Operator Variational Inference

Rajesh Ranganath[†], Jaan Altosaar[†], Dustin Tran[‡], David Blei[‡]

[†]Princeton University, [‡]Columbia University

Summary

- All variational inference requires statistical and computational tradeoffs. How do we formalize these tradeoffs?
- We use *operators*, or functions of functions, to design variational objectives. Operators enable us to analyze these tradeoffs.
- For example, we demonstrate *variational programs*—a rich class of posterior approximations that does not require a tractable density.

Variational Objectives

- Variational inference is an umbrella term for algorithms that cast Bayesian inference as optimization.
- We want to compute the posterior $p(\mathbf{z} \,|\, \mathbf{x})$, for latent variables $\mathbf{z} = (\mathbf{z}_1, \ldots, \mathbf{z}_d)$ and data $\mathbf{x}$.
- The evidence lower bound (ELBO) is the most popular objective,

$$\mathbb{E}_{q(\mathbf{z})}[\log p(\mathbf{x}, \mathbf{z}) - \log q(\mathbf{z})].$$

- Optimizing the ELBO imposes specific properties on $q \in \mathcal{Q}$.
- We aim to study objectives which trade off different properties.

Operator Variational Objectives

- We define a new class of variational objectives.
- There are three ingredients that form an *operator objective*:

1. An operator $O^{p,q}$ that depends on $p(\mathbf{z} \,|\, \mathbf{x})$ and $q(\mathbf{z})$.
2. A family of test functions $f \in \mathcal{F}$, where each $f(\mathbf{z}) : \mathbb{R}^d \to \mathbb{R}^d$.
3. A distance function $t(a) : \mathbb{R} \to [0, \infty)$.

$$\sup_{f \in \mathcal{F}} t(\ \mathbb{E}_{q(\mathbf{z})}[(O^{p,q} f)(\mathbf{z})]\ )$$

- It is the worst-case expected value among all functions $f \in \mathcal{F}$.
- To use these objectives, we impose two conditions:

1. *Closeness*. Its minimum is achieved at the posterior,

$$\mathbb{E}_{p(\mathbf{z}|\mathbf{x})}[(O^{p,p} f)(\mathbf{z})] = 0 \text{ for all } f \in \mathcal{F}.$$

2. *Tractability*. The operator $O^{p,q}$—originally in terms of $p(\mathbf{z} \,|\, \mathbf{x})$ and $q(\mathbf{z})$—can be written in terms of $p(\mathbf{x}, \mathbf{z})$ and $q(\mathbf{z})$.

- We parameterize $q(\mathbf{z}; \lambda)$ with standard approaches.
- We parameterize $f(\mathbf{z}; \theta)$ with a neural network.

Example: Langevin-Stein Operator Objective

For $f \in \mathcal{F}$, the operator is

$$(O^p f)(\mathbf{z}) = \nabla_z \log p(\mathbf{x}, \mathbf{z})^\top f(\mathbf{z}) + \nabla^\top f, \quad \nabla^\top f = \textstyle\sum_{i=1}^d \nabla_{z_i} f(\mathbf{z}).$$

With distance function $t(a) = a^2$, the objective is

$$\sup_{f \in \mathcal{F}} (\ \mathbb{E}_{q(\mathbf{z})}[\nabla_z \log p(\mathbf{x}, \mathbf{z})^\top f(\mathbf{z}) + \nabla^\top f]\ )^2.$$

Example: A Discrete Operator Objective

Langevin-Stein operators have a discrete analog. For example, consider a one-dimensional latent variable with support $z \in \{0, \ldots, c\}$. Then

$$(O^p f)(z) = \frac{f(z+1) p(z+1, \mathbf{x}) - f(z) p(z, \mathbf{x})}{p(z, \mathbf{x})}.$$

where f is a function such that $f(0) = 0$.

Example: KL Divergence as an Operator Objective

The operator is $(O^{p,q} f)(z) = \log q(\mathbf{z}) - \log p(\mathbf{x}, \mathbf{z}) \quad \forall f \in \mathcal{F}$.
With distance function $t(a) = a$, the objective is

$$\mathbb{E}_{q(\mathbf{z})}[\log q(\mathbf{z}) - \log p(\mathbf{x}, \mathbf{z})].$$

Operator Variational Inference

The operator objective is

$$\min_\lambda \max_\theta t(\ \mathbb{E}_\lambda[(O^{p,q} f_\theta)(z)]\ )$$

Fix $t(a) = a^2$; the case of $t(a) = a$ easily applies.
Gradient with respect to λ. (Variational approximation)

$$\nabla_\lambda \mathcal{L}_\theta = 2\ \mathbb{E}_\lambda[(O^{p,q} f_\theta)(Z)]\ \nabla_\lambda \mathbb{E}_\lambda[(O^{p,q} f_\theta)(Z)]$$

Gradient with respect to θ. (Test function)

$$\nabla_\theta \mathcal{L}_\lambda = 2\ \mathbb{E}_\lambda[(O^{p,q} f_\theta)(z)]\ \mathbb{E}_\lambda[\nabla_\theta O^{p,q} f_\theta(z)]$$

We use black box gradients with two sets of Monte Carlo estimates.

Characterizing Objectives: Variational Programs

The family $q \in \mathcal{Q}$ is typically limited by a tractable density.
We design operators that do not depend on q, $O^{p,q} = O^p$, such as

$$\sup_{f \in \mathcal{F}} (\ \mathbb{E}_{q(\mathbf{z})}[\nabla_z \log p(\mathbf{x}, \mathbf{z})^\top f(\mathbf{z}) + \nabla^\top f]\ )^2.$$

Variational programs enable a larger class of approximating families.

For example, consider a generative program of latent variables,

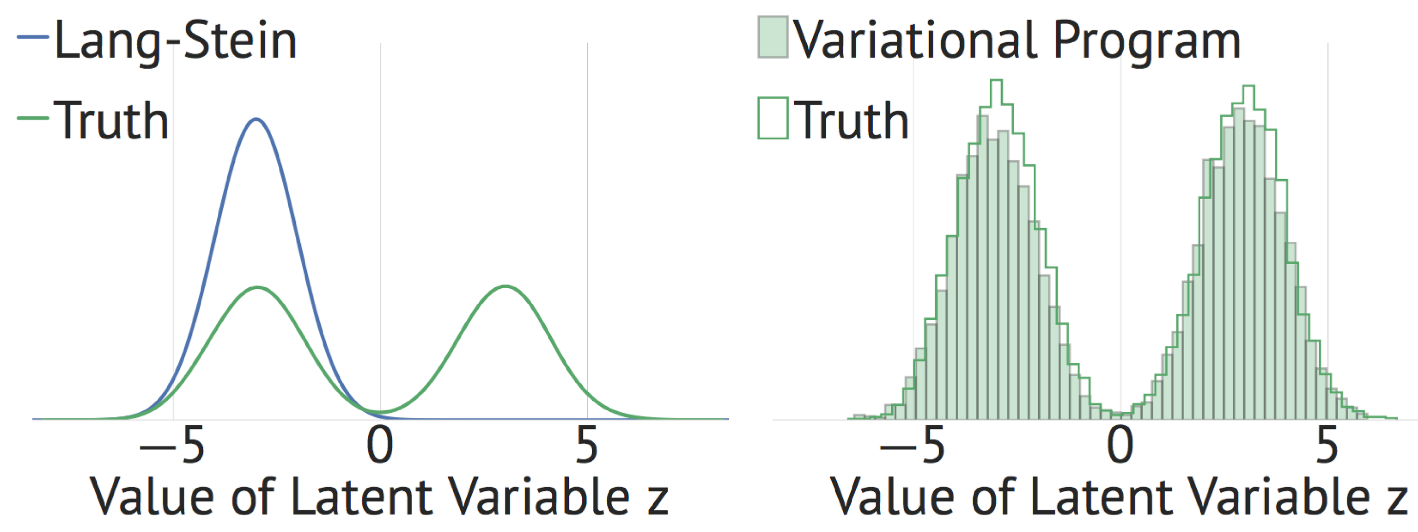
$$\boldsymbol{\epsilon} \sim \text{Normal}(0, 1), \quad \mathbf{z} = G(\boldsymbol{\epsilon}; \lambda),$$

where G is a neural network. The program is differentiable and generates samples for $\mathbf{z}$. Its density does not have to be tractable.

Experiments: 1-D Mixture of Gaussians

We posit the variational program $z \sim q$:

1. Draw $\epsilon, \epsilon' \sim \text{Normal}(0, 1)$.
2. If $\epsilon' > 0$, return $G_1(\epsilon; \lambda_1)$; else if $\epsilon' \leq 0$, return $G_2(\epsilon; \lambda_2)$.



Langevin-Stein with a Gaussian family fits a mode. Langevin-Stein with a variational program approaches the truth.

Experiments: Binarized MNIST

We model binarized MNIST, $\mathbf{x}_n \in \{0, 1\}^{28 \times 28}$, with

$$\mathbf{z}_n \sim \text{Normal}(0, 1),$$
$$\mathbf{x}_n \sim \text{Bernoulli}(\text{logistic}(\mathbf{z}_n^\top \mathbf{W} + \mathbf{b})),$$

where $\mathbf{z}_n$ has latent dimension 10 and with parameters $\{\mathbf{W}, \mathbf{b}\}$.

We posit the variational program $\mathbf{z} \sim q$:

$$\boldsymbol{\epsilon} \sim \text{Normal}(0, I)$$
$$\mathbf{h}_0 = \text{ReLU}({\mathbf{W}_0^q}^\top \boldsymbol{\epsilon} + \mathbf{b}_0^q)$$
$$\mathbf{h}_1 = \text{ReLU}({\mathbf{W}_1^q}^\top \mathbf{h}_0 + \mathbf{b}_1^q)$$
$$\mathbf{z} = {\mathbf{W}_2^q}^\top \mathbf{h}_1 + \mathbf{b}_2^q,$$

with parameters $\{\mathbf{W}_0^q, \mathbf{b}_0^q, \mathbf{W}_1^q, \mathbf{b}_1^q, \mathbf{W}_2^q, \mathbf{b}_2^q\}$.

At test time, we throw away half the pixels and impute them using different objectives. We compare the log-likelihood of the completed image.

Inference method	Completed data log-likelihood
Mean-field Gaussian + KL$(q \| p)$	-59.3
Mean-field Gaussian + LS	-75.3
Variational Program + LS	-58.9

The variational program performs better than KL without directly optimizing for likelihoods.