Black Box Variational Inference

Rajesh Ranganath, Sean Gerrish, and David M. Blei (2014)

Report by Miguel Biron Lattes

Abstract

The purpose of this report is to give a thorough assessment of the Black Box Variational Inference (BBVI) method. We will describe its derivation in detail, and then show how it performs using simulations from a simple Gaussian mixture model, which will help demonstrate the advantages as well as the pitfalls of the algorithm. Later, we will compare the performance of BBVI in a linear mixed effects model for a longitudinal dataset, against an alternative automatic procedure for Bayesian inference. Finally, we will propose improvements to the original paper based on our findings.

1 Introduction to Variational Inference

Consider the general Bayesian inference framework: having postulated a likelihood for observed data x, up to a (possibly infinitely dimensional) parameter z, and a prior distribution over it, we know we can find the posterior distribution of z given x using Bayes' Theorem:

$$p(z|x) = \frac{\overbrace{p(x|z)}^{\text{likelihood}}\overbrace{p(z)}^{\text{prior}}}{\underbrace{p(x)}_{\text{evidence}}} = \frac{\overbrace{p(x,z)}^{\text{joint}}}{\int_z p(x,z)dz} \tag{1}$$

Note that we will note be able to evaluate $p(z|x)$ until we compute the integral in the right hand side (RHS) to obtain the evidence [1]. Even if we are able to estimate this with good accuracy, we will still require to evaluate other integrals to do any sort of statistical inference. For example:

- Marginal distributions: $p(z_k|x) = \int_{z_{-k}} p(z|x)dz_{-k}$
- Expectations for arbitrary measurable f: $\mathbb{E}_{z\sim p(z|x)}[f(z)] = \int_z f(z)p(z|x)dz$
- Predictive distributions: $p(x^{\text{new}}|x) = \int_z p(x^{\text{new}}|z)p(z|x)dz$

Although there exists special cases when these integrals can be solved analytically (e.g., conjugate priors), there is the need for procedures that allow inferences in more general setups.

Variational Inference tackles this problem by finding a distribution q^* among a pre-specified family $\mathcal{Q}$, which best approximates the target posterior $p(z|x)$ under the Kullback-Leibler (KL) divergence criterion:

$$q^* = \arg\min_{q \in \mathcal{Q}} KL(q \| p(z|x)) \tag{2}$$

The KL divergence from any density π_2 to another π_1, such that π_1 is absolutely continuous with respect to π_2, is defined as:

$$KL(\pi_1 \| \pi_2) \triangleq \int_u \log\left(\frac{\pi_1(u)}{\pi_2(u)}\right) \pi_1(u) du = \mathbb{E}_{u \sim \pi_1}\left[\log\left(\frac{\pi_1(u)}{\pi_2(u)}\right)\right] \tag{3}$$

This quantity has two relevant properties:

- $KL(\pi_1 \| \pi_2) \geq 0$ for all densities π_1, π_2
- $KL(\pi_1 \| \pi_2) = 0 \Leftrightarrow \pi_1 = \pi_2$ almost everywhere

Once the optimization problem has been solved and q^* found, the analyst can use it in place of $p(z|x)$ to carry out the inferences needed. However, attempting to directly solve 2 will run into the problem that $p(z|x)$ cannot be computed directly, because the evidence $p(x)$ is unknown (see 1). Of course, one could try to estimate this quantity by using the fact that:

$$p(x) = \int_z p(x, z) dz = \int_z p(x|z) p(z) dz$$

However, this integration is as hard as the ones mentioned above for inferences using the posterior distribution. Nevertheless, there is an alternative way to approach this problem. Starting from the definition of the divergence 3:

$$\begin{aligned}
KL(q \| p(z|x)) &= \int_z \log\left(\frac{q(z)}{p(z|x)}\right) q(z) dz \\
&= \int_z q(z) \log q(z) dz - \int_z q(z) \log p(z|x) dz \\
&= \int_z q(z) \log q(z) dz - \int_z q(z) \log\left(\frac{p(x, z)}{p(x)}\right) dz \\
&= \int_z q(z) \log q(z) dz - \int_z q(z) \log p(x, z) dz + \log p(x) \underbrace{\int_z q(z) dz}_{=1} \\
&= \int_z q(z) \log q(z) dz - \int_z q(z) \log p(x, z) dz + \log p(x) \\
&= \log p(x) - \int_z q(z) [\log p(x, z) - \log q(z)] dz \\
&= \log p(x) - \mathbb{E}_q [\log p(x, z) - \log q(z)]
\end{aligned}$$

Let:

$$\mathcal{L}(q) \triangleq \mathbb{E}_q[\log p(x,z) - \log q(z)] \tag{4}$$

Then, by rearranging the last line of the derivation above, we obtain:

$$\log p(x) = KL(q\|p(z|x)) + \mathcal{L}(q) \tag{5}$$

The quantity $\mathcal{L}$ is known as the Evidence Lower Bound (ELBO). Its name stems from the fact that it is a lower bound for $\log p(x)$. Indeed, given that $KL(\pi_1\|\pi_2) \geq 0$ for any distributions π_1, π_2, it follows from 5 that:

$$\log p(x) \geq \mathcal{L}(q)$$

Note that the left hand side (LHS) of 5 does not depend on q. Thus, any movements in $\mathcal{L}(q)$ induced by changes in q will be exactly compensated by $KL(q\|p(z|x))$. Hence, we conclude that minimizing $KL(q\|p(z|x))$ with respect to q is equivalent to maximizing $\mathcal{L}(q)$:

$$\arg\min_{q\in\mathcal{Q}} KL(q\|p(z|x)) = \arg\max_{q\in\mathcal{Q}} \mathcal{L}(q)$$

Assuming that the members q of the family of approximating distributions $\mathcal{Q}$ are specified up to a parameter vector $\lambda \in \Lambda \subset \mathbb{R}^d$, then 2 can be recast in terms of these free variables:

$$q^* = q(\lambda^*) = \arg\max_{\lambda\in\Lambda} \mathcal{L}(q(\lambda)) \tag{6}$$

In general, though, the optimization problem posed by 6 can be as difficult as directly integrating the joint $p(x,z)$. Therefore, researchers often restrict $\mathcal{Q}$ to a narrow class of approximating families, with the purpose of reducing the complexity of the optimization [1]. Next, we will introduce the most popular choice: the mean-field approximation.

1.1 The mean-field approximating family

Under the mean-field approximation, we assume that the components of z are independent. Therefore, the joint distribution of z can be factorized into the product of marginal distributions, each governed by a parameter $\lambda_i \in \mathbb{R}^{d_i}$ (such that $\sum_i d_i = d$):

$$q(z;\lambda) = \prod_{i=1}^{n} q_i(z_i;\lambda_i) \tag{7}$$

This independence assumption is exploited by the Coordinate Ascent Variational Inference (CAVI) algo-

rithm, which, as its name states, uses coordinate ascent to solve 6. Indeed, starting from 4:

$$
\begin{aligned}
\mathcal{L}(q) &= \mathbb{E}_q[\log p(x,z) - \log q(z)] \\
&= \mathbb{E}_q\left[\log p(x,z) - \sum_j \log q_j(z_j)\right] \\
&= \mathbb{E}_{q_i}\left[\mathbb{E}_{q_{-i}}\left[\log p(x,z) - \sum_j \log q_j(z_j)\right]\right] \\
&= \mathbb{E}_{q_i}\left[\mathbb{E}_{q_{-i}}[\log p(x,z)] - \log q_i(z_i) - \sum_{j\neq i} \mathbb{E}_{q_{-i}}[\log q_j(z_j)]\right] \\
&= \mathbb{E}_{q_i}[\mathbb{E}_{q_{-i}}[\log p(x,z)] - \log q_i(z_i)] - \sum_{j\neq i} \mathbb{E}_{q_{-i}}[\log q_j(z_j)] \\
&= -\mathbb{E}_{q_i}\left[\log \frac{q_i(z_i)}{\exp[\mathbb{E}_{q_{-i}}[\log p(x,z)]]}\right] - \sum_{j\neq i} \mathbb{H}(q_j) \\
&= -KL(q_i \| \alpha \exp[\mathbb{E}_{q_{-i}}[\log p(x,z)]]) - \sum_{j\neq i} \mathbb{H}(q_j)
\end{aligned}
$$

where α is a normalizing constant to ensure that the second term is a valid distribution for z_i. We have shown that, in order to maximize $\mathcal{L}$ with respect to q_i, we need to minimize the term associated with the Kullback-Leibler divergence (the sum of entropies does not depend on q_i). This term is minimized when we choose:

$$
q_i \propto \exp[\mathbb{E}_{q_{-i}}[\log p(x,z)]] \tag{8}
$$

CAVI solves the VI optimization problem by iteratively applying 8 to each of the marginal components q_i. It is guaranteed to converge to a (possibly) local maximum.

However, applying CAVI in practice to an arbitrary model can be very time consuming, because each of the updates for the components q_i have to be analitically derived. Moreover, any change in the model joint $p(x,z)$ or in the variational approximation q can likely yield the CAVI updates invalid, and then new updates have to be derived. This difficults the exploration of many alternative models.

Because of this shortcoming, researchers have aimed to provide automatic procedures for solving VI while retaining theoretical guarantees of convergence to good solutions. Black Box Variational Inference is one of these attempts.

2 Black Box Variational Inference

2.1 Basic definition of the algorithm

Black Box Variational Inference (BBVI) [2] is a method aimed to avoid the "painstaking derivations" needed to obtain optimal CAVI updates. At its core, BBVI solves 6 by using stochastic optimization. Applying the first order condition to 6, we have:

$$
\begin{aligned}
0 &= \nabla_\lambda \mathcal{L}(q(\lambda)) \\
&= \nabla_\lambda [\mathbb{E}_q[\log p(x,z) - \log q(z)]] \\
&= \nabla_\lambda \int_z q(z;\lambda)(\log p(x,z) - \log q(z;\lambda))dz \\
&= \int_z \nabla_\lambda [q(z;\lambda)(\log p(x,z) - \log q(z;\lambda))]dz \\
&= \int_z \nabla_\lambda [q(z;\lambda)](\log p(x,z) - \log q(z;\lambda))dz + \int_z q(z;\lambda)(\underbrace{\nabla_\lambda[\log p(x,z)]}_{0} - \nabla_\lambda[\log q(z;\lambda)])dz \\
&= \int_z q(z;\lambda)\nabla_\lambda[\log q(z;\lambda)](\log p(x,z) - \log q(z;\lambda))dz - \int_z q(z;\lambda)\nabla_\lambda[q(z;\lambda)]\frac{1}{q(z;\lambda)}dz \\
&= \mathbb{E}_q[\nabla_\lambda[\log q(z;\lambda)](\log p(x,z) - \log q(z;\lambda))] - \nabla_\lambda \left[\int_z q(z;\lambda)dz\right] \\
&= \mathbb{E}_q[\nabla_\lambda[\log q(z;\lambda)](\log p(x,z) - \log q(z;\lambda))] - \nabla_\lambda[1] \\
&= \mathbb{E}_q[\nabla_\lambda[\log q(z;\lambda)](\log p(x,z) - \log q(z;\lambda))]
\end{aligned}
$$

Appendix A offers a discussion on the conditions needed to obtain the fourth line of this derivation (interchanging the gradient with the integration). Also, the replacement used in the first term of line 6, which follows from standard differential calculus:

$$\nabla_\lambda[q(z;\lambda)] = q(z;\lambda)\nabla_\lambda[\log q(z;\lambda)] \Leftrightarrow \nabla_\lambda[\log q(z;\lambda)] = \frac{\nabla_\lambda[q(z;\lambda)]}{q(z;\lambda)}$$

is usually referred to as the "REINFORCE trick" in the Machine Learning (ML) literature, originating in [3]. It is a "trick" in the sense that it rewrites the term as an expectation with respect to $q(z;\lambda)$.

Assuming the above derivation is correct, the first order condition becomes:

$$0 = \mathbb{E}_q[\nabla_\lambda[\log q(z;\lambda)](\log p(x,z) - \log q(z;\lambda))] \tag{9}$$

In theory, we could now implement a gradient ascent procedure based on this expression. In practice, however, the expectation will be unknown for arbitrary models $p(x,z)$ and approximating family $\mathcal{Q}$. Stochastic optimization solves this by instead replacing the true gradient with an unbiased estimator

$\hat{\nabla}_\lambda \mathcal{L}(\lambda)$, i.e.:

$$\mathbb{E}_q[\hat{\nabla}_\lambda \mathcal{L}(\lambda)] = \nabla_\lambda \mathcal{L}(q(\lambda)) \ , \ \forall \lambda \in \Lambda$$

Observe that the expectation in the expression $\nabla_\lambda \mathcal{L}(q(\lambda))$ is taken with respect to the variational approximation q. If we choose $\mathcal{Q}$ such that we have a procedure to simulate iid samples from any member q, then:

$$\hat{\nabla}_\lambda \mathcal{L}(\lambda) = \frac{1}{S} \sum_{s=1}^{S} \nabla_\lambda [\log q(z^s; \lambda)](\log p(x, z^s) - \log q(z^s; \lambda)) \tag{10}$$

is an unbiased estimator of $\nabla_\lambda \mathcal{L}(q(\lambda))$, provided that:

$$z^1, ..., z^S \overset{iid}{\sim} q(z; \lambda)$$

The proof is straightforward:

$$\begin{aligned}
\mathbb{E}_q[\hat{\nabla}_\lambda \mathcal{L}(\lambda)] &= \mathbb{E}_q \left[\frac{1}{S} \sum_{s=1}^{S} \nabla_\lambda [\log q(z^s; \lambda)](\log p(x, z^s) - \log q(z^s; \lambda)) \right] \\
&= \frac{1}{S} \sum_{s=1}^{S} \mathbb{E}_q \left[\nabla_\lambda [\log q(z^s; \lambda)](\log p(x, z^s) - \log q(z^s; \lambda)) \right] \\
&= \frac{1}{S} \sum_{s=1}^{S} \nabla_\lambda \mathcal{L}(q(\lambda)) \\
&= \frac{1}{S} S \nabla_\lambda \mathcal{L}(q(\lambda)) \\
&= \nabla_\lambda \mathcal{L}(q(\lambda))
\end{aligned}$$

Note that for this derivation we only used the "identically distributed" part of the iid statement. However, the independence is needed for obtaining the standard $O(1/S)$ variance of Monte Carlo (MC) estimates.

Algorithm 1 shows what we refer to as the "naive" version of BBVI (the reasons for this pejorative name will be discussed in the following section). The stopping criteria is suggested by the authors, which is different from the standard VI practice of monitoring the ELBO (either in the training set or in a held-out set). Although the authors don't explicitly state this, we may speculate that the reason they chose it is that it does not require model specific calculations, making it suitable for a black box approach (this will be revisited in Section 5).

Algorithm 1: Naive BBVI

input : Model, tolerance τ, batch size S

Randomly initialize λ^0

Set $t \leftarrow 0$, $\Delta \leftarrow \infty$

while $\Delta > \tau$ **do**

$t \leftarrow t + 1$

$z^1 ... z^S \overset{iid}{\sim} q(z; \lambda^{t-1})$

$\hat{\nabla}_\lambda \mathcal{L}(\lambda^{t-1}) \leftarrow \frac{1}{S} \sum_{s=1}^{S} \nabla_\lambda [\log q(z^s; \lambda^{t-1})](\log p(x, z^s) - \log q(z^s; \lambda^{t-1}))$

$\lambda^t \leftarrow \lambda^{t-1} + \rho^t \hat{\nabla}_\lambda \mathcal{L}(\lambda^{t-1})$

$\Delta \leftarrow \frac{\|\lambda^t - \lambda^{t-1}\|}{\|\lambda^{t-1}\|}$

output: $\lambda^* = \lambda^t$

Now, if the sequence of step sizes ρ^t satisfies the Robbins-Monro conditions:

$$\sum_{t=1}^{\infty} \rho^t = \infty \qquad\qquad \sum_{t=1}^{\infty} (\rho^t)^2 < \infty$$

then Algorithm 1 is guaranteed to eventually converge to a local maximum. One simple example of such sequence could be $\rho^t = \frac{\rho}{t}$, for some $\rho > 0$. On the other hand, if we choose a constant $\rho^t = \rho$, then Algorithm 1 is not guaranteed to converge. In fact, λ^t will bounce around a local optimum indefinitely [4]. Nevertheless, this need not be a problem, as long as we include an additional stopping criterion, like a maximum number of iterations.

2.2 Understanding the variance of naive BBVI

Although Algorithm 1 is appealing because of its simplicity and its theoretical guarantee of convergence, in practice it does not produce meaningful results for non-trivial models. As many have noted [5, 6, 7, 8], the naive estimator of the gradient has high variance, making convergence in finite time difficult to achieve.

Since we know that the variance of $\hat{\nabla}_\lambda \mathcal{L}(\lambda)$ decreases as $1/S$, let us focus on the case when $S = 1$. We will focus on the gradient for λ_i. Furthermor, wlog, assume $d_i = 1$ (otherwise, we could focus on any of its sub-components). Then:

$$\widehat{\frac{\partial}{\partial \lambda_i} \mathcal{L}(\lambda)} = \frac{\partial}{\partial \lambda_i} [\log q(z; \lambda)](\log p(x, z) - \log q(z; \lambda))$$

with $z \sim q(z; \lambda)$. Assume that q is from the mean field family, and also that the joint admits a

factorization:

$$\log p(x, z) = \sum_{f=1}^{F} \log p_f(x, z_{(f)})$$

Then, we see that:

$$\widehat{\frac{\partial}{\partial \lambda_i} \mathcal{L}(\lambda)} = \frac{\partial}{\partial \lambda_i} [\log q_i(z_i; \lambda_i)] \left(\sum_{f=1}^{F} \log p_f(x, z) - \sum_{j=1}^{n} \log q_j(z_j; \lambda_j) \right)$$

Following the analysis by [5] (Appendix D), the variance of $\widehat{\frac{\partial}{\partial \lambda_i} \mathcal{L}(\lambda)}$ will grow as the number of terms in the summations; i.e., $O(F + n)$. In other words, the variance of BBVI grows linearly with the number of latent variables in the model. Hence, a good tactic to reduce the variance of the estimator would be to reduce the number of terms in said summation that do not yield meaningful information, while keeping the number of latent variables constant.

2.3 Rao-Blackwellization of gradient of the ELBO

2.3.1 Description of the method

Probably guided by the insight from the previous paragraph, the authors apply the technique known as Rao-Blackwellization to $\hat{\nabla}_\lambda \mathcal{L}(\lambda)$ in order to reduce its variance. Rao-Blackwellization reduces the variance of any estimator $J(X, Y)$, where X and Y are random variables, by defining another estimator

$$\hat{J}(X) \triangleq \mathbb{E}[J(X, Y)|X]$$

Using iterated expectations, we see that $\hat{J}(X)$ preserves the expectation of $J(X, Y)$:

$$\mathbb{E}[\hat{J}(X)] = \mathbb{E}[\mathbb{E}[J(X, Y)|X]] = \mathbb{E}[J(X, Y)]$$

The variance of this new estimator becomes:

$$\begin{aligned}
Var(\hat{J}(X)) &= \mathbb{E}[\hat{J}(X)^2] - \mathbb{E}[\hat{J}(X)]^2 \\
&= \mathbb{E}[\hat{J}(X)^2] - \mathbb{E}[J(X, Y)]^2 \\
&= Var(J(X, Y)) + \mathbb{E}[\hat{J}(X)^2] - \mathbb{E}[J(X, Y)^2] \\
&= Var(J(X, Y)) - \mathbb{E}[(J(X, Y) - \hat{J}(X))^2] + 2\mathbb{E}[\hat{J}(X)^2] - 2\mathbb{E}[J(X, Y)\hat{J}(X)] \\
&= Var(J(X, Y)) - \mathbb{E}[(J(X, Y) - \hat{J}(X))^2] + 2\mathbb{E}[\hat{J}(X)(\hat{J}(X) - J(X, Y))]
\end{aligned}$$

Let us focus on the third term:

$$\begin{aligned}\mathbb{E}[\hat{J}(X)(\hat{J}(X) - J(X,Y))] &= \mathbb{E}[\mathbb{E}[\hat{J}(X)(\hat{J}(X) - J(X,Y))|X]] \\ &= \mathbb{E}[\hat{J}(X)(\hat{J}(X) - \mathbb{E}[J(X,Y)|X])] \\ &= \mathbb{E}[\hat{J}(X)(\hat{J}(X) - \hat{J}(X))] \\ &= 0\end{aligned}$$

Therefore:

$$Var(\hat{J}(X)) = Var(J(X,Y)) - \mathbb{E}[(J(X,Y) - \hat{J}(X))^2] \tag{11}$$

In general, $Var(\hat{J}(X)) < Var(J(X,Y))$, unless $J(X,Y) = \hat{J}(X)$ almost everywhere, which in practice corresponds to the pathological case when $J(X,Y)$ does not actually depend on Y.

Recall that $\mathbb{E}[\hat{J}(X)] = \mathbb{E}[J(X,Y)]$. Another way to see this is that:

$$\mathbb{E}[J(X,Y) - \hat{J}(X)] = 0$$

Hence, Rao-Blackwellization removes elements from $J(X,Y)$ that in expectation are equal to 0. Moreover, because of 11, the reduction in variance achieved by $\hat{J}(X)$ is equal to the variance of these terms.

2.3.2 Application to the naive gradient estimator

Back to BBVI, assume that the approximating family follows the mean-field assumption. Also, suppose that the model joint $p(x,z)$ admits a factorization of the form:

$$p(x,z) = p_i(x, z_{(i)})p_{-i}(x, z_{-i}) \tag{12}$$

where $p_i(x, z_{(i)})$ contains all the terms in $p(x,z)$ that depend on z_i, and $p_{-i}(x, z_{-i})$ does not contain any such term. We use $z_{(i)}$ to denote the collection of latent variables that appear in p_i (including z_i).

Then, we can write the true gradient of the ELBO for z_i as follows:

$$\begin{aligned}
\nabla_{\lambda_i}\mathcal{L}(\lambda) &= \mathbb{E}_q[\nabla_{\lambda_i}[\log q_i(z_i;\lambda_i)](\log p(x,z) - \log q(z;\lambda))] \\
&= \mathbb{E}_{q_i}\mathbb{E}_{q_{-i}}\left[\nabla_{\lambda_i}[\log q_i(z_i;\lambda_i)](\log p_i(x,z_{(i)}) + \log p_{-i}(x,z_{-i}) - \sum_j \log q_j(z_j;\lambda_j))\right] \\
&= \mathbb{E}_{q_i}\Big[\nabla_{\lambda_i}[\log q_i(z_i;\lambda_i)]\left(\mathbb{E}_{q_{-i}}[\log p_i(x,z_{(i)})] + \mathbb{E}_{q_{-i}}[\log p_{-i}(x,z_{-i})] - \log q_i(z_i;\lambda_i)\right. \\
&\qquad \left.\left. -\mathbb{E}_{q_{-i}}\left[\sum_{j\neq i}\log q_j(z_j;\lambda_j)\right]\right)\right]
\end{aligned}$$

Note that the term:

$$C \triangleq \mathbb{E}_{q_{-i}}[\log p_{-i}(x,z_{-i})] - \mathbb{E}_{q_{-i}}\left[\sum_{j\neq i}\log q_j(z_j;\lambda_j)\right] \tag{13}$$

does not depend on any z. Hence:

$$\begin{aligned}
\nabla_{\lambda_i}\mathcal{L}(\lambda) &= \mathbb{E}_{q_i}[\nabla_{\lambda_i}[\log q_i(z_i;\lambda_i)](\mathbb{E}_{q_{-i}}[\log p_i(x,z_{(i)})] - \log q_i(z_i;\lambda_i) + C)] \\
&= \mathbb{E}_{q_i}[\nabla_{\lambda_i}[\log q_i(z_i;\lambda_i)](\mathbb{E}_{q_{-i}}[\log p_i(x,z_{(i)})] - \log q_i(z_i;\lambda_i))] + C\underbrace{\mathbb{E}_{q_i}[\nabla_{\lambda_i}[\log q_i(z_i;\lambda_i)]}_{0} \\
&= \mathbb{E}_{q_i}[\nabla_{\lambda_i}[\log q_i(z_i;\lambda_i)](\mathbb{E}_{q_{-i}}[\log p_i(x,z_{(i)})] - \log q_i(z_i;\lambda_i))] \\
&= \mathbb{E}_{q_i}\mathbb{E}_{q_{-i}}[\nabla_{\lambda_i}[\log q_i(z_i;\lambda_i)](\log p_i(x,z_{(i)}) - \log q_i(z_i;\lambda_i))] \\
&= \mathbb{E}_{q_{(i)}}[\nabla_{\lambda_i}[\log q_i(z_i;\lambda_i)](\log p_i(x,z_{(i)}) - \log q_i(z_i;\lambda_i))]
\end{aligned}$$

In the last line, we write $\mathbb{E}_{q_{(i)}}$ because the terms inside only depend on $z_{(i)}$. Let:

$$\hat{\nabla}^{RB}_{\lambda_i}\mathcal{L}(\lambda_i) \triangleq \frac{1}{S}\sum_s \nabla_{\lambda_i}[\log q_i(z_i^s;\lambda_i)](\log p_i(x,z_{(i)}^s) - \log q_i(z_i^s;\lambda_i)) \tag{14}$$

Then $\hat{\nabla}^{RB}_{\lambda_i}\mathcal{L}(\lambda_i)$ is a Rao-Blackwellized $\hat{\nabla}_{\lambda_i}\mathcal{L}(\lambda)$, because:

$$\mathbb{E}_q[\hat{\nabla}_{\lambda_i}\mathcal{L}(\lambda) - \hat{\nabla}^{RB}_{\lambda_i}\mathcal{L}(\lambda_i)] = C\mathbb{E}_{q_i}[\nabla_{\lambda_i}[\log q_i(z_i;\lambda_i)] = 0$$

where C was defined in 13.

We can analyze the variance of this new estimator following the same procedure as in 2.2. Assume the worst case, in which z_i appears in all of the factors of the joint, so that $p_i(x,z_{(i)}) = p(x,z)$. For example, this would be the case for a global parameter in a hierarchical Bayesian model. Even in this case, the number of factors has been reduced by $n-1$, since we eliminated all terms $\log q_j(z_j,\lambda_j)$

for $j \neq i$. Hence, following the same notation as before, the variance of this estimator grows as $O(F + n - n + 1) = O(F)$.

On the other hand, assume the best case scenario, where $p_i(x, z_{(i)})$ is non-trivial and, moreover, it does not grow (in number of factors) as more terms are added to z. This would be true for the nodes near the leaves in a hierarchical model. Then, the variance of the new estimator becomes $O(1)$.

We conclude that Rao-Blackwellization has indeed improved the quality of the estimator, although the amount of improvement depends on the particular role of each z_i in the graphical structure of the model.

2.4 Control variates

2.4.1 Description of the method

Control variates is another variance reduction technique, which the authors of the original BBVI paper apply to $\hat{\nabla}^{RB}_{\lambda_i}\mathcal{L}(\lambda_i)$ to further improve its accuracy. There are many related methods that fall in the category of control variates, but we will focus on the one that the authors use, which is also the most popular, and is known as the regression estimator [9].

Let u be a random variable, $\mu \in \mathbb{R}$ an unknown quantity, and a function f which is an unbiased estimator of μ; i.e., such that $\mathbb{E}(f(u)) = \mu$. Now, suppose that we can solve a similar problem, in the sense that we have a function $h(u) \in \mathbb{R}$, such that $h(u) \approx f(u)$, and $\mathbb{E}(h(u)) = \theta$ is known. The precise meaning of $h(u) \approx f(u)$ is not clear now but will be made clearer later.

We define a new estimator of μ:

$$g(u) \triangleq f(u) - \beta(h(u) - \theta) \tag{15}$$

First, note that $\forall \beta \in \mathbb{R}$, $g(u)$ is unbiased:

$$\mathbb{E}(g(u)) = \underbrace{\mathbb{E}(f(u))}_{\mu} - \beta(\underbrace{\mathbb{E}(h(u)) - \theta}_{0}) = \mu$$

Its variance is given by the expression:

$$Var(g(u)) = Var(f(u)) + \beta^2 Var(h(u)) - 2\beta Cov(f(u), h(u))$$

Now, assume f and h are fixed. Because $Var(g(u))$ is a quadratic convex expression on β, we can find

its unique minimizer by using the first order condition:

$$0 = \frac{\partial}{\partial \beta} Var(g(u)) = 2\beta Var(h(u)) - 2Cov(f(u), h(u))$$
$$\Rightarrow \beta^* = \frac{Cov(f(u), h(u))}{Var(h(u))}$$

Note that β^* is equal to the Ordinary Least Squares (OLS) solution for the linear regression:

$$f(u) = \mu + \beta(h(u) - \theta)$$

Not surprisingly, this is the reason for the name of the estimator. In fact, this procedure can be trivially extended to multiple control variates by solving for the vector β using the OLS solution for multiple linear regression. Understanding control variates as a regression procedure helps us gain an intuition of its workings. Indeed, because of 15, $g(u)$ can be identified with the residual of the regression. Thus, the more of $f(u)$ that can be captured by $h(u)$, the lower the variance of the new estimator $g(u)$.

The previous statement can be formalized. Plugging the optimal β back in the expression for the variance, we obtain:

$$\begin{aligned} Var(g(u)) &= Var(f(u)) + \frac{Cov(f(u), h(u))^2}{Var(h(u))^2} Var(h(u)) - 2\frac{Cov(f(u), h(u))}{Var(h(u))} Cov(f(u), h(u)) \\ &= Var(f(u)) - \frac{Cov(f(u), h(u))^2}{Var(h(u))} \\ &= Var(f(u)) - \frac{\rho_{fh}^2 Var(f(u)) Var(h(u))}{Var(h(u))} \\ &= Var(f(u))(1 - \rho_{fh}^2) \\ &\leq Var(f(u)) \end{aligned}$$

where ρ_{fh} is the correlation between $f(u)$ and $h(u)$. Hence, the contribution of the regression estimator will be higher when $|\rho_{fh}|$ is higher. This is what we meant in the previous paragraph by $h(u) \approx f(u)$.

Finally, note that nor $Cov(f(u), h(u))$ nor $Var(h(u))$ will be known a priori for arbitrary f, h. Thus, they have to be estimated from simulations in order to obtain an approximation of β. When we use $g(u)$ with this approximation, we lose the unbiasedness guarantee, although this bias is usually small [9].

2.4.2 Application to the RB estimator of the gradient

The authors of BBVI propose to use the score function $\nabla_{\lambda_i} \log q_i(z_i; \lambda_i)$ as a control variate for $\hat{\nabla}^{RB}_{\lambda_i}\mathcal{L}(\lambda_i)$. They base their choice on two arguments:

1. The score function only depends on the variational distribution, thus not requiring custom analytical calculations for different models.

2. Its expectation is known (and equal to 0 as we showed in Section 1), for every variational distribution q.

It is difficult to argue against their choice, since both of the above requirements are needed for a truly black box implementation of VI. And in fact, there is an additional benefit of choosing this control variate which the authors don't explicitly mention. According to [9], since evaluating a control variate adds computational time per iteration, a particular choice of control variate is worth the additional cost only if:

$$(1 - \rho^2_{fh})\frac{c_f + c_h}{c_f} < 1$$

where c_f is the cost of evaluating f and c_h is the additional cost of evaluating h given that we have already computed f. Now, because the score function needs to be evaluated regardless, since it is an input to $\hat{\nabla}^{RB}_{\lambda_i}\mathcal{L}(\lambda_i)$, we have that $c_h = 0$, and thus the inequality is true for any ρ^2_{fh}. In truth, this analysis does not consider the cost of estimating β, which could significantly alter the conclusions.

There is one additional step needed to apply the regression estimator to $\hat{\nabla}^{RB}_{\lambda_i}\mathcal{L}(\lambda_i)$. In the previous section, we only considered the case $f(u) \in \mathbb{R}$, and therefore we need to extend it to the multidimensional case. One natural way to do it is simply apply the procedure in an element-wise fashion. However, the authors choose a different path. They propose to use a single $\beta_i \in \mathbb{R}$ for all the components of $\hat{\nabla}^{RB}_{\lambda_i}\mathcal{L}(\lambda_i)$, obtained as follows:

$$\beta_i = \frac{\sum_{d=1}^{d_i} Cov(f_i^d, h_i^d)}{\sum_{d=1}^{d_i} Var(h_i^d)} \tag{16}$$

Although the authors state that is the "optimal" way to obtain the coefficient in the multivariate case, they do not specify according to which criterion this is an optimal choice. Nevertheless, it can be shown that β_i minimizes the trace of the covariance matrix:

$$\beta_i = \arg\min_{\beta \in \mathbb{R}} \mathbf{tr}(Var(g(\beta)))$$

Indeed, let $g(u) = f(u) - \beta(h(u) - \theta)$, with $g(u) \in \mathbb{R}^V$. Then, the variances of each of its components $g_v(u)$ can be written as:

$$Var(g_v(u)) = Var(f_v(u)) + \beta^2 Var(h_v(u)) - 2\beta Cov(f_v(u), h_v(u))$$

Summing over v:

$$\mathbf{tr}(Var(g(\beta))) = \sum_v Var(g_v(u)) = \sum_v Var(f_v(u)) + \beta^2 \sum_v Var(h_v(u)) - 2\beta \sum_v Cov(f_v(u), h_v(u))$$

We recognize that this equation has the same form in terms of β as the one in the univariate case, so its unique minimizer is:

$$\beta^* = \frac{\sum_v Cov(f_v, h_v)}{\sum_v Var(h_v)}$$

Even though this choice of β is optimal in the above sense, it could be problematic when the covariance terms add to 0, even though their magnitudes may be arbitrarily large. In this case, the control variate would be rendered useless (this will be revisited in Section 5).

2.5 Improved BBVI algorithm

Algorithm 2: RB+CV BBVI

input : Model, tolerance τ, batch size S
Randomly initialize λ^0
Set $t \leftarrow 0$, $\Delta \leftarrow \infty$
while $\Delta > \tau$ **do**
  $t \leftarrow t + 1$
  $z^1 ... z^S \overset{iid}{\sim} q(z; \lambda^{t-1})$
  for $i \leftarrow 1$ **to** n **do**
    $f_i \leftarrow \frac{1}{S} \sum_s \nabla_{\lambda_i} [\log q_i(z_i^s; \lambda_i^{t-1})](\log p_i(x, z_{(i)}^s) - \log q_i(z_i^s; \lambda_i^{t-1}))$
    $h_i \leftarrow \frac{1}{S} \sum_s \nabla_{\lambda_i} [\log q_i(z_i^s; \lambda_i^{t-1})]$
    $\hat{\beta}_i \leftarrow \frac{\sum_d \widehat{Cov}(f_i^d, h_i^d)}{\sum_d \widehat{Var}(h_i^d)}$
    $g_i \leftarrow f_i - \beta_i h_i$
    $\lambda_i^t \leftarrow \lambda_i^{t-1} + \rho_i^t g_i$
  $\Delta \leftarrow \frac{\|\lambda^t - \lambda^{t-1}\|}{\|\lambda^{t-1}\|}$
output: $\lambda^* = \lambda^t$

Algorithm 2 presents the improved version of the naive Algorithm 1, which incorporates both the Rao-Blackwellized gradients and the control variates.

Notice that, since $\mathbb{E}[h_i^d] = 0$, then $Cov(f_i^d, h_i^d) = \mathbb{E}(f_i^d h_i^d)$ and $Var(h_i^d) = \mathbb{E}((h_i^d)^2)$. With these, we construct the MC estimators as:

$$\widehat{Cov}(f_i^d, h_i^d) = \frac{1}{S} \sum_s f_i^d(z^s) h_i^d(z^s) \qquad \widehat{Var}(h_i^d) = \frac{1}{S} \sum_s (h_i^d(z^s))^2$$

2.5.1 Additional extensions

The authors propose two additional extensions to Algorithm 2:

AdaGrad: set the sequence of step sizes according to the (diagonal) AdaGrad method [10]:

$$G^t = G^{t-1} + g^t (g^t)^T$$
$$\rho^t = \eta \text{diag}((G^t)^{-1/2})$$

where g^t is the RB+CV gradient at iteration t, $G^0 = 0$ and η is an adjustable parameter. In practice, since we only use the diagonal of G, we don't store the full outer product matrix, and only keep track of its diagonal.

Since the BBVI paper was published, there have been works questioning the true contribution of adaptive methods in ML applications, most notably [11]. Therefore, both in Sections 3 and 4, we compare a simple Stochastic Gradient Descent (SGD) implementation to the AdaGrad version.

Scalability in large hierarchical models: consider a model with the following log joint distribution:

$$\log p(x, \gamma, z) = \log p(\gamma) + \log p(z|\gamma) + \sum_{i=1}^{n} \log p(x_i|z_i, \gamma)$$

Notice that all the factors in the joint depend on the latent variable γ, so that $p_i(x, z_{(i)}) = p(x, z)$. In particular, every data point depends on it. Therefore, every evaluation of the gradient with respect to γ will require a full pass over the dataset. The authors argue that this situation does not scale well, and therefore propose to replace every occurrence of $\sum_{i=1}^{n} \log p(x_i|z_i, \gamma)$ with $n \log p(x_j|z_j, \gamma)$, with $j \sim \text{Uniform}(1, ..., n)$., which is an unbiased estimator of the former. They refer to this approach as "doubly stochastic".

3 Simulations

In order to explore the workings of BBVI, we apply it to a simple enough model: a univariate Gaussian mixture model:

$$\mu_k \overset{iid}{\sim} N(0, \sigma^2)$$
$$c_i \overset{iid}{\sim} \text{Categorical}(1/K, ..., 1/K)$$
$$x_i|c_i, \mu \overset{indep}{\sim} N(c_i^T \mu, 1)$$

where $k = 1...K$ are the distinct clusters, there are N observations x_i, and σ^2 is a known parameter. Also,

c_i is encoded as a one-hot vector. We propose the following mean-field variational approximations:

$$q(\mu_k; m_k, s_k^2) = N(m_k, s_k^2) \qquad\qquad q(c_i; \phi) = \text{Categorical}(\phi_i)$$

where $\mu_k \in \mathbb{R}$, $s_k^2 > 0$, and ϕ_i lies in the interior of the standard simplex in $\mathbb{R}^K$. Hence, $\lambda \in \Lambda \subsetneq \mathbb{R}^{K(2+N)}$. For the positive parameters s^2 and ϕ, we had to apply hard-thresholding of 0.01 (and also re-normalization for ϕ) so as to ensure that the gradient ascent updates stayed within Λ.

The model score functions are:

$$\nabla_{m_k, s_k^2}[\log q(\mu_k; m_k, s_k^2)] = \left(\frac{\mu_k - m_k}{s_k^2}, -\frac{1}{2s_k^2} + \frac{(\mu_k - m_k)^2}{2s_k^4} \right)^T$$
$$\nabla_{\phi_i}[\log q(c_i; \phi)] = \left(\frac{c_{i1}}{\phi_{i1}}, ..., \frac{c_{iK}}{\phi_{iK}} \right)^T$$

The Rao-Blackwellized gradients are:

$$\nabla_{m_k, s_k^2} \mathcal{L}(\lambda) = \mathbb{E}_{m_k, s_k^2} \left[\nabla_{m_k, s_k^2}[\log q(\mu_k; m_k, s_k^2)] \left(-\frac{\mu_k^2}{2\sigma^2} - \sum_{i=1}^{N} c_{ik} \frac{(x_i - \mu_k)^2}{2} + \frac{(\mu_k - m_k)^2}{2s_k^2} \right) \right]$$
$$\nabla_{\phi_i} \mathcal{L}(\lambda) = \mathbb{E}_{\phi_i} \left[\nabla_{\phi_i}[\log q(c_i; \phi_i)] \left(-\frac{(x_i - c_i^T \mu)^2}{2} - \sum_{k=1}^{K} c_{ik} \log \phi_{ik} \right) \right]$$

The BBVI algorithm to solve this model was implemented in R. We sampled synthetic data with $N = 100$, $K = 2$, $\mu = (-2, 2)$, and assumed $\sigma^2 = 25$. We then used BBVI to find the approximating distributions. We distinguished between the naive, the RB, and the RB+CV implementations, in order to assess the marginal contributions of each of them. We set $S = 1000$ as the authors did, and used a tolerance for convergence $\Delta = 0.0001$. Because this was a rather small model in terms of latent variables, we did not require the use of the doubly stochastic approach proposed by the authors.

Figure 1 shows the trajectories for (m, s^2), the squared correlation between the true allocations to cluster 1 c_{i1} and assignment probabilities ϕ_{i1}^t, and the ELBO, which was analytically derived for this simple model[1]. We compared increasingly complex versions of BBVI, starting with the naive algorithm, then adding RB gradients, then the control variates, and finally replacing the constant step-size SGD with the AdaGrad schedule. The naive procedure is not shown, as it did not produce meaningful results in 5,000 iterations.

The first thing to notice is that all 3 alternatives converge to a neighborhood around the optimum in less than 100 iterations, which take less than a minute. As expected, the SGD implementations never converge, but oscillate around the optimum. In contrast, the AdaGrad version stops, and actually

[1]Because the ELBO is available, we used it to check convergence instead of the original criterion proposed by the authors.

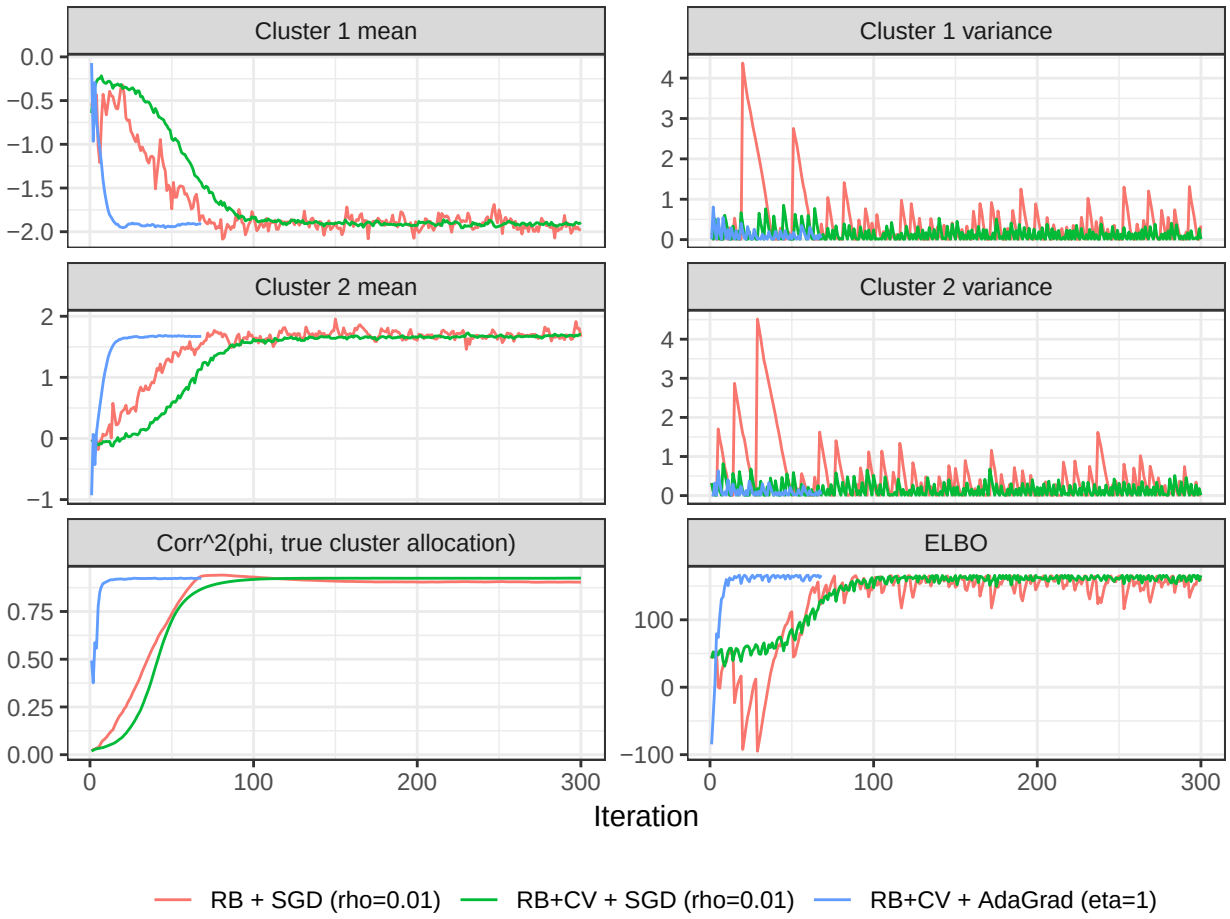

Figure 1: First 300 iterations for different implementations of BBVI (naive not shown as it didn't produce meaningful results). The bottom left plot shows the squared correlation between the true allocations to cluster 1 c_{i1}, and the probabilities ϕ_{i1}^t. The ELBO was derived analytically.

approaches the optimum much faster.

The second thing to notice is the improvement in variance when the control variate is used. This can be seen both in the cluster means and the variances, but it's much more relevant in the latter. Indeed, the trajectories for the variances in the RB+CV version are much less spiky than in the RB case. These spikes occur because the partial derivative of the score function with respect to the variance has terms $1/s_k^2$, which explode as the variance approaches the hard threshold of 0.01. In fact, if one sets the threshold to 0.0001, then the neither the RB nor the RB+CV version produce meaningful results, as the spikes become much more problematic (not shown).

In the same way, notice how AdaGrad shows the lowest volatility in the cluster variances' trajectories. It is sensible to argue that this is achieved by taking advantage of the exploding gradients near zero, which rapidly decrease the step size for these components without changing the ones for rest of the

variational parameters. In fact, AdaGrad achieves convergences even in the case with the hard-threshold set at 0.0001 (not shown), although it stops before reaching the optimum.

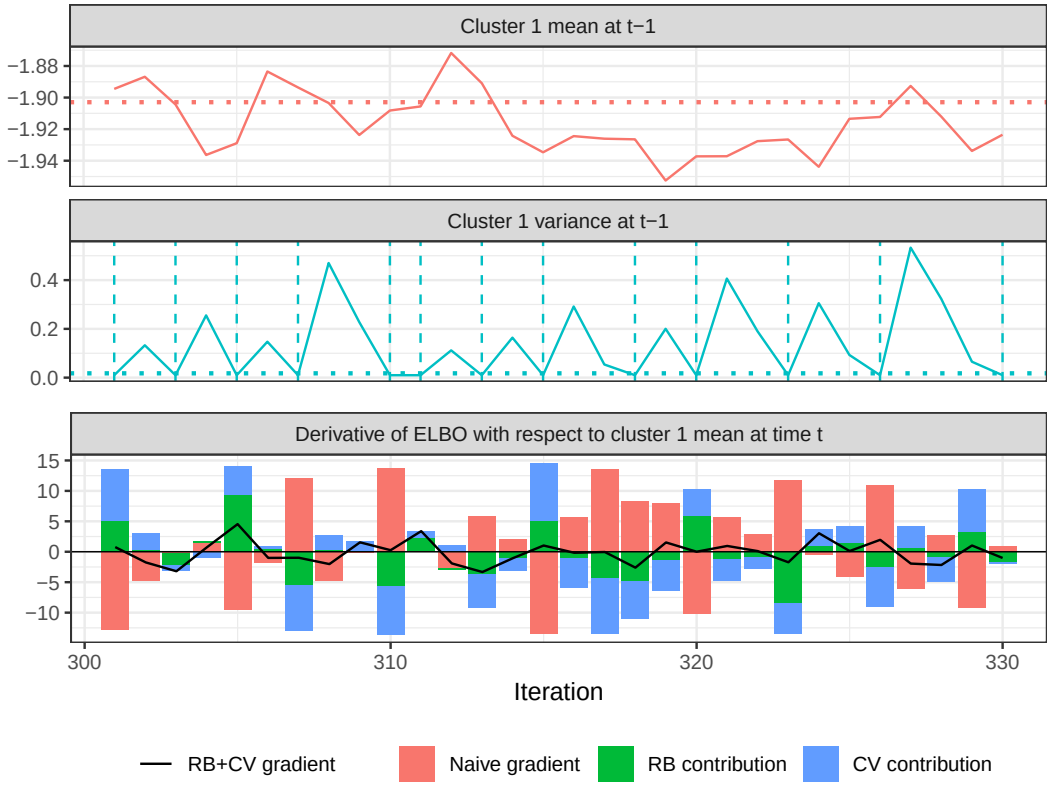

Figure 2: Zoom-in into the trajectories of the cluster 1 variational parameters between iteration 301 and 330. The horizontal dotted lines show the true parameters. The vertical dashed lines in panel 2 show the times when the variance is at the threshold level (0.01). The bottom plot shows the contributions of the naive gradient and each of the variance reduction methods to the total RB+CV gradient for the mean (using SGD with $\rho = 0.01$). The total gradient is the sum of the bars, and is depicted as a solid black line.

In spite of the above, it is interesting to note that, even though the gradients of the scores with respect to the means also have terms of the form $1/s_k^2$, both the RB and the RB+CV (SGD) trajectories for these parameters are much less spiky than for the variances. As a way to understand how this is achieved, we show in Figure 2 the trajectories of the variational parameters for cluster 1 at the steady state; i.e., when SGD has already arrived at a close neighborhood near the optimum. The horizontal dotted lines show the true parameters, while the vertical dashed lines in panel 2 show the times when the variance is at the threshold level (in this case 0.01). Notice how close to 0 is the true value for the variance (≈ 0.02).

We also show in the bottom plot the following decomposition for the RB+CV gradient:

$$\widehat{\frac{\partial \mathcal{L}}{\partial m}}^{RB+CV} = \widehat{\frac{\partial \mathcal{L}}{\partial m}}^{\text{naive}} + \underbrace{\left(\widehat{\frac{\partial \mathcal{L}}{\partial m}}^{RB} - \widehat{\frac{\partial \mathcal{L}}{\partial m}}^{\text{naive}}\right)}_{\text{RB contribution}} + \underbrace{\left(\widehat{\frac{\partial \mathcal{L}}{\partial m}}^{RB+CV} - \widehat{\frac{\partial \mathcal{L}}{\partial m}}^{RB}\right)}_{\text{CV contribution}} \tag{17}$$

Of course, 17 is a simple identity. Still, it helps us identify the marginal contributions of RB and CV. These components are shown in the stacked columns. The total RB+CV derivative is shown in a solid line. First, note how almost all of the vertical dashed lines above coincide with jumps in the naive derivative for the mean. However, the total gradient does not become greatly affected. And second, we see that most of the reduction in volatility is being achieved by CV, because the RB gradient for m_k does not remove terms associated with s_k^2 (in fact, the absolute value of the RB contribution is more or less constant across iterations at this stage). The CV method works simply because both the score function and the gradient are exploding, and so they can cancel each other.

We will revisit the problem of spiking trajectories in Section 5 where we will propose an alternative way to deal with this issue.

4 Application to a longitudinal dataset

In this section, we investigate the performance of BBVI in a real dataset, while comparing it to an established technique for Bayesian Inference.

4.1 Description of the dataset

The data corresponds to the "Panel Study Income Dynamics" (PSID), available in the `R` package `faraway` [12], which itself is a companion to [13]. According to the description in the package, the PSID is a longitudinal study of a representative sample of U.S. individuals, which begun in 1968. The study was conducted at the Survey Research Center, Institute for Social Research, University of Michigan and is still continuing. The data represents a small subset of the total data.

Table 1: Summary of the PSID dataset

Variable	Mean	Min	p5	p25	p50	p75	p95	Max
Age	32	25	25	28	34	36	38	39
Years of education	12	3	7	10	12	13	16	16
Income	13,575	3	1,000	4,300	9,000	18,050	38,560	180,000

The dataset contains information for 85 people (39 females and 46 males), from the year 1968 to 1990.

The dataset is unbalanced, with each person appearing between 11 and 23 times. A summary of the continuous covariates included is summarized in Table 1.

4.2 Description of the model

We follow [13] for the specification of the model. The response variable becomes the logarithm of the income, in order to address its positive skew, which can be appreciated in Table 1. We use a linear mixed effects model:

$$y_{tp} = \beta^T X_{tp} + \alpha_p + \gamma_p t + \varepsilon_{tp}$$

where α_p is a random intercept, and γ_p is a random slope with respect to the time variable (year of the study). The explanatory variables in X include an intercept, age, years of education, and an interaction between gender and time. The above can be translated into:

$$\begin{aligned}
\beta, \sigma_\alpha, \sigma_\gamma, \sigma_\varepsilon &\sim \text{improper uniform prior} \\
\alpha_p &\overset{iid}{\sim} N(0, \sigma_\alpha^2) \\
\gamma_p &\overset{iid}{\sim} N(0, \sigma_\gamma^2) \\
y_{tp} | \alpha_p, \gamma_p &\overset{indep}{\sim} N(\hat{y}_{tp}, \sigma_\varepsilon^2)
\end{aligned}$$

where $\hat{y}_{tp} \triangleq \beta^T X_{tp} + \alpha_p + \gamma_p t$.

4.3 BBVI for the linear mixed effects model

Table 2: Variational families and components needed to apply BBVI to the PSID model

z_i	$q_i(z_i)$	$\log q_i(z_i)$	$\nabla_{\lambda_i} \log q_i(z_i)$	$p_i(x, z_{(i)})$
β_j	$N(m_{\beta_j}, s_{\beta_j})$	$\log \phi(\beta_j; m_{\beta_j}, s_{\beta_j})$	$\nabla \log \phi(\beta_j; m_{\beta_j}, s_{\beta_j})$	$\sum_{t,p} \log \phi(y_{tp}; \hat{y}_{tp}, \sigma_\varepsilon^2)$
α_p	$N(m_{\alpha_p}, s_{\alpha_p})$	$\log \phi(\alpha_p; m_{\alpha_p}, s_{\alpha_p})$	$\nabla \log \phi(\alpha_p; m_{\alpha_p}, s_{\alpha_p})$	$\log \phi(\alpha_p; 0, \sigma_\alpha^2) + \sum_t \log \phi(y_{tp}; \hat{y}_{tp}, \sigma_\varepsilon^2)$
γ_p	$N(m_{\gamma_p}, s_{\gamma_p})$	$\log \phi(\gamma_p; m_{\gamma_p}, s_{\gamma_p})$	$\nabla \log \phi(\gamma_p; m_{\gamma_p}, s_{\gamma_p})$	$\log \phi(\gamma_p; 0, \sigma_\gamma^2) + \sum_t \log \phi(y_{tp}; \hat{y}_{tp}, \sigma_\varepsilon^2)$
σ_α	$\text{Gamma}(a_{\sigma_\alpha}, b_{\sigma_\alpha})$	$\log G(\sigma_\alpha; a_{\sigma_\alpha}, b_{\sigma_\alpha})$	$\nabla \log G(\sigma_\alpha; a_{\sigma_\alpha}, b_{\sigma_\alpha})$	$\sum_p \log \phi(\alpha_p; 0, \sigma_\alpha)$
σ_γ	$\text{Gamma}(a_{\sigma_\gamma}, b_{\sigma_\gamma})$	$\log G(\sigma_\gamma; a_{\sigma_\gamma}, b_{\sigma_\gamma})$	$\nabla \log G(\sigma_\gamma; a_{\sigma_\gamma}, b_{\sigma_\gamma})$	$\sum_p \log \phi(\gamma_p; 0, \sigma_\gamma)$
σ_ε	$\text{Gamma}(a_{\sigma_\varepsilon}, b_{\sigma_\varepsilon})$	$\log G(\sigma_\varepsilon; a_{\sigma_\varepsilon}, b_{\sigma_\varepsilon})$	$\nabla \log G(\sigma_\varepsilon; a_{\sigma_\varepsilon}, b_{\sigma_\varepsilon})$	$\sum_{t,p} \log \phi(y_{tp}; \hat{y}_{tp}, \sigma_\varepsilon^2)$

The mean-field variational approximations chosen for each latent variable, along with the necessary calculations for applying BBVI to the model, are summarized in Table 2. There, $\phi(m, s)$ refers to the

normal pdf, and therefore:

$$\nabla \log \phi(x; m, s) = \left(\frac{x-m}{s^2}, -\frac{1}{s} + \frac{(x-m)^2}{s^3} \right)^T$$

Additionally, $G(g; a, b)$ refers to the Gamma pdf, and:

$$\nabla \log G(g; a, b) = \left(\log b - \psi(a) + \log g, \frac{a}{b} - g \right)^T$$

where $\psi(\cdot)$ is the digamma function.

In our implementation, we didn't require the use of the doubly stochastic extension, because the algorithm converged in reasonable time using the full dataset.

4.4 Results

We implemented BBVI (Algorithm 2) for model described earlier in `R`. We run it for 1 hour, after which the algorithm had completed a bit more than 2,000 iterations. For each of them, we computed the predictive log likelihood for the response variable, using a trimmed average[2] of 30 independent additional draws of the latent variables from q with the variational parameters for the iteration.

Our implementation of BBVI was compared to the No-U-Turn Sampler (NUTS) implemented in Stan [14], which we use through its interface for `R` (the package `rstan`). For the linear mixed effects model, we used the code provided in [15], but modified it to obtain the predictive log likelihoods at each iteration. We run a single chain of the algorithm, to obtain a total of 2,000 iterations (including warm-up)[3].

The results of the tests can be seen in Figure 3. It is important to note that our BBVI implementation is working correctly, slowly achieving increasing values for the predictive log likelihood, even though it starts an order of magnitude worse than Stan. However, the noisy curve for BBVI shows just how much variance is embedded in its calculations. Recall that these are averages for 30 samples, whereas the line for Stan is evaluated using only one sample (the current state of the chain).

These results are somewhat contradictory with the claims in the BBVI paper that BBVI achieves better performance than a "Metropolis-Hastings-within-Gibbs" (MH-Gibbs) sampler [16]. This is an extension of the Gibbs sampler for cases in which we cannot simulate directly from the full conditionals. In spite of this, it is still guaranteed that its stationary distribution is the true posterior, so it is an exact inference procedure. Therefore, it is surprising that BBVI surpassed it in accuracy.

[2]We trimmed the lower and upper 25% of the sums, because of the high variance, which was still not completely eliminated, as the noisy red line of Figure 3 shows.

[3]We previously run it with multiple chains in order to diagnose problems that could result in chains not mixing. We didn't find any such problems.

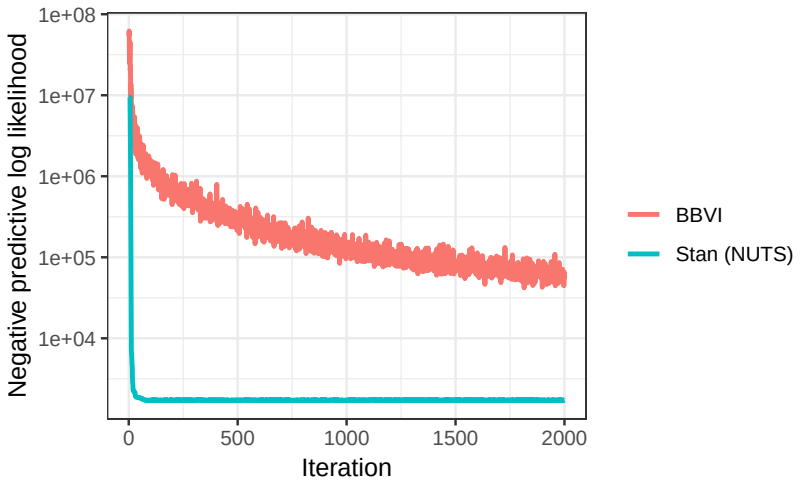

Figure 3: Negative predictive log likelihood for the response variable of the longitudinal data experiment, for 2,000 iterations of both BBVI and Stan (NUTS).

Nevertheless, the results they show could be explained by the fact that MH does not scale well (see [17] for a geometric explanation of this failure), even though in this case it is embedded within a Gibbs sampler. Indeed, if the model is large enough, the full conditionals could become high dimensional themselves, leading to the same problems that standard MH faces.

5 Extensions

5.1 Lessons from the experiments

Both the simulations of Section 3, and the application of Section 4, show that actually getting BBVI to produce meaningful results requires much more coding and tweaking than what is explicitly stated in Algorithm 2. The reasons for this behavior are multiple:

1. As was mentioned in Section 3, it is crucial to carefully establish lower bounds for bounded parameters, in order to avoid exploding gradients. It is difficult because there is a trade-off between accuracy (the true parameter could actually be arbitrarily close to the boundary), and stability.

2. Even for unbounded variational parameters, gradients can become very unstable for large values of their inputs. Therefore, in practice it is also necessary to restrict these parameters to a "reasonable" set, a task which faces the same trade-off as above.

3. For the above reasons, an adaptive step-size like AdaGrad becomes extremely useful. However, it adds the additional task of tuning, which is model (and probably even dataset) specific.

In fact, before applying BBVI to the longitudinal dataset, we attempted to fit the Latent Dirichlet Allocation (LDA) [18] for performing topic modeling on the rather small dataset "KOS blog entries" [19], which barely contains 3,430 documents and a vocabulary of 6,906 words.

Even though LDA is a conjugate model, so that a Gibbs sampler is easy to derive, the comparison to BBVI, although not completely fair, was still interesting, because of the claim in the original papar that BBVI achieved better performance than MH-Gibbs (recall the discussion in Section 4.4).

However, although we successfully implemented a Gibbs sampler that converged in a couple of minutes for 20 topics, we weren't able to get BBVI to produce reasonable results after hours of iterations, for just 2 topics. Although it is possible that this was an implementation error, it is also true that the Gibbs sampler requires more or less the same coding expertise to be correctly applied. As a future reference, we left the details of the calculations for using BBVI to fit the LDA in Appendix B.

5.2 Proposed extensions

5.2.1 Change of variables for bounded variational parameters

Let $f : S \subsetneq \mathbb{R}^p \to \mathbb{R}$ be a differentiable function, with S open. Consider the un-constrained optimization problem:

$$\max_{s \in S} f(s)$$

which we solve by gradient descent:

$$s^{u+1} = s^u + \rho^u \nabla_s f(s^u)$$

for some step-size sequence ρ^u. Now, assume there exists a bijective map $T : \mathbb{R}^p \to S$, such that its Jacobian is defined. Then, $s = T(t)$ is a change of variable, which we can use to solve the equivalent problem:

$$\max_{t \in \mathbb{R}^P} f(T(t))$$

We can further assume that this change of variable is of the form $s_p = T_p(t_p)$, for all p. Then, the Jacobian of T is a diagonal matrix:

$$J_T(t) = \text{diag} \underbrace{\left(\frac{dT_1(t_1)}{dt_1}, ..., \frac{dT_P(t_P)}{dt_P} \right)}_{\triangleq D_T(t)}$$

where $D_T(t)$ is the vector of stacked derivatives. Therefore, by the chain rule, the gradient descent

iteration for the equivalent optimization problem can be stated as:

$$t^{u+1} = t^u + \rho^u \nabla_s f(T(t^u)) \odot D_T(t^u)$$

where $\odot$ denotes the Hadamard or element-wise product.

We could apply the above method to BBVI in order to get rid of the need to tune appropriate thresholds for bounded variables. In practice, the transformations would be basically of three kinds:

- The identity function if s_p is unbounded in $\mathbb{R}$
- $s_p = e^{t_p}$ for positive variables.
- A logistic transformation for s_p defined on bounded intervals.

In fact, this simple change of variables is used in the Automatic Differentiation Variational Inference (ADVI) method [20]. This approach is similar to BBVI (two authors are the same). At its core, it differs from BBVI in that, besides using the change of variable, it substitutes the REINFORCE trick with the reparameterization gradient trick [5, 21, 22], and then uses automatic differentation to obtain all the necessary gradients.

5.2.2 BBVI within a probabilistic programming language

It is true that BBVI is a technique that is applicable for a large class of statistical models. In fact, by allowing for discrete latent variables, it can tackle more models than other black box VI methods that require differentiable $p(x, z)$ (like the ones based on the re-parameterization trick).

However, it is also true that, in order to be implemented, BBVI still requires some (albeit simple) analytical calculations. Recall that the RB gradient exploits the factorization of $p(x, z)$ in order to work. Hence, it is not sufficient to have a software library that simply has the pdf's of many approximating families and their corresponding gradients. We also need an environment capable of capturing the graphical model associated with a probabilistic model. In other words, this software would need more than just access to evaluations of $p(x, z)$; it would require a richer description of the model.

With the above, the ease with which a practitioner would be able to carry out BBVI on his models would be considerably increased. Other features of this software library that could certainly help would be automatically tuning of the AdaGrad parameter, along with any thresholds necessary in order to avoid numerical problems (recall the discussion in 5.1).

5.2.3 MC estimates of the ELBO as alternative stopping criterion

Recall the definition of the ELBO from Equation 4. Algorithm 2 could be modified in order to obtain MC estimates of this quantity:

$$\mathcal{L}(q) \approx \frac{1}{S} \sum_{s} (\log p(x, z^s) - \log q(z^s))$$

Note that this would not require additional computations, as all the factors that compose both $p(x, z)$ and $q(z)$ are already computed for the RB gradients. With this estimator, we could replace the stopping criterion of Algorithm 2 that assesses convergence in the variational parameters. This criterion has the disadvantage that parameters may differ in scale, so that the norm of λ may be influenced by just a couple of parameters with large absolute values.

5.2.4 Separate estimates for the CV parameter

As we mentioned in 2.4.2, the fact that the authors choose the β estimator of Equation 16 for the cases when λ is not univariate, might be problematic when the covariances of each component are large but with opposite signs. This could result $\hat{\beta} \approx 0$, rendering the control variate useless.

This situation could not arise if we had element-wise β coefficients. Moreover, this does not require additional computations, as they would use the same inputs that the unique β needs (namely, the element-wise covariances and variances).

6 Conclusions

In this report, we conducted a thorough analysis of the BBVI paper, replicating and completing all of their derivations, while also testing its proposed algorithm in both simulated and real data. These implementations helped us understand the practical limitations of a method that is otherwise theoretically sound. The main conclusion from this work is that there is still some way to go in order for BBVI to become a truly useful automatic approach for variational inference. Hopefully, the extensions that we propose could address some of the issues we described.

References

[1] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational inference: A review for statisticians," *Journal of the American Statistical Association*, vol. 112, no. 518, pp. 859–877, 2017. [Online]. Available: https://doi.org/10.1080/01621459.2017.1285773

[2] R. Ranganath, S. Gerrish, and D. Blei, "Black box variational inference," in *Artificial Intelligence and Statistics*, 2014, pp. 814–822.

[3] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine learning*, vol. 8, no. 3-4, pp. 229–256, 1992.

[4] S. Mandt, M. D. Hoffman, and D. M. Blei, "Stochastic gradient descent as approximate bayesian inference," *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 4873–4907, 2017.

[5] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," in *Proceedings of the 31st International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, E. P. Xing and T. Jebara, Eds., vol. 32, no. 2. Bejing, China: PMLR, 22–24 Jun 2014, pp. 1278–1286. [Online]. Available: http://proceedings.mlr.press/v32/rezende14.html

[6] C. J. Maddison, A. Mnih, and Y. Whye Teh, "The Concrete Distribution: A Continuous Relaxation of Discrete Random Variables," *ArXiv e-prints*, Nov. 2016.

[7] T. Salimans, D. Kingma, and M. Welling, "Markov chain monte carlo and variational inference: Bridging the gap," in *Proceedings of the 32nd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, F. Bach and D. Blei, Eds., vol. 37. Lille, France: PMLR, 07–09 Jul 2015, pp. 1218–1226. [Online]. Available: http://proceedings.mlr.press/v37/salimans15.html

[8] A. Miller, N. Foti, A. D'Amour, and R. P. Adams, "Reducing reparameterization gradient variance," in *Advances in Neural Information Processing Systems*, 2017, pp. 3708–3718.

[9] A. B. Owen, *Monte Carlo theory, methods and examples*, 2013.

[10] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, no. Jul, pp. 2121–2159, 2011.

[11] A. C. Wilson, R. Roelofs, M. Stern, N. Srebro, and B. Recht, "The Marginal Value of Adaptive Gradient Methods in Machine Learning," *ArXiv e-prints*, May 2017.

[12] J. Faraway, *faraway: Functions and Datasets for Books by Julian Faraway*, 2016, r package version 1.0.7. [Online]. Available: https://CRAN.R-project.org/package=faraway

[13] ——, *Extending the linear model with R: generalized linear, mixed effects and nonparametric regression models*. CRC press, 2016, vol. 124.

[14] B. Carpenter, A. Gelman, M. Hoffman, D. Lee, B. Goodrich, M. Betancourt, M. Brubaker, J. Guo, P. Li, and A. Riddell, "Stan: A probabilistic programming language," *Journal of Statistical Software, Articles*, vol. 76, no. 1, pp. 1–32, 2017. [Online]. Available: https://www.jstatsoft.org/v076/i01

[15] J. Faraway, "Stan analysis of a longitudinal model," http://www.maths.bath.ac.uk/~jjf23/stan/longitudinal.html, accessed: 2018-10-02.

[16] W. R. Gilks, "Full conditional distributions," *Markov chain Monte Carlo in practice*, pp. 75–88.

[17] M. Betancourt, "A Conceptual Introduction to Hamiltonian Monte Carlo," *ArXiv e-prints*, Jan. 2017.

[18] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.

[19] D. Dua and E. Karra Taniskidou, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[20] A. Kucukelbir, D. Tran, R. Ranganath, A. Gelman, and D. M. Blei, "Automatic differentiation variational inference," *Journal of Machine Learning Research*, vol. 18, no. 14, pp. 1–45, 2017. [Online]. Available: http://jmlr.org/papers/v18/16-107.html

[21] T. Salimans and D. A. Knowles, "Fixed-Form Variational Posterior Approximation through Stochastic Linear Regression," *ArXiv e-prints*, Jun. 2012.

[22] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," *ArXiv e-prints*, Dec. 2013.

[23] K. C. Border, "Lecture notes on leibniz' rule," December 2016 (accessed September 21, 2018). [Online]. Available: http://www.its.caltech.edu/~kcborder/Notes/LeibnizRule.pdf

Appendices

A Notes on the derivation of the gradient of the ELBO

In deriving 9, we have used Leibniz' integral rule to claim that:

$$\nabla_\lambda \int_z q(z;\lambda)(\log p(x,z) - \log q(z;\lambda))dz = \int_z \nabla_\lambda [q(z;\lambda)(\log p(x,z) - \log q(z;\lambda))]dz \tag{18}$$

For this to hold, we need to show that the function $f(\lambda, z) : \Lambda \times \mathbb{R}^n \to \mathbb{R}$ defined by:

$$f(\lambda, z) = q(z;\lambda)(\log p(x,z) - \log q(z;\lambda))$$

satisfies the following conditions [23]:

1. Λ must be an open subset of $\mathbb{R}^d$
2. $\forall \lambda \in \Lambda$, the function $z \mapsto f(\lambda, z)$ must be measurable
3. $\nabla_\lambda f(\lambda, z)$ exists for all $\lambda \in \Lambda$ and almost all z
4. $\exists \{g_i\}_{i=1\dots p}$ measurable such that $\forall \lambda \in \Lambda, \forall i : \left|\frac{\partial}{\partial \lambda_i} f(\lambda, z)\right| < g_i(z)$, for almost all z (i.e., the partial derivatives are bounded)

Under the above conditions, the Dominated Convergence Theorem can be applied to show that 18 holds. Let us then check the appropriateness of these conditions:

1. If the variational parameters are unbounded, then $\Lambda = \mathbb{R}^d$ and we're done. Otherwise, in order for this condition to hold, we will need to restrict our search to $\text{Int}(\Lambda)$.

2. To comply with this condition, we need ensure that the members of $\mathcal{Q}$ are differentiable wrt the variational parameters λ, which is the case for virtually all interesting models (a counterexample would be the Binomial(n, p) distribution, because it requires $n \in \mathbb{N}$).

3. Note that this restriction is equivalent to require that the ELBO be a measurable function. In turn, by 5, this is the same as requiring that $KL(q\|p(z|x))$ exists. For this to hold, it is sufficient to choose a family $\mathcal{Q}$ such that $\forall q \in \mathcal{Q}$, q and $p(z|x)$ agree on the sets of measure zero. For example, if in our model $z_i \in (0, \infty)$, we should only search among approximating distributions whose support is precisely that set.

4. The intermediate steps for the derivation of the first order conditions, tell us that:

$$\nabla_\lambda f(\lambda, z) = q(z; \lambda)\nabla_\lambda[\log q(z; \lambda)](\log p(x, z) - \log q(z; \lambda))$$

   which is equivalent to:

$$\frac{\partial}{\partial \lambda_i} f(\lambda, z) = q(z; \lambda)\frac{\partial \log q(z; \lambda)}{\partial \lambda_i}(\log p(x, z) - \log q(z; \lambda)) \ , \ \forall i$$

   Hence, we need to show that there exist measurable functions $g_i(z)$, such that:

$$\left|\frac{\partial}{\partial \lambda_i} f(\lambda, z)\right| = q(z; \lambda)\left|\frac{\partial \log q(z; \lambda)}{\partial \lambda_i} \log \frac{p(x, z)}{q(z; \lambda)}\right| \leq g_i(z)$$

B BBVI for LDA

In its simplest form, the LDA can be expressed as:

$$\begin{aligned}
\pi_d &\overset{iid}{\sim} \text{Dirichlet}(\alpha) \\
z_{dw}|\pi_d &\overset{indep}{\sim} \text{Categorical}(\pi_d) \\
b_k &\overset{iid}{\sim} \text{Dirichlet}(\gamma) \\
y_{dw}|z_{dw}, b &\overset{indep}{\sim} \text{Categorical}(b_{z_{dw}})
\end{aligned}$$

where:

- $\pi_d \in \mathbb{R}^K$ is the distribution of the d-th document over the K topics.
- z_{dw} gives the topic allocation for the w-th token in document d. It is important to understand that a token can appear multiple times in a document, and that the pair (d, w) refers to all those occurrences.
- $b_k \in \mathbb{R}^V$ is the distribution of the k-th topic over the V words in the vocabulary.
- y_{dw} is the actual word in the vocabulary that gets assigned to the w-th token in document d. This is the only observed data (along with the counts, which in this formulation are assumed constant).

α and γ are the concentration parameters of the Dirichlet distributions. When they are less than 1, the produced discrete distributions tend to be more sparse, which is one would usually expect in topic modeling. We set them both equal to 0.5.

The joint distribution can be written as follows:

$$p(\pi, b, z, y) = \prod_{d=1}^{D} \prod_{w=1}^{W_d} \prod_{k=1}^{K} \pi_{dk}^{\alpha-1+\mathbb{I}(z_{dw}=k)} \prod_{v=1}^{V} b_{kv}^{\gamma-1+\mathbb{I}(z_{dw}=k, y_{dw}=v)} \tag{19}$$

B.1 Gibbs sampler: full conditionals

We start with π_d. From 19:

$$\begin{aligned} p(\pi_d | b, z, y, \pi_{-d}) &\overset{\pi_d}{\propto} p(\pi, b, z, y) \\ &\overset{\pi_d}{\propto} \prod_{w=1}^{W_d} \prod_{k=1}^{K} \pi_{dk}^{\alpha-1+\mathbb{I}(z_{dw}=k)} \\ &= \prod_{k=1}^{K} \pi_{dk}^{\alpha-1+\sum_{w=1}^{W_d} \mathbb{I}(z_{dw}=k)} \end{aligned}$$

Therefore, the full conditional for π_d becomes:

$$\pi_d | z \overset{indep}{\sim} \text{Dirichlet} \begin{pmatrix} \alpha + \sum_{w=1}^{W_d} \mathbb{I}(z_{dw} = 1) \\ \vdots \\ \alpha + \sum_{w=1}^{W_d} \mathbb{I}(z_{dw} = K) \end{pmatrix}$$

Next, we obtain the full conditional for z_{dw}:

$$\begin{aligned} p(z_{dw} | b, z_{-dw}, y, \pi) &\overset{z_{dw}}{\propto} p(\pi, b, z, y) \\ &\overset{z_{dw}}{\propto} \prod_{k=1}^{K} \pi_{dk}^{\mathbb{I}(z_{dw}=k)} \prod_{v=1}^{V} b_{kv}^{\mathbb{I}(z_{dw}=k, y_{dw}=v)} \\ &= \prod_{k=1}^{K} \pi_{dk}^{\mathbb{I}(z_{dw}=k)} \left(\underbrace{\prod_{v=1}^{V} b_{kv}^{\mathbb{I}(y_{dw}=v)}}_{b_{k y_{dw}}} \right)^{\mathbb{I}(z_{dw}=k)} \\ &= \prod_{k=1}^{K} (\pi_{dk} b_{k y_{dw}})^{\mathbb{I}(z_{dw}=k)} \end{aligned}$$

Hence:

$$z_{dw} | b, y_{dw}, \pi_d \overset{indep}{\sim} \text{Categorical}(\pi_d \odot b_{\cdot, y_{dw}}^T)$$

where $\odot$ denotes the Hadamard or element-wise product. Finally, we see that for b_k:

$$\begin{aligned} p(b_k|b_{-k}, z, y, \pi) &\overset{b_k}{\propto} p(\pi, b, z, y) \\ &\overset{b_k}{\propto} \prod_{d=1}^{D} \prod_{w=1}^{W_d} \prod_{v=1}^{V} b_{kv}^{\gamma-1+\mathbb{I}(z_{dw}=k, y_{dw}=v)} \\ &= \prod_{v=1}^{V} b_{kv}^{\gamma-1+\sum_{d=1}^{D}\sum_{w=1}^{W_d}\mathbb{I}(z_{dw}=k, y_{dw}=v)} \end{aligned}$$

from which we conclude that:

$$b_k|z, y \overset{indep}{\sim} \text{Dirichlet}\begin{pmatrix} \gamma + \sum_{d=1}^{D}\sum_{w=1}^{W_d}\mathbb{I}(z_{dw}=k, y_{dw}=1) \\ \vdots \\ \gamma + \sum_{d=1}^{D}\sum_{w=1}^{W_d}\mathbb{I}(z_{dw}=k, y_{dw}=V) \end{pmatrix}$$

B.2 BBVI for LDA

Table 3: Components needed to apply BBVI to LDA

z_i	$q_i(z_i)$	$\log q_i(z_i)$	$\nabla_{\boldsymbol{\lambda}_i} \log q_i(z_i)$	$p_i(x, z_{(i)})$
π_d	Dirichlet(μ_d)	$\sum_k(\mu_{dk}-1)\log\pi_{dk}$	$\psi(\sum_k \mu_{dk}) - \psi(\mu_d) + \log\pi_d$	$\sum_k\left[\alpha - 1 + \sum_w \mathbb{I}(z_{dw}=k)\right]\log\pi_{dk}$
z_{dw}	Categorical(J_{dw})	$J_{dwz_{dw}}$	$(0...0, 1/J_{dwz_{dw}}, 0...0)$	$\log[\pi_{dz_{dw}} b_{z_{dw}y_{dw}}]$
b_k	Dirichlet(η_k)	$\sum_v(\eta_{kv}-1)\log b_{kv}$	$\psi(\sum_v \eta_{kv}) - \psi(\eta_k) + \log b_k$	$\sum_v\left[\gamma - 1 + \sum_{dw}\mathbb{I}(z_{dw}=k, y_{dw}=v)\right]$

To obtain all the components we need to apply BBVI to LDA, we can reuse most of the work that we did in the previous section. Therefore, we simply state the summarized results in Table 3. Here, $\psi(\cdot)$ is the digamma function.