Stochastic Variational Inference

Mark van der Wilk

Department of Computing
Imperial College London

@markvanderwilk
m.vdwilk@imperial.ac.uk

February 25, 2022

Recap: Variational Inference

- KL measures discrepancy between distributions

$$\mathrm{KL}[q(\mathbf{z})||p(\mathbf{z}\,|\,\mathbf{x})] \geqslant 0 \qquad \text{with equality iff } q(\mathbf{z}) = p(\mathbf{z}\,|\,\mathbf{x}) \tag{1}$$

- Find approx $q_{\mathbf{v}}(\mathbf{z}) \approx p(\mathbf{z}\,|\,\mathbf{x})$ by minimising KL divergence:

$$\mathbf{v}^* = \operatorname*{argmin}_{\mathbf{v}} \mathrm{KL}[q_{\mathbf{v}}(\mathbf{z})||p(\mathbf{z}\,|\,\mathbf{x})] \tag{2}$$

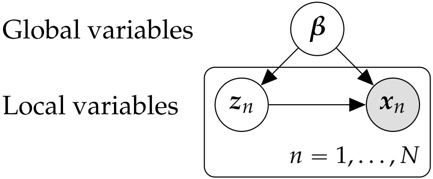
- Equivalent to maximising lower bound (ELBO) $\mathcal{L}$ since

$$\mathrm{KL}[q_{\mathbf{v}}(\mathbf{z})||p(\mathbf{z}\,|\,\mathbf{x})] = \log p(\mathbf{x}) - \mathcal{L}(\mathbf{v}) \tag{3}$$

$$\implies \mathbf{v}^* = \operatorname*{argmax}_{\mathbf{v}} \mathcal{L}(\mathbf{v}) \tag{4}$$

VI for Conditionally Conjugate Models

For the class of **conditionally conjugate models**, i.e. models with complete conditionals in exponential family (e.g. Bernoulli, Beta, Gamma, Gaussian, ...) and **mean-field** (independent) variational approximations.



- We have **closed-form** expression for ELBO
- Coordinate-ascent algorithm for maximising ELBO
- Important if you want to be a VI researcher, but not enough time.

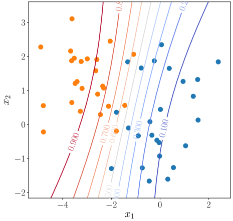
Overview of today

- Limitations of Conditionally-Conjugate VI
- Black-box variational inference
- Gradients of expectations

Limitation 1: Non-conjugate models

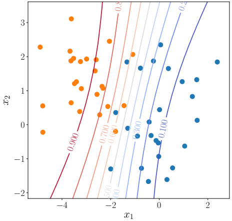
Example: Bayesian Logistic Regression

- Binary classification
- Inputs $x \in \mathbb{R}$, labels $y \in \{0, 1\}$
- Model parameter z (normally denoted by θ)



Example: Bayesian Logistic Regression

- Binary classification
- Inputs $x \in \mathbb{R}$, labels $y \in \{0, 1\}$
- Model parameter z (normally denoted by θ)

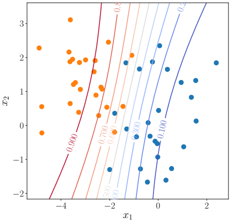


Prior on model parameter: $p(z) = \mathcal{N}\left(0,\, 1\right)$

Likelihood: $p(y_n|x_n, z) = \mathrm{Ber}(\sigma(zx_n))$

Example: Bayesian Logistic Regression

- Binary classification
- Inputs $x \in \mathbb{R}$, labels $y \in \{0, 1\}$
- Model parameter z (normally denoted by θ)



$$\text{Prior on model parameter: } p(z) = \mathcal{N}\left(0,\, 1\right)$$
$$\text{Likelihood: } p(y_n | x_n, z) = \mathrm{Ber}(\sigma(z x_n))$$

- Assume we have a single data point (x, y)
- Goal: Approximate the intractable posterior distribution $p(z|x, y)$ using variational inference

Example: Bayesian Logistic Regression (2)

- Choose Gaussian variational approximation:
  $q_{\mathbf{v}}(z) = \mathcal{N}\left(z; \mu, \sigma^2\right)$ ⏩ $\boldsymbol{\nu} =$

Example: Bayesian Logistic Regression (2)

- Choose Gaussian variational approximation:
  $q_{\mathbf{v}}(z) = \mathcal{N}\left(z; \mu, \sigma^2\right)$ ⏩ $\boldsymbol{\nu} = \{\mu, \sigma^2\}$

Example: Bayesian Logistic Regression (2)

- Choose Gaussian variational approximation:

  $q_{\mathbf{v}}(z) = \mathcal{N}\left(z; \mu, \sigma^2\right)$ ⏩ $\boldsymbol{\nu} = \{\mu, \sigma^2\}$
- Objective function: ELBO $\mathcal{F}(\boldsymbol{\nu})$

$$\mathcal{F}(m, \sigma^2) = \mathbb{E}_q[\log p(z) - \log q(z) + \log p(y|x, z)]$$

Example: Bayesian Logistic Regression (2)

- Choose Gaussian variational approximation: $q_{\mathbf{v}}(z) = \mathcal{N}\left(z; \mu,\, \sigma^2\right)$ ⏩ $\boldsymbol{\nu} = \{\mu, \sigma^2\}$
- Objective function: ELBO $\mathcal{F}(\boldsymbol{\nu})$

$$\begin{aligned}\mathcal{F}(m, \sigma^2) &= \mathbb{E}_q[\log p(z) - \log q(z) + \log p(y|x,z)] \\ &= -\tfrac{1}{2}(\mu^2 + \sigma^2) + \tfrac{1}{2}\log\sigma^2 + \mathbb{E}_q[\log p(y|x,z)] + \mathrm{c}\end{aligned}$$

Example: Bayesian Logistic Regression (2)

- Choose Gaussian variational approximation:

  $q_{\mathbf{v}}(z) = \mathcal{N}\left(z; \mu,\, \sigma^2\right)$ ⏩ $\boldsymbol{\nu} = \{\mu, \sigma^2\}$
- Objective function: ELBO $\mathcal{F}(\boldsymbol{\nu})$

$$\begin{aligned}\mathcal{F}(m, \sigma^2) &= \mathbb{E}_q[\log p(z) - \log q(z) + \log p(y|x,z)] \\ &= -\tfrac{1}{2}(\mu^2 + \sigma^2) + \tfrac{1}{2}\log \sigma^2 + \mathbb{E}_q[\log p(y|x,z)] + \mathrm{c}\end{aligned}$$

$$\mathbb{E}_q[\log p(y|x,z)] =$$

Example: Bayesian Logistic Regression (2)

- Choose Gaussian variational approximation:

  $q_{\mathbf{v}}(z) = \mathcal{N}\left(z; \mu, \sigma^2\right)$ ▶▶ $\boldsymbol{\nu} = \{\mu, \sigma^2\}$
- Objective function: ELBO $\mathcal{F}(\boldsymbol{\nu})$

$$
\begin{aligned}
\mathcal{F}(m, \sigma^2) &= \mathbb{E}_q[\log p(z) - \log q(z) + \log p(y|x, z)] \\
&= -\tfrac{1}{2}(\mu^2 + \sigma^2) + \tfrac{1}{2}\log\sigma^2 + \mathbb{E}_q[\log p(y|x, z)] + \mathrm{c} \\
\mathbb{E}_q[\log p(y|x, z)] &= \mathbb{E}_q[y \log\sigma(xz) + (1 - y)\log(1 - \sigma(xz))]
\end{aligned}
$$

Example: Bayesian Logistic Regression (2)

- Choose Gaussian variational approximation:

  $q_{\mathbf{v}}(z) = \mathcal{N}\left(z; \mu, \sigma^2\right)$ ⏩ $\boldsymbol{\nu} = \{\mu, \sigma^2\}$
- Objective function: ELBO $\mathcal{F}(\boldsymbol{\nu})$

$$\begin{aligned}\mathcal{F}(m, \sigma^2) &= \mathbb{E}_q[\log p(z) - \log q(z) + \log p(y|x,z)] \\ &= -\tfrac{1}{2}(\mu^2 + \sigma^2) + \tfrac{1}{2}\log\sigma^2 + \mathbb{E}_q[\log p(y|x,z)] + \mathrm{c} \\ \mathbb{E}_q[\log p(y|x,z)] &= \mathbb{E}_q[y\log\sigma(xz) + (1-y)\log(1-\sigma(xz))] \\ &= \mathbb{E}_q[yxz] - \mathbb{E}_q[y\log(1+\exp(xz))] \\ &\quad + \mathbb{E}_q\left[(1-y)\log\left(1 - \frac{\exp(xz)}{1+\exp(xz)}\right)\right]\end{aligned}$$

with

$$\sigma(xz) = \frac{\exp(xz)}{1+\exp(xz)}$$

Computing the Expected Log-Likelihood

$$
\begin{aligned}
\mathbb{E}_q[\log p(y|x,z)] &= \mathbb{E}_q[yxz] - \mathbb{E}_q[y\log(1+\exp(xz))] \\
&\quad + \mathbb{E}_q[(1-y)\log\left(1-\frac{\exp(xz)}{1+\exp(xz)}\right)]
\end{aligned}
$$

Computing the Expected Log-Likelihood

$$
\begin{aligned}
\mathbb{E}_q[\log p(y|x,z)] &= \mathbb{E}_q[yxz] - \mathbb{E}_q[y \log(1 + \exp(xz))] \\
&\quad + \mathbb{E}_q[(1-y)\log\left(1 - \frac{\exp(xz)}{1+\exp(xz)}\right)] \\
&= yx\mu - \mathbb{E}_q[y \log(1 + \exp(xz))] \\
&\quad + \mathbb{E}_q[(1-y)\log\left(\frac{1}{1+\exp(xz)}\right)]
\end{aligned}
$$

Computing the Expected Log-Likelihood

$$\begin{aligned}
\mathbb{E}_q[\log p(y|x,z)] &= \mathbb{E}_q[yxz] - \mathbb{E}_q[y\log(1+\exp(xz))] \\
&\quad + \mathbb{E}_q[(1-y)\log\left(1 - \frac{\exp(xz)}{1+\exp(xz)}\right)] \\
&= yx\mu - \mathbb{E}_q[y\log(1+\exp(xz))] \\
&\quad + \mathbb{E}_q[(1-y)\log\left(\frac{1}{1+\exp(xz)}\right)] \\
&= yx\mu - \mathbb{E}_q[y\log(1+\exp(xz))] \\
&\quad - \mathbb{E}_q[\log(1+\exp(xz))] + \mathbb{E}_q[y\log(1+\exp(xz))]
\end{aligned}$$

Computing the Expected Log-Likelihood

$$\begin{aligned}
\mathbb{E}_q[\log p(y|x,z)] &= \mathbb{E}_q[yxz] - \mathbb{E}_q[y\log(1+\exp(xz))] \\
&\quad + \mathbb{E}_q[(1-y)\log\left(1 - \frac{\exp(xz)}{1+\exp(xz)}\right)] \\
&= yx\mu - \mathbb{E}_q[y\log(1+\exp(xz))] \\
&\quad + \mathbb{E}_q[(1-y)\log\left(\frac{1}{1+\exp(xz)}\right)] \\
&= yx\mu - \mathbb{E}_q[y\log(1+\exp(xz))] \\
&\quad - \mathbb{E}_q[\log(1+\exp(xz))] + \mathbb{E}_q[y\log(1+\exp(xz))] \\
&= yx\mu - \mathbb{E}_q[\log(1+\exp(xz))]
\end{aligned}$$

Example: Bayesian Logistic Regression (ctd.)

- Choose Gaussian variational approximation:
  $q_{\mathbf{v}}(z) = \mathcal{N}\left(z; \mu, \sigma^2\right)$ ⏩ $\boldsymbol{\nu} = \{\mu, \sigma^2\}$
- Objective function: ELBO $\mathcal{F}(\boldsymbol{\nu})$

$$
\begin{aligned}
\mathcal{F}(\mu, \sigma^2) &= \mathbb{E}_q[\log p(z) + \log p(y|x, z) - \log q(z)] \\
&= -\tfrac{1}{2}(\mu^2 + \sigma^2) + \tfrac{1}{2}\log\sigma^2 + \mathbb{E}_q[\log p(y|x, z)] + \mathrm{c} \\
&= -\tfrac{1}{2}(\mu^2 + \sigma^2) + \tfrac{1}{2}\log\sigma^2 + yx\mu - \mathbb{E}_q[\log(1 + \exp(xz))]
\end{aligned}
$$

Example: Bayesian Logistic Regression (ctd.)

- Choose Gaussian variational approximation:
  $q_{\mathbf{v}}(z) = \mathcal{N}\left(z; \mu, \sigma^2\right)$ ⏩ $\boldsymbol{\nu} = \{\mu, \sigma^2\}$
- Objective function: ELBO $\mathcal{F}(\boldsymbol{\nu})$

$$
\begin{aligned}
\mathcal{F}(\mu, \sigma^2) &= \mathbb{E}_q[\log p(z) + \log p(y|x, z) - \log q(z)] \\
&= -\tfrac{1}{2}(\mu^2 + \sigma^2) + \tfrac{1}{2}\log \sigma^2 + \mathbb{E}_q[\log p(y|x, z)] + \mathrm{c} \\
&= -\tfrac{1}{2}(\mu^2 + \sigma^2) + \tfrac{1}{2}\log \sigma^2 + yx\mu - \mathbb{E}_q[\log(1 + \exp(xz))]
\end{aligned}
$$

- **Expectation cannot be computed in closed form**
- We want to optimise w.r.t. variational parameters μ, σ^2.

Example: Bayesian Logistic Regression (ctd.)

- Choose Gaussian variational approximation:
  $q_{\mathbf{v}}(z) = \mathcal{N}\left(z; \mu, \sigma^2\right)$ ⏩ $\boldsymbol{\nu} = \{\mu, \sigma^2\}$
- Objective function: ELBO $\mathcal{F}(\boldsymbol{\nu})$

$$\begin{aligned}\mathcal{F}(\mu, \sigma^2) &= \mathbb{E}_q[\log p(z) + \log p(y|x,z) - \log q(z)] \\ &= -\tfrac{1}{2}(\mu^2 + \sigma^2) + \tfrac{1}{2}\log\sigma^2 + \mathbb{E}_q[\log p(y|x,z)] + \mathrm{c} \\ &= -\tfrac{1}{2}(\mu^2 + \sigma^2) + \tfrac{1}{2}\log\sigma^2 + yx\mu - \mathbb{E}_q[\log(1+\exp(xz))]\end{aligned}$$

- **Expectation cannot be computed in closed form**
- We want to optimise w.r.t. variational parameters μ, σ^2.
- How can we optimise quantities that we cannot compute in closed-form?

Non-Conjugate Models

- Nonlinear time series models
- Deep latent Gaussian models
- Attention models (e.g., DRAW)
- Generalized linear models (e.g., logistic regression)
- Bayesian neural networks
- ...

Non-Conjugate Models

- Nonlinear time series models
- Deep latent Gaussian models
- Attention models (e.g., DRAW)
- Generalized linear models (e.g., logistic regression)
- Bayesian neural networks
- ...

There are many interesting non-conjugate models

⏩ Look for a solution that is not model specific

⏩ **Black-Box Variational Inference**

Limitation 2: Large datasets

Example: Bayesian Logistic Regression

Usual formulation:

$$p(y_n \mid \mathbf{x}_n, \mathbf{z}) = \mathrm{Ber}(\sigma(\boldsymbol{\theta}^\top \mathbf{x}_n)) \tag{5}$$

$$p(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}; 0, \mathbf{I}) \tag{6}$$

ELBO:

$$\begin{aligned}\mathcal{L} &= \mathbb{E}_{q(\boldsymbol{\theta})}\left[\log \prod_{n=1}^{N} p(y_n \mid \mathbf{x}_n, \boldsymbol{\theta})\right] - \mathrm{KL}[q(\boldsymbol{\theta})||p(\boldsymbol{\theta})] \\ &= \sum_{n=1}^{N} \mathbb{E}_{q(\boldsymbol{\theta})}[\log p(y_n \mid \mathbf{x}_n, \boldsymbol{\theta})] - \mathrm{KL}[q(\boldsymbol{\theta})||p(\boldsymbol{\theta})] \end{aligned} \tag{7}$$

Big data

$$\mathcal{L} = \sum_{n=1}^{N} \mathbb{E}_{q(\boldsymbol{\theta})}[\log p(y_n \mid \mathbf{x}_n, \boldsymbol{\theta})] - \mathrm{KL}[q(\boldsymbol{\theta}) || p(\boldsymbol{\theta})] \tag{8}$$

In “big data” applications, N may be millions or billions.

⏩ Summing over all datapoints at **each** optimisation iteration for $q(\boldsymbol{\theta})$ is **too slow**.

Stochastic Optimisation

Stochastic Optimisation

$$\mathcal{L} = \sum_{n=1}^{N} \mathbb{E}_{q(\boldsymbol{\theta})}[\log p(y_n \mid \mathbf{x}_n, \boldsymbol{\theta})] - \mathrm{KL}[q(\boldsymbol{\theta})||p(\boldsymbol{\theta})] \tag{9}$$

We can trivially find an **unbiased estimator** of the ELBO and its gradient by subsampling the data points! (solves problem 2)

Stochastic Optimisation

$$\mathcal{L} = \sum_{n=1}^{N} \mathbb{E}_{q(\boldsymbol{\theta})}[\log p(y_n \,|\, \mathbf{x}_n, \boldsymbol{\theta})] - \mathrm{KL}[q(\boldsymbol{\theta})||p(\boldsymbol{\theta})] \tag{9}$$

We can trivially find an **unbiased estimator** of the ELBO and its gradient by subsampling the data points! (solves problem 2)

$$\hat{\mathcal{L}} = \frac{N}{M} \sum_{n \in \mathcal{M}} \mathbb{E}_{q_{\mathbf{v}}(\boldsymbol{\theta})}[\log p(y_n \,|\, \mathbf{x}_n, \boldsymbol{\theta})] - \mathrm{KL}[q_{\mathbf{v}}(\boldsymbol{\theta})||p(\boldsymbol{\theta})] \tag{10}$$

$$\frac{\partial \hat{\mathcal{L}}}{\partial \mathbf{v}} = \frac{N}{M} \sum_{n \in \mathcal{M}} \frac{\partial}{\partial \mathbf{v}} \mathbb{E}_{q_{\mathbf{v}}(\boldsymbol{\theta})}[\log p(y_n \,|\, \mathbf{x}_n, \boldsymbol{\theta})] - \frac{\partial}{\partial \mathbf{v}} \mathrm{KL}[q_{\mathbf{v}}(\boldsymbol{\theta})||p(\boldsymbol{\theta})] \tag{11}$$

Can we still optimise with estimated gradients?

Stochastic Optimisation

$$\mathcal{L} = \sum_{n=1}^{N} \mathbb{E}_{q(\boldsymbol{\theta})}[\log p(y_n \mid \mathbf{x}_n, \boldsymbol{\theta})] - \mathrm{KL}[q(\boldsymbol{\theta})||p(\boldsymbol{\theta})] \tag{9}$$

We can trivially find an **unbiased estimator** of the ELBO and its gradient by subsampling the data points! (solves problem 2)

$$\hat{\mathcal{L}} = \frac{N}{M} \sum_{n \in \mathcal{M}} \mathbb{E}_{q_{\mathbf{v}}(\boldsymbol{\theta})}[\log p(y_n \mid \mathbf{x}_n, \boldsymbol{\theta})] - \mathrm{KL}[q_{\mathbf{v}}(\boldsymbol{\theta})||p(\boldsymbol{\theta})] \tag{10}$$

$$\frac{\partial \hat{\mathcal{L}}}{\partial \mathbf{v}} = \frac{N}{M} \sum_{n \in \mathcal{M}} \frac{\partial}{\partial \mathbf{v}} \mathbb{E}_{q_{\mathbf{v}}(\boldsymbol{\theta})}[\log p(y_n \mid \mathbf{x}_n, \boldsymbol{\theta})] - \frac{\partial}{\partial \mathbf{v}} \mathrm{KL}[q_{\mathbf{v}}(\boldsymbol{\theta})||p(\boldsymbol{\theta})] \tag{11}$$

Can we still optimise with estimated gradients? (Yes)

Stochastic Gradient Descent (MML / Comp Opt)

Goal: $\mathbf{v}^* = \operatorname{argmax}_{\mathbf{v}} \mathcal{L}(\mathbf{v})$

Stochastic Gradient Descent (MML / Comp Opt)

Goal: $\mathbf{v}^* = \operatorname{argmax}_{\mathbf{v}} \mathcal{L}(\mathbf{v})$
Normal gradient descent:

$$\mathbf{v}_t = \mathbf{v}_{t-1} + \rho_t \nabla_{\mathbf{v}} \mathcal{L}(\mathbf{v}_{t-1}) \tag{12}$$

$$\mathbf{v}_t \to \mathbf{v}^* \text{ as } t \to \infty \tag{13}$$

Stochastic Gradient Descent (MML / Comp Opt)

Goal: $\mathbf{v}^* = \operatorname{argmax}_{\mathbf{v}} \mathcal{L}(\mathbf{v})$
Normal gradient descent:

$$\mathbf{v}_t = \mathbf{v}_{t-1} + \rho_t \nabla_{\mathbf{v}} \mathcal{L}(\mathbf{v}_{t-1}) \tag{12}$$

$$\mathbf{v}_t \to \mathbf{v}^* \text{ as } t \to \infty \tag{13}$$

Stochastic gradient descent (Robbins & Monro, 1951):

$$\text{if } \mathbb{E}[\hat{g}_t] = \nabla_{\mathbf{v}} \mathcal{L}(\mathbf{v}_t) \tag{14}$$

$$\mathbf{v}_t = \mathbf{v}_{t-1} + \rho_t \hat{g}_t \tag{15}$$

$$\mathbf{v}_t \to \mathbf{v}^* \text{ as } t \to \infty \qquad \text{if } \sum_{t=1}^{\infty} \rho_t = \infty \text{ and } \sum_{t=1}^{\infty} \rho_t^2 < \infty \tag{16}$$

$$\text{e.g. } \rho_t = 1/t \tag{17}$$

Stochastic Gradient Descent (MML / Comp Opt)

Goal: $\mathbf{v}^* = \operatorname{argmax}_{\mathbf{v}} \mathcal{L}(\mathbf{v})$
Normal gradient descent:

$$\mathbf{v}_t = \mathbf{v}_{t-1} + \rho_t \nabla_{\mathbf{v}} \mathcal{L}(\mathbf{v}_{t-1}) \tag{12}$$

$$\mathbf{v}_t \to \mathbf{v}^* \text{ as } t \to \infty \tag{13}$$

Stochastic gradient descent (Robbins & Monro, 1951):

$$\text{if } \mathbb{E}[\hat{g}_t] = \nabla_{\mathbf{v}} \mathcal{L}(\mathbf{v}_t) \tag{14}$$

$$\mathbf{v}_t = \mathbf{v}_{t-1} + \rho_t \hat{g}_t \tag{15}$$

$$\mathbf{v}_t \to \mathbf{v}^* \text{ as } t \to \infty \qquad \text{if } \sum_{t=1}^{\infty} \rho_t = \infty \text{ and } \sum_{t=1}^{\infty} \rho_t^2 < \infty \tag{16}$$

$$\text{e.g. } \rho_t = 1/t \tag{17}$$

Having a small $\mathbb{V}[\hat{g}_t]$ is crucial to ensure fast convergence.

Stochastic Optimisation

- Stochastic optimisation solves problem 2.
- Still stuck with problem 1: Intractable integrals in VI.

Stochastic Optimisation

- Stochastic optimisation solves problem 2.
- Still stuck with problem 1: Intractable integrals in VI.

Since we're using stochastic gradient estimates anyway...

Stochastic Optimisation

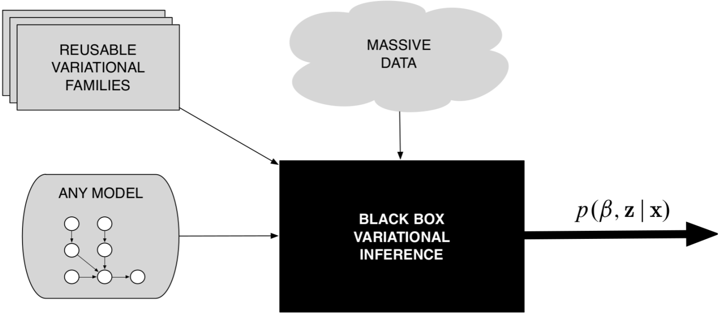
- Stochastic optimisation solves problem 2.
- Still stuck with problem 1: Intractable integrals in VI.

Since we're using stochastic gradient estimates anyway...

Can we not also find Monte Carlo approximations to the gradients of intractable integrals?

Black-Box Variational Inference (BBVI)

Black-Box Variational Inference


REUSABLE VARIATIONAL FAMILIES
MASSIVE DATA
ANY MODEL
BLACK BOX VARIATIONAL INFERENCE
$p(\beta, \mathbf{z} \mid \mathbf{x})$

From Blei et al.'s NIPS-2016 tutorial

- Any model (limitation 1)
- Massive data (limitation 2)
- Some general assumptions on the approximating family

Black-Box Variational Inference

Problem 1: Intractable integral of the expected log-likelihood term

$$\mathbb{E}_{q(\mathbf{z})}[\log p(\mathbf{x} \mid \mathbf{z})] \,. \tag{18}$$

For stochastic optimisation we need an estimator of its **gradient** $\hat{g}_t$, such that

$$\mathbb{E}[\hat{g}_t] = \nabla_{\mathbf{v}} \mathcal{L}(\mathbf{v}) \tag{19}$$

Black-Box Variational Inference

Problem 1: Intractable integral of the expected log-likelihood term

$$\mathbb{E}_{q(\mathbf{z})}[\log p(\mathbf{x} \mid \mathbf{z})] . \tag{18}$$

For stochastic optimisation we need an estimator of its **gradient** $\hat{g}_t$, such that

$$\mathbb{E}[\hat{g}_t] = \nabla_{\mathbf{v}} \mathcal{L}(\mathbf{v}) \tag{19}$$

Can we find such unbiased estimates?

Black-Box Variational Inference

Problem 1: Intractable integral of the expected log-likelihood term

$$\mathbb{E}_{q(\mathbf{z})}[\log p(\mathbf{x} \mid \mathbf{z})] \,. \tag{18}$$

For stochastic optimisation we need an estimator of its **gradient** $\hat{g}_t$, such that

$$\mathbb{E}[\hat{g}_t] = \nabla_{\mathbf{v}} \mathcal{L}(\mathbf{v}) \tag{19}$$

Can we find such unbiased estimates?

- Score function estimator
- Reparameterisation estimator

Problem statement

We have intractable terms that can be written as:

$$\mathbb{E}_{q_{\mathbf{v}}(\mathbf{z})}[h(\mathbf{z}, \mathbf{v})] \tag{20}$$

Goal: Find estimator $\hat{g}$ with property

$$\mathbb{E}[\hat{g}] = \nabla_{\mathbf{v}} \mathbb{E}_{q_{\mathbf{v}}(\mathbf{z})}[h(\mathbf{z}, \mathbf{v})] \tag{21}$$

Problem statement

We have intractable terms that can be written as:

$$\mathbb{E}_{q_{\mathbf{v}}(\mathbf{z})}[h(\mathbf{z}, \mathbf{v})] \tag{20}$$

Goal: Find estimator $\hat{g}$ with property

$$\mathbb{E}[\hat{g}] = \nabla_{\mathbf{v}} \mathbb{E}_{q_{\mathbf{v}}(\mathbf{z})}[h(\mathbf{z}, \mathbf{v})] \tag{21}$$

Remember:

- It's easy to find a MC estimate of the objective.
- But we need an MC estimate of the gradients!

Approach

$$g(\mathbf{v}) = \nabla_{\mathbf{v}} \mathbb{E}_q[h(\mathbf{z}, \mathbf{v})] \tag{22}$$

- Switch order to integration first, then differentiation (Monte Carlo estimates need expectations, and expectations are integrals)

Approach

$$g(\mathbf{v}) = \nabla_{\mathbf{v}} \mathbb{E}_q[h(\mathbf{z}, \mathbf{v})] \tag{22}$$

- Switch order to integration first, then differentiation (Monte Carlo estimates need expectations, and expectations are integrals)
- Write integration as expectation again

Approach

$$g(\mathbf{v}) = \nabla_{\mathbf{v}} \mathbb{E}_q[h(\mathbf{z}, \mathbf{v})] \tag{22}$$

- Switch order to integration first, then differentiation
  (Monte Carlo estimates need expectations, and expectations are integrals)
- Write integration as expectation again
- Approximate the expectation after having taken the gradient
  ⏩ Monte Carlo estimator (ideally with low variance)

Approach

$$g(\mathbf{v}) = \nabla_{\mathbf{v}} \mathbb{E}_q[h(\mathbf{z}, \mathbf{v})] \tag{22}$$

- Switch order to integration first, then differentiation
  (Monte Carlo estimates need expectations, and expectations are integrals)
- Write integration as expectation again
- Approximate the expectation after having taken the gradient
  ⏩ Monte Carlo estimator (ideally with low variance)
- Stochastic optimization

Approach

$$g(\mathbf{v}) = \nabla_{\mathbf{v}} \mathbb{E}_q[h(\mathbf{z}, \mathbf{v})] \tag{22}$$

- Switch order to integration first, then differentiation
  (Monte Carlo estimates need expectations, and expectations are integrals)
- Write integration as expectation again
- Approximate the expectation after having taken the gradient
  ⏩ Monte Carlo estimator (ideally with low variance)
- Stochastic optimization

⏩ Require: general way to compute gradients of expectations

Log-Derivative Trick

Log-Derivative Trick

$$\nabla_{\nu} \log q_{\mathbf{v}}(z) =$$

Log-Derivative Trick

Log-Derivative Trick

$$\nabla_{\nu} \log q_{\mathbf{v}}(z) = \frac{\nabla_{\nu} q_{\mathbf{v}}(z)}{q_{\mathbf{v}}(z)}$$

$$\iff \nabla_{\nu} q_{\mathbf{v}}(z) = q_{\mathbf{v}}(z) \nabla_{\nu} \log q_{\mathbf{v}}(z)$$

Log-Derivative Trick

Log-Derivative Trick

$$\nabla_{\nu} \log q_{\mathbf{v}}(z) = \frac{\nabla_{\nu} q_{\mathbf{v}}(z)}{q_{\mathbf{v}}(z)}$$

$$\iff \nabla_{\nu} q_{\mathbf{v}}(z) = q_{\mathbf{v}}(z) \nabla_{\nu} \log q_{\mathbf{v}}(z)$$

- Therefore:

$$\begin{aligned} \int \nabla_{\nu} q_{\mathbf{v}}(z) f(z) dz &= \int q_{\mathbf{v}}(z) \nabla_{\nu} \log q_{\mathbf{v}}(z) f(z) dz \\ &= \mathbb{E}_q[\nabla_{\nu} \log q_{\mathbf{v}}(z) f(z)] \end{aligned}$$

Log-Derivative Trick

Log-Derivative Trick

$$\nabla_{\nu} \log q_{\mathbf{v}}(z) = \frac{\nabla_{\nu} q_{\mathbf{v}}(z)}{q_{\mathbf{v}}(z)}$$

$$\iff \nabla_{\nu} q_{\mathbf{v}}(z) = q_{\mathbf{v}}(z) \nabla_{\nu} \log q_{\mathbf{v}}(z)$$

- Therefore:

$$\begin{aligned}\int \nabla_{\nu} q_{\mathbf{v}}(z) f(z) dz &= \int q_{\mathbf{v}}(z) \nabla_{\nu} \log q_{\mathbf{v}}(z) f(z) dz \\ &= \mathbb{E}_q[\nabla_{\nu} \log q_{\mathbf{v}}(z) f(z)]\end{aligned}$$

- If we can sample from q, this expectation can be evaluated easily (Monte Carlo estimation)

Gradients of Expectations: Approach 1

$$\text{ELBO} = \mathcal{F}(\boldsymbol{\nu}) = \mathbb{E}_q[h(\boldsymbol{z}, \boldsymbol{\nu})]\,, \quad h(\boldsymbol{z}, \boldsymbol{\nu}) = \log p(\boldsymbol{x}, \boldsymbol{z}) - \log q(\boldsymbol{z}|\boldsymbol{\nu})$$

- Need gradient of ELBO w.r.t. variational parameters $\boldsymbol{\nu}$

Gradients of Expectations: Approach 1

$$\text{ELBO} = \mathcal{F}(\boldsymbol{\nu}) = \mathbb{E}_q[h(\boldsymbol{z},\boldsymbol{\nu})]\,, \quad h(\boldsymbol{z},\boldsymbol{\nu}) = \log p(\boldsymbol{x},\boldsymbol{z}) - \log q(\boldsymbol{z}|\boldsymbol{\nu})$$

- Need gradient of ELBO w.r.t. variational parameters $\boldsymbol{\nu}$

$$\nabla_{\boldsymbol{\nu}}\mathcal{F} = \nabla_{\boldsymbol{\nu}}\mathbb{E}_q[h(\boldsymbol{z},\boldsymbol{\nu})]$$

$$= \mathbb{E}_q[\nabla_{\boldsymbol{\nu}} \log q(\boldsymbol{z}|\boldsymbol{\nu})h(\boldsymbol{z},\boldsymbol{\nu}) + \nabla_{\boldsymbol{\nu}}h(\boldsymbol{z},\boldsymbol{\nu})]$$

Gradients of Expectations: Approach 1

$$\text{ELBO} = \mathcal{F}(\boldsymbol{\nu}) = \mathbb{E}_q[h(\boldsymbol{z}, \boldsymbol{\nu})]\,, \quad h(\boldsymbol{z}, \boldsymbol{\nu}) = \log p(\boldsymbol{x}, \boldsymbol{z}) - \log q(\boldsymbol{z}|\boldsymbol{\nu})$$

- Need gradient of ELBO w.r.t. variational parameters $\boldsymbol{\nu}$

$$\nabla_{\boldsymbol{\nu}} \mathcal{F} = \nabla_{\boldsymbol{\nu}} \mathbb{E}_q[h(\boldsymbol{z}, \boldsymbol{\nu})] = \nabla_{\boldsymbol{\nu}} \int h(\boldsymbol{z}, \boldsymbol{\nu}) q_{\mathbf{v}}(\boldsymbol{z}) d\boldsymbol{z}$$

$$= \mathbb{E}_q[\nabla_{\boldsymbol{\nu}} \log q(\boldsymbol{z}|\boldsymbol{\nu}) h(\boldsymbol{z}, \boldsymbol{\nu}) + \nabla_{\boldsymbol{\nu}} h(\boldsymbol{z}, \boldsymbol{\nu})]$$

Gradients of Expectations: Approach 1

$$\text{ELBO} = \mathcal{F}(\boldsymbol{\nu}) = \mathbb{E}_q[h(\boldsymbol{z}, \boldsymbol{\nu})], \quad h(\boldsymbol{z}, \boldsymbol{\nu}) = \log p(\boldsymbol{x}, \boldsymbol{z}) - \log q(\boldsymbol{z}|\boldsymbol{\nu})$$

- Need gradient of ELBO w.r.t. variational parameters $\boldsymbol{\nu}$

$$\nabla_{\boldsymbol{\nu}} \mathcal{F} = \nabla_{\boldsymbol{\nu}} \mathbb{E}_q[h(\boldsymbol{z}, \boldsymbol{\nu})] = \nabla_{\boldsymbol{\nu}} \int h(\boldsymbol{z}, \boldsymbol{\nu}) q_{\mathbf{v}}(\boldsymbol{z}) d\boldsymbol{z}$$

$$= \int \big(\nabla_{\boldsymbol{\nu}} h(\boldsymbol{\nu}, \boldsymbol{z})\big) q_{\mathbf{v}}(\boldsymbol{z}) + h(\boldsymbol{\nu}, \boldsymbol{z}) \nabla_{\boldsymbol{\nu}} q_{\mathbf{v}}(\boldsymbol{z}) d\boldsymbol{z}$$

product rule

$$= \mathbb{E}_q[\nabla_{\boldsymbol{\nu}} \log q(\boldsymbol{z}|\boldsymbol{\nu}) h(\boldsymbol{z}, \boldsymbol{\nu}) + \nabla_{\boldsymbol{\nu}} h(\boldsymbol{z}, \boldsymbol{\nu})]$$

Gradients of Expectations: Approach 1

$$\text{ELBO} = \mathcal{F}(\boldsymbol{\nu}) = \mathbb{E}_q[h(\boldsymbol{z}, \boldsymbol{\nu})]\,, \quad h(\boldsymbol{z}, \boldsymbol{\nu}) = \log p(\boldsymbol{x}, \boldsymbol{z}) - \log q(\boldsymbol{z}|\boldsymbol{\nu})$$

- Need gradient of ELBO w.r.t. variational parameters $\boldsymbol{\nu}$

$$\begin{aligned}
\nabla_{\boldsymbol{\nu}}\mathcal{F} &= \nabla_{\boldsymbol{\nu}}\mathbb{E}_q[h(\boldsymbol{z}, \boldsymbol{\nu})] = \nabla_{\boldsymbol{\nu}} \int h(\boldsymbol{z}, \boldsymbol{\nu}) q_{\mathbf{v}}(\boldsymbol{z}) d\boldsymbol{z} \\
&= \int \big(\nabla_{\boldsymbol{\nu}} h(\boldsymbol{\nu}, \boldsymbol{z})\big) q_{\mathbf{v}}(\boldsymbol{z}) + h(\boldsymbol{\nu}, \boldsymbol{z}) \nabla_{\boldsymbol{\nu}} q_{\mathbf{v}}(\boldsymbol{z}) d\boldsymbol{z} && \boxed{\text{product rule}} \\
&= \int q_{\mathbf{v}}(\boldsymbol{z}) \nabla_{\boldsymbol{\nu}} h(\boldsymbol{z}, \boldsymbol{\nu}) + q_{\mathbf{v}}(\boldsymbol{z}) \nabla_{\boldsymbol{\nu}} \log q_{\mathbf{v}}(\boldsymbol{z}) h(\boldsymbol{z}, \boldsymbol{\nu}) d\boldsymbol{z} && \boxed{\text{log-deriv. trick}} \\
&= \mathbb{E}_q[\nabla_{\boldsymbol{\nu}} \log q(\boldsymbol{z}|\boldsymbol{\nu}) h(\boldsymbol{z}, \boldsymbol{\nu}) + \nabla_{\boldsymbol{\nu}} h(\boldsymbol{z}, \boldsymbol{\nu})]
\end{aligned}$$

Gradients of Expectations: Approach 1

$$\text{ELBO} = \mathcal{F}(\boldsymbol{\nu}) = \mathbb{E}_q[h(\boldsymbol{z}, \boldsymbol{\nu})]\,, \quad h(\boldsymbol{z}, \boldsymbol{\nu}) = \log p(\boldsymbol{x}, \boldsymbol{z}) - \log q(\boldsymbol{z}|\boldsymbol{\nu})$$

- Need gradient of ELBO w.r.t. variational parameters $\boldsymbol{\nu}$

$$\begin{aligned}
\nabla_{\boldsymbol{\nu}} \mathcal{F} = \nabla_{\boldsymbol{\nu}} \mathbb{E}_q[h(\boldsymbol{z}, \boldsymbol{\nu})] &= \nabla_{\boldsymbol{\nu}} \int h(\boldsymbol{z}, \boldsymbol{\nu}) q_{\mathbf{v}}(\boldsymbol{z}) d\boldsymbol{z} \\
&= \int \big(\nabla_{\boldsymbol{\nu}} h(\boldsymbol{\nu}, \boldsymbol{z})\big) q_{\mathbf{v}}(\boldsymbol{z}) + h(\boldsymbol{\nu}, \boldsymbol{z}) \nabla_{\boldsymbol{\nu}} q_{\mathbf{v}}(\boldsymbol{z}) d\boldsymbol{z} && \boxed{\text{product rule}} \\
&= \int q_{\mathbf{v}}(\boldsymbol{z}) \nabla_{\boldsymbol{\nu}} h(\boldsymbol{z}, \boldsymbol{\nu}) + q_{\mathbf{v}}(\boldsymbol{z}) \nabla_{\boldsymbol{\nu}} \log q_{\mathbf{v}}(\boldsymbol{z}) h(\boldsymbol{z}, \boldsymbol{\nu}) d\boldsymbol{z} && \boxed{\text{log-deriv. trick}} \\
&= \mathbb{E}_q[\nabla_{\boldsymbol{\nu}} \log q(\boldsymbol{z}|\boldsymbol{\nu}) h(\boldsymbol{z}, \boldsymbol{\nu}) + \nabla_{\boldsymbol{\nu}} h(\boldsymbol{z}, \boldsymbol{\nu})]
\end{aligned}$$

- We successfully swapped gradient and expectation
- q known
  - ⏩ Sample from q and use Monte Carlo estimation

Score Function

- Score function: Derivative of a log-likelihood with respect to the parameter vector $\boldsymbol{\nu}$:

Score Function

- Score function: Derivative of a log-likelihood with respect to the parameter vector $\boldsymbol{\nu}$:

Score Function

$$\text{score} = \nabla_{\boldsymbol{\nu}} \log q_{\mathbf{v}}(z) = \frac{1}{q_{\mathbf{v}}(z)} \nabla_{\boldsymbol{\nu}} q_{\mathbf{v}}(z)$$

Score Function

- Score function: Derivative of a log-likelihood with respect to the parameter vector $\boldsymbol{\nu}$:

Score Function

$$\text{score} = \nabla_{\boldsymbol{\nu}} \log q_{\mathbf{v}}(z) = \frac{1}{q_{\mathbf{v}}(z)} \nabla_{\boldsymbol{\nu}} q_{\mathbf{v}}(z)$$

- Measures the sensitivity of the log-likelihood w.r.t. $\boldsymbol{\nu}$

Score Function (2)

$$\text{score} = \nabla_{\boldsymbol{\nu}} \log q_{\mathbf{v}}(z) = \frac{1}{q_{\mathbf{v}}(z)} \nabla_{\boldsymbol{\nu}} q_{\mathbf{v}}(z)$$

- Important property:

$$\mathbb{E}_{q_{\mathbf{v}}(z)}[\text{score}] =$$

Score Function (2)

$$\text{score} = \nabla_{\nu} \log q_{\mathbf{v}}(z) = \frac{1}{q_{\mathbf{v}}(z)} \nabla_{\nu} q_{\mathbf{v}}(z)$$

- Important property:

$$\begin{aligned}
\mathbb{E}_{q_{\mathbf{v}}(z)}[\text{score}] &= \mathbb{E}_{q_{\mathbf{v}}(z)}\left[\frac{1}{q_{\mathbf{v}}(z)} \nabla_{\nu} q_{\mathbf{v}}(z)\right] \\
&= \int \frac{1}{q_{\mathbf{v}}(z)} q_{\mathbf{v}}(z) \nabla_{\nu} q_{\mathbf{v}}(z) dz \\
&= \int \nabla_{\nu} q_{\mathbf{v}}(z) dz = \nabla_{\nu} \int q_{\mathbf{v}}(z) dz = \nabla_{\nu} 1 = 0
\end{aligned}$$

▶▶ Mean of the score function is 0

Score Function Gradient Estimator

$$\text{ELBO} = \mathbb{E}_q[h(\boldsymbol{z}, \boldsymbol{\nu})] = \mathbb{E}_q[\log p(\boldsymbol{x}, \boldsymbol{z}) - \log q_{\mathbf{v}}(\boldsymbol{z})]$$

- Gradient of ELBO:

$$\nabla_{\boldsymbol{\nu}} \text{ELBO} =$$

Score Function Gradient Estimator

$$\text{ELBO} = \mathbb{E}_q[h(\boldsymbol{z}, \boldsymbol{\nu})] = \mathbb{E}_q[\log p(\boldsymbol{x}, \boldsymbol{z}) - \log q_{\mathbf{v}}(\boldsymbol{z})]$$

- Gradient of ELBO:

$$\nabla_{\boldsymbol{\nu}}\text{ELBO} = \mathbb{E}_q[\nabla_{\boldsymbol{\nu}} \log q_{\mathbf{v}}(\boldsymbol{z}) h(\boldsymbol{z}, \boldsymbol{\nu})] + \mathbb{E}_q[\nabla_{\boldsymbol{\nu}} h(\boldsymbol{z}, \boldsymbol{\nu})]$$

Score Function Gradient Estimator

$$\text{ELBO} = \mathbb{E}_q[h(\boldsymbol{z}, \boldsymbol{v})] = \mathbb{E}_q[\log p(\boldsymbol{x}, \boldsymbol{z}) - \log q_{\mathbf{v}}(\boldsymbol{z})]$$

- Gradient of ELBO:

$$\begin{aligned}\nabla_{\boldsymbol{v}}\text{ELBO} &= \mathbb{E}_q[\nabla_{\boldsymbol{v}} \log q_{\mathbf{v}}(\boldsymbol{z}) h(\boldsymbol{z}, \boldsymbol{v})] + \mathbb{E}_q[\nabla_{\boldsymbol{v}} h(\boldsymbol{z}, \boldsymbol{v})] \\ &= \mathbb{E}_q[\nabla_{\boldsymbol{v}} \log q_{\mathbf{v}}(\boldsymbol{z}) h(\boldsymbol{z}, \boldsymbol{v})] \\ &\quad + \mathbb{E}_q[\underbrace{\nabla_{\boldsymbol{v}} \log p(\boldsymbol{x}, \boldsymbol{z})}_{=0} - \underbrace{\nabla_{\boldsymbol{v}} \log q_{\mathbf{v}}(\boldsymbol{z})}_{\text{score}}]\end{aligned}$$

Score Function Gradient Estimator

$$\text{ELBO} = \mathbb{E}_q[h(\boldsymbol{z}, \boldsymbol{\nu})] = \mathbb{E}_q[\log p(\boldsymbol{x}, \boldsymbol{z}) - \log q_{\mathbf{v}}(\boldsymbol{z})]$$

- Gradient of ELBO:

$$\begin{aligned}\nabla_{\boldsymbol{\nu}}\text{ELBO} &= \mathbb{E}_q[\nabla_{\boldsymbol{\nu}} \log q_{\mathbf{v}}(\boldsymbol{z}) h(\boldsymbol{z}, \boldsymbol{\nu})] + \mathbb{E}_q[\nabla_{\boldsymbol{\nu}} h(\boldsymbol{z}, \boldsymbol{\nu})] \\ &= \mathbb{E}_q[\nabla_{\boldsymbol{\nu}} \log q_{\mathbf{v}}(\boldsymbol{z}) h(\boldsymbol{z}, \boldsymbol{\nu})] \\ &\quad + \mathbb{E}_q[\underbrace{\nabla_{\boldsymbol{\nu}} \log p(\boldsymbol{x}, \boldsymbol{z})}_{=0} - \underbrace{\nabla_{\boldsymbol{\nu}} \log q_{\mathbf{v}}(\boldsymbol{z})}_{\text{score}}]\end{aligned}$$

- Exploit that the mean of the score function is 0. Then:

$$\begin{aligned}\nabla_{\boldsymbol{\nu}}\text{ELBO} &= \mathbb{E}_q[\nabla_{\boldsymbol{\nu}} \log q_{\mathbf{v}}(\boldsymbol{z}) h(\boldsymbol{z}, \boldsymbol{\nu})] \\ &= \mathbb{E}_q[\nabla_{\boldsymbol{\nu}} \log q_{\mathbf{v}}(\boldsymbol{z})(\log p(\boldsymbol{x}, \boldsymbol{z}) - \log q_{\mathbf{v}}(\boldsymbol{z}))]\end{aligned}$$

Score Function Gradient Estimator

$$\text{ELBO} = \mathbb{E}_q[h(\boldsymbol{z}, \boldsymbol{\nu})] = \mathbb{E}_q[\log p(\boldsymbol{x}, \boldsymbol{z}) - \log q_{\mathbf{v}}(\boldsymbol{z})]$$

- Gradient of ELBO:

$$\begin{aligned}\nabla_{\boldsymbol{\nu}}\text{ELBO} &= \mathbb{E}_q[\nabla_{\boldsymbol{\nu}} \log q_{\mathbf{v}}(\boldsymbol{z}) h(\boldsymbol{z}, \boldsymbol{\nu})] + \mathbb{E}_q[\nabla_{\boldsymbol{\nu}} h(\boldsymbol{z}, \boldsymbol{\nu})] \\ &= \mathbb{E}_q[\nabla_{\boldsymbol{\nu}} \log q_{\mathbf{v}}(\boldsymbol{z}) h(\boldsymbol{z}, \boldsymbol{\nu})] \\ &\quad + \mathbb{E}_q[\underbrace{\nabla_{\boldsymbol{\nu}} \log p(\boldsymbol{x}, \boldsymbol{z})}_{=0} - \underbrace{\nabla_{\boldsymbol{\nu}} \log q_{\mathbf{v}}(\boldsymbol{z})}_{\text{score}}]\end{aligned}$$

- Exploit that the mean of the score function is 0. Then:

$$\begin{aligned}\nabla_{\boldsymbol{\nu}}\text{ELBO} &= \mathbb{E}_q[\nabla_{\boldsymbol{\nu}} \log q_{\mathbf{v}}(\boldsymbol{z}) h(\boldsymbol{z}, \boldsymbol{\nu})] \\ &= \mathbb{E}_q[\nabla_{\boldsymbol{\nu}} \log q_{\mathbf{v}}(\boldsymbol{z})(\log p(\boldsymbol{x}, \boldsymbol{z}) - \log q_{\mathbf{v}}(\boldsymbol{z}))]\end{aligned}$$

- Likelihood ratio gradient (Glynn, 1990)
- REINFORCE gradient (Williams, 1992)

Using Noisy Stochastic Gradients

- Gradient of the ELBO

$$\nabla_{\boldsymbol{\nu}} \text{ELBO} = \mathbb{E}_q[\nabla_{\boldsymbol{\nu}} \log q_{\mathbf{v}}(\boldsymbol{z})(\log p(\boldsymbol{x}, \boldsymbol{z}) - \log q_{\mathbf{v}}(\boldsymbol{z}))]$$

is an expectation

Using Noisy Stochastic Gradients

- Gradient of the ELBO

$$\nabla_{\nu}\text{ELBO} = \mathbb{E}_q[\nabla_{\boldsymbol{\nu}} \log q_{\mathbf{v}}(\boldsymbol{z})(\log p(\boldsymbol{x}, \boldsymbol{z}) - \log q_{\mathbf{v}}(\boldsymbol{z}))]$$

  is an expectation

- Require that $q_{\mathbf{v}}(\boldsymbol{z})$ is differentiable w.r.t. $\boldsymbol{\nu}$

Using Noisy Stochastic Gradients

- Gradient of the ELBO

$$\nabla_{\boldsymbol{\nu}} \text{ELBO} = \mathbb{E}_q[\nabla_{\boldsymbol{\nu}} \log q_{\mathbf{v}}(\boldsymbol{z})(\log p(\boldsymbol{x}, \boldsymbol{z}) - \log q_{\mathbf{v}}(\boldsymbol{z}))]$$

  is an expectation
- Require that $q_{\mathbf{v}}(\boldsymbol{z})$ is differentiable w.r.t. $\boldsymbol{\nu}$
- Get noisy unbiased gradients using Monte Carlo by sampling from q:

$$\frac{1}{S}\sum_{s=1}^{S} \nabla_{\boldsymbol{\nu}} \log q_{\mathbf{v}}(\boldsymbol{z}^{(s)})(\log p(\boldsymbol{x}, \boldsymbol{z}^{(s)}) - \log q_{\mathbf{v}}(\boldsymbol{z}^{(s)}))\,, \quad \boldsymbol{z}^{(s)} \sim q_{\mathbf{v}}(\boldsymbol{z})$$

Using Noisy Stochastic Gradients

- Gradient of the ELBO

$$\nabla_{\boldsymbol{\nu}} \mathrm{ELBO} = \mathbb{E}_q[\nabla_{\boldsymbol{\nu}} \log q_{\mathbf{v}}(\boldsymbol{z})(\log p(\boldsymbol{x}, \boldsymbol{z}) - \log q_{\mathbf{v}}(\boldsymbol{z}))]$$

  is an expectation
- Require that $q_{\mathbf{v}}(\boldsymbol{z})$ is differentiable w.r.t. $\boldsymbol{\nu}$
- Get noisy unbiased gradients using Monte Carlo by sampling from q:

$$\frac{1}{S}\sum_{s=1}^{S} \nabla_{\boldsymbol{\nu}} \log q_{\mathbf{v}}(\boldsymbol{z}^{(s)})(\log p(\boldsymbol{x}, \boldsymbol{z}^{(s)}) - \log q_{\mathbf{v}}(\boldsymbol{z}^{(s)}))\,, \quad \boldsymbol{z}^{(s)} \sim q_{\mathbf{v}}(\boldsymbol{z})$$

- Sampling from q is easy (we choose q)

Using Noisy Stochastic Gradients

- Gradient of the ELBO

$$\nabla_{\boldsymbol{\nu}} \text{ELBO} = \mathbb{E}_q[\nabla_{\boldsymbol{\nu}} \log q_{\mathbf{v}}(\boldsymbol{z})(\log p(\boldsymbol{x}, \boldsymbol{z}) - \log q_{\mathbf{v}}(\boldsymbol{z}))]$$

  is an expectation
- Require that $q_{\mathbf{v}}(\boldsymbol{z})$ is differentiable w.r.t. $\boldsymbol{\nu}$
- Get noisy unbiased gradients using Monte Carlo by sampling from q:

$$\frac{1}{S}\sum_{s=1}^{S} \nabla_{\boldsymbol{\nu}} \log q_{\mathbf{v}}(\boldsymbol{z}^{(s)})(\log p(\boldsymbol{x}, \boldsymbol{z}^{(s)}) - \log q_{\mathbf{v}}(\boldsymbol{z}^{(s)}))\,, \quad \boldsymbol{z}^{(s)} \sim q_{\mathbf{v}}(\boldsymbol{z})$$

- Sampling from q is easy (we choose q)
- Use this within SVI to converge to a local optimum

Summary: BBVI procedure

Black Box Variational Inference

1. Input: model $p(\boldsymbol{x}, \boldsymbol{z})$, variational approximation $q_{\mathbf{v}}(\boldsymbol{z})$
2. Repeat

   2.1 Draw S samples $\boldsymbol{z}^{(s)} \sim q_{\mathbf{v}}(\boldsymbol{z})$

   2.2 Update variational parameters

$$\boldsymbol{\nu}_{t+1} = \boldsymbol{\nu}_t + \rho_t \underbrace{\frac{1}{S} \sum_{s=1}^{S} \nabla_{\nu} \log q(\boldsymbol{z}^{(s)} | \boldsymbol{\nu}) (\log p(\boldsymbol{x}, \boldsymbol{z}^{(s)}) - \log q(\boldsymbol{z}^{(s)} | \boldsymbol{\nu}))}_{\text{MC estimate of the score-function gradient of the ELBO}}$$

   2.3 $t = t + 1$

Requirements for Inference

Similar to MCMC in that it makes **few** requirements

- Computing the noisy gradient of the ELBO requires:
  - Sampling from q. We choose q so that this is possible.
  - Evaluate the score function $\nabla_{\boldsymbol{\nu}} \log q_{\mathbf{v}}(\boldsymbol{z})$
  - Evaluate $\log q_{\mathbf{v}}(\boldsymbol{z})$ and $\log p(\boldsymbol{x}, \boldsymbol{z}) = \log p(\boldsymbol{z}) + \log p(\boldsymbol{x}|\boldsymbol{z})$

⏩ No model-specific computations for optimization

(computations are only specific to the choice of the variational approximation)

Issue: Variance of the Gradients

- Stochastic optimization ⏩ **Gradients are noisy (high variance)**
- The noisier the gradients, the slower the convergence
- Possible solutions:
  - Control variates (with the score function as control variate)
  - Rao-Blackwellization
  - Importance sampling

Issues with score function estimator

We can simplify the gradient estimator further:

- Score-function gradient estimator only requires general assumptions
- Noisy gradients are a problem
- Address this issue by making some additional assumptions (not too strict)

  ▸▸ Pathwise gradient estimators

Approach

$$g(\mathbf{v}) = \nabla_{\mathbf{v}} \mathbb{E}_q[h(\mathbf{z}, \mathbf{v})] \tag{23}$$

- Switch order to integration first, then differentiation

Approach

$$g(\mathbf{v}) = \nabla_{\mathbf{v}} \mathbb{E}_q[h(\mathbf{z}, \mathbf{v})] \tag{23}$$

- Switch order to integration first, then differentiation
- Write integration as expectation again

Approach

$$g(\mathbf{v}) = \nabla_{\mathbf{v}} \mathbb{E}_q[h(\mathbf{z}, \mathbf{v})] \tag{23}$$

- Switch order to integration first, then differentiation
- Write integration as expectation again
- Approximate the expectation after having taken the gradient

  ⏩ Monte Carlo estimator (ideally with low variance)

Approach

$$g(\mathbf{v}) = \nabla_{\mathbf{v}} \mathbb{E}_q[h(\mathbf{z}, \mathbf{v})] \tag{23}$$

- Switch order to integration first, then differentiation
- Write integration as expectation again
- Approximate the expectation after having taken the gradient
  ⏩ Monte Carlo estimator (ideally with low variance)
- Stochastic optimization

Approach

$$g(\mathbf{v}) = \nabla_{\mathbf{v}} \mathbb{E}_q[h(\mathbf{z}, \mathbf{v})] \tag{23}$$

- Switch order to integration first, then differentiation
- Write integration as expectation again
- Approximate the expectation after having taken the gradient
  ⏩ Monte Carlo estimator (ideally with low variance)
- Stochastic optimization

⏩ Require: general way to compute gradients of expectations

Change of Variables

Some distributions can be sampled using a **change of variables**, i.e.

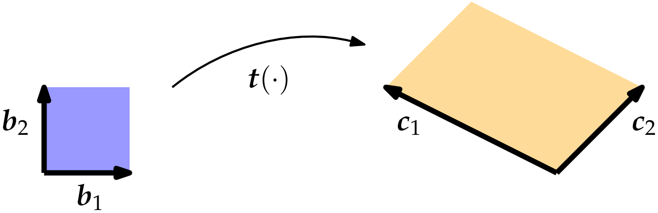
$$\mathbf{z} = \boldsymbol{t}(\boldsymbol{\epsilon}) \qquad \text{with } \boldsymbol{\epsilon} \sim p(\boldsymbol{\epsilon}) \implies p(\mathbf{z}) \text{ some desired distribution}$$

Densities are related

$$p(\boldsymbol{\epsilon}) = p(\mathbf{z} = t(\boldsymbol{\epsilon}))\frac{\partial t(\boldsymbol{\epsilon})}{\partial \boldsymbol{\epsilon}}$$

Integrals are related

$$\int h(\mathbf{z})p(\mathbf{z})\mathrm{d}\mathbf{z} = \int h(t(\boldsymbol{\epsilon}))p(\mathbf{z} = t(\boldsymbol{\epsilon}))\frac{\partial t(\boldsymbol{\epsilon})}{\partial \boldsymbol{\epsilon}}\mathrm{d}\boldsymbol{\epsilon} = \int h(t(\boldsymbol{\epsilon}))p(\boldsymbol{\epsilon})\mathrm{d}\boldsymbol{\epsilon}$$



Gradients of Expectations: Approach 2

$$\nabla_{\boldsymbol{\nu}} \text{ELBO} = \nabla_{\boldsymbol{\nu}} \mathbb{E}_q[g(\boldsymbol{z}, \boldsymbol{\nu})]$$

Gradients of Expectations: Approach 2

$$\begin{aligned}\nabla_{\boldsymbol{\nu}}\text{ELBO} &= \nabla_{\boldsymbol{\nu}}\mathbb{E}_q[g(\boldsymbol{z},\boldsymbol{\nu})]\\ &= \nabla_{\boldsymbol{\nu}}\int g(\boldsymbol{z},\boldsymbol{\nu})q_{\mathbf{v}}(\boldsymbol{z})d\boldsymbol{z}\end{aligned}$$

Gradients of Expectations: Approach 2

$$\begin{aligned}\nabla_{\boldsymbol{\nu}}\text{ELBO} &= \nabla_{\boldsymbol{\nu}}\mathbb{E}_q[g(\boldsymbol{z},\boldsymbol{\nu})] \\ &= \nabla_{\boldsymbol{\nu}}\int g(\boldsymbol{z},\boldsymbol{\nu})q_{\mathbf{v}}(\boldsymbol{z})d\boldsymbol{z} \\ &= \nabla_{\boldsymbol{\nu}}\int g(\boldsymbol{z},\boldsymbol{\nu})q(\boldsymbol{\epsilon})d\boldsymbol{\epsilon}\end{aligned}$$

$$\boxed{q(\boldsymbol{z})d\boldsymbol{z} = q(\boldsymbol{\epsilon})d\boldsymbol{\epsilon}}$$

Gradients of Expectations: Approach 2

$$
\begin{aligned}
\nabla_{\boldsymbol{\nu}}\text{ELBO} &= \nabla_{\boldsymbol{\nu}}\mathbb{E}_q[g(\boldsymbol{z},\boldsymbol{\nu})] \\
&= \nabla_{\boldsymbol{\nu}} \int g(\boldsymbol{z},\boldsymbol{\nu}) q_{\mathbf{v}}(\boldsymbol{z}) d\boldsymbol{z} \\
&= \nabla_{\boldsymbol{\nu}} \int g(\boldsymbol{z},\boldsymbol{\nu}) q(\boldsymbol{\epsilon}) d\boldsymbol{\epsilon} && \boxed{q(\boldsymbol{z})d\boldsymbol{z} = q(\boldsymbol{\epsilon})d\boldsymbol{\epsilon}} \\
&= \nabla_{\boldsymbol{\nu}} \int g(t(\boldsymbol{\epsilon},\boldsymbol{\nu}),\boldsymbol{\nu}) q(\boldsymbol{\epsilon}) d\boldsymbol{\epsilon} && \boxed{\boldsymbol{z} = t(\boldsymbol{\epsilon},\boldsymbol{\nu})}
\end{aligned}
$$

Gradients of Expectations: Approach 2

$$
\begin{aligned}
\nabla_{\boldsymbol{\nu}} \text{ELBO} &= \nabla_{\boldsymbol{\nu}} \mathbb{E}_q[g(\boldsymbol{z}, \boldsymbol{\nu})] \\
&= \nabla_{\boldsymbol{\nu}} \int g(\boldsymbol{z}, \boldsymbol{\nu}) q_{\mathbf{v}}(\boldsymbol{z}) d\boldsymbol{z} \\
&= \nabla_{\boldsymbol{\nu}} \int g(\boldsymbol{z}, \boldsymbol{\nu}) q(\boldsymbol{\epsilon}) d\boldsymbol{\epsilon} && \boxed{q(\boldsymbol{z}) d\boldsymbol{z} = q(\boldsymbol{\epsilon}) d\boldsymbol{\epsilon}} \\
&= \nabla_{\boldsymbol{\nu}} \int g(t(\boldsymbol{\epsilon}, \boldsymbol{\nu}), \boldsymbol{\nu}) q(\boldsymbol{\epsilon}) d\boldsymbol{\epsilon} && \boxed{\boldsymbol{z} = t(\boldsymbol{\epsilon}, \boldsymbol{\nu})} \\
&= \int \nabla_{\boldsymbol{\nu}} g(t(\boldsymbol{\epsilon}, \boldsymbol{\nu}), \boldsymbol{\nu}) q(\boldsymbol{\epsilon}) d\boldsymbol{\epsilon} && \boxed{\nabla_{\boldsymbol{\nu}} \int_{\boldsymbol{\epsilon}} = \int_{\boldsymbol{\epsilon}} \nabla_{\boldsymbol{\nu}}}
\end{aligned}
$$

Gradients of Expectations: Approach 2

$$
\begin{aligned}
\nabla_{\boldsymbol{\nu}}\mathrm{ELBO} &= \nabla_{\boldsymbol{\nu}}\mathbb{E}_q[g(\boldsymbol{z},\boldsymbol{\nu})] \\
&= \nabla_{\boldsymbol{\nu}}\int g(\boldsymbol{z},\boldsymbol{\nu})q_{\mathbf{v}}(\boldsymbol{z})d\boldsymbol{z} \\
&= \nabla_{\boldsymbol{\nu}}\int g(\boldsymbol{z},\boldsymbol{\nu})q(\boldsymbol{\epsilon})d\boldsymbol{\epsilon} && \boxed{q(\boldsymbol{z})d\boldsymbol{z} = q(\boldsymbol{\epsilon})d\boldsymbol{\epsilon}} \\
&= \nabla_{\boldsymbol{\nu}}\int g(t(\boldsymbol{\epsilon},\boldsymbol{\nu}),\boldsymbol{\nu})q(\boldsymbol{\epsilon})d\boldsymbol{\epsilon} && \boxed{\boldsymbol{z} = t(\boldsymbol{\epsilon},\boldsymbol{\nu})} \\
&= \int \nabla_{\boldsymbol{\nu}}g(t(\boldsymbol{\epsilon},\boldsymbol{\nu}),\boldsymbol{\nu})q(\boldsymbol{\epsilon})d\boldsymbol{\epsilon} && \boxed{\nabla_{\boldsymbol{\nu}}\int_{\boldsymbol{\epsilon}} = \int_{\boldsymbol{\epsilon}}\nabla_{\boldsymbol{\nu}}} \\
&= \mathbb{E}_{q(\boldsymbol{\epsilon})}[\nabla_{\boldsymbol{\nu}}g(t(\boldsymbol{\epsilon},\boldsymbol{\nu}),\boldsymbol{\nu})]
\end{aligned}
$$

Gradients of Expectations: Approach 2

$$
\begin{aligned}
\nabla_{\boldsymbol{\nu}} \text{ELBO} &= \nabla_{\boldsymbol{\nu}} \mathbb{E}_q[g(\boldsymbol{z}, \boldsymbol{\nu})] \\
&= \nabla_{\boldsymbol{\nu}} \int g(\boldsymbol{z}, \boldsymbol{\nu}) q_{\mathbf{v}}(\boldsymbol{z}) d\boldsymbol{z} \\
&= \nabla_{\boldsymbol{\nu}} \int g(\boldsymbol{z}, \boldsymbol{\nu}) q(\boldsymbol{\epsilon}) d\boldsymbol{\epsilon} & \boxed{q(\boldsymbol{z}) d\boldsymbol{z} = q(\boldsymbol{\epsilon}) d\boldsymbol{\epsilon}} \\
&= \nabla_{\boldsymbol{\nu}} \int g(t(\boldsymbol{\epsilon}, \boldsymbol{\nu}), \boldsymbol{\nu}) q(\boldsymbol{\epsilon}) d\boldsymbol{\epsilon} & \boxed{\boldsymbol{z} = t(\boldsymbol{\epsilon}, \boldsymbol{\nu})} \\
&= \int \nabla_{\boldsymbol{\nu}} g(t(\boldsymbol{\epsilon}, \boldsymbol{\nu}), \boldsymbol{\nu}) q(\boldsymbol{\epsilon}) d\boldsymbol{\epsilon} & \boxed{\nabla_{\boldsymbol{\nu}} \int_{\boldsymbol{\epsilon}} = \int_{\boldsymbol{\epsilon}} \nabla_{\boldsymbol{\nu}}} \\
&= \mathbb{E}_{q(\boldsymbol{\epsilon})}[\nabla_{\boldsymbol{\nu}} g(t(\boldsymbol{\epsilon}, \boldsymbol{\nu}), \boldsymbol{\nu})]
\end{aligned}
$$

▶▶ Turned gradient of an expectation into expectation of a gradient (and sampling from $q(\boldsymbol{\epsilon})$ is very easy).

Reparametrization Trick

Reparametrization Trick

Base distribution $p(\boldsymbol{\epsilon})$ and a deterministic transformation $\boldsymbol{z} = t(\boldsymbol{\epsilon}, \boldsymbol{\nu})$ so that $\boldsymbol{z} \sim q_{\mathbf{v}}(\boldsymbol{z})$. Then:

$$\nabla_{\boldsymbol{\nu}} \mathbb{E}_{q_{\mathbf{v}}(\boldsymbol{z})}[f(\boldsymbol{z})] = \mathbb{E}_{p(\boldsymbol{\epsilon})}[\nabla_{\boldsymbol{\nu}} f(t(\boldsymbol{\epsilon}, \boldsymbol{\nu}))]$$

▶▶ Expectation taken w.r.t. base distribution

Reparametrization Trick

Reparametrization Trick

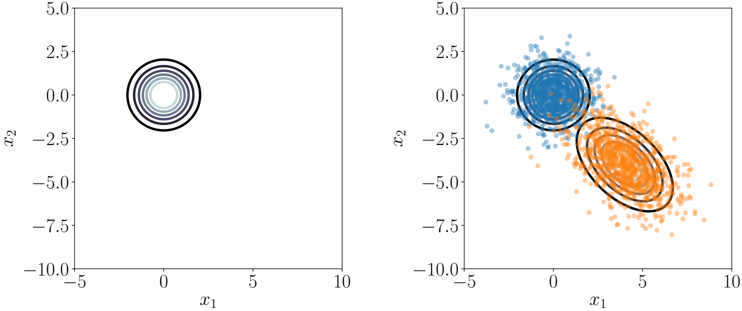
Base distribution $p(\boldsymbol{\epsilon})$ and a deterministic transformation $\boldsymbol{z} = t(\boldsymbol{\epsilon}, \boldsymbol{\nu})$ so that $\boldsymbol{z} \sim q_{\mathbf{v}}(\boldsymbol{z})$. Then:

$$\nabla_{\boldsymbol{\nu}} \mathbb{E}_{q_{\mathbf{v}}(\boldsymbol{z})}[f(\boldsymbol{z})] = \mathbb{E}_{p(\boldsymbol{\epsilon})}[\nabla_{\boldsymbol{\nu}} f(t(\boldsymbol{\epsilon}, \boldsymbol{\nu}))]$$

⏩ Expectation taken w.r.t. base distribution

- Key idea: change of variables using a deterministic transformation

Example



$$
\begin{aligned}
\boldsymbol{\nu} &:= \{\boldsymbol{\mu}, \boldsymbol{R}\}, \quad \boldsymbol{R}\boldsymbol{R}^\top = \boldsymbol{\Sigma} \\
p(\boldsymbol{\epsilon}) &= \mathcal{N}\left(\mathbf{0}, \boldsymbol{I}\right) \\
t(\boldsymbol{\epsilon}, \boldsymbol{\nu}) &= \boldsymbol{\mu} + \boldsymbol{R}\boldsymbol{\epsilon} \\
\implies p(\boldsymbol{z}) &= \mathcal{N}\left(\boldsymbol{z} \,|\, \boldsymbol{\mu}, \boldsymbol{\Sigma}\right)
\end{aligned}
$$

Pathwise Gradients

$$\begin{aligned} g(\boldsymbol{z}, \boldsymbol{\nu}) &= \log p(\boldsymbol{x}, \boldsymbol{z}) - \log q(\boldsymbol{z}|\boldsymbol{\nu}) \\ \boldsymbol{z} &= t(\boldsymbol{\epsilon}, \boldsymbol{\nu}) \end{aligned}$$

Simplify gradient of the ELBO:

Pathwise Gradients

$$\begin{aligned} g(\boldsymbol{z}, \boldsymbol{\nu}) &= \log p(\boldsymbol{x}, \boldsymbol{z}) - \log q(\boldsymbol{z}|\boldsymbol{\nu}) \\ \boldsymbol{z} &= t(\boldsymbol{\epsilon}, \boldsymbol{\nu}) \end{aligned}$$

Simplify gradient of the ELBO:

$$\begin{aligned} \nabla_{\boldsymbol{\nu}} \text{ELBO} &= \mathbb{E}_{p(\boldsymbol{\epsilon})}[\nabla_{\boldsymbol{\nu}} g(t(\boldsymbol{\epsilon}, \boldsymbol{\nu}), \boldsymbol{\nu})] \\ &= \mathbb{E}_{p(\boldsymbol{\epsilon})}[\nabla_{\boldsymbol{\nu}} \log p(\boldsymbol{x}, t(\boldsymbol{\epsilon}, \boldsymbol{\nu})) - \nabla_{\boldsymbol{\nu}} \log q(t(\boldsymbol{\epsilon}, \boldsymbol{\nu})|\boldsymbol{\nu})] \quad \boxed{\text{Def. of } g} \\ &= \mathbb{E}_{p(\boldsymbol{\epsilon})}[\nabla_{\boldsymbol{z}} \log p(\boldsymbol{x}, \boldsymbol{z}) \nabla_{\boldsymbol{\nu}} t(\boldsymbol{\epsilon}, \boldsymbol{\nu}) \\ &\quad - \nabla_{\boldsymbol{z}} \log q(\boldsymbol{z}|\boldsymbol{\nu}) \nabla_{\boldsymbol{\nu}} t(\boldsymbol{\epsilon}, \boldsymbol{\nu}) - \underbrace{\nabla_{\boldsymbol{\nu}} \log q(t(\boldsymbol{\epsilon}, \boldsymbol{\nu})|\boldsymbol{\nu})}_{\text{score}}] \quad \boxed{\text{Chain rule}} \\ &= \mathbb{E}_{p(\boldsymbol{\epsilon})}[\nabla_{\boldsymbol{z}} \big(\log p(\boldsymbol{x}, \boldsymbol{z}) - \log q_{\mathbf{v}}(\boldsymbol{z})\big) \nabla_{\boldsymbol{\nu}} t(\boldsymbol{\epsilon}, \boldsymbol{\nu})] \quad \boxed{\text{Score property}} \end{aligned}$$

- Pathwise gradient
- Reparametrization gradient

Variance Comparison

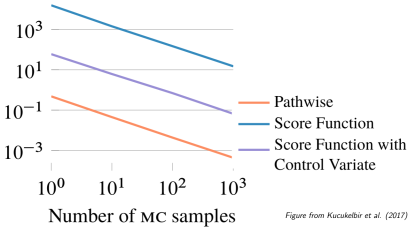


Figure from Kucukelbir et al. (2017)

- Drastically reduced variance compared to score-function gradient estimation

Variance Comparison

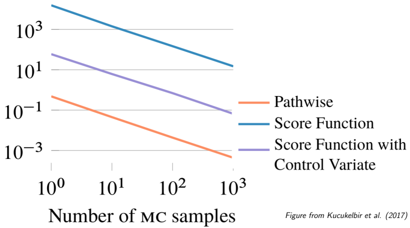


Figure from Kucukelbir et al. (2017)

- Drastically reduced variance compared to score-function gradient estimation
- Restricted class of models (compared with score function estimator)

Score Function vs Pathwise Gradients

$$\text{ELBO} = \int g(z, \nu) q_{\mathbf{v}}(z) dz$$

$$g(z, \nu) = \log p(x, z) - \log q(z|\mu)$$

- Score function gradient:

$$\nabla_\nu \text{ELBO} = \mathbb{E}_q[\left(\nabla_\nu \log q(z|\nu)\right) g(z, \nu)]$$

⏩ Gradient of the variational distribution

- Reparametrization gradient:

$$\nabla_\nu \text{ELBO} = \mathbb{E}_{p(\epsilon)}[\left(\nabla_z g(z, \nu)\right) \nabla_\nu t(\epsilon, \nu)]$$

⏩ Gradient of the model and the variational distribution

- Often, $\mathbb{E}_{q_{\mathbf{v}}(\mathbf{z})}[\log q_{\mathbf{v}}(\mathbf{z})]$ can be computed in closed form, and is excluded from MC estimation. (Skill to recognise when.)

Summary

- Score function
  - Works for all models (continuous and discrete)
  - Works for a large class of variational approximations
  - Variance can be high ⏩ Slow convergence
- Pathwise gradient estimator
  - Requires differentiable models
  - Requires the variational approximation to be expressed as a deterministic transformation $\boldsymbol{z} = t(\boldsymbol{\epsilon}, \boldsymbol{\nu})$
  - Generally lower variance

References I