Variational Boosting: Iteratively Refining Posterior Approximations

Andrew C. Miller [1] **Nicholas J. Foti** [2] **Ryan P. Adams** [1 3]

Abstract

We propose a black-box variational inference method to approximate intractable distributions with an increasingly rich approximating class. Our method, *variational boosting*, iteratively refines an existing variational approximation by solving a sequence of optimization problems, allowing a trade-off between computation time and accuracy. We expand the variational approximating class by incorporating additional covariance structure and by introducing new components to form a mixture. We apply variational boosting to synthetic and real statistical models, and show that the resulting posterior inferences compare favorably to existing variational algorithms.

1. Introduction

Variational inference (VI) is a family of methods to approximate an intractable *target* distribution (typically known only up to a constant) with a tractable *surrogate* distribution (Blei et al., 2016; Jordan et al., 1999; Wainwright & Jordan, 2008). VI procedures typically minimize the Kullback-Leibler (KL) divergence between the approximation and target distributions by maximizing a tractable lower bound on the marginal likelihood. The approximating family is often fixed, and typically excludes the neighborhood surrounding the target distribution, which prevents the approximation from becoming arbitrarily close to the true posterior. In the context of Bayesian inference, this mismatch between the variational family and the true posterior often manifests as underestimating the posterior variances of the model parameters and the inability to capture posterior correlations (Wainwright & Jordan, 2008).

An alternative approach to posterior inference uses Markov chain Monte Carlo (MCMC) methods that approximate a target distribution with samples drawn from a Markov chain constructed to admit the target distribution as the stationary distribution. MCMC enables a trade-off between computation and accuracy: drawing more samples makes the approximation closer to the target distribution. However, MCMC algorithms typically must be run iteratively and it can be difficult to assess convergence to the true target. Furthermore, correctly specifying MCMC moves can be more algorithmically restrictive than optimization-based approaches.

To alleviate the mismatch between tractable variational approximations and complicated posterior distributions, we propose a variational inference method that *iteratively* allows the approximating family of distributions to become more complex. Under certain conditions, the proposed approximations are eventually expressive enough to represent the true target arbitrarily well (though we do not prove our algorithm attains such a universal approximation here). Thus, the practitioner can trade time fitting a posterior approximation for increased accuracy of posterior estimates. Our algorithm grows the complexity of the approximating class in two ways: 1) incorporating rich covariance structure, and 2) sequentially adding new components to the approximating distribution. Our method builds on black-box variational inference methods using the *re-parameterization trick* by adapting it to be used with mixture distributions. This allows our method to be applied to a variety of target distributions including those arising from non-conjugate model specifications (Kingma & Welling, 2014; Ranganath et al., 2014; Salimans & Knowles, 2013). We demonstrate empirically that our algorithm improves posterior estimates over other variational methods for several practical Bayesian models.

2. Variational Inference

Given a *target distribution* with density[1] $\pi(x)$ for a *continuous* random variable $x \in \mathcal{X} \subseteq \mathbb{R}^D$, variational inference approximates $\pi(x)$ with a tractable distribution, $q(x; \lambda)$, from which we can efficiently draw samples and form sample-based estimates of functions of x. Variational methods minimize the KL-divergence, $\mathrm{KL}(q||\pi)$, between $q(\cdot; \lambda)$ and the true π as a function of *variational parameter* λ (Bishop, 2006). Although direct minimization of

[1]Harvard University, Cambridge, MA, USA [2]University of Washington, Seattle, WA, USA [3]Google Brain, Cambridge, MA, USA. Correspondence to: Andrew C. Miller <acm@seas.harvard.edu>, Nicholas J. Foti <nfoti@uw.edu>.

Proceedings of the 34^{th} International Conference on Machine Learning, Sydney, Australia, PMLR 70, 2017. Copyright 2017 by the author(s).

[1] We assume $\pi(x)$ is known up to a constant, $\tilde{\pi}(x) = \mathcal{C}\pi(x)$.

$\mathrm{KL}(q||\pi)$ is often intractable, we can derive a tractable objective based on properties of the KL-divergence. This objective is known as the *evidence lower bound* (ELBO):

$$\begin{aligned}\mathcal{L}(\lambda) &= \mathbb{E}_{q_\lambda}\left[\ln \pi(x) - \ln q(x;\lambda)\right] + \ln \mathcal{C} \\ &= \ln \mathcal{C} - \mathrm{KL}(q_\lambda||\pi) \leq \ln \mathcal{C} = \ln \int \tilde{\pi}(x)dx\end{aligned}$$

which, due to the positivity of $\mathrm{KL}(q||\pi)$, is a lower bound on $\mathcal{C} = \log \pi(x)$, i.e., the marginal likelihood.

Variational methods typically fix a family of distributions $Q = \{q(\cdot;\lambda) : \lambda \in \Lambda\}$ parameterized by λ, and *maximize* the ELBO with respect to $\lambda \in \Lambda$. Often there exists some (possibly non-unique) $\lambda^* \in \Lambda$ for which $\mathrm{KL}(q||\pi)$ is minimized. However, when the family Q does not include π then $\mathrm{KL}(q_{\lambda^*}||\pi) > 0$ which will result in biased estimates of functions $f(x)$, $\mathbb{E}_{x\sim q_{\lambda^*}}[f(x)] \neq \mathbb{E}_{x\sim\pi}[f(x)]$.

The primary alternative to variational methods for approximate inference is Markov chain Monte Carlo (MCMC), which constructs a Markov chain such that the target distribution remains invariant. Expectations with respect to the target distribution can be calculated as an average with respect to these correlated samples. MCMC typically enjoys nice asymptotic properties; as the number of samples grows, MCMC samplers represent the true target distribution with increasing fidelity. However, rules for constructing correct Markov steps are restrictive. With a few exceptions, most MCMC algorithms require evaluating a log-likelihood that touches all data at each step in the chain (Maclaurin & Adams, 2014; Welling & Teh, 2011). This becomes problematic during statistical analyses of large amounts of data — MCMC is often considered unusable because of this computational bottleneck. Notably, variational methods can avoid this bottleneck by sub-sampling the data (Ranganath et al., 2016a), as unbiased estimates of the *log-likelihood* can often be straight-forwardly used with optimization methods.

In the next section, we propose an algorithm that iteratively grows the approximating class Q and reframes the VI procedure as a series of optimization problems, resulting in a practical inference method that can both represent arbitrarily complex distributions and scale to large data sets.

3. Variational Boosting

We define our class of approximating distributions to be mixtures of C simpler component distributions:

$$q^{(C)}(x;\lambda,\rho) = \sum_{c=1}^{C} \rho_c q_c(x;\lambda_c)\,,\ \text{s.t. } \rho_c \geq 0, \sum_c \rho_c = 1,$$

where we denote the full mixture as $q^{(C)}$, mixing proportions $\rho = (\rho_1, \ldots, \rho_C)$, and component distributions $q_c(\cdot;\lambda_c)$ parameterized by $\lambda = (\lambda_1, \ldots, \lambda_C)$. The $q_c(\cdot;\lambda_c)$s can be any distribution over $\mathcal{X} \subseteq \mathbb{R}^D$ from which we can efficiently draw samples using a continuous mapping parameterized by λ_c (e.g., multivariate normal (Jaakkola & Jordan, 1998), or a composition of invertible maps (Rezende & Mohamed, 2015)).

When posterior expectations and variances are of interest, mixture distributions provide tractable summaries. Expectations are easily expressed in terms of component expectations:

$$\mathbb{E}_{q^{(C)}}\left[f(x)\right] = \int q^{(C)}(x)f(x)dx = \sum_c \rho_c \mathbb{E}_{q_c}\left[f(x)\right].$$

In the case of multivariate normal components, the mean and covariance of a mixture are easy to compute, as are marginal distributions along any set of dimensions.

Variational boosting (VBoost) begins with a single mixture component, $q^{(1)}(x;\lambda) = q_1(x;\lambda_1)$ with $C = 1$. We fix $\rho_1 = 1$ and use existing black-box variational inference methods to fit the first component parameter, λ_1. At the next iteration $C = 2$, we fix λ_1 and introduce a new component into the mixture, $q_2(x;\lambda_2)$. We define a new ELBO objective as a function of new component parameters, λ_2, and a new mixture weight, ρ_2. We then optimize this objective with respect to λ_2 and ρ_2 until convergence. At each subsequent round, c, we introduce new component parameters and a mixing weight, (λ_c, ρ_c), which are then optimized according to a new ELBO objective. The name *variational boosting* is inspired by methods that iteratively construct strong learners from ensembles of weak learners. We apply VBoost to target distributions via black-box variational inference with the *re-parameterization trick* to fit each component and mixture weights (Kingma & Welling, 2014; Ranganath et al., 2014; Salimans & Knowles, 2013). However, using mixtures as the variational approximation complicates the use of the re-parameterization trick.

3.1. The re-parameterization trick and mixtures

The re-parameterization trick is used to compute an unbiased estimate of the gradient of an objective that is expressed as an intractable expectation with respect to a continuous-valued random variable. This situation arises in variational inference when the ELBO cannot be evaluated analytically. We form an unbiased estimate as:

$$\begin{aligned}\mathcal{L}(\lambda) &= \mathbb{E}_q\left[\ln \pi(x) - \ln q(x;\lambda)\right] &(1)\\ &\approx \frac{1}{L}\sum_{\ell=1}^{L}\left[\ln \pi(x^{(\ell)}) - \ln q(x^{(\ell)};\lambda)\right] &(2)\end{aligned}$$

where $x^{(\ell)} \sim q(x;\lambda)$. To obtain a Monte Carlo estimate of the gradient of $\mathcal{L}(\lambda)$ using the re-parameterization trick, we first separate the randomness needed to generate $x^{(\ell)}$ from the parameters λ, by defining a deterministic map $x^{(\ell)} \triangleq f_q(\epsilon;\lambda)$ such that $\epsilon \sim p(\epsilon)$ implies

$x^{(\ell)} \sim q(x;\lambda)$. Note that $p(\epsilon)$ does not depend on λ. We then differentiate Eq. (2) with respect to λ through the map f_q to obtain an estimate of $\nabla_\lambda \mathcal{L}(\lambda)$.

When $q(\cdot;\lambda)$ is a mixture, applying the re-parameterization trick is not straightforward. The typical sampling procedure for a mixture model includes a discrete random variable that indicates a mixture component, which complicates differentiation. We circumvent this by re-writing the variational objective as a weighted combination of expectations with respect to individual mixture components:

$$\begin{aligned}\mathcal{L}(\lambda,\rho) &= \int \left(\sum_{c=1}^{C} \rho_c q_c(x;\lambda_c)\right) [\ln \pi(x) - \ln q(x;\lambda)]\, dx \\ &= \sum_{c=1}^{C} \rho_c \int q_c(x;\lambda_c)\, [\ln \pi(x) - \ln q(x;\lambda)]\, dx \\ &= \sum_{c=1}^{C} \rho_c \mathbb{E}_{q_c} [\ln \pi(x) - \ln q(x;\lambda)]\end{aligned}$$

which is a weighted sum of component-specific ELBOs. If the q_c are continuous and there exists some function $f_c(\epsilon;\lambda)$ such that $x = f_c(\epsilon;\lambda)$ and $x \sim q_c(\cdot;\lambda)$ when $\epsilon \sim p(\epsilon)$, then we can apply the re-parameterization trick to each component to obtain gradients of the ELBO :

$$\begin{aligned}\nabla_{\lambda_c} \mathcal{L}(\lambda,\rho) &= \nabla_{\lambda_c} \sum_{c=1}^{C} \rho_c \mathbb{E}_{x\sim q(x;\lambda)} [\ln \pi(x) - \ln q(x;\lambda)] \\ &= \sum_{c=1}^{C} \rho_c \mathbb{E}_{\epsilon \sim p(\epsilon)} \big[\nabla_{\lambda_c} \ln \pi(f_c(\epsilon;\lambda_c)) \\ &\qquad\qquad - \nabla_{\lambda_c} \ln q(f_c(\epsilon;\lambda_c);\lambda)\big] .\end{aligned}$$

VBoost leverages the above formulation of $\nabla_{\lambda_c}\mathcal{L}(\lambda,\rho)$ to use the re-parameterization trick in a component-by-component manner, allowing us to improve the variational approximation as we incorporate new components.

3.2. Incorporating New Components

Next, we describe how to incrementally add components during the VBoost procedure.

The first component VBoost starts by fitting a approximation to $\pi(x)$ consisting of a single component, q_1. We do this by maximizing the first ELBO objective

$$\mathcal{L}^{(1)}(\lambda_1) = \mathbb{E}_{q_1} [\ln \pi(x) - \ln q_1(x;\lambda_1)] \quad (3)$$

$$\lambda_1^* = \arg\max_{\lambda_1} \mathcal{L}^{(1)}(\lambda_1) . \quad (4)$$

Depending on the forms of π and q_1, optimizing $\mathcal{L}^{(1)}$ can be accomplished by various methods — an obvious choice being black-box VI with the re-parameterization trick. After convergence we fix λ_1 to be λ_1^*.

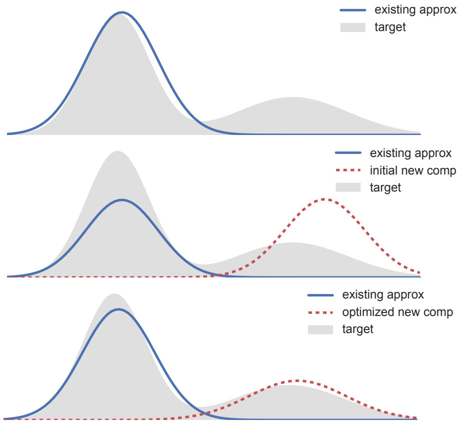


Figure 1. One-dimensional illustration of the VBoost procedure. *Top*: Initial single-component approximation (solid blue). *Middle*: A new component (dotted red) is initialized. *Bottom*: New component parameters and mixing weights are optimized using Monte Carlo gradients of the ELBO. Note that the mass of the existing components can rise and fall, but not shift in space.

Component $C+1$ After iteration C, our current approximation to $\pi(x)$ is a mixture distribution with C components:

$$q^{(C)}(x;\lambda,\rho) = \sum_{c=1}^{C} \rho_c q_c(x;\lambda_c). \quad (5)$$

Adding a component to Eq. (5) introduces a new component parameter, λ_{C+1}, and a new mixing weight, ρ_{C+1}. In this section, the mixing parameter $\rho_{C+1} \in [0,1]$ mixes between the new component, $q_{C+1}(\cdot;\lambda_{C+1})$ and the existing approximation, $q^{(C)}$. The new approximate distribution is

$$\begin{aligned}&q^{(C+1)}(x;\lambda,\rho) \\ &= (1-\rho_{C+1}) q^{(C)}(x) + \rho_{C+1} q_{C+1}(x;\lambda_{C+1}) .\end{aligned}$$

The new ELBO, as a function of ρ_{C+1} and λ_{C+1}, is:

$$\begin{aligned}&\mathcal{L}^{(C+1)}(\rho_{C+1},\lambda_{C+1}) \\ &= \mathbb{E}_{x\sim q^{(C+1)}} \left[\ln \pi(x) - \ln q^{(C+1)}(x;\lambda_{C+1},\rho_{C+1})\right] \\ &= (1-\rho_{C+1}) \mathbb{E}_{q^{(C)}} \left[\ln \pi(x) - \ln q^{(C+1)}(x;\lambda_{C+1},\rho_{C+1})\right] \\ &\quad + \rho_{C+1} \mathbb{E}_{q_{C+1}} \left[\ln \pi(x) - \ln q^{(C+1)}(x;\lambda_{C+1},\rho_{C+1})\right] .\end{aligned}$$

Crucially, we have separated out two expectations: one with respect to the existing approximation, $q^{(C)}$ (which is fixed), and the other with respect to the new component distribution, q_{C+1}. Because we have fixed $q^{(C)}$, we only need to optimize the new component parameters, λ_{C+1} and ρ_{C+1}, allowing us to use the re-parameterization trick to obtain gradients of $\mathcal{L}^{(C+1)}$. Note that evaluating the gradient requires sampling from the existing components which may result in larger variance than typical black-box variational methods. To mitigate the extra variance we use many samples to estimate the gradient and leave variance reduction to future work.

Figure 1 illustrates the algorithm on a simple one-dimensional example — the initialization of a new component and the resulting mixture after optimizing the second objective, $\mathcal{L}^{(2)}(\rho_2, \lambda_2)$. Figure 2 depicts the result of VBoost on a two-dimensional, multi-modal target distribution. In both cases, the component distributions are Gaussians with diagonal covariance.

3.3. Structured Multivariate Normal Components

Though our method can use any component distribution that can be sampled using a continuous mapping, a sensible choice of component distribution is a multivariate normal

$$\begin{aligned} q(x;\lambda) &= \mathcal{N}(x; \mu_\lambda, \Sigma_\lambda) \\ &= |2\pi\Sigma_\lambda|^{-1/2} \exp\left(-\tfrac{1}{2}(x-\mu_\lambda)^\intercal \Sigma_\lambda^{-1}(x-\mu_\lambda)\right) \end{aligned}$$

where the variational parameter λ is transformed into a mean vector μ_λ and covariance matrix Σ_λ.

Specifying the structure of the covariance matrix is a choice that largely depends on the dimensionality of $\mathcal{X} \subseteq \mathbb{R}^D$ and the correlation structure of the target distribution. A common choice of covariance is a diagonal matrix, $\Sigma_\lambda = \texttt{diag}(\sigma_1^2, \ldots, \sigma_D^2)$, which implies that x is independent across dimensions. When the approximation only consists of one component, this structure is commonly referred to as the *mean field* family. While computationally efficient, mean field approximations cannot model posterior correlations, which often leads to underestimation of marginal variances. Additionally, when diagonal covariances are used as the component distributions in Eq. (5) the resulting mixture may require a large number of components to represent the strong correlations (see Fig. 2). Furthermore, independence constraints can actually introduce local optima in the variational objective (Wainwright & Jordan, 2008).

On the other end of the spectrum, we can parameterize the entire covariance matrix using the Cholesky decomposition, L, such that $LL^\intercal = \Sigma$. This allows Σ to be any positive semi-definite matrix, enabling q to have the full flexibility of a D-dimensional multivariate normal distribution. However, this introduces $D(D+1)/2$ parameters, which can be computationally cumbersome when D is even moderately large. Furthermore, only a few pairs of variables may exhibit posterior correlations, particularly in multi-level models or neural networks where different parameter types may be nearly independent in the posterior.

As such, we would like to incorporate *some* capacity to capture correlations between dimensions of x without overparameterizing the approximation. The next subsection discusses a covariance specification that provides this tradeoff, while remaining computationally tractable.

Low-rank plus diagonal covariance Black-box variational inference methods with the re-parameterization trick require sampling from the variational distribution and efficiently computing (or approximating) the entropy of the variational distribution. For multivariate normal distributions, the entropy is a function of the determinant of the covariance matrix, Σ, while computing the log likelihood requires computing Σ^{-1}. When the dimensionality of the target, D, is large, computing determinants and inverses will have $O(D^3)$ time complexity and therefore may be prohibitively expensive to compute at every iteration.

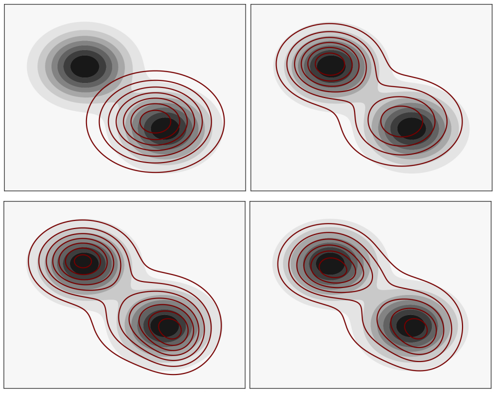


Figure 2. Sequence of increasingly complex approximate posteriors, with $C = 1, 2, 3, 4$ isotropic Gaussian components. The background (grey/black) contours depict the target distribution, and the foreground (red) contours depict the approximations.

However, it may be unnecessary to represent all $D(D-1)/2$ possible correlations in the target distribution, particularly if certain dimensions are close to independent. One way to increase the capacity of $q(x;\lambda)$ is to model the covariance as a *low-rank plus diagonal* (LR+D) matrix

$$\Sigma = FF^\intercal + \text{diag}(\exp(v)) \tag{6}$$

where $F \in \mathbb{R}^{D \times r}$ is a matrix of off diagonal factors, and $v \in \mathbb{R}^D$ is the log-diagonal component. This is effectively approximating the target via a *factor analysis* model.

The choice of the rank r presents a tradeoff: with a larger rank, the variational approximation can be more flexible; with a lower rank, the computations necessary for fitting the variational approximation are more efficient. As a concrete example, in Section 4 we present a $D = 37$ dimensional posterior resulting from a non-conjugate hierarchical model, and we show that a "rank $r = 2$ plus diagonal" covariance does an excellent job capturing all $D(D-1)/2 = 780$ pairwise correlations and D marginal variances. Incorporating more components using the VBoost framework further improves the approximation of the distribution.

To use the re-parameterization trick with this low rank covariance, we can simulate from q in two steps

$$\begin{gathered} z^{(\text{lo})} \sim \mathcal{N}(0, I_r) \qquad z^{(\text{hi})} \sim \mathcal{N}(0, I_D) \\ x = Fz^{(\text{lo})} + \mu + \mathcal{I}(v/2) z^{(\text{hi})} \end{gathered}$$

where $z^{(\text{lo})}$ generates the randomness due to the low-rank structure, and $z^{(\text{hi})}$ generates the randomness due to the diagonal structure. We use the operator $\mathcal{I}(a) = \text{diag}(\exp(a))$ for notational brevity. This generative process can be differentiated, yielding Monte Carlo estimates of the gradient with respect to F and v suitable for stochastic optimization.

In order to use LR+D covariance structure within VBoost, we will need to efficiently compute the determinant and inverse of Σ. The matrix determinant lemma expresses the determinant of Σ as the product of two determinants

$$\begin{aligned}|FF^{\intercal} + \mathcal{I}(v))| &= |\mathcal{I}(v))||I_r + F^{\intercal}\mathcal{I}(\text{-}v)F| \\ &= \exp\left(\sum_d v_d\right)|I_r + F^{\intercal}\mathcal{I}(\text{-}v)F|\end{aligned}$$

where the left term is simply the product of the diagonal component, and the right term is the determinant of an $r \times r$ matrix, computable in $O(r^3)$ time (Harville, 1997).

To compute Σ^{-1}, the Woodbury matrix identity states that

$$\begin{aligned}&(FF^{\intercal} + \mathcal{I}(v))^{-1} \\ &= \mathcal{I}(\text{-}v) - \mathcal{I}(\text{-}v)F(I_r + F^{\intercal}\mathcal{I}(\text{-}v)F)^{-1}F^{\intercal}\mathcal{I}(\text{-}v)\end{aligned}$$

which involves the inversion of a smaller, $r \times r$ matrix and can be done in $O(r^3)$ time (Golub & Van Loan, 2013). Importantly, for $r \ll D$ the above operations are efficiently differentiable and amenable for use in the BBVI framework.

Fitting the rank To specify the ELBO objective, we need to choose a rank r for the component covariance. There are many ways to decide on the rank of the variational approximation, some more appropriate for certain settings given dimensionality and computational constraints. For instance, we can greedily incorporate new rank components. Alternatively, we can fit a sequence of ranks $r = 1, 2, \ldots, r_{\text{max}}$, and choose the best result (in terms of KL). In the Bayesian neural network model, we report results for a fixed schedule of ranks. In the hierarchical Poisson model, we monitor the change in marginal variances to decide the appropriate rank. See Section B of the supplement for further discussion.

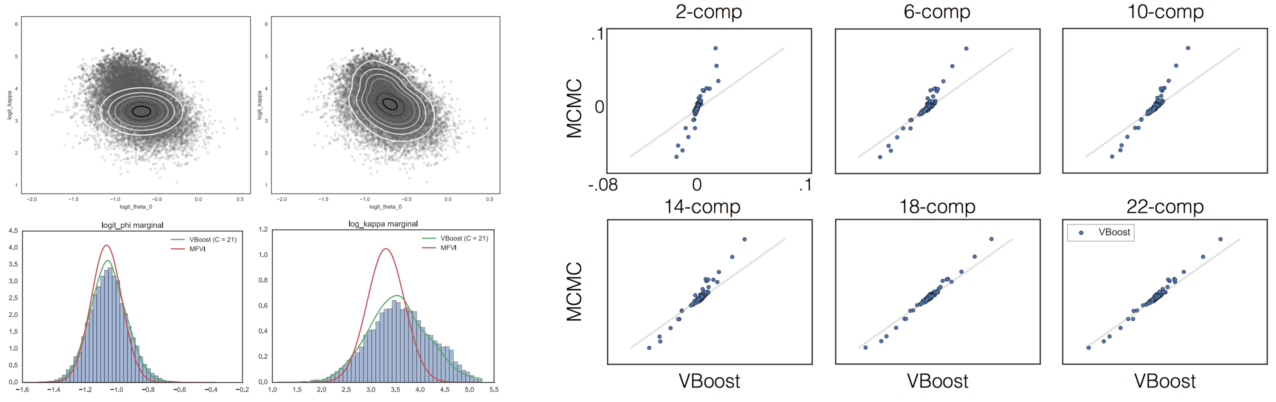
Initializing new component parameters When we add a new component, we must first initialize the component parameters. We find that the VBoost optimization procedure can be sensitive to initialization, so we devise a cheap importance sampling-based algorithm to generate a good starting point. This initialization procedure is detailed in Section A and Algorithm 1 of the supplement.

3.4. Related Work

Mixtures of mean field approximations were introduced in Jaakkola & Jordan (1998) where mean field-like updates were developed using a bound on the entropy term and model-specific parameter updates. Nonparametric variational inference, introduced in Gershman et al. (2012), is a black-box variational inference algorithm that approximates a target distribution with a mixture of equally-weighted isotropic normals. The authors use a lower-bound on the entropy term in the ELBO to make the optimization procedure tractable. Similarly, Salimans & Knowles (2013) present a method for fitting mixture distributions as an approximation. However, their method is restricted to mixture component distributions within the exponential family, and a joint optimization procedure. Mixture distributions are a type of hierarchical variational model (Ranganath et al., 2016b), where the component identity can be thought of as latent variables in the variational distribution. While in Ranganath et al. (2016b), the authors optimize a lower bound on the ELBO to fit general hierarchical variational models, our approach integrates out the discrete latent variables, allowing us to directly optimize the ELBO.

Sequential maximum-likelihood estimation of mixture models has been studied previously where the error between the sequentially learned model and the optimal model where all components and weights are jointly learned is bounded by $O(1/C)$ where C is the number of mixture components (Li & Barron, 1999; Li, 1999; Rakhlin et al., 2006). A similar bound was proven in Zhang (2003) using arguments from convex analysis. More recently, sequentially constructing a mixture of deep generative models has been shown to achieve the same $O(1/C)$ error bound when trained using an adversarial approach (Tolstikhin et al., 2016). Though these ideas show promise for deriving error bounds for variational boosting, there are difficulties in applying them.

In concurrent work, Guo et al. (2016) developed a boosting procedure to construct flexible approximations to posterior distributions. In particular, they use gradient-boosting to determine candidate component distributions and then optimize the mixture weight for the new component (Friedman, 2000). However, Guo et al. (2016) assume that the gradient-boosting procedure is able to find the optimal new component so that the arguments in Zhang (2003) apply, which is not true in general. We note that if we make the similar assumption that at each step of VBoost the component parameters λ_C^* are found exactly, then the optimization of ρ_C is convex and can be optimized exactly. We can then appeal to the same arguments in Zhang (2003) and obtain an $O(1/C)$ error bound. The work in Guo et al. (2016) provides important first steps in the theoretical development of boosting methods applied to variational inference, however, we note that developing a comprehensive theory that deals with the difficulties of multimodality and the non-joint-convexity of KL divergence in λ and ρ is still needed. Recently, Moore (2016) began to address issues of multimodality from model symmetry in variational inference. However, the question remains whether the entire distribution is being explored.



(a) `baseball` marginals

(b) `baseball` covariances

Figure 3. Left: Comparison of bivariate (top) and univariate (bottom) marginals for the `baseball` model. Histograms/scatterplots depict 20,000 NUTS samples. The top left depicts $(\ln \kappa, \theta_0)$ marginal samples and a mean field approximation (MFVI). The Top Right shows the same bivariate marginal, and the VBoost approximation with isotropic components. The bottom panels compare NUTS, MFVI, and VBoost on univariate marginals (ϕ and $\ln \kappa$). *Right*: Comparison of posterior covariances for the $D = 20$-dimensional `baseball` model. Each plot compares covariance estimates of VBoost (x-axis) with increasing numbers of components and MCMC samples (y-axis). As more components are added, the VBoost estimates more closely match the MCMC covariance estimates.

Seeger (2010) explored the use of low-rank covariance Gaussians as variational approximations using a PCA-like algorithm. Additionally, concurrent work has proposed the use a LR+D matrices as the covariances of Gaussian posterior approximations (Ong et al., 2017). We have also found that though the LR+D covariance approximation is useful for capturing posterior correlations, combining the idea with boosting new components to capture non-Gaussian posteriors yields superior posterior inferences.

4. Experiments and Analysis

To supplement the previous synthetic examples, we use VBoost to approximate various challenging posterior distributions arising from real statistical models of interest.[2]

Binomial Regression We first apply VBoost to a non-conjugate hierarchical binomial regression model.[3] The model describes the binomial rates of success (batting averages) of baseball players using a hierarchical model (Efron & Morris, 1975), parameterizing the "skill" of each player:

$$\begin{aligned} \theta_j &\sim \texttt{Beta}(\phi \cdot \kappa, (1-\phi) \cdot \kappa) && \text{player } j \text{ prior} \\ y_j &\sim \texttt{Binomial}(K_j, \theta_j) && \text{player } j \text{ hits} \ , \end{aligned}$$

where y_j is the number of successes (hits) player j has attempted in K_j attempts (at bats). Each player has a latent success rate θ_j, which is governed by two global variables κ and ϕ. We specify the priors $\phi \sim \texttt{Unif}(0, 1)$ and $\kappa \sim \texttt{Pareto}(1, 1.5)$. There are 18 players in this example, creating a posterior distribution with $D = 20$ parameters. For each round of VBoost, we estimate $\nabla_{\lambda,\rho} \mathcal{L}^{(C+1)}$ using 400 samples each for q_{C+1} and q_C. We use 1,000 iterations of `adam` with default parameters to update ρ_{C+1} and λ_{C+1} (Kingma & Ba, 2014).

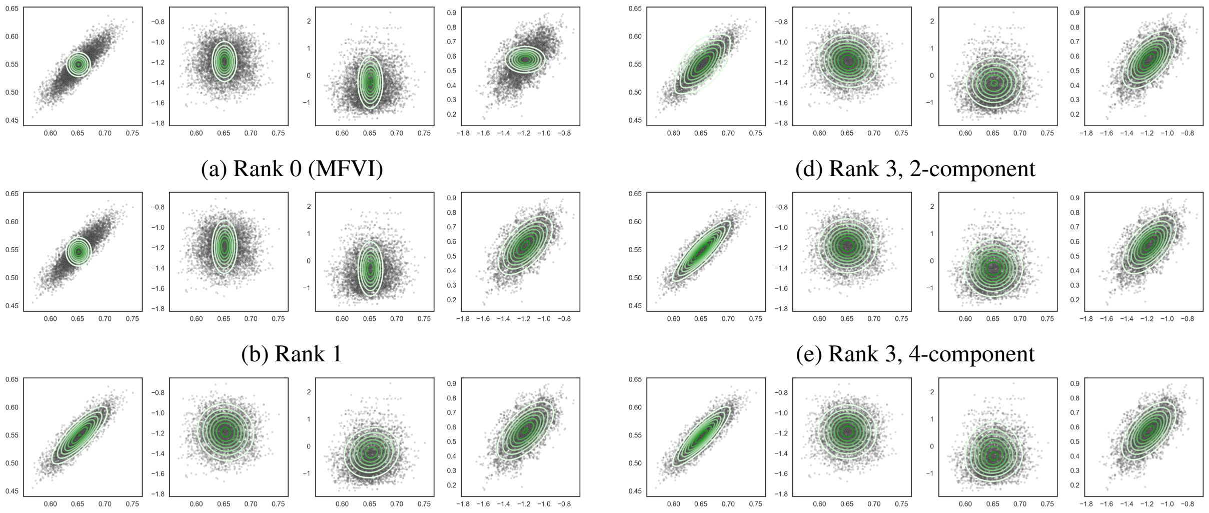
In all experiments, we use `autograd` to obtain gradients with respect to new component parameters (Maclaurin et al., 2015b;a). To highlight the fidelity of our method, we compare VBoost with rank-1 components to mean field VI (MFVI) and the No-U-Turn Sampler (NUTS) (Hoffman & Gelman, 2014). The empirical distribution resulting from 20k NUTS samples is considered the "ground truth" posterior in this example. Figure 3a compares a selection of univariate and bivariate posterior marginals. We see that VBoost is able to closely match the NUTS posteriors, improving upon the MFVI approximation. Figure 3b compares the VBoost covariance estimates to the "ground truth" estimates of MCMC at various stages of the algorithm. We see that VBoost is able to capture pairwise covariances with increasing accuracy as the number of components increases.

Multi-level Poisson GLM We use VBoost to approximate the posterior of a hierarchical Poisson GLM, a common non-conjugate Bayesian model. Here, we focus on a specific model that was formulated to measure the relative rates of stop-and-frisk events for different ethnicities and in different precincts (Gelman et al., 2007), and has been used as an illustrative example of multi-level modeling (Gelman & Hill, 2006). The model uses a precinct and ethnicity effect to describe the relative rate of stop-and-frisk events

$$\begin{aligned} \alpha_e &\sim \mathcal{N}(0, \sigma_\alpha^2) && \text{ethnicity effect} \\ \beta_p &\sim \mathcal{N}(0, \sigma_\beta^2) && \text{precinct effect} \\ \ln \lambda_{ep} &= \mu + \alpha_e + \beta_p + \ln N_{ep} && \text{log rate} \\ Y_{ep} &\sim \mathcal{P}(\lambda_{ep}) && \text{stop-and-frisk events} \end{aligned}$$

[2] Code available at `https://github.com/andymiller/vboost`.

[3] Model and data from the `mc-stan` case studies



(a) Rank 0 (MFVI)

(b) Rank 1

(c) Rank 2

(d) Rank 3, 2-component

(e) Rank 3, 4-component

(f) Rank 3, 8-component

Figure 4. Left: A sampling of bivariate marginals for a single Gaussian component approximation for the $D = 37$-dimensional `frisk` model. Each row incorporates more covariance structure. Though there are a total of 666 covariances to be approximated, only a few directions in the D-dimensional parameter space exhibit non-trivial correlations. *Right*: The same marginals with a mixture approximation using rank-3 Gaussians at various stages of the VBoost algorithm. Introducing new mixture components allows the posterior to take a non-Gaussian shape, most exhibited in the third column.

dataset	pbp	rank 5	vboost 2	vboost 6	vboost 10
wine	-0.990 (± 0.08)	-0.972 (± 0.05)	**-0.971** (± 0.05)	-0.978 (± 0.06)	-0.994 (± 0.06)
boston	-2.902 (± 0.64)	-2.670 (± 0.16)	-2.651 (± 0.16)	**-2.599** (± 0.16)	-2.628 (± 0.16)
concrete	-3.162 (± 0.15)	-3.247 (± 0.06)	-3.228 (± 0.06)	-3.169 (± 0.07)	**-3.134** (± 0.08)
power-plant	-2.798 (± 0.04)	-2.814 (± 0.03)	-2.811 (± 0.03)	-2.800 (± 0.03)	**-2.793** (± 0.03)
yacht	-0.990 (± 0.08)	-0.972 (± 0.05)	**-0.971** (± 0.05)	-0.978 (± 0.06)	-0.994 (± 0.06)
energy-eff.	**-1.971** (± 0.11)	-2.452 (± 0.12)	-2.422 (± 0.11)	-2.345 (± 0.11)	-2.299 (± 0.12)

Table 1. Test log probability for PBP and VBoost with varying number of components (fixed rank of 5). Each entry shows the average predictive performance of the model and the standard deviation across the 20 trials — bold indicates the best average (though not necessarily "statistical significance").

where Y_{ep} are the number of stop-and-frisk events within ethnicity group e and precinct p over some fixed period of time; N_{ep} is the total number of arrests of ethnicity group e in precinct p over the same period of time; α_e and β_p are the ethnicity and precinct effects. The prior over the mean offset and group variances is given by $\mu, \ln \sigma^2_\alpha, \ln \sigma^2_\beta \sim \mathcal{N}(0, 10^2)$.

As before, we simulate 20k NUTS samples, and compare various variational approximations. Because of the high posterior correlations present in this example, VBoost with *diagonal* covariance components is inefficient in its representation of this structure. As such, this example relies on the *low-rank* approximation to shape the posterior. Figure 4 shows how posterior accuracy is affected by incorporating covariance structure (left) and adding more components (right). Figures 6 and 7 in the supplement compare VBoost covariances to MCMC samples, showing that increased posterior rank capacity and number of components yield more accurate marginal variance and covariance estimates. These results indicate that while incorporating covariance structure increases the accuracy of estimating marginal variances, the non-Gaussianity afforded by the use of mixture components allows for a better posterior approximation translating into more accurate moment estimates.

Bayesian Neural Network We apply our method to a Bayesian neural network (BNN) regression model, which admits a high-dimensional, non-Gaussian posterior. We compare predictive performance of VBoost to Probabilistic Backpropagation (PBP) (Hernández-Lobato & Adams, 2015). Mimicking the experimental setup of Hernández-Lobato & Adams (2015), we use a single 50-unit hidden layer, with ReLU activation functions. We place a normal prior over each weight in the neural network, governed by the same variance and an inverse Gamma prior over the observation variance yielding the model:

$$\begin{aligned} w_i &\sim \mathcal{N}(0, 1/\alpha) && \text{weights} \\ y|x, w, \tau &\sim \mathcal{N}(\phi(x, w), 1/\tau) && \text{output distribution} \end{aligned}$$

where $w = \{w_i\}$ is the set of weights, and $\phi(x, w)$ is a multi-layer perceptron that maps input x to output y as a function of parameters w. Both α and τ are given $\text{Gamma}(1, .1)$ priors. We denote the set of parameters as $\theta \triangleq (w, \alpha, \tau)$. We approximate the posterior $p(w, \alpha, \tau|\mathcal{D})$, where $\mathcal{D}$ is the training set of $\{x_n, y_n\}_{n=1}^N$ input-output pairs. We then use the posterior predictive distribution to compute the distribution for a new input x^*

$$p(y|x^*, \mathcal{D}) = \int p(y|x^*, \theta)p(\theta|\mathcal{D})d\theta \tag{7}$$

$$\approx \frac{1}{L}\sum_{\ell=1}^{L} p(y|x^*, \theta^{(\ell)}), \quad \theta^{(\ell)} \sim p(\theta|\mathcal{D}) \tag{8}$$

and report average predictive log probabilities for held out data, $p(Y = y^*|x^*, \mathcal{D})$. For a dataset with input dimension P, the posterior has dimension $D = (P+2)\cdot 50+3$ (between $D = 303$ and $D = 753$ for the data sets considered).

We report held-out predictive performance for different approximate posteriors for six datasets. For each dataset, we perform the following training procedure 20 times. First, we create a random partition into a 90% training set and 10% testing set. We then apply VBoost, adding rank 5 components. We allow each additional component only 200 iterations. To save time on initialization, we draw 100 samples from the existing approximation, and initialize the new component with the sample with maximum weight. For comparison, Probabilistic back-propagation is given 1000 passes over the training data — empirically, sufficient for the algorithm to converge.

Table 3 in the supplement presents out-of-sample log probability for single-component multivariate Gaussian approximations with varying rank structure. Table 1 presents out-of-sample log probability for additional rank 5 components added using VBoost. We note that though we do not see much improvement as rank structure is added, we do see predictive improvement as components are added. Our results suggest that incorporating and adapting new mixture components is a recipe for a more expressive posterior approximation, translating into better predictive results. In fact, for all datasets we see that incorporating a new component improves test log probability, and we see further improvement with additional components for most of the datasets. Furthermore, in five of the datasets we see predictive performance surpass probabilistic back-propagation as new components are added. This highlights VBoost's ability to trade computation for improved accuracy. These empirical results suggest that augmenting a Gaussian approximation to include additional capacity can improve predictive performance in a BNN while retaining computational tractability.

4.1. Comparison to NPVI

We also compare VBoost to nonparametric variational inference (NPVI) (Gershman et al., 2012), a similar mixture based black-box variational method. NPVI derives a tractable lower bound to the ELBO which is then approximately maximized. NPVI requires computing the Hessian of the model for the ELBO approximation, so we limit our comparison to the lower dimensional hierarchical models.

num comps	1	2	5	10	20
VBoost	-702.97	-700.92	-699.69	-699.07	-698.88
NPVI	-718.47	-717.86	-717.09	-716.36	-715.86

Table 2. ELBO values for VBoost and NPVI (higher is better). Note that VBoost with 1 component is MFVI. All ELBO values are computed using a Monte Carlo estimate with $L = 100k$ samples from the variational distribution. In NPVI, each component is a spherical gaussian with a single σ^2 shared across all dimensions — this limits the capacity of the approximation, requiring more components. Note, VBoost greedily incorporates components, while NPVI is re-run using a different number of components.

We also note that the NPVI ELBO approximation does not fully integrate the $\ln \pi(x)$ term against the variational approximation, $q(x; \lambda)$ when optimizing the mean parameters of the approximation components. When we applied NPVI to the `baseball` model, we discovered an instability in the optimization of these mean parameters (which we verified by finding that map optimization diverges). Black box VI, VBoost, and MCMC were not susceptible to this pathology. Consequently, we only compare NPVI to VBoost on the `frisk` model. Because NPVI uses diagonal components, we restrict VBoost to use purely diagonal components ($r = 0$). In Table 2 we show marginal likelihood lower bounds, comparing NPVI to VBoost with a varying number of components. Even with a single component, the NPVI objective tends to underperform. The NPVI component variance is spherical, limiting its capacity to represent posterior correlations. Further, NPVI is approximately optimizing a looser lower bound to the marginal likelihood. These two factors explain why NPVI fails to match MFVI and VBoost.

5. Discussion and Conclusion

We proposed VBoost, a practical variational inference method that constructs an increasingly expressive posterior approximation and is applicable to a variety of Bayesian models. We demonstrated the ability of VBoost to learn rich representations of complex, high-dimensional posteriors on a variety of real world statistical models. One avenue for future work is incorporating flexible component distributions such as compositions of invertible maps (Rezende & Mohamed, 2015) or auxiliary variable variational models (Maaløe et al., 2016). We also plan to study approximation guarantees of the VBoost method and variance reduction techniques for our reparameterization gradient approach. Also, when optimizing parameters of a variational family, recent work has shown that the natural gradient can be more robust and lead to better optima (Hoffman et al., 2013; Johnson et al., 2016). Deriving and applying natural gradient updates for mixture approximations could make VBoost more efficient.

Acknowledgements

The authors would like to thank Arjumand Masood, Mike Hughes, and Finale Doshi-Velez for helpful feedback. ACM is supported by the Applied Mathematics Program within the Office of Science Advanced Scientific Computing Research of the U.S. Department of Energy under contract No. DE-AC02-05CH11231. NJF is supported by a Washington Research Foundation Innovation Postdoctoral Fellowship in Neuroengineering and Data Science. RPA is supported by NSF IIS-1421780 and the Alfred P. Sloan Foundation.

References

Bishop, C. *Pattern Recognition and Machine Learning*. Springer-Verlag, 2006.

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. Variational inference: A review for statisticians. *arXiv preprint arXiv:1601.00670*, 2016.

Dempster, A. P., Laird, N. M., and Rubin, D. B. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (methodological)*, pp. 1–38, 1977.

Efron, B. and Morris, C. Data analysis using Stein's estimator and its generalizations. *Journal of the American Statistical Association*, 70(350):311–319, 1975.

Friedman, J. H. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29:1189–1232, 2000.

Gelman, A. and Hill, J. *Data Analysis Using Regression and Multilevel/Hierarchical Models*. Cambridge University Press, 2006.

Gelman, A., Fagan, J., and Kiss, A. An analysis of the NYPD's stop-and-frisk policy in the context of claims of racial bias. *Journal of the American Statistical Association*, 102:813–823, 2007.

Gershman, S., Hoffman, M., and Blei, D. M. Nonparametric variational inference. In *International Conference on Machine Learning*, 2012.

Golub, G. H. and Van Loan, C. F. *Matrix Computations*. JHU Press, 2013.

Guo, F., Wang, X., Fan, K., Broderick, T., and Dunson, D. B. Boosting variational inference. arXiv:1611.05559, 2016.

Harville, D. A. *Matrix Algebra from a Statistician's Perspective*. Springer-Verlag, 1997.

Hernández-Lobato, J. M. and Adams, R. P. Probabilistic backpropagation for scalable learning of Bayesian neural networks. 2015.

Hoffman, M. D. and Gelman, A. The No-U-turn sampler: adaptively setting path lengths in Hamiltonian monte carlo. *Journal of Machine Learning Research*, 15(1): 1593–1623, 2014.

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

Jaakkola, T. S. and Jordan, M. I. Improving the mean field approximation via the use of mixture distributions. In *Learning in Graphical Models*, pp. 163–173. Springer, 1998.

Johnson, M. J., Duvenaud, D. K., Wiltschko, A. B., and Datta, S. R.and Adams, R. P. Composing graphical models with neural networks for structured representations and fast inference. In *Advances in Neural Information Processing Systems*, 2016.

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.

Kingma, D. and Ba, J. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2014.

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

Li, Q. *Estimation of Mixture Models*. PhD thesis, Yale University, May 1999.

Li, Q. J. and Barron, A. R. Mixture density estimation. In *Advances in Neural Information Processing Systems*, 1999.

Maaløe, L., Sønderby, C. K., Sønderby, S. K., and Winther, O. Auxiliary deep generative models. In *International Conference on Machine Learning*, 2016.

Maclaurin, D. and Adams, R. P. Firefly Monte Carlo: Exact MCMC with subsets of data. In *Uncertainty in Artificial Intelligence*, 2014.

Maclaurin, D., Duvenaud, D., and Adams, R. P. Autograd: Reverse-mode differentiation of native python. *ICML workshop on Automatic Machine Learning*, 2015a.

Maclaurin, D., Duvenaud, D., Johnson, M., and Adams, R. P. Autograd: Reverse-mode differentiation of native Python, 2015b. URL `http://github.com/HIPS/autograd`.

Moore, D. A. Symmetrized variational inference. In *NIPS Workshop on Advances in Approximate Bayesian Inferece*, 2016.

Ong, V. M.-H., Nott, D. J., and Smith, M. S. Gaussian variational approximation with factor covariance structure. *arXiv preprint arXiv:1701.03208*, 2017.

Rakhlin, A., D., Panchenko, and S., Mukherjee. Risk bounds for mixture density estimation. *ESAIM: Probability and Statistics*, 9:220–229, 2006.

Ranganath, R., Gerrish, S., and Blei, D. M. Black box variational inference. In *International Conference on Artificial Intelligence and Statistics*, pp. 814–822, 2014.

Ranganath, R., Altosaar, J., Tran, D., and Blei, D. M. Operator variational inference. In *Advances in Neural Information Processing Systems*, 2016a.

Ranganath, R., Tran, D., and Blei, D. M. Hierarchical variational models. In *International Conference on Machine Learning*, 2016b.

Rezende, D. and Mohamed, S. Variational inference with normalizing flows. In *International Conference on Machine Learning*, pp. 1530–1538, 2015.

Salimans, T. and Knowles, D. A. Fixed-form variational posterior approximation through stochastic linear regression. *Bayesian Analysis*, 8(4):837–882, 2013.

Seeger, M. W. Gaussian covariance and scalable variational inference. In *International Conference on Machine Learning*, 2010.

Tolstikhin, I., Gelly, S., Bousquet, O., Simon-Gabriel, C.-J., and Schoelkopf, B. Adagan: Boosting generative models. *arXiv preprint arXiv:1701.02386*, 2016.

Wainwright, M. J. and Jordan, M. I. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1-2):1–305, 2008.

Welling, M. and Teh, Y. W. Bayesian learning via stochastic gradient Langevin dynamics. In *International Conference on Machine Learning*, 2011.

Zhang, T. Sequential greedy approximation for certain convex optimization problems. *IEEE Transactions on Information Theory*, 49:682–691, 2003.