum_{k=1}^{K} \sum_{v=1}^{V} (\hat{g}^L_i - g_i)_{kv}^2.
\]
In Fig. 2, we plot those quantities as a function of iteration steps (time). As argued before, we arrive at a drastic variance reduction (middle panel) when choosing large values of \( L \). In contrast, the mean squared bias (left) typically increases with \( L \). It shows a complex time-evolution as it maintains memory of \( L \) previous steps. For example, the kinks in the bias curves (left panel of Fig. 2) occur at times 3, 10, 30, 100 and 300, i.e. they correspond to the values of \( L \) characterizing the corresponding stochastic gradient run. Those are the times from which on the smoothed gradient loses memory of its initial state, typically carrying a large bias. Also, the variance of the corresponding optimization runs becomes stationary at those times (middle panel). Those are the times where the initialization process ends and the queue \( Q \) in algorithm 1 has reached its maximal length \( L \). The right panel shows the mean squared bias, being affected both by variance and bias.

Due to the long-time memory of the optimization runs, one can associate some "inertia" or "momentum" to each value of \( L \). The larger \( L \), the smaller the variance and the larger the inertia. In
a non-convex optimization setup with many local minima as in our case, too much inertia can be harmful: the optimization run can be driven out of a local optimum. This effect can be seen for the $L = 100$ and $L = 300$ runs in Fig. 2, where the mean squared bias and error curves bend upwards at long times. Think of a marble rolling in a wavy landscape: having too much momentum it runs the danger of passing through the global minimum and eventually getting trapped in high-lying local minima. This picture suggests that the optimal value of $L$ depends on the “ruggedness” of the potential landscape of the optimization problem at hand. Our empirical study suggest that choosing $L$ between 10 and 100 produces the smallest mean squared error.

Aside: connection to gradient averaging Our algorithm was inspired by various gradient averaging schemes. However, we cannot easily used averaged gradients in SVI. To see the drawbacks of gradient averaging, let us consider $L$ stochastic gradients $\hat{g}_i, \hat{g}_{i-1}, \hat{g}_{i-2}, \ldots, \hat{g}_{i-L+1}$ and replace

$$\hat{g}_i \rightarrow \frac{1}{L} \sum_{j=0}^{L-1} \hat{g}_{i-j}. \quad (18)$$

in the coordinate update of SVI. One arrives at the following parameter update for $\lambda_i$:

$$\lambda_{i+1} = (1 - \rho_i)\lambda_i + \rho_i \left( \eta + \frac{1}{L} \sum_{j=0}^{L-1} \hat{S}_{i-j} - \frac{1}{L} \sum_{j=0}^{L-1} (\lambda_{i-j} - \lambda_i) \right) \quad (19)$$

The above update can lead to the violation of optimization constraints, namely to a negative variational parameter $\lambda$. Note that for $L = 1$ (the case of SVI), the third term is zero, guaranteeing positivity of the update. This is no longer guaranteed for $L > 1$, and the gradient updates will eventually become negative (we found this in practice). Furthermore, we find that there is an extra contribution to the bias compared to Eq. (14).

$$\mathbb{E}[(g_i^L - g_i)^2] = \mathbb{E} \left[ \left( \frac{1}{L} \sum_{j=0}^{L-1} (\lambda_i - \lambda_{i-j}) + \frac{1}{L} \sum_{j=0}^{L-1} (S_{i-j} - S_i) \right)^2 \right]. \quad (20)$$

Hence, gradient averaging carries an additional bias in lambda - it is the same term that may make our coordinate updates negative. In contrast, the variance term is the same as in sufficient statistics averaging. Compared to gradient averaging, the smoothed stochastic gradient has a smaller bias while profiting from the same variance reduction.

3 Empirical study

We tested SVI for LDA, using the smoothed stochastic gradients, on two large corpora:

- 882K abstracts from the Arxiv repository (ARX), using a vocabulary of 14K words.
- 1.7M articles from the New York Times (NYT), using a vocabulary of 8K words.

Our algorithm is an extension of Matthew Hoffman’s Python code for online LDA [22]. We set the minibatch size to $B = 300$ and furthermore set number of topics to $K = 100$, and the hyperparameters $\alpha = \eta = 0.5$. We fixed the learning rate to $\rho = 10^{-3}$ (we also compared our results to a decreasing learning rate and found the same behavior).

To illustrate the effect of running online LDA on the Arxiv corpus, we show the top 10 words from 8 out of 100 topics in Fig. 1. For a quantitative test of model fitness, we evaluate the predictive probability over the vocabulary [11]. To this end, we separate a test set from the training set. This test set is furthermore split into two parts: half of it is used to obtain the local variational parameters (i.e. the topic proportions, $\gamma^{(d)} = \mathbb{E}[\Theta^{(d)}] \gamma^{(d)}$) by fitting LDA with the fixed global parameters $\lambda$. The second part is used to compute the likelihoods of the contained words:

$$p(w_{new} | w_{old}, D) \approx \int \left( \sum_{k=1}^{K} \bar{\Theta}_k \beta_k \gamma^{(d)} \right) q(\Theta) q(\beta) d\Theta d\beta = \sum_{k=1}^{K} \bar{\beta}_k w_{new} \gamma^{(d)}_k. \quad (21)$$

where $\bar{\beta}_k = \mathbb{E}[\beta_k] = \lambda_k / |\lambda_k|$. We show the predictive probabilities as a function of time in Fig. 3 for the New York Times (left) and Arxiv (right) corpus, respectively. More precisely, time is measured in terms of effective passes through the data set (minibatch size $\times$ iterations / size of corpus). Within each plot, we compare different numbers of stored sufficient statistics, $L \in \{
Figure 3: Per-word predictive probability as a function of the effective number of passes through the data (minibatch size * iterations / size of corpus). We compare results for the New York Times (left) and Arxiv (right) data sets. Each plot shows data for different values of \( L \). We used a constant learning rate of \( 10^{-3} \), and set a time budget of 24 hours. Highest likelihoods are obtained for \( L \) between 10 and 100, after which strong bias effects set in.

\{1, 10, 100, 1000, 10000, \infty\}. The last value of \( L = \infty \) corresponds to a version of the algorithm where we average over all previous sufficient statistics, which is related to averaged gradients (AG), but which has a bias too large to compete with small and finite values of \( L \). The maximal values of 5 and 15 effective passes through the complete data set for the NYT and ARX data sets, respectively, correspond to a run time larger than 24 hours, which we set as a hard cutoff in our study.

Interestingly, highest held-out likelihoods are obtained for intermediate values of \( L \). E.g., averaging only over 10 subsequent sufficient statistics results in much faster convergence and higher likelihoods at very little extra storage costs. As discussed before, we attribute this fact to the best tradeoff between variance and bias for those runs.

4 Discussion and Conclusions

SVI scales up Bayesian inference, but suffers from noisy stochastic gradients. To reduce the mean squared error relative to the full gradient, we averaged the sufficient statistics of SVI successively over \( L \) iteration steps. The resulting smoothed gradient is, however, biased, and the performance of the method is governed by the competition between bias and variance. We argued theoretically and showed empirically that intermediate values of the number of stored sufficient statistics \( L \) give the highest held-out likelihoods.

Proving convergence for our algorithm is still an open problem, which is non-trivial especially because the variational objective is non-convex. To guarantee convergence, however, one can simply reduce the number of stored gradients to one close to the optimum, in which case SVI is recovered.

Acknowledgements We thank Laurent Charlin, Alp Kucukelbir, Prem Gopolan, Rajesh Ranganath, Linpeng Tang, Neil Houlsby, Marius Kloft, and Matthew Hoffman for stimulating and insightful discussions. We acknowledge financial support by NSF CAREER NSF IIS-0745520, NSF BIGDATA NSF IIS-1247664, NSF NEURO NSF IIS-1009542, ONR N00014-11-1-0651, the Alfred P. Sloan foundation, DARPA FA8750-14-2-0009 and the NSF MRSEC program through the Princeton Center for Complex Materials Fellowship (DMR-0819860).
References

[1] C. M. Bishop. Pattern Recognition and Machine Learning. Springer 2006.
[2] J. Hensman, N. Fusi, N. D. Lawrence. Gaussian Processes for Big Data. Uncertainty in Artificial Intelligence 2013
[3] P. Gopalan, J. Hofman, D.M. Blei. Scalable recommendation with Poisson factorization. arXiv preprint. http://arxiv.org/abs/1311.1704, 2013
[4] P. Gopalan and D. Blei. Efficient discovery of overlapping communities in massive networks. PNAS, 110 (36) 14534-14539, 2013.
[5] E M. Airoldi, D. Blei, S. E. Fienberg E. P. Xing. Mixed Membership Stochastic Blockmodels. The Journal of Machine Learning Research Volume 9, 6/1/2008 p. 1981-2014.
[6] H. Robbins and S. Monro. A stochastic approximation method. The Annals of Mathematical Statistics, 22(3):400407, 1951.
[7] M. J. Wainwright, M. I. Jordan. Graphical Models, Exponential Families, and Variational Inference. Now Publishers Inc, 2008.
[8] P. Tseng. An incremental gradient(-projection) method with momentum term and adaptive stepsize rule. SIAM Journal on Optimization, 8(2):506531, 1998.
[9] Y. Nesterov. Primal-dual subgradient methods for convex problems. Mathematical programming, 120(1):221259, 2009.
[10] M. Wahabzada, K. Kersting. Larger Residuals, Less Work: Active Document Scheduling for Latent Dirichlet Allocation. ECML PKDD 2011.
[11] M. Hoffman, D. Blei, J. Paisley, and C. Wang. Stochastic variational inference. Journal of Machine Learning Research, 14:1303-1347, 2013.
[12] M. Sato. Online model selection based on the variational Bayes. Neural Computation, 13(7):1649 1681, 2001.
[13] D. Blei, A. Ng, and M. Jordan. Latent Dirichlet allocation. Journal of Machine Learning Research, 3:993-1022, January 2003.
[14] M. Hoffman, D. Blei, and F. Bach. Online learning for latent Dirichlet allocation. Neural Information Processing Systems, 2010.
[15] M. Schmidt, N. Le Roux, F. Bach. Minimizing Finite Sums with the Stochastic Average Gradient. Technical report, HAL 00860051, 2013.
[16] F. Bach, E. Moulines. Non-Asymptotic Analysis of Stochastic Approximation Algorithms for Machine Learning. Advances in Neural Information Processing Systems (NIPS), 2011.
[17] F. Bach and E. Moulines. Non-strongly-convex smooth stochastic approximation with convergence rate O(1/n). Technical report, HAL 00831977, 2013. To appear in Advances in Neural Information Processing Systems (NIPS).
[18] C. Wang X. Chen, A. Smola E. Xing. Variance Reduction for Stochastic Gradient Optimization. Neural Information Processing Systems 2013.
[19] R. Johnson. T. Zhang: Accelerating Stochastic Gradient Descent using Predictive Variance Reduction. Neural Information Processing Systems 2013.
[20] D. Blei and J. Lafferty. Topic Models. In A. Srivastava and M. Sahami, editors, Text Mining: Classification, Clustering, and Applications . Chapman & Hall/CRC Data Mining and Knowledge Discovery Series, 2009.
[21] A. Nedic and D. Bertsekas. Convergence rate of incremental subgradient algorithms. In Stochastic Optimization: Algorithms and Applications, pages 263304. Kluwer Academic, 2000.
[22] M. Hoffman, online LDA code in Python, downloaded from https://www.cs.princeton.edu/blei/topicmodeling.html