Approximate inference (part one); Exponential families

CPSC 440/550: Advanced Machine Learning

cs.ubc.ca/~dsuth/440/23w2

University of British Columbia, on unceded Musqueam land

2023-24 Winter Term 2 (Jan–Apr 2024)

Last time: Empirical Bayes

- MLE can do weird things
  - Might pick highly "unlikely" model that exactly fits training data
- MAP helps by adding a prior, but still commits to one parameter

- Bayesian inference makes optimal decisions if your likelihood/prior are "correct"
  - "Right thing to do" if the model (prior + likelihood) is good
  - Computation can be tough: today's topic!
- Empirical Bayes uses data to find a good prior, $\arg\max_\alpha p(\mathbf{X} \mid \alpha)$
  - Tends to be less sensitive to overfitting than normal MLE
  - Compared to cross-validation: can be easier to compute, no data splitting
  - Can still overfit; it's just MLE in a "less sensitive" model!

- But maybe we should use a hyper-prior to pick good hyper-parameters...
  - Computation can be **really** tough

Overview of Bayesian Inference Tasks

- Bayesian inference requires computing expectations with respect to posterior,

$$\mathbb{E}[f(\theta)] = \int_\theta f(\theta)\, p(\theta \mid x) \mathrm{d}\theta$$

  - If $f(\theta) = \theta$, we get posterior mean of θ
  - If $f(\theta) = p(\tilde{x} \mid \theta)$, we get posterior predictive
  - If $f(\theta) = \mathbb{1}(\theta \in S)$ we get probability of S (e.g., marginals or conditionals)
  - If $f(\theta) = 1$ and we use $\tilde{p}(\theta \mid x)$ instead of $p(\theta \mid x)$, we get marginal likelihood

- But posterior often doesn't have a closed-form expression
  - Bayesian linear regression – $w \sim \mathcal{N}(m, v)$; $y \mid x, w \sim \mathcal{N}(w^\top x, \sigma^2)$ – does
  - Bayesian logistic regression – change to $p(y \mid x, w) = \frac{1}{1+\exp(-y\, w^\top x)}$ – doesn't
  - More complex models almost never do

- Our two main tools for approximate inference:
  1. Monte Carlo methods
  2. Variational methods
- Classic ideas from statistical physics that revolutionized Bayesian stats

Approximate Inference

Two main strategies for approximate inference:

1. Monte Carlo methods:
   - Approximate p with empirical distribution over samples,

$$p(x) \approx \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}(x^{(i)} = x)$$

   - Turns inference into sampling
2. Variational methods:
   - Approximate p with "closest" distribution q from a tractable family,

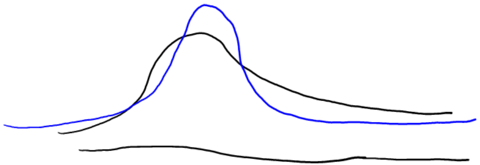
$$p(x) \approx q(x)$$

     - Gaussian, product of Bernoulli, any other model with easy inference. . . .
   - Turns inference into optimization

Outline

Variational Inference Illustration

- Approximate non-Gaussian p by a Gaussian q:



- Variational methods try to find simple distribution q that is closest to target p
- Unlike Monte Carlo, does not converge to true solution
  - A Gaussian may not be able to perfectly model posterior
- Variational methods quickly give an approximate solution
  - Sometimes all we need
  - Sometimes, approximation is better than any reasonable amount of Monte Carlo!

Laplace Approximation

- The classic, simplest variational method is the Laplace approximation
  1. Find an x that maximizes $p(x)$,

$$x^* \in \arg\min_x \{-\log p(x)\}$$

  2. Compute second-order Taylor expansion of $f(x) = -\log p(x)$ at x^*

$$-\log p(x) \approx f(x^*) + \underbrace{\nabla f(x^*)}_{0}^{\mathsf{T}}(x - x^*) + \tfrac{1}{2}(x - x^*)^{\mathsf{T}}\, \nabla^2 f(x^*)\,(x - x^*)$$

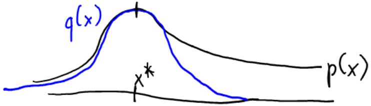
  3. Use distribution q that has this $-\log q(x)$ everywhere:

$$-\log q(x) = f(x^*) + \frac{1}{2}(x - x^*)\nabla^2 f(x^*)(x - x^*)$$

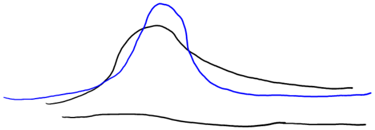
  This means the distribution q is exactly $\mathcal{N}(x^*, [\nabla^2 f(x^*)]^{-1})$
    - Same approximation as used by Newton's method in optimization

Laplace Approximation

- Laplace approximation replaces a complicated p with a Gaussian q
  - Centered at the mode, and agrees with 1st/2nd-derivatives of log-likelihood there:



  - In the $n \to \infty$ limit, "nicely behaved" posteriors are asymptotically normal
    - Bernstein-von Mises theorem
- Now you only need to compute Gaussian integrals (linear algebra for many f)
  - Very fast: just maximize + find one Hessian (compared to super-slow Monte Carlo)
  - Bad approximation if posterior is heavy-tailed, multi-modal, skewed, etc
- It might not even give you the "best" Gaussian approximation:



- We'll discuss fancier variational methods later in the course

Outline

Motivating problem: Bayesian Logistic Regression

- A classic way to fit a binary classifier is L2-regularized logistic loss,

$$\hat{w} \in \arg\max_{w} \sum_{i=1}^{n} \log(1 + \exp(-y^{(i)}\, w^\mathsf{T} x^{(i)})) + \frac{\lambda}{2}\|w\|^2$$

- This corresponds to using a sigmoid likelihood and Gaussian prior,

$$p(y \mid x, w) = \frac{1}{1 + \exp(-y\, w^\mathsf{T} x)}, \quad w \sim \mathcal{N}\left(0, \frac{1}{\lambda}\mathbf{I}\right)$$

- In Bayesian logistic regression, we'd work with the posterior
  - But the posterior isn't Gaussian: so this isn't a conjugate prior
  - We don't have a nice expression for the posterior predictive or marginal likelihood
- Laplace approximation would use $\mathcal{N}(\hat{w}_{\mathsf{MAP}}, [\nabla^2 f(x^*)]^{-1})$
  - Not the correct distribution for finite n; will give a (somewhat) wrong answer

Motivation: Monte Carlo for Bayesian Logistic Regression

- Posterior predictive in Bayesian logistic regression has the form

$$p(\tilde{y} \mid \tilde{x}, \mathbf{X}, \mathbf{y}, \lambda) = \int_w p(\tilde{y} \mid \tilde{x}, w)\, p(w \mid \mathbf{X}, \mathbf{y}, \lambda)\, \mathrm{d}w$$
$$= \mathop{\mathbb{E}}_w \left[p(\tilde{y} \mid \tilde{x}, w) \mid \mathbf{X}, \mathbf{y}, \lambda \right]$$

- Given w, we can compute $p(\tilde{y} \mid \tilde{x}, w) = 1/\left(1 + \exp\left(-\tilde{y}\, w^\mathsf{T} \tilde{x}\right)\right)$ just fine
- If we could sample from the posterior for w, we could estimate with Monte Carlo!
  - But we don't know how to generate IID samples from this posterior

- Soon, we'll cover MCMC, which is a standard method in scenarios like this

- But we'll start simpler: rejection sampling and importance sampling
- These methods assume you can generate from a simple distribution q
  - for example, a Gaussian
- but you really want to solve an integral for a complicated distribution p
  - for example, the posterior for Bayesian logistic regression

Rejection Sampling for Conditionals

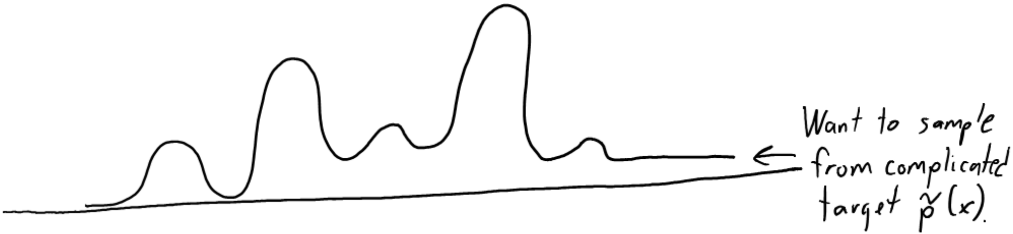
- We already mentioned rejection sampling for conditional sampling:
  - Example: sampling from a Gaussian conditional on knowing $x \in [-1, 1]$

  - Generate Gaussian samples, throw out ("reject") the ones that aren't in $[-1, 1]$
  - The remaining samples will follow the conditional distribution

- Can be used to generate IID samples from conditional distributions

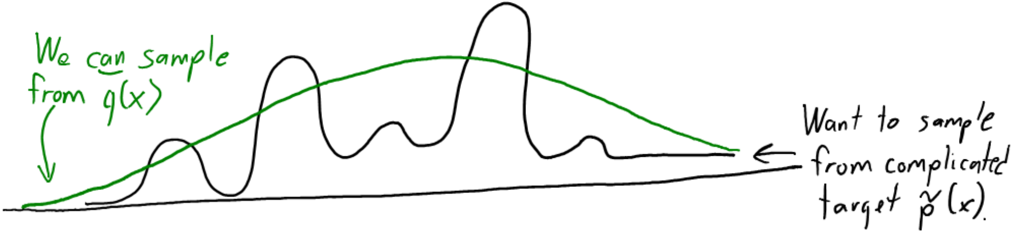
General Rejection Sampling Algorithm

- General rejection sampling algorithm tries to "sample area under the graph":



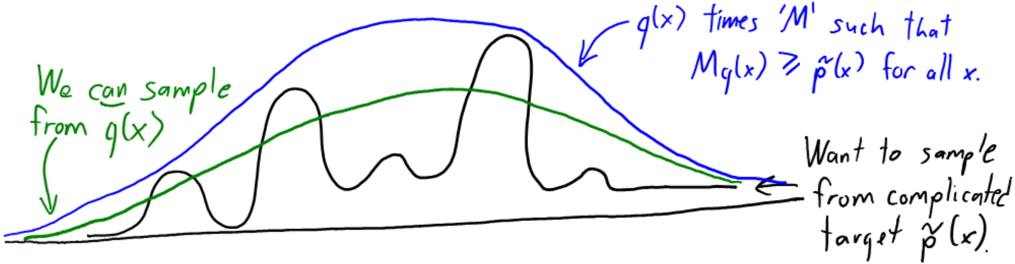
General Rejection Sampling Algorithm

- General rejection sampling algorithm tries to "sample area under the graph":



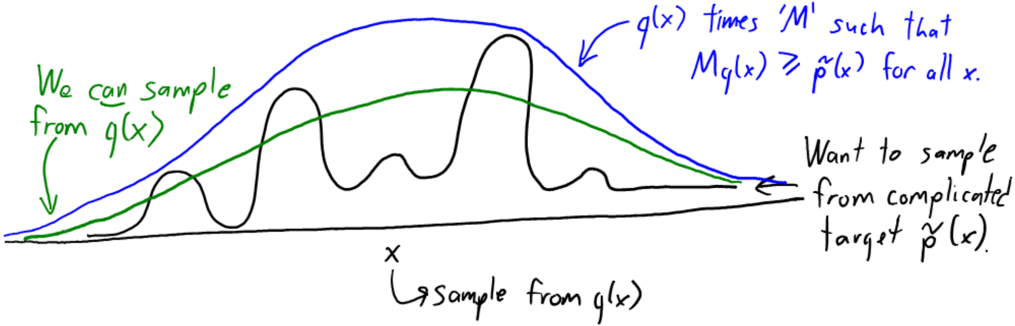
General Rejection Sampling Algorithm

- General rejection sampling algorithm tries to "sample area under the graph":



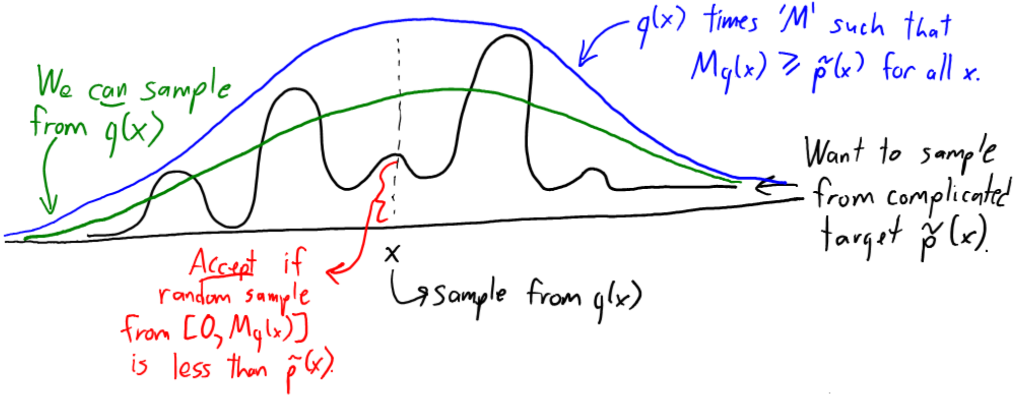
General Rejection Sampling Algorithm

- General rejection sampling algorithm tries to “sample area under the graph”:



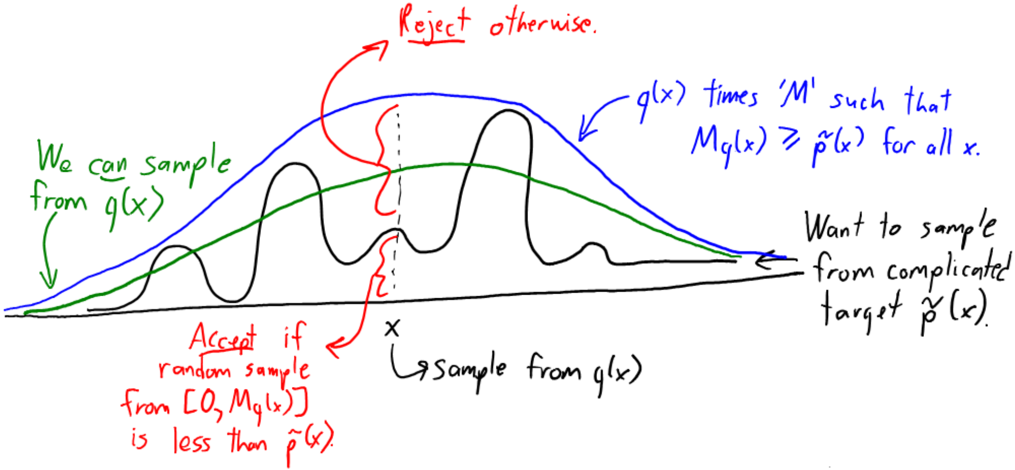
General Rejection Sampling Algorithm

- General rejection sampling algorithm tries to "sample area under the graph":



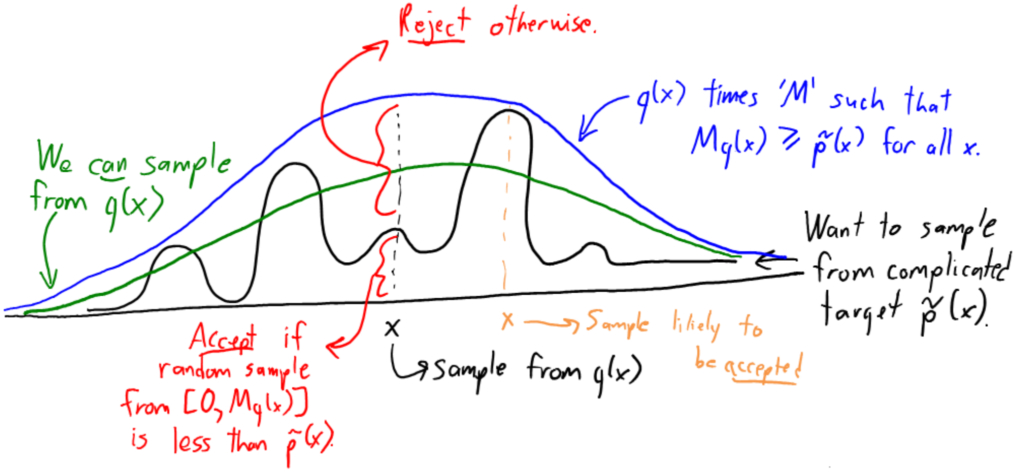
General Rejection Sampling Algorithm

- General rejection sampling algorithm tries to "sample area under the graph":



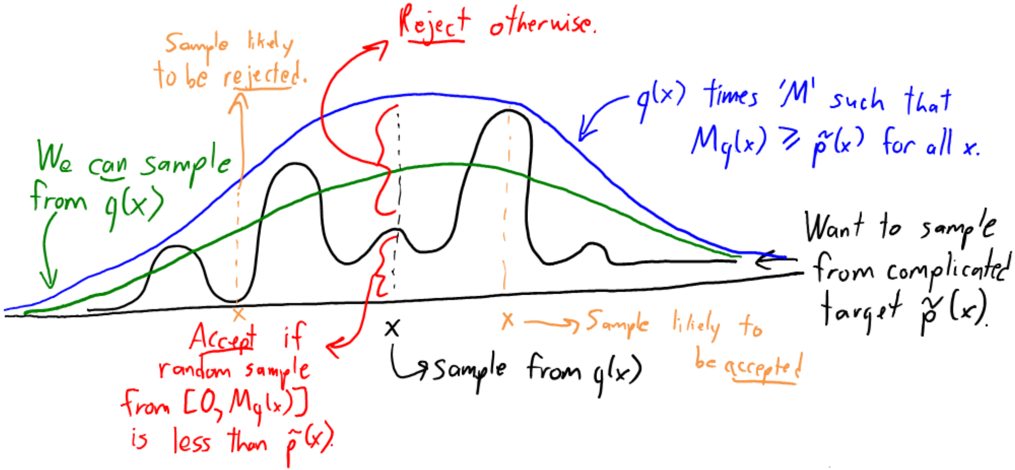
General Rejection Sampling Algorithm

- General rejection sampling algorithm tries to "sample area under the graph":



General Rejection Sampling Algorithm

- General rejection sampling algorithm tries to "sample area under the graph":



General Rejection Sampling Algorithm

- Ingredients of the general rejection sampling algorithm:
  1. Ability to evaluate an unnormalized $\tilde{p}(x)$, so that $p(x) = \tilde{p}(x)/Z$
  2. A distribution q that we can sample from
  3. An upper bound M on $\tilde{p}(x)/q(x)$

- Rejection sampling algorithm:
  1. Sample x from $q(x)$
  2. Keep the sample with probability $\tilde{p}(x)/(Mq(x))$:
     - Sample u from $\text{Unif}([0,1])$, keep the sample if $u \leq \tilde{p}(x) \,/\, (Mq(x))$
- The accepted samples will be from $p(x)$, as long as M is a valid upper bound

- Then can use the accepted samples in Monte Carlo:

$$\mathop{\mathbb{E}}_{x \sim p} f(x) \approx \frac{1}{\sum_{i=1}^{m} \mathbb{1}\left(\text{accepted } x^{(i)}\right)} \sum_{i=1}^{m} \mathbb{1}\left(\text{accepted } x^{(i)}\right) f\left(x^{(i)}\right)$$

General Rejection Sampling Algorithm

- For Bayesian logistic regression, we could propose samples from the prior:

$$\tilde{p}(w \mid \mathbf{X}, \mathbf{y}) = p(\mathbf{y} \mid \mathbf{X}, w)\, p(w) \qquad\qquad q(w) = p(w)$$

$$\frac{\tilde{p}(w \mid \mathbf{y}, \mathbf{X})}{q(w)} = \frac{p(\mathbf{y} \mid \mathbf{X}, w)p(w)}{p(w)} = p(\mathbf{y} \mid \mathbf{X}, w) \leq 1$$

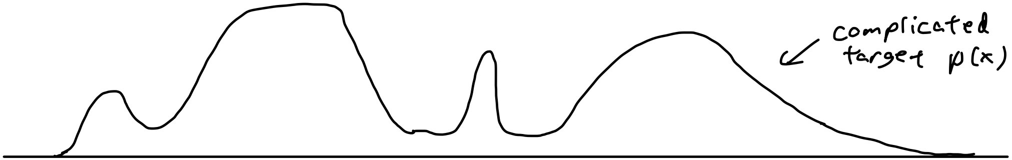
  - Recall $\mathbf{y}$ is discrete here, so $p(\mathbf{y} \mid \mathbf{X}, w) \leq 1$: we can use $M = 1$
  - w sampled from prior would tend to be kept if they explain the data well

- Drawbacks of rejection sampling:
  - You need to know a bound M on $\tilde{p}(x)/q(x)$ (may be hard/impossible to find)
    - If x is unbounded and p has heavier tails than q, no M exists
  - You may reject a large number of samples
    - Most samples are rejected for high-dimensional complex distributions, or if q is bad

Outline

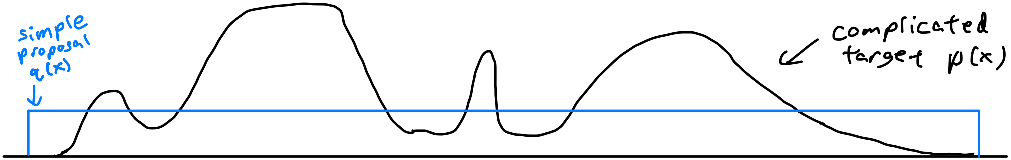
Alternate approach: importance sampling

- Instead of rejection, importance sampling re-weights q samples to look like p



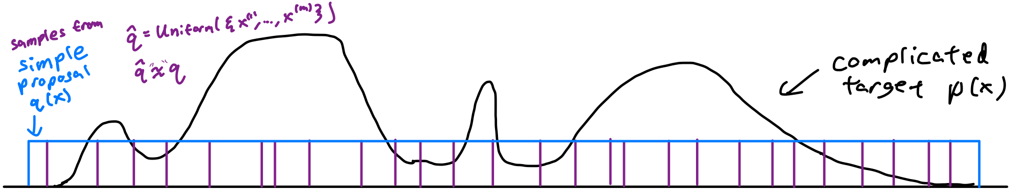
Alternate approach: importance sampling

- Instead of rejection, importance sampling re-weights q samples to look like p



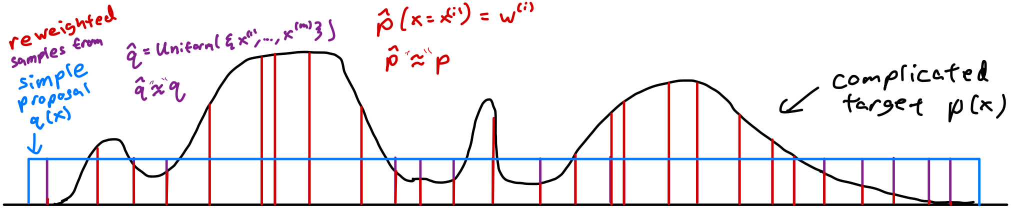
Alternate approach: importance sampling

- Instead of rejection, importance sampling re-weights q samples to look like p



Alternate approach: importance sampling

- Instead of rejection, importance sampling re-weights q samples to look like p



Alternate approach: importance sampling

- Instead of rejection, importance sampling re-weights q samples to look like p

- Derivation:

$$\begin{aligned}
\mathop{\mathbb{E}}_{x\sim p}[f(x)] &= \int p(x)f(x)\,\mathrm{d}x \\
&= \int q(x)\frac{p(x)}{q(x)}f(x)\,\mathrm{d}x \\
&= \mathop{\mathbb{E}}_{x\sim q}\left[\frac{p(x)}{q(x)}f(x)\right] \approx \frac{1}{n}\sum_{i=1}^{n}\frac{p(x^{(i)})}{q(x^{(i)})}f(x^{(i)}),
\end{aligned}$$

  using a Monte Carlo approximation with IID samples from q
  - Replace integral with a sum for discrete distributions
- We can sample from q, but reweight by $p(x)/q(x)$ to compute expectation
- Only assumption is that for all x with nonzero p, q is also nonzero

Self-Normalized Importance Sampling

- What if we only have $\tilde{p}$, with $p(x) = \tilde{p}(x)/Z$?

$$\mathop{\mathbb{E}}_{x\sim p}[f(x)] = \int p(x) f(x)\,\mathrm{d}x = \frac{1}{Z}\int q(x)\frac{\tilde{p}(x)}{q(x)} f(x)\,\mathrm{d}x$$

$$= \frac{\mathbb{E}_{x\sim q}\left[\frac{\tilde{p}(x)}{q(x)} f(x)\right]}{\int \tilde{p}(x)\,\mathrm{d}x} = \frac{\mathbb{E}_{x\sim q}\left[\frac{\tilde{p}(x)}{q(x)} f(x)\right]}{\int q(x)\frac{\tilde{p}(x)}{q(x)}\,\mathrm{d}x} = \frac{\mathbb{E}_{x\sim q}\left[\frac{\tilde{p}(x)}{q(x)} f(x)\right]}{\mathbb{E}_{x\sim q}\left[\frac{\tilde{p}(x)}{q(x)}\right]}$$

- Can use Monte Carlo estimator based on m samples from q:

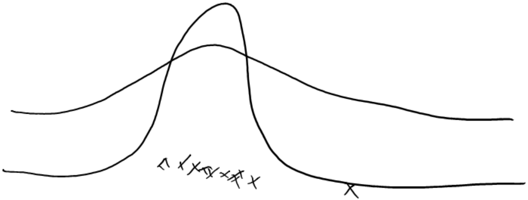
$$\mathop{\mathbb{E}}_{x\sim p}[f(x)] \approx \frac{\frac{1}{n}\sum_{i=1}^{m}\frac{\tilde{p}(x^{(i)})}{q(x^{(i)})} f(x^{(i)})}{\frac{1}{m}\sum_{i=1}^{m}\frac{\tilde{p}(x^{(i)})}{q(x^{(i)})}}$$

  - Weighted mean, normalized by $\tilde{p}(x^{(i)})/q(x^{(i)})$
  - Biased estimator: $\mathbb{E}\,\frac{1}{\hat{Z}} > \frac{1}{Z}$ for non-constant distributions (Jensen's inequality)

Importance Sampling

- Importance sampling is only efficient if q is close to p
- Otherwise, weights will be huge for a small number of samples
  - Even though unbiased, variance can be huge

- Can be problematic if q has lighter "tails" than p:
  - You rarely sample the tails, so those samples get huge weights



- As with rejection sampling, does not tend to work well in high dimensions
  - There's room, though, to cleverly design q
    - e.g. "alternate between sampling two Gaussians with different variances"

Summary

- Laplace approximation: simple way to find a Gaussian approximation to posterior
  - Fast and easy, but not always accurate
- Rejection sampling: generate exact samples from complicated distributions
  - Tends to reject too many samples in high dimensions
- Importance sampling: re-weights samples from the wrong distribution
  - Tends to have high variance in high dimensions

- Next time: all in the (exponential) family