Variational Inference

STATS 305B: Applied Statistics II

Scott Linderman

February 3, 2025

Recap

We've covered several approaches for Bayesian inference...

- **Exact inference:** for simple models (e.g. conjugate exponential family models) where the posterior is available in closed form.
- **Laplace Approximation:** for unimodal posteriors where you can find the mode and local curvature.
- **Markov Chain Monte Carlo:** for sampling from the posterior when it's difficult to compute or approximate.
  - **Metropolis-Hastings:** a very general MCMC algorithm, and the building block for many other MCMC techniques.
  - **Gibbs sampling:** an MCMC algorithm that iteratively samples conditional distributions for one variable at a time. This works well for conditionally conjugate models with weak correlations.
  - **Hamiltonian Monte Carlo:** an MCMC algorithm to draw samples from the posterior by leveraging gradients of the log joint probability. This works well for more general posteriors over continuous variables.

Outline

Today we'll talk about another approach to approximate Bayesian inference called **variational inference (VI)**.

The key idea is to approximate the posterior with the closest member of a parametric family.

Instead of a sampling problem, this is an optimization problem.

Why Variational Inference?

MCMC methods are asymptotically unbiased (though for finite samples there is a transient bias that shrinks as $O(M^{-1})$). The real issue is variance: it only shrinks as $O(M^{-1/2})$.

Motivation: With finite computation, can we get better posterior estimates by trading asymptotic bias for smaller variance?

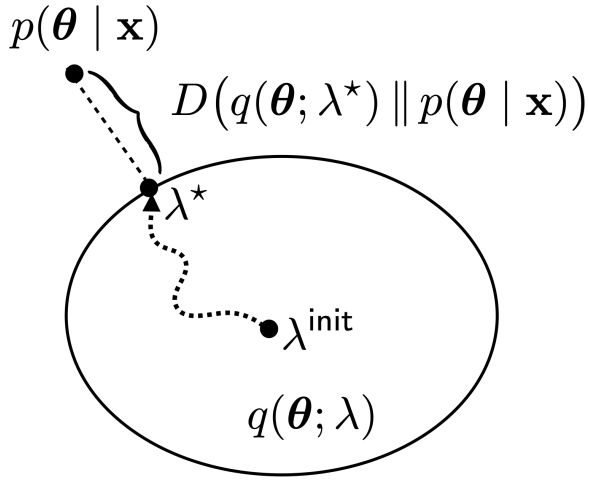
Idea: approximate the posterior by with a simple, parametric form (though not strictly a Gaussian on the mode!). Optimize to find the approximation that is as "close" as possible to the posterior.

Notation

This notation could be a bit confusing. Let,

- $\boldsymbol{\theta} \in \mathbb{R}^D$ denote **all of latent variables and parameters** we wish to infer.
- $p(\boldsymbol{\theta} \mid \boldsymbol{x})$ denote the true posterior distribution we want to approximate.
- $q(\boldsymbol{\theta}; \boldsymbol{\lambda})$ denote a parametric *variational approximation* to the posterior where...
- $\boldsymbol{\lambda}$ denotes the *variational parameters* that we will optimize.
- $D(q \,||\, p)$ denote a *divergence measure* that takes in two distributions q and p and returns a measure of how similar they are.

A view of variational inference



Three key questions

- *What parametric family should we use?*
- *How should we measure closeness?*
- *How do we find the closest distribution in that family?*

Coordinate Ascent Variational Inference (CAVI)

- *What parametric family should we use?*
  - The **mean-field family**.
- *How should we measure closeness?*
  - The **Kullback-Leibler (KL)** divergence.
- *How do we find the closest distribution in that family?*
  - **Coordinate ascent**, assuming we have a conditionally conjugate model.

Gradient-based Variational Inference

- *What parametric family should we use?*
  - Pretty much any q, as long as we can sample from it and evaluate the log density.
- *How should we measure closeness?*
  - The **Kullback-Leibler (KL)** divergence.
- *How do we find the closest distribution in that family?*
  - **Stochastic gradient ascent** using Monte Carlo estimates of the ELBO and its gradient.

Gradient-based VI methods go under a few different names: black-box VI (BBVI), automatic differentiation VI (ADVI), fixed-form VI...

The Kullback-Leibler (KL) divergence

The KL divergence is a measure of closeness between two distributions. It is defined as,

$$\begin{aligned} D_{\mathrm{KL}}\left(q(\boldsymbol{\theta};\boldsymbol{\lambda}) \,\|\, p(\boldsymbol{\theta} \mid \boldsymbol{x})\right) &= \mathbb{E}_{q(\boldsymbol{\theta};\boldsymbol{\lambda})}\left[\log \frac{q(\boldsymbol{\theta};\boldsymbol{\lambda})}{p(\boldsymbol{\theta} \mid \boldsymbol{x})}\right] \\ &= \mathbb{E}_{q(\boldsymbol{\theta};\boldsymbol{\lambda})}\left[\log q(\boldsymbol{\theta};\boldsymbol{\lambda})\right] - \mathbb{E}_{q(\boldsymbol{\theta};\boldsymbol{\lambda})}\left[\log p(\boldsymbol{\theta} \mid \boldsymbol{x})\right] \end{aligned}$$

It has some nice properties:

- It is non-negative.
- It is zero iff $q(\boldsymbol{\theta};\boldsymbol{\lambda}) \equiv p(\boldsymbol{\theta} \mid \boldsymbol{x})$.
- It is defined in terms of expectations wrt q.

But it's also a bit weird...

- It's asymmetric ($D_{\mathrm{KL}}\left(q \,\|\, p\right) \neq D_{\mathrm{KL}}\left(p \,\|\, q\right)$).

The evidence lower bound (ELBO) from another angle

More concerning, the KL divergence involves the posterior $p(\boldsymbol{\theta} \mid \boldsymbol{x})$, which we cannot compute!

But notice that...

$$
\begin{aligned}
D_{\mathrm{KL}}\left(q(\boldsymbol{\theta};\boldsymbol{\lambda}) \,\|\, p(\boldsymbol{\theta} \mid \boldsymbol{x}\right) &= \mathbb{E}_{q(\boldsymbol{\theta};\boldsymbol{\lambda})}\left[\log q(\boldsymbol{\theta};\boldsymbol{\lambda})\right] - \mathbb{E}_{q(\boldsymbol{\theta};\boldsymbol{\lambda})}\left[\log p(\boldsymbol{\theta} \mid \boldsymbol{x})\right] \\
&= \mathbb{E}_{q(\boldsymbol{\theta};\boldsymbol{\lambda})}\left[\log q(\boldsymbol{\theta};\boldsymbol{\lambda})\right] - \mathbb{E}_{q(\boldsymbol{\theta};\boldsymbol{\lambda})}\left[\log p(\boldsymbol{\theta},\boldsymbol{x})\right] + \mathbb{E}_{q(\boldsymbol{\theta};\boldsymbol{\lambda})}\left[\log p(\boldsymbol{x})\right] \\
&= \underbrace{\mathbb{E}_{q(\boldsymbol{\theta};\boldsymbol{\lambda})}\left[\log q(\boldsymbol{\theta};\boldsymbol{\lambda})\right] - \mathbb{E}_{q(\boldsymbol{\theta};\boldsymbol{\lambda})}\left[\log p(\boldsymbol{\theta},\boldsymbol{x})\right]}_{\text{negative ELBO}, -\mathscr{L}(\boldsymbol{\lambda})} + \underbrace{\log p(\boldsymbol{x})}_{\text{evidence}}
\end{aligned}
$$

The first term involves the log joint, which we can compute, and the last term is independent of the variational parameters!

Rearranging, we see that $\mathscr{L}(\boldsymbol{\lambda})$ is a lower bound on the marginal likelihood, aka the evidence,

$$
\mathscr{L}(\boldsymbol{\lambda}) = \log p(\boldsymbol{x}) - D_{\mathrm{KL}}\left(q(\boldsymbol{\theta};\boldsymbol{\lambda}) \,\|\, p(\boldsymbol{\theta} \mid \boldsymbol{x}\right) \leq \log p(\boldsymbol{x}).
$$

That's why we call it the **evidence lower bound (ELBO)**.

Viewer discretion advised...

https://www.youtube.com/watch?v=jugUBL4rEIM

Setup

The optimal approximation is,

$$q^\star(\boldsymbol{\theta};\boldsymbol{\lambda}) = \arg\min_{q\in\mathcal{Q}} D_{\mathrm{KL}}\left(q(\boldsymbol{\theta};\boldsymbol{\lambda}) \parallel p(\boldsymbol{\theta} \mid \boldsymbol{X})\right)$$

or equivalently

$$\begin{aligned}\boldsymbol{\lambda}^\star &= \arg\min_{\boldsymbol{\lambda}\in\boldsymbol{\Lambda}} D_{\mathrm{KL}}\left(q(\boldsymbol{\theta};\boldsymbol{\lambda}) \parallel p(\boldsymbol{\theta} \mid \boldsymbol{X})\right) \\ &= \arg\max_{\boldsymbol{\lambda}\in\boldsymbol{\Lambda}} \mathcal{L}(\boldsymbol{\lambda})\end{aligned}$$

where $\mathcal{L}(\boldsymbol{\lambda})$ denotes the **evidence lower bound (ELBO)**,

$$\mathcal{L}(\boldsymbol{\lambda}) = \mathbb{E}_{q(\boldsymbol{\theta};\boldsymbol{\lambda})}\left[\log p(\boldsymbol{X},\boldsymbol{\theta}) - \log q(\boldsymbol{\theta};\boldsymbol{\lambda})\right]$$

Optimizing the ELBO with coordinate ascent

We want to find the variational parameters $\boldsymbol{\lambda}$ that minimize the KL divergence or, equivalently, maximize the ELBO.

For the mean-field family, we can typically do this via **coordinate ascent**.

Consider optimizing the parameters for one factor $q(\theta_d; \lambda_d)$. As a function of λ_d, the ELBO is,

$$\begin{aligned}
\mathscr{L}(\boldsymbol{\lambda}) &= \mathbb{E}_{q(\theta_d;\lambda_d)}\left[\mathbb{E}_{q(\boldsymbol{\theta}_{\neg d};\boldsymbol{\lambda}_{\neg d})}\left[\log p(\boldsymbol{\theta}, \boldsymbol{x})\right]\right] - \mathbb{E}_{q(\theta_d;\lambda_d)}[\log q(\theta_d;\lambda_d)] + c \\
&= \mathbb{E}_{q(\theta_d;\lambda_d)}\left[\mathbb{E}_{q(\boldsymbol{\theta}_{\neg d};\boldsymbol{\lambda}_{\neg d})}\left[\log p(\theta_d \mid \boldsymbol{\theta}_{\neg d}, \boldsymbol{x})\right]\right] - \mathbb{E}_{q(\theta_d;\lambda_d)}[\log q(\theta_d;\lambda_d)] + c' \\
&= -D_{\mathrm{KL}}\left(q(\theta_d;\lambda_d) \,\|\, \tilde{p}(\theta_d)\right) + c''
\end{aligned}$$

where

$$\tilde{p}(\theta_d) \propto \exp\left\{\mathbb{E}_{q(\boldsymbol{\theta}_{\neg d};\boldsymbol{\lambda}_{\neg d})}\left[\log p(\theta_d \mid \boldsymbol{\theta}_{\neg d}, \boldsymbol{x})\right]\right\}$$

The ELBO is maximized wrt λ_d when this KL is minimized; i.e. when $q(\theta_d;\lambda_d) = \tilde{p}(\theta_d)$, the exponentiated expected log conditional probability, holding all other factors fixed.

Optimizing the ELBO with stochastic gradient ascent

- **Idea:** Assume the variational parameters $\boldsymbol{\Lambda}$ are unconstrained (i.e., $\boldsymbol{\Lambda} = \mathbb{R}^Q$), then perform (stochastic) gradient ascent.
- If the parameters are unconstrained and the ELBO is differentiable, we can use **gradient ascent**. Repeat:

$$\boldsymbol{\lambda} \leftarrow \boldsymbol{\lambda} + \alpha \nabla_{\boldsymbol{\lambda}} \mathscr{L}(\boldsymbol{\lambda})$$

  with **step size** α. Typically, you decrease the step size over iterations so that $\alpha_1 \geq \alpha_2 \geq \dots$
- More generally, we can use ***stochastic*** **gradient ascent** with an estimate of the gradient, $\widehat{\nabla}_{\boldsymbol{\lambda}} \mathscr{L}(\boldsymbol{\lambda})$, as long as it is unbiased,

$$\mathbb{E}[\widehat{\nabla}_{\boldsymbol{\lambda}} \mathscr{L}(\boldsymbol{\lambda})] = \nabla_{\boldsymbol{\lambda}} \mathscr{L}(\boldsymbol{\lambda}).$$

Optimizing the ELBO with stochastic gradient ascent

- **Idea:** Assume the variational parameters $\boldsymbol{\Lambda}$ are unconstrained (i.e., $\boldsymbol{\Lambda} = \mathbb{R}^Q$), then perform (stochastic) gradient ascent.
- If the parameters are unconstrained and the ELBO is differentiable, we can use **gradient ascent**. Repeat:

$$\boldsymbol{\lambda} \leftarrow \boldsymbol{\lambda} + \alpha \nabla_{\boldsymbol{\lambda}} \mathscr{L}(\boldsymbol{\lambda})$$

  with **step size** α. Typically, you decrease the step size over iterations so that $\alpha_1 \geq \alpha_2 \geq \dots$
- More generally, we can use ***stochastic* gradient ascent** with an estimate of the gradient, $\widehat{\nabla}_{\boldsymbol{\lambda}} \mathscr{L}(\boldsymbol{\lambda})$, as long as it is unbiased,

$$\mathbb{E}[\widehat{\nabla}_{\boldsymbol{\lambda}} \mathscr{L}(\boldsymbol{\lambda})] = \nabla_{\boldsymbol{\lambda}} \mathscr{L}(\boldsymbol{\lambda}).$$

Optimizing the ELBO with stochastic gradient ascent

- **Idea:** Assume the variational parameters $\boldsymbol{\Lambda}$ are unconstrained (i.e., $\boldsymbol{\Lambda} = \mathbb{R}^Q$), then perform (stochastic) gradient ascent.
- If the parameters are unconstrained and the ELBO is differentiable, we can use **gradient ascent**. Repeat:

$$\boldsymbol{\lambda} \leftarrow \boldsymbol{\lambda} + \alpha \nabla_{\boldsymbol{\lambda}} \mathscr{L}(\boldsymbol{\lambda})$$

  with **step size** α. Typically, you decrease the step size over iterations so that $\alpha_1 \geq \alpha_2 \geq \dots$
- More generally, we can use ***stochastic*** **gradient ascent** with an estimate of the gradient, $\widehat{\nabla}_{\boldsymbol{\lambda}} \mathscr{L}(\boldsymbol{\lambda})$, as long as it is unbiased,

$$\mathbb{E}[\widehat{\nabla}_{\boldsymbol{\lambda}} \mathscr{L}(\boldsymbol{\lambda})] = \nabla_{\boldsymbol{\lambda}} \mathscr{L}(\boldsymbol{\lambda}).$$

Monte Carlo gradient estimation

No problem! We'll just use ordinary Monte Carlo to estimate the gradient. But we run into a problem...

$$\begin{aligned}\nabla_{\boldsymbol{\lambda}} \mathscr{L}(\boldsymbol{\lambda}) &= \nabla_{\boldsymbol{\lambda}} \mathbb{E}_{q(\boldsymbol{\theta};\boldsymbol{\lambda})} \left[\log p(\boldsymbol{x}, \boldsymbol{\theta}) - \log q(\boldsymbol{\theta}; \boldsymbol{\lambda})\right] \\ &\neq \mathbb{E}_{q(\boldsymbol{\theta};\boldsymbol{\lambda})} \left[\nabla_{\boldsymbol{\lambda}} \left(\log p(\boldsymbol{x}, \boldsymbol{\theta}) - \log q(\boldsymbol{\theta}; \boldsymbol{\lambda})\right)\right].\end{aligned}$$

Problem: Why can't we simply bring the gradient inside the expectation?

The “score function” gradient estimator

The basic problem is that the variational parameters $\boldsymbol{\lambda}$ determine the distribution we are taking an expectation under. However, there are a few ways to obtain unbiased estimates of the gradient.

One approach is called the **score function gradient estimator** or the **REINFORCE estimator** [**?**]. It is based on the following identity,

$$\nabla_{\boldsymbol{\lambda}} \log q(\boldsymbol{\theta}; \boldsymbol{\lambda}) = \frac{\nabla_{\boldsymbol{\lambda}} q(\boldsymbol{\theta}; \boldsymbol{\lambda})}{q(\boldsymbol{\theta}; \boldsymbol{\lambda})}$$

where the l.h.s. is called the score function of distribution q.

The "score function" gradient estimator

We can use this identity to obtain an unbiased estimate of the gradient of an expectation,

$$\begin{aligned}\nabla_{\boldsymbol{\lambda}} \mathbb{E}_{q(\boldsymbol{\theta};\boldsymbol{\lambda})}\left[h(\boldsymbol{\theta})\right] &= \nabla_{\boldsymbol{\lambda}} \int q(\boldsymbol{\theta};\boldsymbol{\lambda})\, h(\boldsymbol{\theta})\, \mathrm{d}\boldsymbol{\theta} \\ &= \int \left(\nabla_{\boldsymbol{\lambda}} q(\boldsymbol{\theta};\boldsymbol{\lambda})\right) h(\boldsymbol{\theta})\, \mathrm{d}\boldsymbol{\theta} \\ &= \int \left(q(\boldsymbol{\theta};\boldsymbol{\lambda}) \nabla_{\boldsymbol{\lambda}} \log q(\boldsymbol{\theta};\boldsymbol{\lambda})\right) h(\boldsymbol{\theta})\, \mathrm{d}\boldsymbol{\theta} \\ &= \mathbb{E}_{q(\boldsymbol{\theta};\boldsymbol{\lambda})}\left[\left(\nabla_{\boldsymbol{\lambda}} \log q(\boldsymbol{\theta};\boldsymbol{\lambda})\right) h(\boldsymbol{\theta})\right]\end{aligned}$$

From this identity, we can obtain an unbiased Monte Carlo estimate,

$$\widehat{\nabla}_{\boldsymbol{\lambda}} \mathbb{E}_{q(\boldsymbol{\theta};\boldsymbol{\lambda})}[h(\boldsymbol{\theta})] = \frac{1}{M} \sum_{m=1}^{M} \left[\nabla_{\boldsymbol{\lambda}} \log q(\boldsymbol{\theta}^{(m)};\boldsymbol{\lambda})\, h(\boldsymbol{\theta}^{(m)})\right]; \qquad \boldsymbol{\theta}^{(m)} \overset{\text{iid}}{\sim} q(\boldsymbol{\theta};\boldsymbol{\lambda})$$

The “score function” gradient estimator

Notes:

1. The exchange of the gradient and the integral is allowed as long as the dominated convergence theorem holds, and it usually does for ML applications.
2. The score function gradient estimator is broadly applicable; e.g. it works for discrete and continuous latent variables $\boldsymbol{\theta}$. We just need the log density to be continuously differentiable wrt $\boldsymbol{\lambda}$ and to be able to sample from q.
3. If h is a function of both $\boldsymbol{\theta}$ and $\boldsymbol{\lambda}$, you need to apply the product rule. This gives another term,

$$\nabla_{\boldsymbol{\lambda}} \mathbb{E}_{q(\boldsymbol{\theta};\boldsymbol{\lambda})}\left[h(\boldsymbol{\theta},\boldsymbol{\lambda})\right] = \mathbb{E}_{q(\boldsymbol{\theta};\boldsymbol{\lambda})}\left[\left(\nabla_{\boldsymbol{\lambda}} \log q(\boldsymbol{\theta};\boldsymbol{\lambda})\right) h(\boldsymbol{\theta},\boldsymbol{\lambda})\right] + \mathbb{E}_{q(\boldsymbol{\theta};\boldsymbol{\lambda})}\left[\nabla_{\boldsymbol{\lambda}} h(\boldsymbol{\theta},\boldsymbol{\lambda})\right]$$

Control variates

Though broadly applicable, the score function estimator is often too high variance to be useful. This problem can often be mitigated with **control variates**.

Recall that the expectation of the score is zero,

$$\begin{aligned}\mathbb{E}_{q(\boldsymbol{\theta};\boldsymbol{\lambda})}\left[\nabla_{\boldsymbol{\lambda}} \log q(\boldsymbol{\theta};\boldsymbol{\lambda})\right] &= \int q(\boldsymbol{\theta};\boldsymbol{\lambda}) \nabla_{\boldsymbol{\lambda}} \log q(\boldsymbol{\theta};\boldsymbol{\lambda}) \,\mathrm{d}\boldsymbol{\theta} \\ &= \int \nabla_{\boldsymbol{\lambda}} q(\boldsymbol{\theta};\boldsymbol{\lambda}) \,\mathrm{d}\boldsymbol{\theta} \\ &= \nabla_{\boldsymbol{\lambda}} \int q(\boldsymbol{\theta};\boldsymbol{\lambda}) \,\mathrm{d}\boldsymbol{\theta} \\ &= \nabla_{\boldsymbol{\lambda}} 1 = 0.\end{aligned}$$

Thus, we can subtract off any **baseline** from the function of interest without changing the expectation, but potentially reducing variance substantially,

$$\mathbb{E}_{q(\boldsymbol{\theta};\boldsymbol{\lambda})}\left[h(\boldsymbol{\theta}) \nabla_{\boldsymbol{\lambda}} \log q(\boldsymbol{\theta};\boldsymbol{\lambda})\right] = \mathbb{E}_{q(\boldsymbol{\theta};\boldsymbol{\lambda})}\left[(h(\boldsymbol{\theta}) - b) \nabla_{\boldsymbol{\lambda}} \log q(\boldsymbol{\theta};\boldsymbol{\lambda})\right].$$

The pathwise gradient estimator

- The pathwise gradient estimator has more requirements, but often performs better. Suppose $q(\boldsymbol{\theta}; \boldsymbol{\lambda}) = \mathcal{N}(\boldsymbol{\theta}; \boldsymbol{\mu}, \mathrm{diag}(\boldsymbol{\sigma}^2))$, where $\boldsymbol{\lambda} = (\boldsymbol{\mu}, \log \boldsymbol{\sigma}^2)$ are the (unconstrained) variational parameters. Then,

$$\boldsymbol{\theta} \sim q(\boldsymbol{\theta}; \boldsymbol{\lambda}) \quad \Longleftrightarrow \quad \boldsymbol{\theta} = r(\boldsymbol{\lambda}, \boldsymbol{\epsilon}), \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$$

  where $r(\boldsymbol{\lambda}, \boldsymbol{\epsilon}) = \boldsymbol{\mu} + \boldsymbol{\sigma}\boldsymbol{\epsilon}$ is a **reparameterization** of $\boldsymbol{\theta}$ in terms of parameters $\boldsymbol{\lambda}$ and "noise" $\boldsymbol{\epsilon}$.

- We can use the **law of the unconscious statistician** to rewrite the expectations as,

$$\mathbb{E}_{q(\boldsymbol{\theta};\boldsymbol{\lambda})}\left[h(\boldsymbol{\theta}, \boldsymbol{\lambda})\right] = \mathbb{E}_{\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})}\left[h(r(\boldsymbol{\lambda}, \boldsymbol{\epsilon}), \boldsymbol{\lambda})\right]$$

  The distribution that the expectation is taken under no longer depends on the parameters $\boldsymbol{\lambda}$, so we can simply take the gradient inside the expectation,

$$\nabla_{\boldsymbol{\lambda}} \mathbb{E}_{q(\boldsymbol{\theta};\boldsymbol{\lambda})}\left[h(\boldsymbol{\theta}, \boldsymbol{\lambda})\right] = \mathbb{E}_{\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})}\left[\nabla_{\boldsymbol{\lambda}} h(r(\boldsymbol{\lambda}, \boldsymbol{\epsilon}), \boldsymbol{\lambda})\right]$$

- Now we can **use Monte Carlo to obtain an unbiased estimate of the final expectation.**

$$\widehat{\nabla}_{\boldsymbol{\lambda}} \mathbb{E}_{q(\boldsymbol{\theta};\boldsymbol{\lambda})}\left[h(\boldsymbol{\theta}, \boldsymbol{\lambda})\right] = \frac{1}{M} \sum^{M} \nabla_{\boldsymbol{\lambda}} h(r(\boldsymbol{\lambda}, \boldsymbol{\epsilon}_m), \boldsymbol{\lambda}); \qquad \boldsymbol{\epsilon}_m \overset{\text{iid}}{\sim} \mathcal{N}(\mathbf{0}, \boldsymbol{I})$$

The pathwise gradient estimator

- The pathwise gradient estimator has more requirements, but often performs better. Suppose $q(\boldsymbol{\theta}; \boldsymbol{\lambda}) = \mathcal{N}(\boldsymbol{\theta}; \boldsymbol{\mu}, \mathrm{diag}(\boldsymbol{\sigma}^2))$, where $\boldsymbol{\lambda} = (\boldsymbol{\mu}, \log \boldsymbol{\sigma}^2)$ are the (unconstrained) variational parameters. Then,

$$\boldsymbol{\theta} \sim q(\boldsymbol{\theta}; \boldsymbol{\lambda}) \quad \Longleftrightarrow \quad \boldsymbol{\theta} = r(\boldsymbol{\lambda}, \boldsymbol{\epsilon}), \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$$

  where $r(\boldsymbol{\lambda}, \boldsymbol{\epsilon}) = \boldsymbol{\mu} + \boldsymbol{\sigma}\boldsymbol{\epsilon}$ is a **reparameterization** of $\boldsymbol{\theta}$ in terms of parameters $\boldsymbol{\lambda}$ and "noise" $\boldsymbol{\epsilon}$.

- We can use the **law of the unconscious statistician** to rewrite the expectations as,

$$\mathbb{E}_{q(\boldsymbol{\theta};\boldsymbol{\lambda})}\left[h(\boldsymbol{\theta}, \boldsymbol{\lambda})\right] = \mathbb{E}_{\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})}\left[h(r(\boldsymbol{\lambda}, \boldsymbol{\epsilon}), \boldsymbol{\lambda})\right]$$

  The distribution that the expectation is taken under no longer depends on the parameters $\boldsymbol{\lambda}$, so we can simply take the gradient inside the expectation,

$$\nabla_{\boldsymbol{\lambda}} \mathbb{E}_{q(\boldsymbol{\theta};\boldsymbol{\lambda})}\left[h(\boldsymbol{\theta}, \boldsymbol{\lambda})\right] = \mathbb{E}_{\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})}\left[\nabla_{\boldsymbol{\lambda}} h(r(\boldsymbol{\lambda}, \boldsymbol{\epsilon}), \boldsymbol{\lambda})\right]$$

- Now we can **use Monte Carlo to obtain an unbiased estimate of the final expectation.**

$$\widehat{\nabla}_{\boldsymbol{\lambda}} \mathbb{E}_{q(\boldsymbol{\theta};\boldsymbol{\lambda})}\left[h(\boldsymbol{\theta}, \boldsymbol{\lambda})\right] = \frac{1}{M} \sum_{m}^{M} \nabla_{\boldsymbol{\lambda}} h(r(\boldsymbol{\lambda}, \boldsymbol{\epsilon}_m), \boldsymbol{\lambda}); \qquad \boldsymbol{\epsilon}_m \overset{\text{iid}}{\sim} \mathcal{N}(\mathbf{0}, \boldsymbol{I})$$

The pathwise gradient estimator

- The pathwise gradient estimator has more requirements, but often performs better. Suppose $q(\boldsymbol{\theta}; \boldsymbol{\lambda}) = \mathcal{N}(\boldsymbol{\theta}; \boldsymbol{\mu}, \mathrm{diag}(\boldsymbol{\sigma}^2))$, where $\boldsymbol{\lambda} = (\boldsymbol{\mu}, \log \boldsymbol{\sigma}^2)$ are the (unconstrained) variational parameters. Then,

$$\boldsymbol{\theta} \sim q(\boldsymbol{\theta}; \boldsymbol{\lambda}) \quad \Longleftrightarrow \quad \boldsymbol{\theta} = r(\boldsymbol{\lambda}, \boldsymbol{\epsilon}), \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$$

  where $r(\boldsymbol{\lambda}, \boldsymbol{\epsilon}) = \boldsymbol{\mu} + \boldsymbol{\sigma}\boldsymbol{\epsilon}$ is a **reparameterization** of $\boldsymbol{\theta}$ in terms of parameters $\boldsymbol{\lambda}$ and "noise" $\boldsymbol{\epsilon}$.

- We can use the **law of the unconscious statistician** to rewrite the expectations as,

$$\mathbb{E}_{q(\boldsymbol{\theta};\boldsymbol{\lambda})}\left[h(\boldsymbol{\theta}, \boldsymbol{\lambda})\right] = \mathbb{E}_{\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})}\left[h(r(\boldsymbol{\lambda}, \boldsymbol{\epsilon}), \boldsymbol{\lambda})\right]$$

  The distribution that the expectation is taken under no longer depends on the parameters $\boldsymbol{\lambda}$, so we can simply take the gradient inside the expectation,

$$\nabla_{\boldsymbol{\lambda}} \mathbb{E}_{q(\boldsymbol{\theta};\boldsymbol{\lambda})}\left[h(\boldsymbol{\theta}, \boldsymbol{\lambda})\right] = \mathbb{E}_{\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})}\left[\nabla_{\boldsymbol{\lambda}} h(r(\boldsymbol{\lambda}, \boldsymbol{\epsilon}), \boldsymbol{\lambda})\right]$$

- Now we can **use Monte Carlo to obtain an unbiased estimate of the final expectation.**

$$\widehat{\nabla}_{\boldsymbol{\lambda}} \mathbb{E}_{q(\boldsymbol{\theta};\boldsymbol{\lambda})}\left[h(\boldsymbol{\theta}, \boldsymbol{\lambda})\right] = \frac{1}{M} \sum^{M} \nabla_{\boldsymbol{\lambda}} h(r(\boldsymbol{\lambda}, \boldsymbol{\epsilon}_m), \boldsymbol{\lambda}); \qquad \boldsymbol{\epsilon}_m \overset{\text{iid}}{\sim} \mathcal{N}(\mathbf{0}, \boldsymbol{I})$$

Exercises

Exercise: Come up with a reparameterization of an exponential distribution, $q(\theta; \lambda) = \text{Exp}(\theta; \lambda)$

Question: Can you use the pathwise gradient estimator for a Bernoulli posterior, $q(\theta; \lambda) = \text{Bern}(\theta; \lambda)$?

Empirically comparing estimator variances

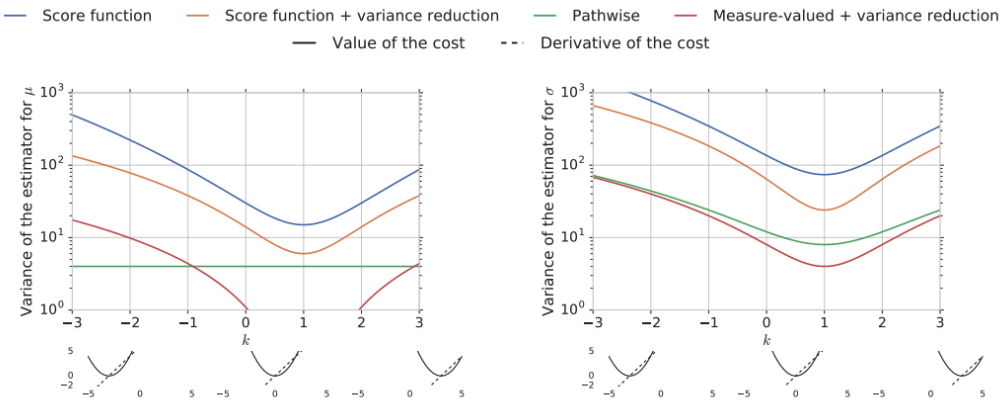


Figure 2: Variance of the stochastic estimates of $\nabla_\theta \mathbb{E}_{\mathcal{N}(x|\mu,\sigma^2)}\left[(x-k)^2\right]$ for $\mu = \sigma = 1$ as a function of k for three different classes of gradient estimators. Left: $\theta = \mu$; right: $\theta = \sigma$. The graphs in the bottom row show the function (solid) and its gradient (dashed) for $k \in \{-3, 0, 3\}$.

Empirical comparisons from **?**.

Working with mini-batches of data

Often, the ELBO involves a sum over data points,

$$\begin{aligned}\mathscr{L}(\boldsymbol{\lambda}) &= \mathbb{E}_q[\log p(\boldsymbol{X}, \boldsymbol{\theta}) - \log q(\boldsymbol{\theta}; \boldsymbol{\lambda}] \\ &= \mathbb{E}_{q(\boldsymbol{\theta};\boldsymbol{\lambda})}\left[\sum_{n=1}^{N} \log p(\boldsymbol{x}_n \mid \boldsymbol{\theta}) + \log p(\boldsymbol{\theta}) - \log q(\boldsymbol{\theta}; \boldsymbol{\lambda})\right] \\ &= \sum_{n=1}^{N} \mathbb{E}_{q(\boldsymbol{\theta};\boldsymbol{\lambda})}[\log p(\boldsymbol{x}_n \mid \boldsymbol{\theta})] - D_{\mathrm{KL}}\left(q(\boldsymbol{\theta}; \boldsymbol{\lambda}) \,\|\, p(\boldsymbol{\theta})\right)\end{aligned}$$

We can view the sum as an "expectation" over data indices,

$$\sum_{n=1}^{N} \mathbb{E}_{q(\boldsymbol{\theta};\boldsymbol{\lambda})}[\log p(\boldsymbol{x}_n \mid \boldsymbol{\theta})] = N\,\mathbb{E}_{n\sim\mathrm{Unif}(1,N)}[\mathbb{E}_{q(\boldsymbol{\theta};\boldsymbol{\lambda})}[\log p(\boldsymbol{x}_n \mid \boldsymbol{\theta})]],$$

and we can use Monte Carlo to approximate both expectations! (The same is true for Monte Carlo estimators of the gradient of the ELBO.)

SGD convergence and extensions

When does SGD work? This is a well studied problem in stochastic optimization [**??**].

Under relatively mild conditions, SGD converges to a **local minimum** if the step sizes obey the **Robbins-Monro condtions**,

$$\sum_{i=0}^{\infty} \alpha_i = \infty \qquad \text{and} \qquad \sum_{i=0}^{\infty} \alpha_i^2 < \infty$$

There have been dozens of extensions to basic SGD including,

- SGD with momentum
- AdaGrad [**?**]
- RMSProp
- Adam [**?**]

References