Marginalized Stochastic Natural Gradients for Black-Box Variational Inference

Geng Ji [1 2] **Debora Sujono** [2] **Erik B. Sudderth** [2]

Abstract

Black-box variational inference algorithms use stochastic sampling to analyze diverse statistical models, like those expressed in probabilistic programming languages, without model-specific derivations. While the popular score-function estimator computes unbiased gradient estimates, its variance is often unacceptably large, especially in models with discrete latent variables. We propose a stochastic natural gradient estimator that is as broadly applicable and unbiased, but improves efficiency by exploiting the curvature of the variational bound, and provably reduces variance by marginalizing discrete latent variables. Our marginalized stochastic natural gradients have intriguing connections to classic coordinate ascent variational inference, but allow parallel updates of variational parameters, and provide superior convergence guarantees relative to naive Monte Carlo approximations. We integrate our method with the probabilistic programming language Pyro and evaluate real-world models of documents, images, networks, and crowd-sourcing. Compared to score-function estimators, we require far fewer Monte Carlo samples and consistently converge orders of magnitude faster.

1. Introduction

Variational inference is widely used to estimate the posterior distributions of hidden variables in probabilistic models (Wainwright & Jordan, 2008). Many previous studies have found that variational inference can have dramatic computational advantages compared to MCMC methods like Gibbs samplers (Gopalan & Blei, 2013; Gan et al., 2015; Gopalan et al., 2016; Ji et al., 2019). Variational bounds are usually optimized via *coordinate ascent variational inference* (CAVI, Jordan et al. (1999)) algorithms that iteratively update single (or small blocks of) variational parameters, while holding all others fixed to their current values. Although CAVI updates can be effective for simple models composed from conjugate priors, for many models the expectations required for exact CAVI updates are intractable: they may require complex integrals for continuous variables, or computation scaling exponentially with the number of dependent discrete variables.

Variational algorithms for models with non-conjugate conditionals have been derived via hand-crafted auxiliary variables that induce looser, but more tractable, bounds on the data log-likelihood (Jordan et al., 1999; Winn & Bishop, 2005). Such bounds typically require complex derivations specialized to the parametric structure of specific distributions (Albert & Chib, 1993; Jaakkola & Jordan, 1999; Polson et al., 2013), and thus do not easily integrate with general-purpose probabilistic inference systems.

To address these limitations, several authors have explored stochastic gradient algorithms that directly optimize a reparameterized bound involving the log-likelihood gradient or score function (Paisley et al., 2012; Wingate & Weber, 2013; Ranganath et al., 2014), as in the classic REINFORCE policy gradient algorithm (Williams, 1992). Unlike other black-box variational methods such as *automatic differentiation variational inference* (ADVI, Kucukelbir et al. (2017)) that require specific variable reparameterizations (Kingma & Welling, 2014; Rezende et al., 2014), REINFORCE provides unbiased gradients for all models including the many practically important ones with discrete latent variables.

Due to its simplicity and generality, REINFORCE has become the "standard" variational inference algorithm for a number of *probabilistic programming languages* (PPLs) including Edward and TensorFlow Probability (Tran et al., 2016; 2018), WebPPL (Goodman & Stuhlmüller, 2014; Ritchie et al., 2016), Pyro (Bingham et al., 2019), and Gen (Cusumano-Towner et al., 2019). However, its gradient estimates may have extremely high variance. An official WebPPL tutorial[1] warns that REINFORCE will produce poor results for the LDA topic model (Blei et al., 2003) due to its discrete assignment variables: "Because WebPPL's implementation of variational inference works much bet-

[1]Facebook AI [2]Department of Computer Science, University of California, Irvine. Correspondence to: Geng Ji <gji@fb.com>, Debora Sujono <dsujono@uci.edu>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

[1]http://probmods.github.io/ppaml2016/chapters/4-3-variational.html

ter with continuous random choices than discrete ones," they produce an alternative model representation by "explicitly integrating out the latent choice of topic per word" so that ADVI may be used. However, this requires model-specific derivations that are not generally tractable; a better black-box variational method for discrete and other non-reparameterizable latent variable models is sorely needed. Titsias & Lázaro-Gredilla (2015); Tucker et al. (2017); Grathwohl et al. (2018); Liu et al. (2019); Yin & Zhou (2019); Dong et al. (2020) have proposed variance reduction methods for REINFORCE that partially address this issue.

In this paper, we analyze the poor convergence behavior of REINFORCE variational gradients on discrete probabilistic models, and contrast it with a natural gradient variant that makes use of local curvature information. Unlike several previous applications of natural gradients in variational inference (Sato, 2001; Hoffman et al., 2013) where expectations are computed analytically, we propose a Monte Carlo variant inspired by the successes of natural policy gradients in reinforcement learning (Kakade, 2001; Schulman et al., 2015). To avoid the large-variance estimators induced by rare configurations of discrete variables, we marginalize their values in the associated gradient dimensions, producing an estimator with provably lower variance.

Like REINFORCE, our *marginalized stochastic natural gradients* (MSNG) do not require model-specific derivations, do not require gradients of the log-probability, and are guaranteed to converge with appropriate learning rates. As observed for more general stochastic optimization problems (Thomas et al., 2020), MSNG convergence is dramatically accelerated via the interplay of variance reduction and geometry adaptation. MSNG updates intuitively reduce to a weighted combination of current variational parameters and unbiased Monte Carlo estimates of ideal CAVI updates.

We integrate our MSNG method with the PPL Pyro (Bingham et al., 2019). On real-world models of documents, images, networks, and crowd-sourcing, it consistently converges orders of magnitude faster than REINFORCE while requiring far fewer samples to estimate expectations. Compared to baselines variational methods using hand-crafted auxiliary bounds, MSNG updates are equivalent or even superior in terms of predictive accuracy and robustness to initialization, while being easier to derive and implement.

2. Discrete Latent Variable Models

We begin by reviewing five probabilistic models that generate observed data x via discrete latent variables z, from some joint distribution $p(z, x) = p(z)p(x \mid z)$. We specify these models in Pyro, a popular PPL that provides flexible but precise semantics for defining probabilistic models and performing inference queries (Bingham et al., 2019). The grand promise of PPLs is that given a generative model

```
1  import torch, pyro
2  from pyro import distributions as dist
3
4  class BN:
5      def __init__(self, hyperparams):
6          self.b, self.W1, self.c1, self.W2, self.c2 = hyperparams
7          self.D_H2, self.D_H1 = self.W2.shape
8
9      @abstractmethod
10     def squash_fun(self, x):
11         raise NotImplementedError
12
13     def model(self, data):
14         dat_axis = pyro.plate('dat_axis', data.shape[0], dim=0)
15         top_axis = pyro.plate('top_axis', self.D_H2, dim=1)
16         mid_axis = pyro.plate('mid_axis', self.D_H1, dim=1)
17         bot_axis = pyro.plate('bot_axis', data.shape[1], dim=1)
18         with dat_axis, top_axis:
19             z_top = pyro.sample('z_top',
20                                 dist.Bernoulli(self.squash_fun(self.b)))
21             wz_top = z_top @ self.W2 + self.c2
22         with dat_axis, mid_axis:
23             z_bot = pyro.sample('z_bot',
24                                 dist.Bernoulli(self.squash_fun(wz_top)))
25             wz_bot = z_bot @ self.W1 + self.c1
26         with dat_axis, bot_axis:
27             pyro.sample('x', dist.Bernoulli(self.squash_fun(wz_bot)),
28                         obs=data)
29
30 class NoisyOrBN(BN):
31     def squash_fun(self, x):
32         return torch.ones([]) - torch.exp(-x)
33
34 class SigmoidBN(BN):
35     def squash_fun(self, x):
36         return torch.sigmoid(x)
```

Figure 1. Pyro specification of three-layer Bayesian networks. By defining different squashing functions (lines 31 and 35), the noisy-OR topic model and sigmoid belief network are easily created from the base class. "Plate" variables are conditionally independent.

specification, appropriate inference code can be automatically generated, enabling rapid model exploration even for non-expert users. For variational inference with discrete latent variables, the standard choice for most PPLs is REINFORCE. But as we show in Sec. 5, REINFORCE converges very slowly for all models reviewed in this section, motivating the novel algorithms developed in Sec. 4.

2.1. Deep Noisy-OR and Sigmoid Belief Networks

Noisy-OR and sigmoid belief networks both generate data via layers of binary latent variables. See Fig. 1 for compact, integrated Pyro code defining these models.

Like the logical OR operator, the noisy-OR conditional distribution (Horvitz et al., 1988) assumes the activation of a binary variable is independently influenced by the state of each parent. As shown in Eq. (1), if a parent $k \in \mathcal{P}(i)$ is active ($z_k = 1$), it will activate its child i with probability $1 - \exp(-w_{ki})$, regardless of the states of other parents:

$$p(z_i = 1 \mid z_{\mathcal{P}(i)}) = 1 - \exp\left(-w_{0i} - \sum\nolimits_{k \in \mathcal{P}(i)} w_{ki} z_k\right). \tag{1}$$

Inactive parents have no influence on z_i, and a small "leak" probability $1 - \exp(-w_{0i})$ allows nodes to occasionally activate even when all parents are off. The noisy-OR distribution has been widely used in bipartite graphs for medical diagnosis, like the QMR-DT system, where observed symptoms in the bottom layer may be caused by multiple latent diseases (Shwe et al., 1991). More recently, Google (Murphy, 2012, Sec. 26.5.4), Liu et al. (2016), and Ji et al. (2019)

use deep noisy-OR Bayesian networks to model topic interactions within documents, in which observed word tokens are generated by their hierarchical latent topic ancestors.

Sigmoid belief networks (Neal, 1992) are layered binary generative models where the activation probability of a node is determined by the sigmoid function $\sigma(r) = \frac{1}{1+\exp(-r)}$. The activation z_{ij} of node j in layer i depends on the states of nodes in the preceding layer z_{i+1}:

$$p(z_{ij} = 1 \mid z_{i+1}) = \sigma(w_{ij}^T z_{i+1} + c_j). \tag{2}$$

The possibly sparse weight vector w_{ij} determines which parents directly influence the activation of z_{ij}. Gan et al. (2015) use two layers of binary latent variables to generate binary digit images x observed at the finest scale.

2.2. Categorical and Binary Relational Models

Stochastic block models (SBM, Holland et al. (1983)) use categorical latent variables to capture community memberships underlying network or relational data. Each entity i is assigned a community $z_i \sim \text{Categorical}(\pi)$. The probability that a link x_{ij} exists between entities i and j is given by the interaction probability $p(x_{ij} = 1|z_i = k, z_j = \ell) = w_{k\ell}$. We assume relations are undirected, so the link matrix x and connectivity probability matrix w are both symmetric.

In addition to stochastic block models, we also consider a simplified version of the binary latent feature relational model of Miller et al. (2009). Each entity i is described by a set of D hidden binary features $z_{id} \sim \text{Bernoulli}(\rho)$. The probability that an undirected link x_{ij} between entities i and j is present depends on the set of shared features:

$$p(x_{ij} = 1 \mid z) = \Phi\left(w_0 + \sum\nolimits_{d=1}^{D} w_d z_{id} z_{jd}\right). \tag{3}$$

In Eq. (3), Φ is the probit function (standard normal CDF). Weight w_d controls the change in link probability when entities share feature d. $\Phi(w_0)$ is the (small) probability of link occurrence when no features are shared. See the supplement for Pyro specifications of these relational models.

2.3. Annotation Models

Annotation models are used to measure the quality of crowd-sourced data labeled by a large collection of unreliable annotators, and correct label errors. We apply the annotation model of Passonneau & Carpenter (2014) to word sense annotation. Each item i belongs to a true category $z_i \in \{1, \dots, K\}$, where π_k is the prior probability of category k. Observation $x_{ij} \in \{1, \dots, K\}$ represents the label that annotator j assigns to item i. The probability that annotator j assigns the label ℓ to an item whose true category is k depends on the annotator's reliability within that category: $p(x_{ij} = \ell | z_i = k) = \theta_{jk\ell}$, where $\theta_{jk} \sim \text{Dirichlet}(\beta_k)$. The observation matrix x is sparse because each annotator only labels a small subset of all items.

3. Limitations of Existing VI Algorithms

Given any latent variable model $p(z, x) = p(z)p(x \mid z)$, our goal is to infer the posterior distribution $p(z \mid x)$. For complex models, exact posterior inference is usually intractable and approximations are thus needed. The popular *mean field variational inference* method seeks an approximate posterior $q(z)$ from a tractable family with simpler dependencies by maximizing the *evidence lower bound* (ELBO):

$$\mathcal{L}(x; q) = \mathbb{E}_{q(z)} [\log p(z, x) - \log q(z)] \le \log p(x). \tag{4}$$

Maximizing the ELBO minimizes the Kullback-Leibler divergence of $q(z)$ from the true posterior $p(z \mid x)$. In this work, we make a "naive" mean-field approximation in which $q(z) = \prod_i q(z_i)$ is fully factorized.

We next review two classic families of VI algorithms for optimizing the ELBO with respect to $q(z)$. Their limitations motivate the new algorithms we develop in Sec. 4.

3.1. Coordinate Ascent Variational Inference

The classic CAVI algorithm (Jordan et al., 1999; Winn & Bishop, 2005) tightens variational bounds by updating single factors $q(z_i)$ of the variational posterior via Eq. (5), where $p(z_i \mid z_{-i}, x)$ is the *complete conditional* (Blei et al., 2017) given all other variables z_{-i} and observations x:

$$q(z_i) \propto \exp\left\{\mathbb{E}_{q(z_{-i})}[\log p(z_i \mid z_{-i}, x)]\right\}. \tag{5}$$

The expectation is with respect to the variational distributions $q(z_{-i}) = \prod_{j \neq i} q(z_j)$ for all other variables at the current iteration. For instance, suppose z_i is binary and $q(z_i)$ is Bernoulli with logit (natural) parameter τ_i:

$$\tau_i \triangleq \log \frac{\mu_i}{1 - \mu_i}, \quad \mu_i = \mathbb{E}_{q(z_i)}[z_i] = q(z_i = 1). \tag{6}$$

Eq. (5) then simplifies to matching the variational logit to the expected logit of the complete conditional:

$$\tau_i = \mathbb{E}_{q(z_{-i})}\left[\log \frac{p(z_i = 1 \mid z_{-i}, x)}{p(z_i = 0 \mid z_{-i}, x)}\right]. \tag{7}$$

Note that CAVI updates only have strong guarantees when run sequentially: all dependent variational parameters in $q(z_{-i})$ must be held fixed when $q(z_i)$ is updated. This condition is generally required for CAVI updates to monotonically increase the ELBO and converge to a (local) maximum. For large or complex models with many dependent latent variables, CAVI iterations may thus be relatively slow.

Another limitation of CAVI is that while it provides a uniform way to optimize the ELBO, it is not computationally tractable for many models with high-degree variable relationships. In particular, for non-conjugate conditionals like those in Eqs. (1,2,3), computing the expectations in Eq. (7) requires enumerating the exponentially many joint configurations of variables in the Markov blanket of z_i.

Such expectations may sometimes be avoided by introducing auxiliary variables into the probabilistic model via data augmentation tricks (Albert & Chib, 1993; Polson et al., 2013), or directly modifying the variational objective (Jaakkola & Jordan, 1999; Šingliar & Hauskrecht, 2006). While these approaches lead to tractable CAVI updates, the resulting bounds are looser than Eq. (4), and often require complex derivations specialized to narrow model families.

Ye et al. (2020) recently applied a stochastic extension of CAVI, which we refer to as SCAVI, to NMR spectroscopy. The SCAVI method can be applied to more general models, and simply approximates the expectation in Eq. (5) via Monte Carlo sampling from $q(z_{-i})$. While simple and intuitive, the theory supporting SCAVI is weak: Ye et al. (2020) show convergence only in the impractical limit where the number of Monte Carlo samples *per iteration* approaches infinity, and only when variables are updated sequentially.

3.2. Gradient-based Variational Inference

Variational bounds may also be optimized via stochastic gradient ascent. REINFORCE optimizes Eq. (4) using unbiased stochastic gradients computed via Monte Carlo sampling from $q(z)$. This method is derived by rewriting the ELBO's gradient as an expectation of $q(z)$ that depends on the gradient of $\log q(z)$, so REINFORCE is also known as the score-function estimator (Zhang et al., 2018).

For simplicity, we describe how REINFORCE is applied to binary variables z_i. We parameterize $q(z_i)$ using natural parameters τ_i (6) to avoid optimization constraints. REINFORCE computes an unbiased estimate of the ELBO's gradient with respect to τ_i using M samples $z_i^{(m)} \sim q(z)$:

$$\frac{\partial \mathcal{L}}{\partial \tau_i} \approx \frac{1}{M} \sum_{m=1}^{M} \frac{\partial \log q(z_i)}{\partial \tau_i}\bigg|_{z_i^{(m)}} \cdot \big(\log p(z_i^{(m)} \mid z_{-i}^{(m)}, x) - \log q(z_i^{(m)})\big), \tag{8}$$

where $p(z_i \mid z_{-i}, x)$ is the complete conditional, and the score-function can be written as

$$\frac{\partial \log q(z_i)}{\partial \tau_i} = \sigma(-\tau_i)^{z_i} \cdot (-\sigma(\tau_i))^{1-z_i}. \tag{9}$$

An important advantage of Eq. (8), as well as REINFORCE gradients with respect to all other distributions, is that it only requires model log-densities $\log p(z, x)$ to be evaluated at particular points. REINFORCE is thus a *black box variational inference* (BBVI) method that may be applied to different probabilistic models without specialized derivations (Ranganath et al., 2014). Unlike ADVI (Kucukelbir et al., 2017) and some other VI algorithms, it does not require the model log-probability to be differentiable, and may thus be applied to discrete variables which cannot be exactly reparameterized for unbiased gradient estimation (Kingma & Welling, 2014; Rezende et al., 2014). Note that the Stan PPL (Carpenter et al., 2017), which integrates ADVI, inflexibly prohibits discrete latent variables.

While broadly applicable, REINFORCE has notoriously slow convergence because its gradient estimates have high variance. The update of Eq. (8) already improves on the most basic REINFORCE implementation by ignoring factors in the joint log-probability that do not depend on z_i. This modification is equivalent to exactly marginalizing, or "Rao-Blackwellizing" (Ranganath et al., 2014), conditionally independent variables that are outside the Markov blanket of z_i. Pyro model specifications allow this simple form of "Rao-Blackwellization" to be exploited by all inference algorithms we compare, including REINFORCE. We then develop more sophisticated marginalized estimators that dramatically reduce gradient variance.

One can also introduce control variates, which preserve target expectations but approximately cancel noise, into REINFORCE gradient estimators to further reduce variance. Wingate & Weber (2013), Ranganath et al. (2014), and Ritchie et al. (2016) set the control variate to be the zero-mean score function scaled by a carefully-chosen constant called the baseline (Greensmith et al., 2004). More complex control variates include (Paisley et al., 2012; Tucker et al., 2017; Grathwohl et al., 2018).

Using the Gumbel-Max trick (Yellott Jr, 1977; Papandreou & Yuille, 2011), Jang et al. (2017) and Maddison et al. (2017) propose *continuous relaxations of discrete* (CONCRETE) variables. For a surrogate ELBO that replaces discrete distributions with continuous CONCRETE relaxations, gradients may then be computed by automatic differentiation. For models where fractional approximations of discrete variables induce valid likelihoods, this approach often leads to low-variance gradient estimates. However, the gradients are biased with respect to the non-relaxed ELBO of the true discrete model, and good performance requires careful tuning of temperature hyperparameters.

4. Marginalized Stochastic Natural Gradients

We now develop a widely-applicable variational method that overcomes weaknesses of prior work summarized in Sec. 3. We first adapt natural gradients to develop a REINFORCE-like estimator that leverages the local curvature of the ELBO. We then provably reduce estimator variance, and accelerate convergence, by marginalizing discrete latent variable z_i when estimating $\partial\mathcal{L}/\partial\tau_i$. Unlike SCAVI, our MSNG method allows variational parameters to be updated in parallel, and has convergence guarantees even when finite sample sets are used to approximate expectations.

4.1. Stochastic Natural Gradients

The natural gradient adjusts the direction of the traditional gradient by accounting for the information geometry of its parameter space, and leads to faster convergence in maximum likelihood estimation problems (Amari, 1998). We use *stochastic natural gradients* (SNG) to optimize the ELBO by multiplying the standard REINFORCE gradient with the inverse Fisher information matrix (Amari, 1982; Kullback & Leibler, 1951) of the variational distribution $q(z)$. For naive mean-field approximations, the Fisher information matrix is block-diagonal because the variational parameters associated with different variables are uncorrelated.

We first consider binary hidden variables z_i. The Fisher information matrix $F(\tau_i)$ of a Bernoulli distributon $q(z_i)$, with natural parameter τ_i defined as in Eq. (6), is

$$\begin{aligned} F(\tau_i) &= \mathbb{E}_{q(z_i)}\big[(\nabla_{\tau_i}\log q(z_i))(\nabla_{\tau_i}\log q(z_i))^T\big] \\ &= \sigma(\tau_i)\sigma(-\tau_i). \end{aligned} \tag{10}$$

By introducing the mean parameter μ_i, we compute the regular ELBO gradient with respect to τ_i via the chain rule:

$$\frac{\partial\mathcal{L}}{\partial\tau_i} = \frac{\partial\mathcal{L}}{\partial\mu_i}\frac{\partial\mu_i}{\partial\tau_i}, \text{ where } \frac{\partial\mathcal{L}}{\partial\mu_i} = \mathbb{E}_{q(z)}\Big[\frac{\partial\log q(z_i)}{\partial\mu_i}\cdot \tag{11}$$
$$\big(\log p(z_i \mid z_{-i}, x) - \log q(z_i)\big)\Big],\ \frac{\partial\mu_i}{\partial\tau_i} = \sigma(\tau_i)\sigma(-\tau_i).$$

When we compute the natural gradient, the Jacobian matrix $\frac{\partial\mu_i}{\partial\tau_i}$ in Eq. (11) cancels with the inverse of the Fisher information matrix $F(\tau_i)$ of Eq. (10). Then using a REINFORCE-like unbiased estimate for the gradient $\frac{\partial\mathcal{L}}{\partial\mu_i}$, our SNG ascent update for τ_i becomes

$$\tau_i^{\text{new}} = \tau_i + \alpha F^{-1}(\tau_i)\frac{\partial\mathcal{L}}{\partial\tau_i} = \tau_i + \alpha\frac{\partial\mathcal{L}}{\partial\mu_i} \tag{12}$$
$$\approx \tau_i + \frac{\alpha}{M}\sum_{m=1}^{M}\frac{\log p(z_i^{(m)} \mid z_{-i}^{(m)}, x) - \log q(z_i^{(m)})}{z_i^{(m)}\sigma(\tau_i) - (1-z_i^{(m)})\sigma(-\tau_i)}.$$

Here $z_i^{(m)} \sim q(z_i)$ and $z_{-i}^{(m)} \sim q(z_{-i})$ are M samples from the variational posterior, and α is the learning rate.

Stochastic natural gradient updates are guaranteed to converge to a (local) optimum of the ELBO, like REINFORCE and related stochastic gradient methods, because the inverse Fisher information matrix pre-multiplying the noisy gradient is positive definite (Bottou, 1998). Hoffman et al. (2013) also use natural gradients for variational inference, but consider a fundamentally different source of randomness. Our SNG method uses samples from the variational distribution to approximate intractable expectations for high-degree or non-conjugate models. Hoffman et al. (2013) instead consider conditionally conjugate models where CAVI updates have simple closed forms, but sample mini-batches of data to allow efficient learning from big datasets.

4.2. Variance Reduction and Connections to CAVI

Stochastic gradient descent can achieve the minimax optimal convergence rate for both convex and non-convex problems (Nemirovsky & Yudin, 1983; Ghadimi & Lan, 2012; Rakhlin et al., 2012; Ghadimi & Lan, 2013; Singer & Vondrák, 2015; Arjevani et al., 2019; Drori & Shamir, 2020). One of the most common assumptions made in these theoretical studies is that the gradient estimator $g(z,\tau)$ of function $\mathcal{L}(\tau)$ is unbiased and has bounded variance:

$$\mathbb{E}_z[g(z,\tau)] = \nabla\mathcal{L}(\tau), \tag{13}$$
$$\mathbb{E}_z[\|g(z,\tau) - \nabla\mathcal{L}(\tau)\|^2] \le \delta^2, \tag{14}$$

where Eq. (14) bounds the sum of variance across all the gradient dimensions. Various results then show that smaller δ values lead to guarantees of faster convergence rates.

We thus improve SNG by reducing the variance of the gradient estimator: when estimating the gradient for variational parameter τ_i, we analytically marginalize the corresponding discrete variable z_i. Via the Rao-Blackwell Theorem (Casella & Robert, 1996), this marginalization provably reduces the sampling variance of gradient estimation.

Again focusing on binary latent variables, we explicitly marginalize out z_i from the partial derivative in Eq. (11):

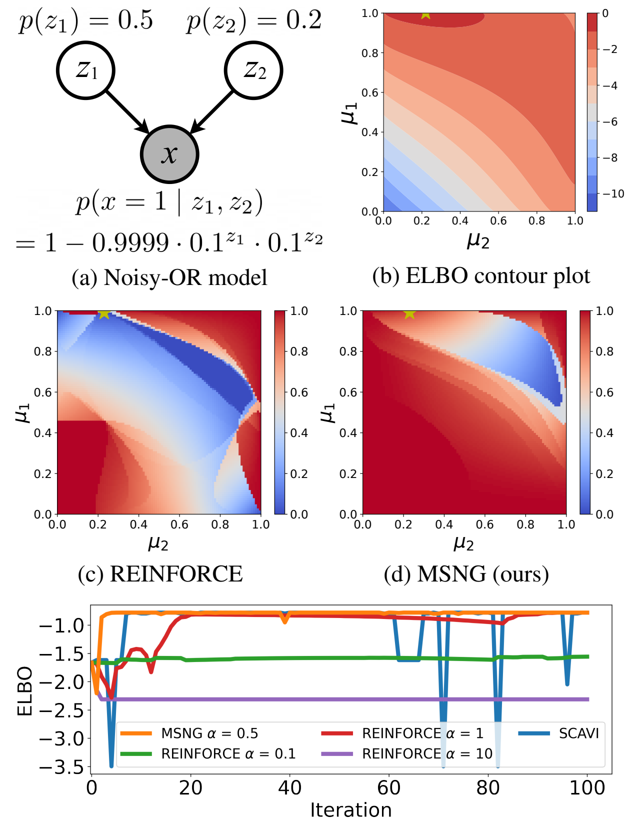
$$\begin{aligned} \frac{\partial\mathcal{L}}{\partial\mu_i} &= \mathbb{E}_{q(z)}\Big[\frac{\partial\log q(z_i)}{\partial\mu_i}\big(\log p(z_i \mid z_{-i}, x) - \log q(z_i)\big)\Big] \\ &= \frac{\mu_i}{\mu_i}\cdot\mathbb{E}_{q(z_{-i})}\big[\log p(z_i = 1 \mid z_{-i}, x) - \log\mu_i\big] + \\ &\quad \frac{1-\mu_i}{\mu_i - 1}\cdot\mathbb{E}_{q(z_{-i})}\big[\log p(z_i = 0 \mid z_{-i}, x) - \log(1-\mu_i)\big] \\ &= \mathbb{E}_{q(z_{-i})}\Big[\log\frac{p(z_i = 1 \mid z_{-i}, x)}{p(z_i = 0 \mid z_{-i}, x)}\Big] - \tau_i. \end{aligned} \tag{15}$$

Given this identity, our *marginalized stochastic natural gradient* (MSNG) variational update becomes:

$$\begin{aligned} \tau_i^{\text{new}} &= \tau_i + \alpha F^{-1}(\tau_i)\frac{\partial\mathcal{L}}{\partial\tau_i} = \tau_i + \alpha\frac{\partial\mathcal{L}}{\partial\mu_i} \\ &= \tau_i + \alpha\Big(\mathbb{E}_{q(z_{-i})}\Big[\log\frac{p(z_i = 1 \mid z_{-i}, x)}{p(z_i = 0 \mid z_{-i}, x)}\Big] - \tau_i\Big) \\ &= \alpha\mathbb{E}_{q(z_{-i})}\Big[\log\frac{p(z_i = 1 \mid z_{-i}, x)}{p(z_i = 0 \mid z_{-i}, x)}\Big] + (1-\alpha)\tau_i \quad (16) \\ &\approx \alpha\frac{1}{M}\sum_{m=1}^{M}\log\frac{p(z_i = 1 \mid z_{-i}^{(m)}, x)}{p(z_i = 0 \mid z_{-i}^{(m)}, x)} + (1-\alpha)\tau_i. \quad (17) \end{aligned}$$

Eq. (16) intuitively updates natural parameters via an average, with weights determined by the learning rate α, of the previous τ_i and the CAVI update (7). Standard CAVI must compute expectations exactly to ensure convergence, which may not be possible for models with many dependent continuous or discrete variables. In contrast, even with finite samples, the MSNG update of Eq. (17) gives unbiased estimates of stochastic natural gradients. Our gradient-based optimization perspective also justifies parallel variational parameter updates; while this may cause CAVI to diverge, MSNG is convergent with appropriate learning rates α. Thus like REINFORCE, MSNG updates all parameters in parallel and only requires pointwise evaluation of log-probabilities, enabling black-box variational inference.

Fig. 2 illustrates the advantages of MSNG over REINFORCE on a toy noisy-OR model. Based on the contour



(a) Noisy-OR model (b) ELBO contour plot

(c) REINFORCE (d) MSNG (ours)

(e) ELBO trace plot of MSNG, REINFORCE, and SCAVI

Figure 2. (a): Graphical illustration of a toy noisy-OR model with two latent variables z and one observation $x = 1$. (b): Contour plot of the model's ELBO as a function of μ_1 and μ_2. The yellow star indicates the global maximum. Although the likelihood is symmetric with respect to z_1 and z_2, the prior of z_1 is higher, so to explain $x = 1$, the optimal $\mu_1^* \approx 0.99$ and μ_2^* similar to its prior. (c): The probability of ELBO increase after a REINFORCE gradient update of Eq. (8). (d): The probability of ELBO increase after a MSNG update of Eq. (17). (e): ELBO trace plot of different methods, with initial $\mu_1 = 0.5, \mu_2 = 0.9$. In (c-e), the sampling budget $M = 2$, and the learning rate $\alpha = 0.5$ in (c) and (d).

plot of the ELBO in Fig. 2(b), this is a simple optimization problem with a global maximum indicated by the yellow star. However, due to its high variance, the noisy REINFORCE gradient is more likely to *decrease* the ELBO than increase it at every point in the blue region of Fig. 2(c). In contrast, the variance of MSNG in Fig. 2(d) is much smaller, and the promising red area around the global maximum makes it more likely to "attract" variational parameters towards the global optimum. The ELBO trace plot in Fig. 2(e) verifies this, where MSNG (orange) converges much faster than REINFORCE even with a well-tuned learning rate (red). We can see that many REINFORCE iterations with relatively high ELBO values actually *decrease* the ELBO, as predicted by the blue region around the optimum in Fig. 2(c). Notice that we adversarially pick our initialization from the blue region in Fig. 2(d), so the ELBOs of MSNG (and other

```python
import torch
from pyro import poutine

class MSNG:
    ...
    def step(self, *args, **kwargs):
        log_p = {}
        for group in self.variational_params:
            log_p[group] = torch.zeros_like(self.variational_params[group])

        for s in range(self.num_samples):
            # Sample z from the variational distribution
            guide_trace = poutine.trace(self.guide).get_trace(*args, **kwargs)

            # Iterate over each variable (each group is a tensor,
            # so variational_params[group][idx] is the actual individual variable)
            for group in self.variational_params:
                for idx in self._get_indices(group):
                    # Remember actual sampled value of z_i
                    k0 = 1 * guide_trace.nodes[group]['value'][idx]

                    for k in range(self._get_K(group)):
                        # Condition z_i = k
                        guide_trace.nodes[group]['value'][idx] = torch.tensor(k)
                        model = poutine.replay(self.model, trace=guide_trace)
                        model_trace = poutine.trace(model).get_trace(*args, **kwargs)

                        # Compute log p(z_i = k, z_j, x)
                        model_log_p = self._compute_log_p(model_trace, group, idx)
                        log_p[group][idx][k] += model_log_p

                    # Change value of z_i back to k0
                    guide_trace.nodes[group]['value'][idx] = k0

        for group in log_p:
            # Compute log odds ratio
            tau = log_p[group] / self.num_samples
            tau_K = tau.index_select(-1, torch.tensor(self._get_K(group) - 1))
            tau -= tau_K

            # Natural gradient ascent update
            tau0 = self.variational_params[group]
            self.variational_params[group] = (1 - lr) * tau0 + lr * tau
```

Figure 3. Pyro implementation of MSNG. This inference code works for any discrete-variable model specified by valid Pyro model and guide functions. See the supplement for further details.

algorithms) initially drop. But after only a few iterations, MSNG robustly increases the ELBO again. Fig. 2(e) also demonstrates that SCAVI (blue) lacks convergence guarantees: even after achieving high ELBO values, "unlucky" samples may regularly cause divergence.

4.3. Generalization to Categorical Variables

Our MSNG algorithm may be easily extended to general categorical latent variables z_i taking one of K discrete states. The natural parameter of $q(z_i)$ now becomes a vector $\tau_i \triangleq \left[\log \frac{\mu_{i1}}{\mu_{iK}}, \ldots, \log \frac{\mu_{ik}}{\mu_{iK}}, \ldots, \log \frac{\mu_{iK-1}}{\mu_{iK}}\right]$ of length $K - 1$. The MSNG update for each entry τ_{ik} is

$$\tau_{ik}^{\text{new}} = \alpha \frac{1}{M} \sum_{m=1}^{M} \log \frac{p(z_i = k \mid z_{-i}^{(m)}, x)}{p(z_i = K \mid z_{-i}^{(m)}, x)} + (1 - \alpha)\tau_{ik}.$$

A detailed derivation of this update is in the supplement, and Fig. 3 shows our general-purpose Pyro implementation.

Our categorical MSNG update again has connections to exact CAVI updates (5), and is equivalent to the binary MSNG update (17) when $K = 2$. Sato (2001) and Hoffman et al. (2013) discuss connections between natural gradients and CAVI for continuous, conditionally conjugate models.

To enable black-box application of our MSNG variational inference method, our experiments focus on models where all latent variables are discrete. While the SNG estimator is easily generalized to continuous latent variables, methods for (approximate) black-box marginalization of continuous variables is left as a promising area for future research.

5. Experiments

We compare our proposed MSNG algorithm with seven other variational methods: the non-marginalized stochastic natural gradients (SNG) of Sec. 4.1, standard score-function gradients (REINFORCE), their improved versions with control variates (SNG+CV and REINFORCE+CV), the heuristic SCAVI method that approximates CAVI expectations in Eq. (5) with Monte Carlo samples, the CONCRETE relaxation based on Gumbel-Max sampling, and model-specific auxiliary-variable methods (AUX, for binary models only). AUX methods are taken from prior work for noisy-OR (Ji et al., 2019) and sigmoid belief networks (Gan et al., 2015), and derived by us (extending Albert & Chib (1993)) for the binary relational model's probit likelihood.

We integrate the black-box methods MSNG, SNG(+CV), and SCAVI into Pyro to make fair comparisons with the already-supported REINFORCE(+CV) and CONCRETE. We use Pyro's standard decaying average baseline[2] to weight control variates, with the default decay rate of 0.9. MSNG is also compatible with control variates, but it does not need a baseline CV because the latent variable associated with the score function in each dimension has been marginalized. More experimental details are provided in the supplement.

5.1. Models and Datasets

Noisy-OR topic graphs for text data. Following Ji et al. (2019), we infer topic activations in a noisy-OR topic model of documents from the "tiny" version of the 20 newsgroups dataset collected by Sam Roweis. We use the same model architecture, which has 44 latent topic nodes within two layers, and 100 observed token nodes. The edge weights w are learned on the training set through the full-model variational training method, with auxiliary bounds for noisy-OR likelihoods, described in the original paper. Their values are then fixed as we compare the different variational inference algorithms on 100 randomly subsampled test documents.

Sigmoid belief networks for image data. On the binarized MNIST training dataset, we learn the edge weights w of a three-layer fully-connected sigmoid belief network using the public data-augmented variational training code by Gan et al. (2015). The top two layers each have 100 hidden nodes, and the observed bottom layer corresponds to the 28×28 pixels. Similar to noisy-OR topic model experiments, we fix the edge weight when testing the different inference algorithms on 100 images randomly selected from the MNIST test set. Fig. 1 shows Pyro specifications of our belief networks.

Relational models for network data. We apply the two relational models described in Sec. 2.2 to two network datasets used by Miller et al. (2009). The first dataset describes various relations between 14 countries between 1950 and 1965 (Rummel, 1976). We choose the "conference" relation, which consists of symmetric connections indicating if two countries co-participate in any international conference. The second dataset is about NeurIPS coauthorship (Globerson et al., 2007), where a link indicates two individuals being coauthors of a paper in one of the first 17 NeurIPS conferences. Following Miller et al. (2009) and Palla et al. (2012), we model the 234 most connected authors. Model parameters and Pyro specifications for these two models are provided in the supplement, as well as an AUX algorithm derived by us for the probit latent-feature model.

Annotation models for crowd-sourcing data. We test the annotation model of Passonneau & Carpenter (2014) on two publicly available Amazon Mechanical Turk datasets (Snow et al., 2008). The first dataset, on *Recognizing Textual Entailment* (RTE), has 800 questions and 164 annotators. For each question, the annotator is presented with two sentences to make a binary choice ($K = 2$) of whether the second hypothesis sentence can be inferred from the first. The second dataset, on *Word Sense Disambiguation* (WSD), has 177 questions and 34 annotators. The annotator is asked to choose the most appropriate sense of a particular word in a sentence out of $K = 3$ possible options.

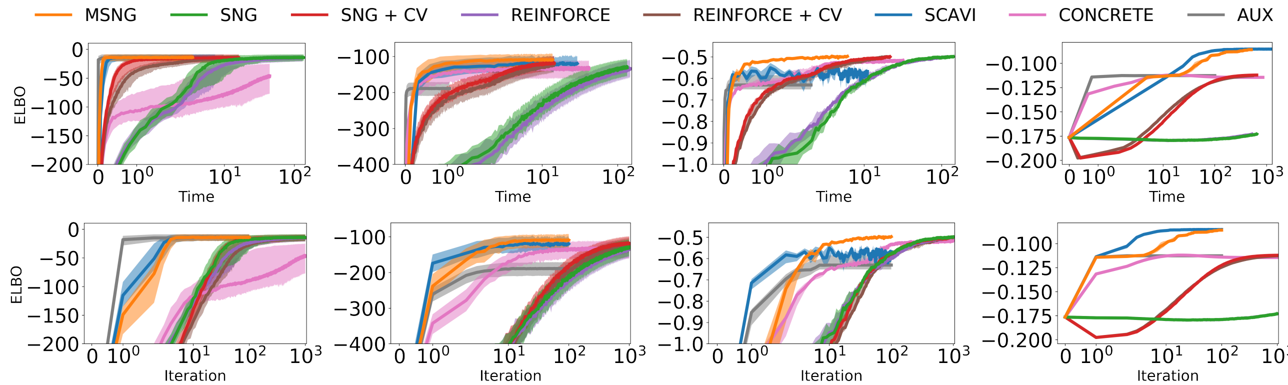
For the annotation model, we use the conjugacy of Dirichlet priors to analytically marginalize the unknown rater accuracy distributions θ_{jk}. This leads to a "collapsed" variational bound that depends only on the posteriors $q(z_i)$ for the true category of each question (item). Collapsing induces high-order dependencies that make classic CAVI updates intractable, but we show that our black-box MSNG updates are effective, while being simpler than previous collapsed variational inference algorithms (Teh et al., 2007). See the supplement for details about model hyperparameters.

5.2. Comparison of Variational Inference Results

Figs. 4 and 5 show the median ELBOs of different methods across repeated runs, where the randomness is caused by Monte Carlo sampling (for stochastic methods) and the order of parameter updates (for sequential methods). Like Gan et al. (2015), the ELBOs are evaluated via Monte Carlo sampling. The algorithms compared with MSNG can be split into four groups: unbiased gradient-based methods, the heuristic SCAVI method, model-dependent AUX methods, and the biased CONCRETE relaxation using continuous variables. For clarity, we focus on one at a time below.

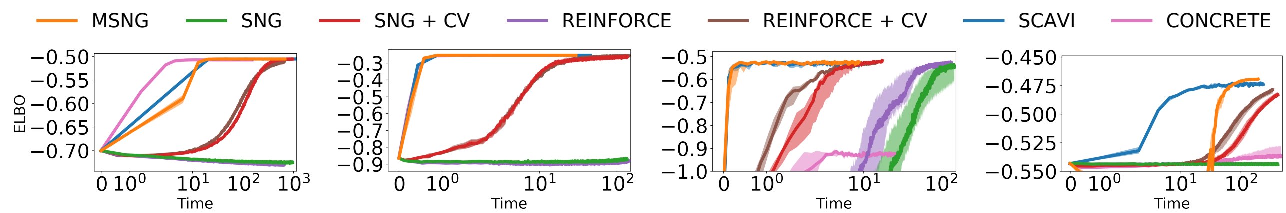
MSNG converges much faster, and requires fewer samples, than competing unbiased BBVI methods. The convergence speed of a stochastic gradient method is influenced by the learning rate and the sampling budget. We find that MSNG is less sensitive to learning rates, and for all but the probit relational model, we choose a fixed rate

[2] https://pyro.ai/examples/svi_part_iii.html#Decaying-Average-Baseline



(a) Newsgroups (noisy-OR) (b) MNIST (sigmoid) (c) Countries (probit) (d) NeurIPS (probit)

Figure 4. Improvement of ELBO (vertical axis) versus runtime (top row) and iteration (bottom row) on various models with binary variables. Lines show the median, while shaded regions show the 25th and 75th percentiles, across 10 repeated runs.



(a) RTE (annotation) (b) WSD (annotation) (c) Countries (SBM) (d) NeurIPS (SBM)

Figure 5. Improvement of ELBO (vertical axis) versus runtime on various models with categorical variables (performance versus iterations is in the supplement). Lines show the median, while shaded regions show the 25th and 75th percentiles, across 10 repeated runs.

via grid search. SNG and REINFORCE (with or without control variates) and CONCRETE show greater sensitivity, so as suggested by Ranganath et al. (2014) we use AdaGrad (Duchi et al., 2011) to adapt learning rates for SNG(+CV), REINFORCE(+CV), CONCRETE, and MSNG (probit only). For each method, we evaluate sample sizes of $M \in \{1, 10, 100\}$, and then report results for the variant that converges with lowest runtime. This leads to a sample size of $M = 1$ for MSNG; 100 for SNG and REINFORCE; and 10 (noisy-OR, sigmoid, Countries probit) or 100 (all else) for SNG+CV, REINFORCE+CV, and CONCRETE.

In Figs. 4 and 5, all methods share the same initialization, and we run them until convergence or for a maximum of 1,000 iterations. Across all eight model-dataset combinations, a general trend of optimization speed is MSNG $\gg$ SNG+CV $\geq$ REINFORCE+CV $\gg$ SNG $\geq$ REINFORCE. Note that plots use a logarithmic horizontal scale, and MSNG often converges tens or even hundreds of times faster than other stochastic gradient methods, in terms of both clock time and number of iterations. Marginalization and natural gradients are both important for peak performance.

In simple problems such as the deep belief networks and relational models of the small countries data, SNG converges faster than REINFORCE, showing the benefit of natural gradients. In harder cases like the annotation models and relational models of the larger NeurIPS data, neither is able to effectively improve the ELBO with a budget of $M = 100$ samples. Meanwhile, methods with control variate (SNG+CV and REINFORCE+CV) are able to make faster improvement to the ELBO, even with 10 times fewer samples. Finally, MSNG always converges much faster than the other methods, even with just 1 sample per iteration.

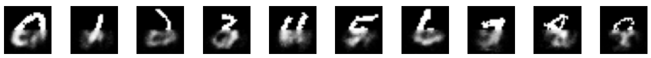
Fig. 6 visualizes MNIST digit completion results using the sigmoid belief network. REINFORCE and REINFORCE+CV are clearly worse than MSNG, with even 10 or 100 times more samples. This performance is consistent with ELBO values of different methods in Fig 4(b).

MSNG is more stable than SCAVI. With only one sample, the heuristic SCAVI method is also able to quickly improve the ELBO in the first few iterations. But its ELBO values in the final iterations are usually worse than MSNG, as shown in Fig. 4 and 5, as well as tables in the supplement that report detailed results for each dataset. Unlike MSNG, SCAVI is not guaranteed to converge, as shown in the examples in Fig. 7. We also observe that while SCAVI may increase the ELBO faster in the first few iterations, it often runs slower than MSNG when measured by the actual clock time. This is because SCAVI has to sequentially update variational parameters, but MSNG is able to compute the gradient updates for all variables in parallel.

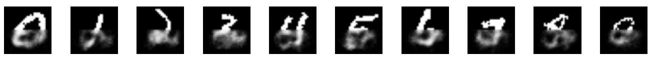
(a) Original images

(b) Digit completion: bottom parts of the images are missing

(c) MSNG (1 sample)



(d) REINFORCE (100 samples)



(e) REINFORCE + CV (10 samples)

(f) SCAVI (1 sample)

(g) CONCRETE (10 samples)

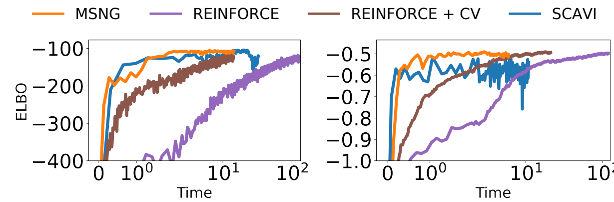
Figure 6. Digit completion results on binarized MNIST images. We use upper halves of the images as observations to infer $q(z)$, and fill in the lower halves by averaging 100 samples of x drawn from $q(z)p(x|z)$. The number of samples used during inference (shown in the parentheses after each method) is matched to the settings for Fig. 4(b). For all methods, the number of inference iterations is 50, and the initial value for each latent node is 0.5.

MSNG optimizes tighter bounds and is more robust to initialization than AUX. In Fig. 4(b-d), model-dependent AUX methods all converge to lower ELBO values than MSNG. This is likely because these methods all optimize looser variational bounds than the original ELBO.

Another possible reason is that by introducing more parameters into the objective, the optimization surface becomes more complicated, and algorithms become more likely to be trapped in local optima during sequential updates. Randomizing update order does not avoid this issue, as shown by the quantiles of AUX performance.

In an additional experiment reported in the supplement, as suggested by Gan et al. (2015), we use the marginal prior (as approximated by Monte Carlo) of each variable in the sigmoid BN to initialize $q(z)$. This engineering does make AUX converge to better ELBOs than the uniform initialization of Fig. 4(b), but MSNG is more robust and converges to superior ELBOs for both initializations.

Finally, compared to hand-crafted auxiliary-variable algorithms, the black-box property of MSNG allows for easy integration with PPL for convenient model selection. Users can easily and quickly fit different models on the same dataset, and pick the best performing candidate with the highest ELBO values. Take the results of the two relational



MNIST (sigmoid) Countries (probit)

Figure 7. ELBO trace plots of MSNG and other black-box variational algorithms. Unlike the curves in Fig. 4 that are summarized across data and repeated runs, the ELBO here is just for one MNIST image (left) and a single run of the probit relation model (right), in order to reveal the non-convergence of SCAVI.

models in Fig. 4(c-d) and Fig. 5(c-d) as an example. While the simple Countries data does not show a clear preference between the two models, the more complicated NeurIPS author data strongly favors the more powerful probit latent-feature model over the stochastic block model.

Biased CONCRETE updates achieve inferior ELBOs. CONCRETE updates more rapidly increase the ELBO than REINFORCE(+CV) in the first few iterations, but much of this advantage is lost when considering computation time. Unlike all other methods we consider, CONCRETE must (automatically) differentiate the model log-likelihood, which has time and memory overhead. CONCRETE is comparable to MSNG and SCAVI for the annotation model; we hypothesize this is because the likelihood depends only on histograms of many discrete variables, so continuous relaxations are more accurate. But for other models, perhaps due to its biased ELBO surrogate and sensitivity to the temperature hyperparameter, CONCRETE results are inconsistent and often dramatically inferior to MSNG. Supplement Tables D.1 and D.2 contain more detailed comparisons.

6. Discussion

We have developed marginalized stochastic natural gradients (MSNG) for black-box variational inference in probabilistic models with discrete latent variables. The MSNG method has better theoretical guarantees, and converges much faster, than REINFORCE. Unlike model-specific auxiliary methods, MSNG directly optimizes a tighter likelihood bound, and is more robust to initialization in spite of being simpler to derive and implement. While our experiments focused on models with only discrete variables, SNG updates are easily extended to models that mix discrete and continuous variables, and we are exploring applications to other model families. Our Pyro-integrated MSNG code provides a compelling method for scalable black-box variational inference.

Acknowledgements

This research supported in part by NSF CAREER Award No. IIS-1758028, NSF RI Award No. IIS-1816365, and a Facebook Probability and Programming research award. We thank Prof. Alexander Ihler for insightful suggestions in early stages of this work.

References

Albert, J. H. and Chib, S. Bayesian analysis of binary and polychotomous response data. *Journal of the American Statistical Association*, 88(422):669–679, 1993.

Amari, S.-I. Differential geometry of curved exponential families-curvatures and information loss. *The Annals of Statistics*, pp. 357–385, 1982.

Amari, S.-I. Natural gradient works efficiently in learning. *Neural computation*, 10(2):251–276, 1998.

Arjevani, Y., Carmon, Y., Duchi, J. C., Foster, D. J., Srebro, N., and Woodworth, B. Lower bounds for non-convex stochastic optimization. *arXiv preprint arXiv:1912.02365*, 2019.

Bingham, E., Chen, J. P., Jankowiak, M., Obermeyer, F., Pradhan, N., Karaletsos, T., Singh, R., Szerlip, P., Horsfall, P., and Goodman, N. D. Pyro: Deep universal probabilistic programming. *Journal of Machine Learning Research*, 20(1):973–978, 2019.

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

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

Bottou, L. Online learning and stochastic approximations. *On-line learning in neural networks*, 17(9):142, 1998.

Carpenter, B., Gelman, A., Hoffman, M. D., Lee, D., Goodrich, B., Betancourt, M., Brubaker, M. A., Guo, J., Li, P., and Riddell, A. Stan: A probabilistic programming language. *Journal of Statistical Software*, 76(1): 1–32, 2017.

Casella, G. and Robert, C. P. Rao-Blackwellisation of sampling schemes. *Biometrika*, 83(1):81–94, 1996.

Cusumano-Towner, M. F., Saad, F. A., Lew, A. K., and Mansinghka, V. K. Gen: A general-purpose probabilistic programming system with programmable inference. In *Conference on Programming Language Design and Implementation*, 2019.

Dong, Z., Mnih, A., and Tucker, G. Disarm: An antithetic gradient estimator for binary latent variables. In *Advances in Neural Information Processing Systems*, 2020.

Drori, Y. and Shamir, O. The complexity of finding stationary points with stochastic gradient descent. In *International Conference on Machine Learning*, 2020.

Duchi, J., Hazan, E., and Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7), 2011.

Gan, Z., Henao, R., Carlson, D., and Carin, L. Learning deep sigmoid belief networks with data augmentation. In *Artificial Intelligence and Statistics*, 2015.

Ghadimi, S. and Lan, G. Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization I: A generic algorithmic framework. *SIAM Journal on Optimization*, 22(4):1469–1492, 2012.

Ghadimi, S. and Lan, G. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.

Globerson, A., Chechik, G., Pereira, F., and Tishby, N. Euclidean embedding of co-occurrence data. *Journal of Machine Learning Research*, 8:2265–2295, 2007.

Goodman, N. D. and Stuhlmüller, A. The Design and Implementation of Probabilistic Programming Languages. http://dippl.org, 2014.

Gopalan, P., Hao, W., Blei, D. M., and Storey, J. D. Scaling probabilistic models of genetic variation to millions of humans. *Nature genetics*, 48(12):1587, 2016.

Gopalan, P. K. and Blei, D. M. Efficient discovery of overlapping communities in massive networks. *Proceedings of the National Academy of Sciences*, 110(36):14534–14539, 2013.

Grathwohl, W., Choi, D., Wu, Y., Roeder, G., and Duvenaud, D. Backpropagation through the void: Optimizing control variates for black-box gradient estimation. In *International Conference on Learning Representations*, 2018.

Greensmith, E., Bartlett, P. L., and Baxter, J. Variance reduction techniques for gradient estimates in reinforcement learning. *Journal of Machine Learning Research*, 5: 1471–1530, 2004.

Hoffman, M. D., Blei, D. M., Wang, C., and Paisley, J. Stochastic variational inference. *Journal of Machine Learning Research*, 14(1):1303–1347, 2013.

Holland, P. W., Laskey, K. B., and Leinhardt, S. Stochastic blockmodels: First steps. *Social networks*, 5(2):109–137, 1983.

Horvitz, E. J., Breese, J. S., and Henrion, M. Decision theory in expert systems and artificial intelligence. *International Journal of Approximate Reasoning*, 2(3):247–302, 1988.

Jaakkola, T. S. and Jordan, M. I. Variational probabilistic inference and the QMR-DT network. *Journal of Artificial Intelligence Research*, 10:291–322, 1999.

Jang, E., Gu, S., and Poole, B. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations*, 2017.

Ji, G., Cheng, D., Ning, H., Yuan, C., Zhou, H., Xiong, L., and Sudderth, E. B. Variational training for large-scale noisy-OR Bayesian networks. In *Conference on Uncertainty in Artificial Intelligence*, 2019.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. An introduction to variational methods for graphical models. *Machine Learning*, 37(2):183–233, 1999.

Kakade, S. M. A natural policy gradient. *Advances in Neural Information Processing Systems*, 2001.

Kingma, D. P. and Welling, M. Auto-encoding variational Bayes. In *International Conference on Learning Representations*, 2014.

Kucukelbir, A., Tran, D., Ranganath, R., Gelman, A., and Blei, D. M. Automatic differentiation variational inference. *Journal of Machine Learning Research*, 18(1): 430–474, 2017.

Kullback, S. and Leibler, R. A. On information and sufficiency. *The annals of mathematical statistics*, 22(1): 79–86, 1951.

Liu, J., Ren, X., Shang, J., Cassidy, T., Voss, C. R., and Han, J. Representing documents via latent keyphrase inference. In *International Conference on World Wide Web*, 2016.

Liu, R., Regier, J., Tripuraneni, N., Jordan, M. I., and McAuliffe, J. Rao-Blackwellized stochastic gradients for discrete distributions. In *International Conference on Machine Learning*, 2019.

Maddison, C. J., Mnih, A., and Teh, Y. W. The concrete distribution: A continuous relaxation of discrete random variables. In *International Conference on Learning Representations*, 2017.

Miller, K., Jordan, M. I., and Griffiths, T. L. Nonparametric latent feature models for link prediction. In *Advances in Neural Information Processing Systems*, 2009.

Murphy, K. P. *Machine Learning: A Probabilistic Perspective*. MIT press, 2012.

Neal, R. M. Connectionist learning of belief networks. *Artificial Intelligence*, 56(1):71–113, 1992.

Nemirovsky, A. S. and Yudin, D. B. *Problem complexity and method efficiency in optimization*. Society for Industrial and Applied Mathematics, 1983.

Paisley, J. W., Blei, D. M., and Jordan, M. I. Variational Bayesian inference with stochastic search. In *International Conference on Machine Learning*, 2012.

Palla, K., Knowles, D. A., and Ghahramani, Z. An infinite latent attribute model for network data. In *International Conference on Machine Learning*, 2012.

Papandreou, G. and Yuille, A. L. Perturb-and-map random fields: Using discrete optimization to learn and sample from energy models. In *International Conference on Computer Vision*, 2011.

Passonneau, R. J. and Carpenter, B. The benefits of a model of annotation. *Transactions of the Association for Computational Linguistics*, 2:311–326, 2014.

Polson, N. G., Scott, J. G., and Windle, J. Bayesian inference for logistic models using Pólya–Gamma latent variables. *Journal of the American statistical Association*, 108(504):1339–1349, 2013.

Rakhlin, A., Shamir, O., and Sridharan, K. Making gradient descent optimal for strongly convex stochastic optimization. In *International Conference on Machine Learning*, 2012.

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

Rezende, D. J., Mohamed, S., and Wierstra, D. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*, 2014.

Ritchie, D., Horsfall, P., and Goodman, N. D. Deep amortized inference for probabilistic programs. *arXiv preprint arXiv:1610.05735*, 2016.

Rummel, R. J. *Attributes of nations and behavior of nation dyads, 1950-1965*. Inter-university Consortium for Political Research, 1976.

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

Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. Trust region policy optimization. In *International Conference on Machine Learning*, 2015.

Shwe, M. A., Middleton, B., Heckerman, D. E., Henrion, M., Horvitz, E. J., Lehmann, H. P., and Cooper, G. F. Probabilistic diagnosis using a reformulation of the INTERNIST-1/QMR knowledge base. *Methods of Information in Medicine*, 30(4):241–255, 1991.

Singer, Y. and Vondrák, J. Information-theoretic lower bounds for convex optimization with erroneous oracles. In *Advances in Neural Information Processing Systems*, 2015.

Šingliar, T. and Hauskrecht, M. Noisy-OR component analysis and its application to link analysis. *Journal of Machine Learning Research*, 7:2189–2213, 2006.

Snow, R., O'connor, B., Jurafsky, D., and Ng, A. Y. Cheap and fast–but is it good? evaluating non-expert annotations for natural language tasks. In *Empirical Methods in Natural Language Processing*, 2008.

Teh, Y. W., Newman, D., and Welling, M. A collapsed variational Bayesian inference algorithm for latent Dirichlet allocation. In *Advances in Neural Information Processing Systems*, 2007.

Thomas, V., Pedregosa, F., Merriënboer, B., Manzagol, P.-A., Bengio, Y., and Le Roux, N. On the interplay between noise and curvature and its effect on optimization and generalization. In *Artificial Intelligence and Statistics*, 2020.

Titsias, M. K. and Lázaro-Gredilla, M. Local expectation gradients for black box variational inference. In *Advances in Neural Information Processing Systems*, 2015.

Tran, D., Kucukelbir, A., Dieng, A. B., Rudolph, M., Liang, D., and Blei, D. M. Edward: A library for probabilistic modeling, inference, and criticism. *arXiv preprint arXiv:1610.09787*, 2016.

Tran, D., Hoffman, M. W., Moore, D., Suter, C., Vasudevan, S., and Radul, A. Simple, distributed, and accelerated probabilistic programming. In *Advances in Neural Information Processing Systems*, 2018.

Tucker, G., Mnih, A., Maddison, C. J., Lawson, J., and Sohl-Dickstein, J. Rebar: Low-variance, unbiased gradient estimates for discrete latent variable models. In *Advances in Neural Information Processing Systems*, 2017.

Wainwright, M. J. and Jordan, M. I. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1:1–305, 2008.

Williams, R. J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3-4):229–256, 1992.

Wingate, D. and Weber, T. Automated variational inference in probabilistic programming. *arXiv preprint arXiv:1301.1299*, 2013.

Winn, J. and Bishop, C. M. Variational message passing. *Journal of Machine Learning Research*, 6:661–694, 2005.

Ye, L., Beskos, A., De Iorio, M., and Hao, J. Monte Carlo co-ordinate ascent variational inference. *Statistics and Computing*, pp. 1–19, 2020.

Yellott Jr, J. I. The relationship between Luce's choice axiom, Thurstone's theory of comparative judgment, and the double exponential distribution. *Journal of Mathematical Psychology*, 15(2):109–144, 1977.

Yin, M. and Zhou, M. ARM: Augment-REINFORCE-merge gradient for stochastic binary networks. In *International Conference on Learning Representations*, 2019.

Zhang, C., Bütepage, J., Kjellström, H., and Mandt, S. Advances in variational inference. *Pattern Analysis and Machine Intelligence*, 41(8):2008–2026, 2018.