A trust-region method for stochastic variational inference with applications to streaming data

Lucas Theis LUCAS@BETHGELAB.ORG
Centre for Integrative Neuroscience, Otfried-Müller-Str. 25, BW 72076 Germany

Matthew D. Hoffman MATHOFFM@ADOBE.COM
Adobe Research, 601 Townsend St., San Francisco, CA 94103 USA

Abstract

Stochastic variational inference allows for fast posterior inference in complex Bayesian models. However, the algorithm is prone to local optima which can make the quality of the posterior approximation sensitive to the choice of hyperparameters and initialization. We address this problem by replacing the natural gradient step of stochastic varitional inference with a trust-region update. We show that this leads to generally better results and reduced sensitivity to hyperparameters. We also describe a new strategy for variational inference on streaming data and show that here our trust-region method is crucial for getting good performance.

1. Introduction

Stochastic variational inference (SVI; Hoffman et al., 2013) has enabled variational inference on massive datasets for a large class of complex Bayesian models. It has been applied to, for example, topic models (Hoffman et al., 2010), nonparametric models (Wang et al., 2011; Paisley et al., 2012), mixture models (Hughes & Sudderth, 2013), and matrix factorizations (Gopalan et al., 2014). However, it has been observed that SVI can be sensitive to the choice of hyperparameters and is prone to local optima (Ranganath et al., 2013; Hughes & Sudderth, 2013; Hoffman & Blei, 2014). Successful attempts to improve its performance include the automatic tuning of learning rates (Ranganath et al., 2013) and variance reduction techniques (Mandt & Blei, 2014). The results of Hoffman & Blei (2014) suggest that local optima in the objective function caused by a mean-field approximation (Wainwright & Jordan, 2008) can be eliminated by means of more complex families of approximating distributions. They also find that the standard search heuristics used in variational inference consistently fail to find the best local optima of the mean-field objective function, suggesting that variational inference algorithms could be improved by employing more robust optimization algorithms. In the first part of this paper we propose a new learning algorithm that replaces the natural gradient steps at the core of SVI with trust-region updates. We show that these trust-region updates are feasible in practice and lead to a more robust algorithm generally yielding better performance.

In the second part of the paper we study the setting of continuous streams of data. SVI is a promising candidate for fitting Bayesian models to data streams, yet the dependence of its updates on the dataset size and its underlying assumption of uniform sampling from the entire dataset have hampered its application in practice. We therefore propose a new strategy for applying SVI in the streaming setting. We show that this strategy fails when used with natural gradient steps but works well with our trust-region method. When applied to latent Dirichlet allocation (LDA; Blei et al., 2003), our method is able to continuously integrate new data points not only at the document level but also at the word level, which existing methods for variational inference on streaming data cannot do.

2. A trust-region method for stochastic variational inference

In the following section, after introducing some notation and outlining the model class studied in this paper, we briefly review stochastic variational inference (SVI) in order to facilitate comparison with our trust-region extension.

2.1. Model assumptions

Our basic model assumptions are summarized by[1]

$$p(\boldsymbol{\beta}) = h(\boldsymbol{\beta}) \exp\left(\boldsymbol{\eta}^\top t(\boldsymbol{\beta}) - a(\boldsymbol{\eta})\right), \tag{1}$$

Proceedings of the 32^{nd} International Conference on Machine Learning, Lille, France, 2015. JMLR: W&CP volume 37.

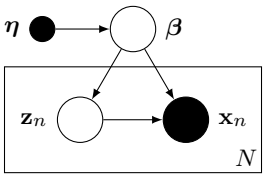

Figure 1. A graphical model representation of the model class considered in this paper.

$$p(\mathbf{x}, \mathbf{z} \mid \boldsymbol{\beta}) = \prod_n h(\mathbf{x}_n, \mathbf{z}_n) \exp\left(t(\boldsymbol{\beta})^\top f(\mathbf{x}_n, \mathbf{z}_n)\right). \quad (2)$$

That is, we have *global parameters* $\boldsymbol{\beta}$ whose prior distribution is an exponential family governed by natural parameters $\boldsymbol{\eta}$, and N conditionally independent pairs of *local parameters* $\mathbf{z}_n$ and observations $\mathbf{x}_n$ whose exponential-family distribution is controlled by $\boldsymbol{\beta}$ (Figure 1). While the basic strategy presented in this paper might also be applied in the non-conjugate case, for simplicity we assume conjugacy between the two exponential-family distributions.

Instances of this model class considered in greater detail in this paper are latent Dirichlet allocation (Blei et al., 2003) and mixture models. Other instances include (but are not limited to) hidden Markov models, probabilistic matrix factorizations, and hierarchical linear and probit regression (Hoffman & Blei, 2014).

2.2. Mean-field variational inference

Mean-field variational inference approximates the posterior distribution over latent variables with a factorial distribution, which here we take to be of the form

$$q(\boldsymbol{\beta}, \mathbf{z}) = q(\boldsymbol{\beta}) \prod_{n,m} q(z_{nm}), \quad (3)$$

$$q(\boldsymbol{\beta}) = h(\boldsymbol{\beta}) \exp\left(\boldsymbol{\lambda}^\top t(\boldsymbol{\beta}) - a(\boldsymbol{\lambda})\right). \quad (4)$$

We further assume that $q(z_{nm})$ is controlled by parameters $\boldsymbol{\phi}_{nm}$. Inference proceeds by alternatingly updating each factor to maximize the *evidence lower bound* (ELBO),

$$\mathcal{L}(\boldsymbol{\lambda}, \boldsymbol{\phi}) = \mathbb{E}_q\left[\log \frac{p(\boldsymbol{\beta})}{q(\boldsymbol{\beta})}\right] + \sum_{n=1}^{N} \mathbb{E}_q\left[\log \frac{p(\mathbf{x}_n, \mathbf{z}_n \mid \boldsymbol{\beta})}{q(\mathbf{z}_n)}\right], \quad (5)$$

which is equivalent to minimizing the Kullback-Leibler divergence between q and the true posterior distribution over $\boldsymbol{\beta}$ and $\mathbf{z}$. While the trust-region method described below might also be applied to more complex approximations, we are particularly interested to see how much a mean-field approximation can be improved via better learning algorithms. As mentioned above, the results of Hoffman & Blei (2014) suggest that mean-field approximations might often perform poorly due to local optima rather than the inflexibility of the approximating family (see also Wainwright & Jordan, 2008). We therefore focus our attention on the mean-field approximation in the following.

[1]The log-normalizer of the likelihood $p(\mathbf{x}, \mathbf{z} \mid \boldsymbol{\beta})$ is absorbed into $t(\boldsymbol{\beta})$ and $f(\mathbf{x}_n, \mathbf{z}_n)$ to simplify the notation.

2.3. Stochastic variational inference

Extending the work by Sato (2001), Hoffman et al. (2013) have shown that mean-field variational inference is equivalent to natural gradient ascent on

$$\mathcal{L}(\boldsymbol{\lambda}) = \max_{\boldsymbol{\phi}} \mathcal{L}(\boldsymbol{\lambda}, \boldsymbol{\phi}). \quad (6)$$

This interpretation enables the derivation of stochastic natural gradient algorithms for variational inference, in this context called *stochastic variational inference* (SVI).

For a uniformly at random selected data point n,

$$\mathcal{L}_n(\boldsymbol{\lambda}, \boldsymbol{\phi}_n) = N\mathbb{E}_q\left[\log \frac{p(\mathbf{x}_n, \mathbf{z}_n \mid \boldsymbol{\beta})}{q(\mathbf{z}_n)}\right] + \mathbb{E}_q\left[\log \frac{p(\boldsymbol{\beta})}{q(\boldsymbol{\beta})}\right] \quad (7)$$

represents an unbiased stochastic approximation of the ELBO given in Equation 5. Similarly,

$$\mathcal{L}_n(\boldsymbol{\lambda}) = \mathcal{L}_n(\boldsymbol{\lambda}, \boldsymbol{\phi}_n^*), \quad (8)$$

$$\boldsymbol{\phi}_n^* = \operatorname{argmax}_{\boldsymbol{\phi}_n} \mathcal{L}_n(\boldsymbol{\lambda}, \boldsymbol{\phi}_n) \quad (9)$$

represents an unbiased estimate of the lower bound in Equation 6. For simplicity, in the following we only consider stochastic approximations based on a single data point. An extension to batches of multiple data points is straightforward. A step in the direction of the natural gradient of $\mathcal{L}_n(\boldsymbol{\lambda})$ scaled by a learning rate of ρ_t is given by (Hoffman et al., 2013)

$$\boldsymbol{\lambda}_{t+1} = (1 - \rho_t)\boldsymbol{\lambda}_t + \rho_t\left(\boldsymbol{\eta} + N\mathbb{E}_{\boldsymbol{\phi}_n^*}[f(\mathbf{x}_n, \mathbf{z}_n)]\right). \quad (10)$$

2.4. Trust-region updates

As has been observed before (Hughes & Sudderth, 2013; Hoffman & Blei, 2014) and as we will corroborate further in Section 4, SVI can be prone to local optima. One possible solution to this problem is to use a more flexible family of approximating distributions (Hoffman & Blei, 2014), effectively smoothing the objective function but sacrificing the speed and convenience of a mean-field approximation. In contrast, here we try to address the issue of local optima by improving on the optimization algorithm.

Instead of performing a natural gradient step (Equation 10), we propose the following stochastic update:

$$\boldsymbol{\lambda}_{t+1} = \operatorname{argmax}_{\boldsymbol{\lambda}} \left\{\mathcal{L}_n(\boldsymbol{\lambda}) - \xi_t D_{\mathrm{KL}}(\boldsymbol{\lambda}, \boldsymbol{\lambda}_t)\right\}. \quad (11)$$

Intuitively, this *trust-region step* tries to find the optimal solution to a stochastic approximation of the lower bound,

Algorithm 1 Trust-region SVI

Set $t = 0$, initialize $\boldsymbol{\lambda}_0$ randomly
repeat
  Select $\mathbf{x}_n$ uniformly from the dataset
  Initialize $\boldsymbol{\phi}_n^*$
  repeat
    $\boldsymbol{\lambda} \leftarrow (1-\rho_t)\boldsymbol{\lambda}_t + \rho_t \left(\boldsymbol{\eta} + N\mathbb{E}_{\boldsymbol{\phi}_n^*}[f(\mathbf{x}_n, \mathbf{z}_n)]\right)$
    $\boldsymbol{\phi}_n^* \leftarrow \text{argmax}_{\boldsymbol{\phi}_n} \mathcal{L}_n(\boldsymbol{\lambda}, \boldsymbol{\phi}_n)$
  until convergence
  $\boldsymbol{\lambda}_{t+1} \leftarrow \boldsymbol{\lambda}$
  $t \leftarrow t + 1$
until convergence

regularized to prevent the distribution over parameters from changing too much. The degree of change between distributions is measured by the Kullback-Leibler (KL) divergence D_{KL}, and the parameter $\xi_t > 0$ controls the strength of the regularization. If we could solve Equation 11 in closed form, this algorithm would at least no longer be hampered by local optima in $\mathcal{L}_n$. In general this will not be possible so that we have to resort to the following optimization scheme.

For fixed $\boldsymbol{\phi}_n^*$, the natural gradient of the objective on the right-hand side of Equation 11 is given by

$$\boldsymbol{\eta} + N\mathbb{E}_{\boldsymbol{\phi}_n^*}[f(\mathbf{x}_n, \mathbf{z}_n)] - \boldsymbol{\lambda} + \xi_t(\boldsymbol{\lambda}_t - \boldsymbol{\lambda}). \tag{12}$$

This follows from the the natural gradient of $\mathcal{L}_n$ (Hoffman et al., 2013) and the fact that the gradient of the Kullback-Leibler divergence between two distributions from an expnential family in canonical form is given by

$$\nabla_{\boldsymbol{\lambda}} D_{\text{KL}}(\boldsymbol{\lambda}, \boldsymbol{\lambda}') = I(\boldsymbol{\lambda})(\boldsymbol{\lambda} - \boldsymbol{\lambda}'), \tag{13}$$

where $I(\boldsymbol{\lambda})$ is the Fisher information matrix (see Supplementary Section 1 for a derivation). Setting the gradient to zero yields

$$\boldsymbol{\lambda} = (1-\rho_t)\boldsymbol{\lambda}_t + \rho_t \left(\boldsymbol{\eta} + N\mathbb{E}_{\boldsymbol{\phi}_n^*}[f(\mathbf{x}_n, \mathbf{z}_n)]\right), \tag{14}$$

defining $\rho_t \equiv (1+\xi_t)^{-1}$. We propose to solve Equation 11 via alternating coordinate ascent, that is, by alternatingly computing $\boldsymbol{\phi}_n^*$ (Equation 9) and $\boldsymbol{\lambda}$ (Equation 14) until some stopping criterion is reached. Note that natural gradient ascent can be seen as a special case of this trust-region method where $\boldsymbol{\lambda}$ is initialized with $\boldsymbol{\lambda}_t$ and only one iteration of updates to $\boldsymbol{\phi}_n^*$ and $\boldsymbol{\lambda}$ is performed.

The need to iteratively optimize $\boldsymbol{\lambda}$ makes each trust-region step more costly than a simple natural gradient step. However, the additional overhead is often smaller than one might expect. For many models (such as LDA), the dominant cost is solving the sub-problem of computing $\text{argmax}_{\phi_n} \mathcal{L}(\boldsymbol{\lambda}, \boldsymbol{\phi}_n^*)$, which must be done iteratively. This sub-problem can be initialized with the previous value of $\boldsymbol{\phi}_n^*$; when $\boldsymbol{\lambda}$ is near convergence, this initialization will already be near a (local) optimum, and the sub-problem can be solved quickly.

In the limit of large ξ_t, the solution to Equation 11 will become identical to the natural gradient step in Equation 10 (see Supplementary Section 2). Consequently, the convergence guarantees that hold for SVI carry over to our trust region method. That is, under certain regularity assumptions and for appropriately decaying ρ_t (Bottou, 1998), iteratively applying Equation 11 will converge to a local optimum of $\mathcal{L}$.

For any finite ξ_t, the two update steps will generally be different. A crucial advantage of the trust-region method is that in each iteration we can initialize $\boldsymbol{\lambda}$ and $\boldsymbol{\phi}_n^*$ arbitrarily before applying the alternating optimization scheme. This way we can hope to jump out of local optima. The optimal initialization will depend on the data and the specific model being used. However, in our experiments with LDA and mixture models we found that a generally useful strategy is to initialize $\boldsymbol{\phi}_n^*$ such that the beliefs over $\mathbf{z}_n$ are uniform and to compute the initial $\boldsymbol{\lambda}$ with these beliefs.

The general algorithm is summarized in Algorithm 1. More detailed derivations for LDA and mixture models can be found in the supplementary material.

2.5. Related work

Our optimization resembles *mirror descent* (Nemirovski & Yudin, 1983; Beck & Teboulle, 2003), which applied to the lower bound (Equation 6) corresponds to updates of the form

$$\boldsymbol{\lambda}_{t+1} = \text{argmax}_{\boldsymbol{\lambda}} \left\{\langle \nabla_{\boldsymbol{\lambda}} \mathcal{L}_n(\boldsymbol{\lambda}_t), \boldsymbol{\lambda} - \boldsymbol{\lambda}_t\rangle - \xi_t B(\boldsymbol{\lambda}, \boldsymbol{\lambda}_t)\right\} \tag{15}$$

for some Bregman divergence B. The main difference to our algorithm is that we try to optimize $\mathcal{L}_n$ exactly instead of a first-order approximation.

Trust-region methods have a long history in optimization (Nocedal & Wright, 1999) and are frequently brought to bear on machine learning problems (e.g., Lin et al., 2007; Pascanu et al., 2014). However, we are unaware of any other trust-region method which omits a local approximation (Equation 11).

3. Streaming

Despite its sequential nature, SVI has not found widespread use in the streaming setting. One reason is that it assumes that the dataset is fixed and its size, N, known in advance. In contrast, streaming data only becomes available over time and is potentially infinite. Another disadvantage of a naive application of SVI to the streaming setting where

data points are processed once as they arrive is that it might not make the best use of the available computational resources. In real world applications, data points rarely arrive at a constant rate. Processing data points as they arrive thus means that there will be times when a computer has to quickly process many data points and other times where it will be idle.

In the following, we propose an alternative but similarly straightforward application of SVI to the streaming setting. As we will show in Section 4, this algorithm gives poor results with natural gradient steps as local optima and sensitivity to hyperparameters are particularly problematic in the streaming setting. However, we find that it performs well using our trust-region updates.

3.1. Streaming SVI

Rather than processing a data point once when it arrives, we suggest continuously optimizing an evolving lower bound. Instead of updating parameters directly, each new data point is simply added to a database. At time t, we optimize

$$\mathbb{E}_q\left[\log \frac{p(\boldsymbol{\beta})}{q(\boldsymbol{\beta})}\right] + \max_{\boldsymbol{\phi}} \sum_{n=1}^{N_t} \mathbb{E}_q\left[\log \frac{p(\mathbf{x}_n, \mathbf{z}_n \mid \boldsymbol{\beta})}{q(\mathbf{z}_n)}\right], \quad (16)$$

where N_t is the number of observed data points, by selecting a data point (or batch of data points) from the database uniformly at random and performing either a natural gradient or trust-region update.

Learning algorithms whose performance is robust to changes in hyperparameter settings are especially important in the streaming setting where performing test runs and cross-validation is often not an option. In addition to using trust-region updates we therefore employ empirical Bayes. After updating $\boldsymbol{\lambda}$ by performing a trust-region step (Equation 11), we update $\boldsymbol{\eta}$ by performing one stochastic step in the direction of the natural gradient of the lower bound,

$$\boldsymbol{\eta}_{t+1} = \boldsymbol{\eta}_t + \rho_t \left(E_{\boldsymbol{\lambda}_{t+1}}\left[t(\boldsymbol{\beta})\right] - E_{\boldsymbol{\eta}_t}\left[t(\boldsymbol{\beta})\right]\right). \quad (17)$$

To make the learning algorithm and hyperparameters less dependent on the speed with which new data points arrive, we use the following learning rate schedule,

$$\rho_t = \left(\tau + \tfrac{N_t - N_0}{B}\right)^{-\kappa}, \quad (18)$$

where B is the batch size used. Instead of coupling the learning rate to the number of updates to the model, this schedule couples it to the number of data points added to the database since the start of learning, $N_t - N_0$.

3.2. Streaming variational Bayes

Streaming variational Bayes (SVB) has been proposed as an alternative to SVI for the streaming setting (Broderick et al., 2013; Tank et al., 2014). Unlike SVI, its updates are independent of the dataset size. The basic idea behind SVB is that of assumed density filtering (e.g., Maybeck, 1982; Minka, 2001): the posterior is updated one data point at a time using Bayes rule and approximated between each two updates. Applied to our model class and ignoring the local variables for a moment, an update can be summarized as

$$\tilde{p}_{n+1}(\boldsymbol{\beta}) \propto p(\mathbf{x}_n \mid \boldsymbol{\beta}) q_{\boldsymbol{\lambda}_n}(\boldsymbol{\theta}), \quad (19)$$

$$\boldsymbol{\lambda}_{n+1} = \operatorname{argmin}_{\boldsymbol{\lambda}} D_{\mathrm{KL}}\left[q_{\boldsymbol{\lambda}}(\boldsymbol{\beta}) \,||\, \tilde{p}_{n+1}(\boldsymbol{\beta})\right], \quad (20)$$

where $\boldsymbol{\lambda}_0$ is set to $\boldsymbol{\eta}$.

3.3. Streaming batch algorithm

We will compare our streaming SVI algorithm and SVB to the following simple baseline algorithm. At each iteration, we randomly select a large batch of $\min(B, N_t)$ data points from the database, where N_t is the current number of data points in the database. We then run a batch algorithm to perform mean-field variational inference for a fixed number of iterations (or until it converges). Once the training is complete, we replace the current parameters with the parameters found by the batch algorithm and immediately restart the training on a newly sampled dataset of size B.

4. Experiments

In the following we demonstrate the usefulness of our proposed modifications to SVI. We start with a toy example and continue with increasingly realistic applications.

4.1. MNIST

As a first illustrative example, consider a mixture of multivariate Bernoulli distributions with K components, where the global parameters are given by component parameters $\boldsymbol{\beta}_k \in [0,1]^D$ and mixture weights $\boldsymbol{\pi}$. We assume uniform prior distributions on $\boldsymbol{\beta}_k$ and $\boldsymbol{\pi}$ and beta and Dirichlet distributions for the approximate posterior distributions,

$$q(\boldsymbol{\beta}) \propto \textstyle\prod_{k,i} \beta_{ki}^{a_{ki}-1}(1-\beta_{ki})^{b_{ki}-1}, \quad q(\boldsymbol{\pi}) \propto \textstyle\prod_k \pi_k^{\gamma_k - 1}.$$

Using 40 components, we applied this model to a binarized version of MNIST where each pixel has been randomly set to 0 or 1 according to its grayscale value interpreted as a probability. The a_{ki} and b_{ki} were randomly initialized by sampling from a gamma distribution with shape parameter 100 and scale parameter 0.01, and the γ_k were initialized at 1. We ran SVI with trust-region updates for 10 epochs (passes through the dataset) using 2 iterations for the inner loop (Algorithm 1) and SVI with natural gradient steps for 20 epochs. For both methods we used a batch size of 200 and the learning rate schedule $\rho_t = (\tau + t)^{-\kappa}$, where for κ we used 0.5 and for τ we used 100.

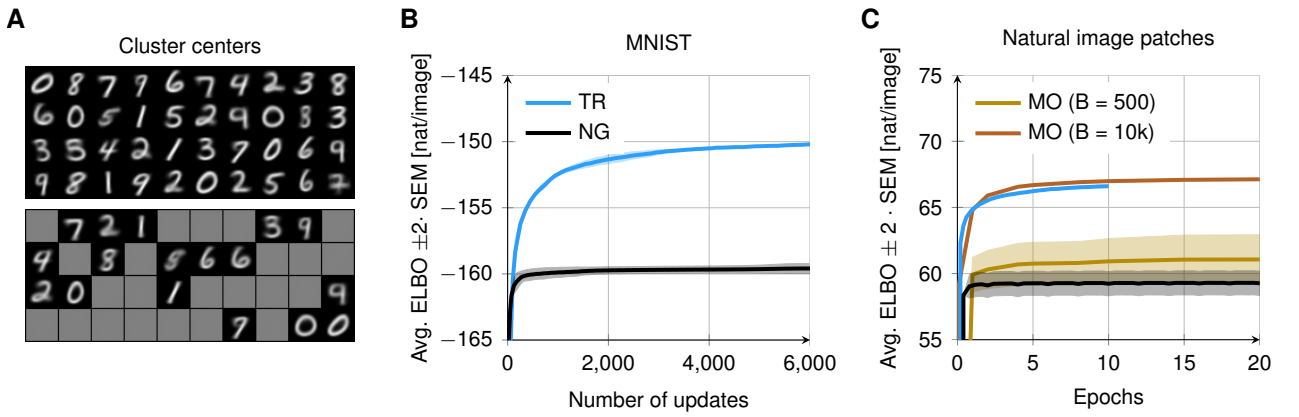

Figure 2. Mixture modeling results. **A**. Expected parameters, $\mathbb{E}[\boldsymbol{\beta}]$, under the mean-field solutions found via trust-region updates (top) and natural gradient steps (bottom). **B**. Lower bound averaged over multiple runs as a function of the number of updates of the parameters. In case of the trust-region method, each step of the inner loop is counted as one update. **C**. Lower bound for 8×8 natural image patches sampled from the van Hateren dataset. The horizontal axis indicates the number of passes through the dataset. An update of the trust-region method takes twice as long as a natural gradient update. A comparison is made to memoized variational inference (MO) with batch sizes of 500 and 10,000. Error bars of TR and MO (B = 10k) are too small to be visible.

Figure 2A shows the expected values of the probabilities β_{ki} under the posterior approximations found by the two methods. The solution found via natural gradient steps makes use of less than half of the mixture components, leading to significantly poorer performance (Figure 2B). This can be explained as follows. A mixture component which is inconsistent with the data will be assigned only a few data points. After updating γ and $\boldsymbol{\lambda}_k$, this mixture component will have lower prior probability and will shrink towards the prior distribution, which makes it less consistent with the data and in turn leads to the assignment of even fewer data points. The trust-region updates alleviate this problem by initializing ϕ_{nk} with $1/K$, that is, assigning the data points in the current batch equally to all mixture components. This forces the mixture components to initially become more consistent with the current batch of data.

4.2. Natural image patches

We next considered the task of modeling natural image patches sampled from the van Hateren image dataset (van Hateren & van der Schaaf, 1998). We used 8×8 image patches, which is a commonly used size in applications such as image compression or restoration (e.g., Zoran & Weiss, 2011). We fit mixtures of Gaussians with 50 components and normal-inverse-Wishart priors on the means and covariances of the components to 500,000 image patches. We used a batch size of 500, $\kappa = .5$, $\tau = 10$, and a fixed number of 5 iterations for the inner loop of the trust-region method. For comparison we also trained models with *memoized variational inference* (MO; Hughes & Sudderth, 2013) using the same settings for the priors and the same initialization of the distributions as with our methods. MO is an online inference algorithm akin to gradient averaging. Its updates are less noisy than those of SVI but it requires memorizing each individual update, generating a considerable memory overhead when the dimensionality of the gradients or the number of batches is large.

Figure 2 shows that SVI with natural gradient steps quickly gets stuck. MO with the same batch size performs slightly better, and MO with a large batch size of 10,000 does very well. Our trust-region method with a batch size of 500 achieves comparable performance and uses much less memory.

4.3. Wikipedia articles

We applied our algorithm to latent Dirichlet allocation (LDA; Blei et al., 2003) and a dataset of approximately 3.8M Wikipedia articles. The model is

$$p(\boldsymbol{\beta}) \propto \prod_{ki} \beta_{ki}^{\eta-1}, \qquad p(\boldsymbol{\theta}_n) \propto \prod_{k} \theta_{nk}^{\alpha_k-1}, \tag{21}$$

$$p(x_{nm}, z_{nm} \mid \boldsymbol{\theta}_n, \boldsymbol{\beta}) = \theta_{nz_{nm}} \beta_{z_{nm}x_{nm}}. \tag{22}$$

Figure 3A shows the performance of our trust-region method plotted against natural gradient ascent for randomly selected hyperparameters. The parameters were selected from a grid with the batch size B and learning rate parameters selected from

$$B \in \{10, 50, 100, 500, 1000\},$$
$$\kappa \in \{0.5, 0.6, 0.7, 0.8, 0.9, 1.0\},$$
$$\tau \in \{1, 10, 100, 1000, 10000\}.$$

For the trust-region method we used m steps in the inner loop of Algorithm 1 and M steps to compute $\boldsymbol{\phi}_n^*$. For

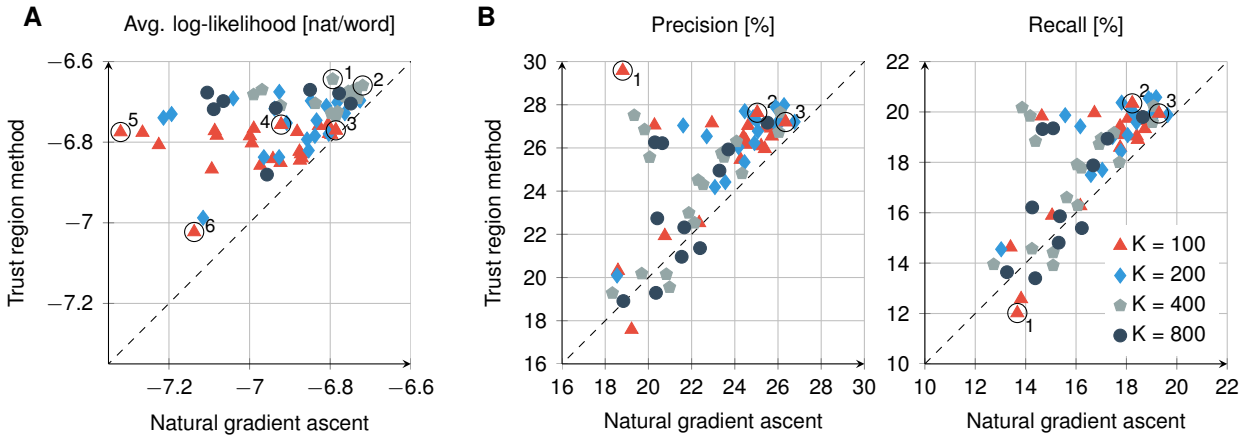

Figure 3. Hyperparameter search results. The circled results' hyperparameters are listed in Table 1 and Table 2. **A**. Performance on a validation set of Wikipedia articles. Shown are estimated log-likelihoods of the expected parameters found via the trust-region method and natural gradient ascent using randomly chosen hyperparameters. **B**. Performance on a validation set of Netflix users.

the corresponding natural gradient ascent we used $mM/2$ steps to compute $\boldsymbol{\phi}_n^*$. The two parameters were selected from $(m, M) \in \{(5, 40), (10, 20), (20, 10), (5, 100)\}$. We ran the trust-region method for 3 epochs and natural gradient ascent for 6 epochs. This gave both methods roughly the same amount of running time. The initial prior parameters were $\eta \in \{0.01, 0.05, 0.1, 0.2, 0.3\}$ and $\alpha_k \in \{0.01, 0.05, 0.1\}$ but were updated via empirical Bayes.

The two methods were evaluated on a separate validation set of 10,000 Wikipedia articles. We computed the expected value of $\boldsymbol{\beta}$ under the variational distribution and estimated the log-likelihood of the data given $\boldsymbol{\beta}$ using the Chib-style estimator described by Wallach et al. (2009). The trust-region method consistently outperformed natural gradient ascent. For 100 topics, the performance of natural gradient ascent varies widely while the trust-region method gave good results in almost all runs. In Table 1 we highlight the hyperparameters of some of the simulations. The table suggests that unlike natural gradient ascent, the trust-region method is able to perform well in particular with small batch sizes of as few as 50 data points.

We next tested our streaming variant of SVI by simulating a data stream as follows. At each iteration of the simulation we took R documents from the Wikipedia dataset and added it to a set of *active documents*, where R was sampled from a Poisson distribution with a fixed rate. We then observed one word of each active document with probability p and added it to a database. This means that at any point in time, the database contained a number of complete and incomplete articles. Streaming SVI only had access to this database. To simulate the gradual accumulation of data we added a small delay after each observation.

We compared our algorithm to streaming variational Bayes (SVB; Broderick et al., 2013) and the streaming batch algorithm described in Section 3.3. For a fair comparison, we extended SVB to also learn the α_k via empirical Bayes. Since in SVB η is only used in the first update to $\boldsymbol{\lambda}$ (Equation 19), we did not estimate η but instead ran SVB for $\eta \in \{0.01, 0.05, 0.1, 0.2, 0.3\}$. The performance we report is the maximum performance over all η. Since SVB can only process one document (or batch of documents) at a time but not one word at a time, SVB was given access to the entire document when the first word was added to the database, putting the other algorithms at a disadvantage. If we naively applied SVB to LDA on a word-by-word basis, the beliefs about $\boldsymbol{\theta}_n$ would only be updated once after seeing the first word due to the mean-field approximation decoupling $\boldsymbol{\theta}_n$ from $\mathbf{x}_n$. Upon arrival of a subsequent word, the target distribution becomes

$$\tilde{p}(\boldsymbol{\theta}_n, \mathbf{z}_n) \propto q(\boldsymbol{\theta}_n) q(\mathbf{z}_n) p(\mathbf{x}_n \mid \mathbf{z}_n), \tag{23}$$

so that an update will not change $q(\boldsymbol{\theta}_n)$. SVB would thus completely ignore dependencies between the topic assignments of different words. In our experiments, we used a batch size of 1,000 with SVB. The streaming batch algorithm was trained for 100 iterations on each randomly selected set of 50,000 data points.

We tested the streaming algorithms on LDA with 100 and 400 topics. For the hyperparameters of SVI we chose the best performing set of parameters found in the non-streaming setting (Figure 3A). That is, for the trust-region method and natural gradient ascent we used hyperparameter sets 4 and 3, and 1 and 2, respectively (see Table 1). We only reduced κ to 0.5 for 100 topics and to 0.4 for 400 topics, finding that keeping the learning rate up for a longer time slightly improves performance. η and the α_k of the streaming batch algorithm were initialized with the same

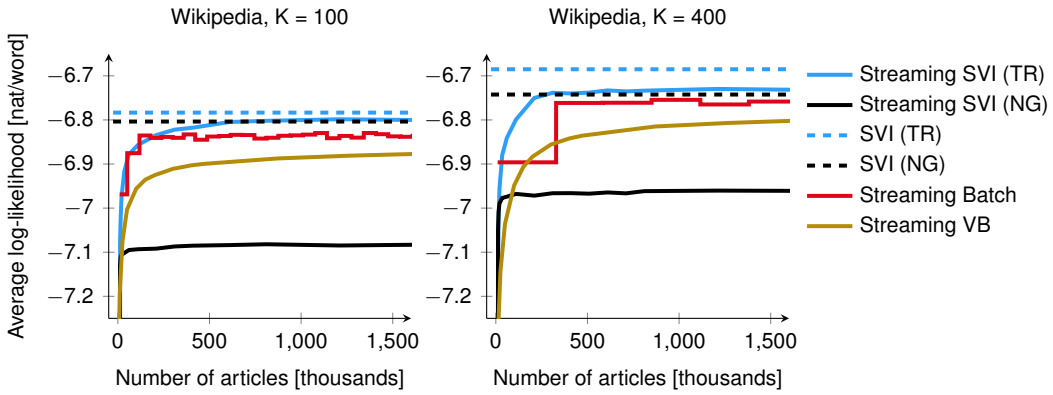

Figure 4. Performance of LDA on streamed Wikipedia articles. The horizontal axis shows the number of articles (complete and incomplete) currently in the database (it is thus monotonically related to time). The vertical axis shows an estimate of the average log-likelihood on a test set separate from the validation set used in Figure 3. Dashed lines indicate the performance of SVI trained on the entire dataset and evaluated on the test set using the same hyperparameters as the streaming variants.

parameters used with the trust-region method.

We find that natural gradient ascent gives extremely poor performance in the streaming setting and gets stuck in a local optimum early on. Using our trust-region method, on the other hand, we were able to achieve a performance at least as good as the best performing natural gradient ascent of the non-streaming setting (Figure 4). The simple streaming batch algorithm also performs well, although its performance is limited by the fixed batch size.

SVB with a batch size of 1,000 performs much better than streaming SVI with natural gradient steps, but worse than our trust-region method or the streaming batch algorithm. We note that by using larger batches, e.g. of size 50,000, SVB can be made to perform at least as well in the limit as the streaming batch algorithm (assuming the data is stationary, as is the case here). However, SVB would only be able to start training once the first 50,000 data points have arrived. The streaming batch algorithm, on the other hand, will simply take all data points available in the database.

4.4. Netflix

We applied LDA to the task of collaborative filtering using the Netflix dataset of 480,000 user ratings on 17,000 movies. Each rating is between 1 and 5. Like Gopalan et al. (2013), we thresholded the ratings and for each user kept only a list of movies with ratings of 4 or 5. For this "implicit" version of the Netflix data, the task is to predict which movies a user likes based on movies the user previously liked. 20% of each user's movies were kept for evaluation purposes and were not part of the training data. Again following Gopalan et al. (2013), we evaluate algorithms by computing *normalized precision-at-M* and *recall-at-M*. Let L be the test set of movies liked by a user and let P be a list of M predicted movies. Then the two measures are given by

$$\text{precision} = \frac{|L \cap P|}{\min\{|L|, |P|\}}, \qquad \text{recall} = \frac{|L \cap P|}{|L|}. \tag{24}$$

Precision is normalized by $\min\{|L|, |P|\}$ rather than $|P|$ so as to keep users with fewer than M movies from having only a small influence on the performance (Gopalan et al., 2013). In the following, we will use $M = 20$.

LDA is trained on the Netflix data as in the case of Wikipedia articles, only that here users play the role of ar-

Table 1. Hyperparameters corresponding to Wikipedia experiments in Figure 3A. B is the batch size and m and M control the number of iterations in the inner loops of each trust-region step (see text for details).

#	B	α	η	κ	τ	m	M
1	500	0.1	0.2	0.6	10	10	20
2	1000	0.1	0.05	0.7	100	20	10
3	500	0.1	0.2	0.7	100	10	20
4	50	0.1	0.2	0.7	10	20	10
5	50	0.1	0.2	0.7	1	20	10
6	10	0.1	0.01	0.5	100	10	20

Table 2. Hyperparameters corresponding to the highlighted Netflix experiments in Figure 3B.

#	B	α	η	κ	τ	m	M
1	10	0.1	0.05	1.0	1	5	40
2	100	0.05	0.3	0.5	100	5	40
3	500	0.1	0.3	0.5	10	10	20

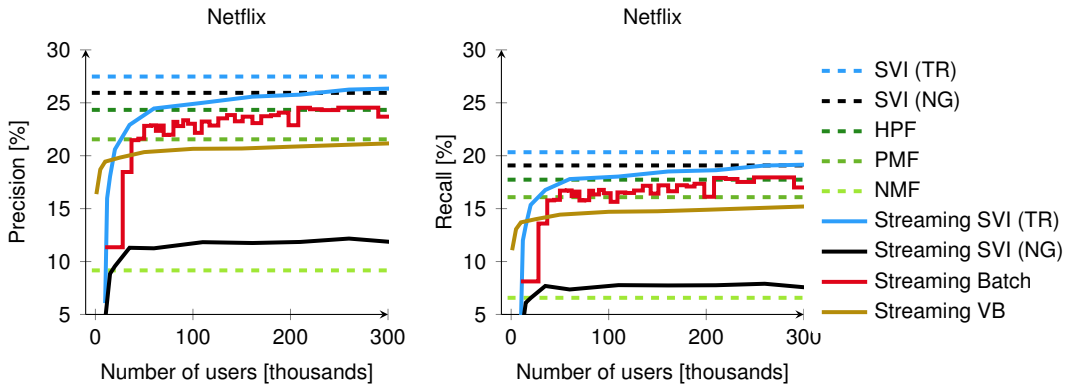

Figure 5. Performance of LDA trained on a stream of Netflix users. The horizontal axis gives the current number of users in the database.

ticles and movies play the role of words. For prediction, we approximated the predictive distribution over the next movie liked by user n,

$$p(x_{ni} \mid \mathbf{x}_{n,<i}) \approx \sum_k \mathbb{E}_q[\theta_{nk}]\mathbb{E}_q[\beta_{kx_{ni}}], \qquad (25)$$

where $\mathbf{x}_{n,<i}$ corresponds to the 80% movies not used for evaluation. We then took the 20 most likely movies under this predictive distribution which were not already in $\mathbf{x}_{n,<i}$ as predictions.

Figure 3B shows results obtained with random hyperparameters using the same grid used with the Wikipedia dataset. Evaluation was performed on a random subset of 10,000 users. Interpretation of the results here is a little bit more challenging since we have to take into account two competing measures. Nevertheless, we find that for the better performing hyperparameter settings, the trust-region method again consistently outperforms natural gradient ascent on both measures. Hyperparameters of the highlighted simulations are given in Table 2.

The streamed Wikipedia articles allowed us to study the effect of a growing dataset on SVI without having to worry about nonstationarities in the data. Here we study a slightly more realistic setting by streaming the users in the order they signed up for Netflix, thus potentially creating a nonstationary data stream. We tested LDA only with 100 topics since using more topics did not seem to help improve performance (Figure 3B). Figure 5 shows the results of the streaming experiment on a test set of 10,000 users different from the ones used in the parameter search. The hyperparameters were again chosen based on the parameter search results (sets 2 and 3 in Table 2) but reducing κ to 0.4. As in the Wikipedia experiment, we find that natural gradient ascent performs poorly while the trust-region method is able to get to the same performance level as the best performing natural gradient ascent in the non-streaming setting.

For comparison, we also include results of Gopalan et al. (2013) on hierarchical Poisson factorization (HPF), probabilistic matrix factorization (PMF), and nonnegative matrix factorization (NMF) to show that our numbers are competitive[2].

5. Discussion

We have proposed a new variant of stochastic variational inference which replaces the natural gradient step with a trust-region step. Our experiments show that this change can make SVI less prone to local optima and less sensitive to the choice of hyperparameters. We only explored an application of the trust-region method to mean-field approximations with conjugate priors. However, the same ideas might be applied in other settings, for example, by combining them with other recent innovations in variational inference such as *structured SVI* (Hoffman & Blei, 2014) and *black box variational inference* (Ranganath et al., 2014).

We further described a simple strategy for applying SVI to streaming data and have shown that the trust-region updates are crucial for good performance in this setting. However, Figures 4 and 5 also reveal room for improvement as our streaming method does not yet reach the performance of the trust-region method applied to the full dataset. Since we used empirical Bayes to tune the parameters of the prior and the trust-region method's performance is not very sensitive to the batch size, the only hyperparameters still requiring some tuning are the ones controlling the learning rate schedule. Here we found that using larger learning rates ($\kappa \leq 0.5$) generally works well. Nevertheless, it would be desirable to find a fully automatic solution. We explored adapting the work of Ranganath et al. (2013) to our trust-region method but found that it did not work well in the streaming setting.

[2]Gopalan et al. (2013) report worse results for LDA with SVI. This may be due to a suboptimal choice of hyperparameters.

Acknowledgments

This research was mostly done while Lucas Theis was an intern at Adobe Research, San Francisco.

References

Beck, A. and Teboulle, M. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31:167–175, 2003.

Blei, D. M., Ng, A. Y., and Jordan, M. I. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3: 993–1022, 2003.

Bottou, L. *Online Learning and Stochastic Approximations*, volume 17, pp. 9–42. Cambridge University Press, 1998.

Broderick, T., Boyd, N., Wibisono, A., Wilson, A. C., and Jordan, M. I. Streaming Variational Bayes. In *Advances in Neural Information Processing Systems 26*, 2013.

Gopalan, P., Hofman, J. M., and Blei, D. M. Scalable Recommendation with Poisson Factorization. *arXiv.org*, abs/1311.1704, 2013.

Gopalan, P., Charlin, L., and Blei, D. M. Content-based recommendations with Poisson factorization. In *Advances in Neural Information Processing Systems 27*, 2014.

Hoffman, M. D. and Blei, D. M. Structured Stochastic Variational Inference. *arXiv.org*, abs/1404.4114, 2014.

Hoffman, M. D., Blei, D. M., and Bach, F. R. Online Learning for Latent Dirichlet Allocation. In *Advances in Neural Information Processing Systems 23*, 2010.

Hoffman, M. D., Blei, D. M., Wang, C., and Paisley, J. Stochastic Variational Inference. *Journal of Machine Learning Research*, 14:1303–1347, 2013.

Hughes, M. C. and Sudderth, E. B. Memoized Online Variational Inference for Dirichlet Process Mixture Models. In *Advances in Neural Information Processing Systems 26*, 2013.

Lin, C.-J., Weng, R. C., and Keerthi, S. S. Trust region newton methods for large-scale logistic regression. In *Proceedings of the 26th International Conference on Machine Learning*, 2007.

Mandt, S. and Blei, D. Smoothed Gradients for Stochastic Variational Inference. In *Advances in Neural Information Processing Systems 27*, 2014.

Maybeck, P. S. *Stochastic models, estimation and control*. Academic Press, 1982.

Minka, T. *A family of algorithms for approximate Bayesian inference*. PhD thesis, MIT Media Lab, MIT, 2001.

Nemirovski, A. and Yudin, D. *Problem Complexity and Method Efficiency in Optimization*. Wiley, New York, 1983.

Nocedal, J. and Wright, S. J. *Numerical Optimization*. Springer, New York, 1999.

Paisley, J., Wang, C., and Blei, D. M. The discrete infinite logistic normal distribution. *Bayesian Analysis*, 7 (2):235–272, 2012.

Pascanu, R., Dauphin, Y. N., Ganguli, S., and Bengio, Y. On the saddle point problem for non-convex optimization. *arXiv.org*, abs/1405.4604, 2014.

Ranganath, R., Wang, C., Blei, D. M., and Xing, E. P. An Adaptive Learning Rate for Stochastic Variational Inference. In *Proceedings of the 30th International Conference on Machine Learning*, 2013.

Ranganath, R., Gerrish, S., and Blei, D. M. Black box variational inference. In *International Conference on Artificial Intelligence and Statistics 17*, 2014.

Sato, M. Online model selection based on the variational bayes. *Neural Computation*, 13(7):1649–1681, 2001.

Tank, A., Foti, N. J., and Fox, E. B. Streaming Variational Inference for Bayesian Nonparametric Mixture Models. *arXiv.org*, abs/1412.0694, 2014.

van Hateren, J. H. and van der Schaaf, A. Independent component filters of natural images compared with simple cells in primary visual cortex. *Proc. of the Royal Society B: Biological Sciences*, 265(1394), 1998.

Wainwright, M. J. and Jordan, M. I. Graphical Models, Exponential Families, and Variational Inference. *Foundations and Trends in Machine Learning*, 1(1), 2008.

Wallach, H. M., Murray, I., Salakhutdinov, R., and Mimno, D. Evaluation Methods for Topic Models. In *Proceedings of the 26th International Conference on Machine Learning*, 2009.

Wang, C., Paisley, J., and Blei, D. Online variational inference for the hierarchical Dirichlet process. In *International Conference on Artificial Intelligence and Statistics 14*, 2011.

Zoran, D. and Weiss, Y. From learning models of natural image patches to whole image restoration. In *International Conference on Computer Vision*, pp. 479–486, 2011.