# CPSC 440: Advanced Machine Learning

## Rejection/Importance Sampling

Mark Schmidt

University of British Columbia

Winter 2021

# Overview of Bayesian Inference Tasks

- In Bayesian approach, we typically work with the posterior

$$p(\theta \mid x) = \frac{1}{Z} p(x \mid \theta) p(\theta),$$

  where $Z$ makes the distribution sum/integrate to $1$.

- Typically, we need to compute expectation of some $f$ with respect to posterior,

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

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

# Need for Approximate Integration

- Bayesian models allow things that aren't possible in other frameworks:
  - Optimize the regularizer (empirical Bayes).
  - Relax IID assumption (hierarchical Bayes).
  - Have clustering happen on multiple levels (topic models).
- But posterior often doesn't have a closed-form expression.
  - We don't just want to flip coins and multiply Gaussians.
- We once again need approximate inference:
  1. Variational methods.
  2. Monte Carlo methods.
- Classic ideas from statistical physics, that revolutionized Bayesian stats/ML.

# Variational Inference vs. Monte Carlo

Two main strategies for approximate inference:

1. Variational methods:
   - Approximate $p$ with "closest" distribution $q$ from a tractable family,

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

   - Turns inference into optimization (need to find best $q$).
     - Called variational Bayes.
2. Monte Carlo methods:
   - Approximate $p$ with empirical distribution over samples,

$$p(x) \approx \frac{1}{n} \sum_{i=1}^{n} \mathcal{I}[x^i = x].$$

   - Turns inference into sampling.
     - For Bayesian methods, we'll typically need to sample from posterior.

# Conjugate Graphical Models: Ancestral and Gibbs Sampling

- For conjugate DAGs, we can use ancestral sampling for unconditional sampling.
  - By using inverse transform to sample 1D conditionals.
- Examples:
  - For Markov chains, sample $x_1$ then $x_2$ and so on.
  - For HMMs, sample the hidden $z_j$ then sample the $x_j$.
  - For LDA, sample $\pi$ then sample the $z_j$ then sample the $x_j$.
- We can also often use Gibbs sampling as an approximate sampler.
  - If neighbours are conjugate in UGMs.
  - To generate conditional samples in conjugate DAGs.
- However, without conjugacy our inverse transform trick doesn't work.
  - We can't even sample from the 1D conditionals with this method.

# Beyond Inverse Transform and Conjugacy

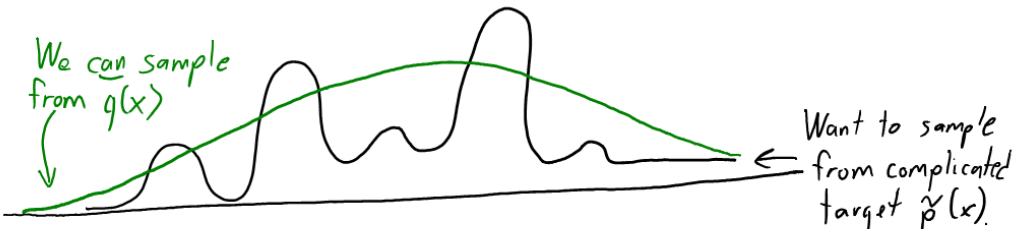
- We want to use simple distributions to sample from complex distributions.
  - Two common strategies are rejection sampling and importance sampling.

- We've previously seen rejection sampling to do conditional sampling:
  - Example: sampling from a Gaussian subject to $x \in [-1, 1]$.

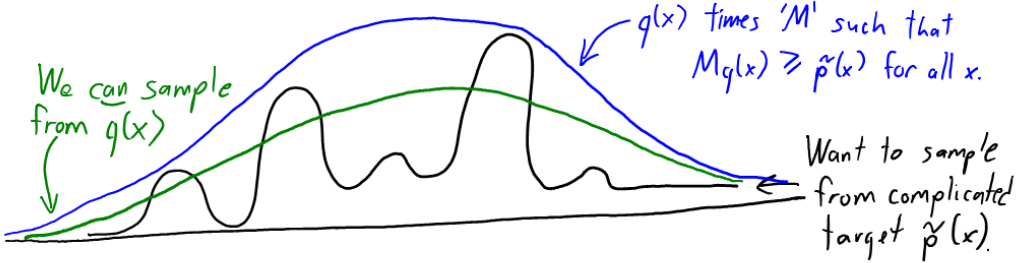

  - Generate unconditional samples, throw out ("reject") the ones that aren't in $[-1, 1]$.

# General Rejection Sampling Algorithm

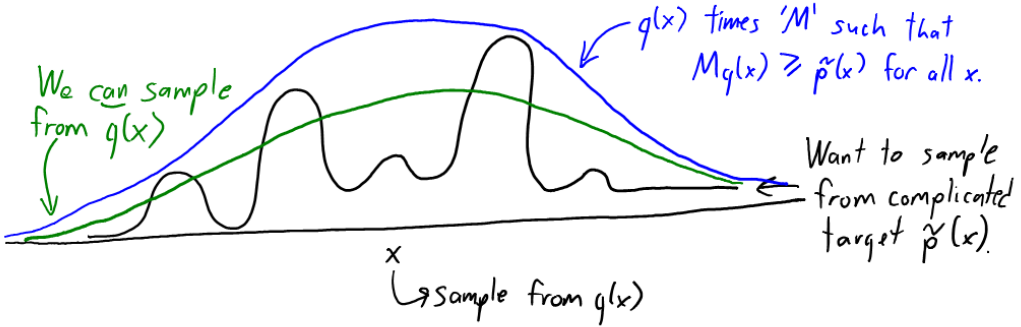
# General Rejection Sampling Algorithm



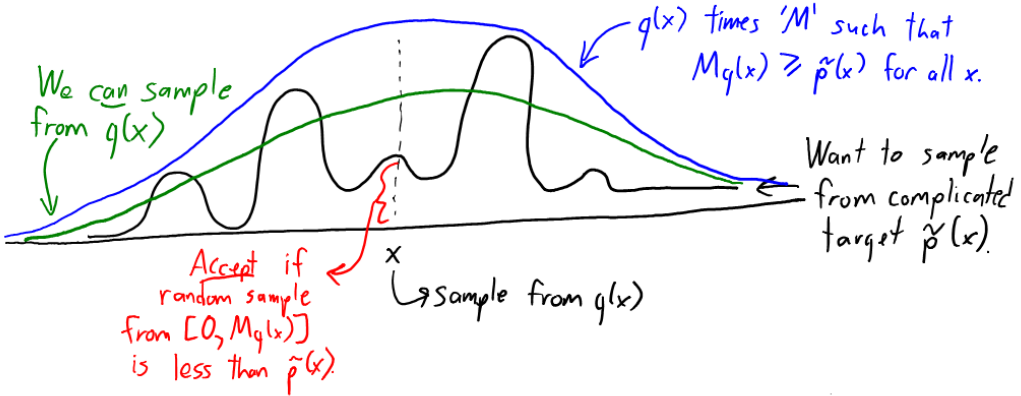
# General Rejection Sampling Algorithm



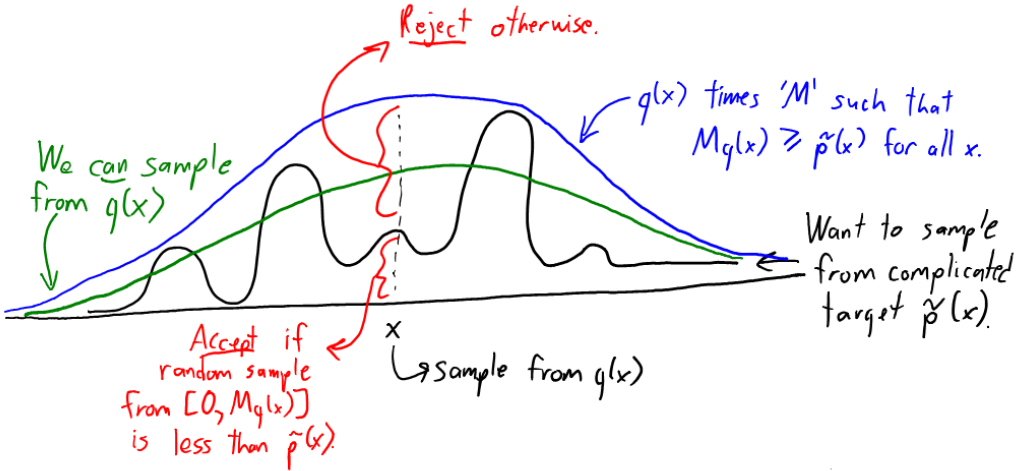
# General Rejection Sampling Algorithm



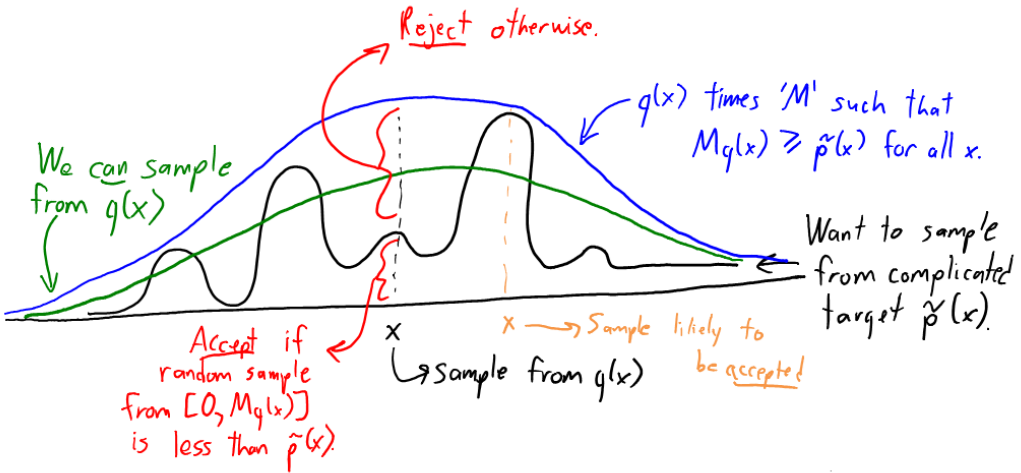
# General Rejection Sampling Algorithm



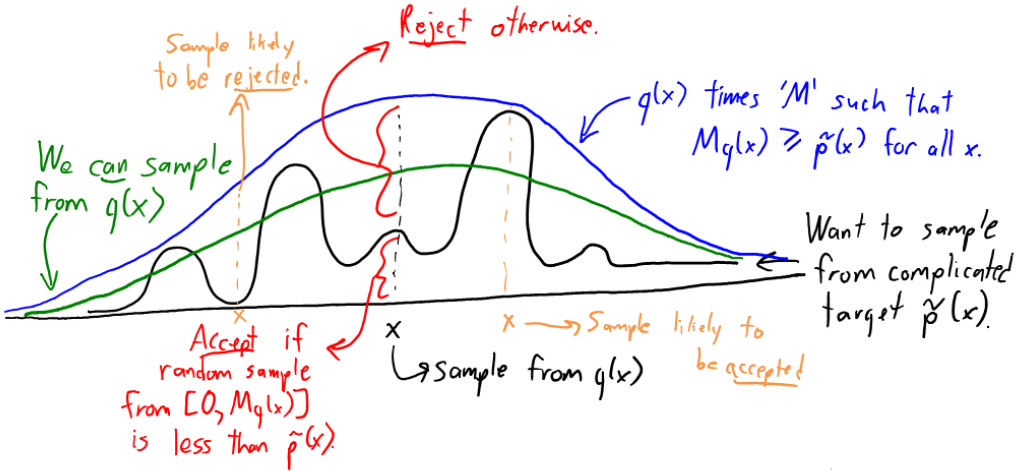
# General Rejection Sampling Algorithm



# General Rejection Sampling Algorithm



# General Rejection Sampling Algorithm



# General Rejection Sampling Algorithm

- Ingredients of a more general rejection sampling algorithm:
  1. Ability to evaluate unnormalized $\tilde{p}(x)$,

$$p(x) = \frac{\tilde{p}(x)}{Z}.$$

  2. A distribution $q$ that is easy to sample from.
  3. An upper bound $M$ on $\tilde{p}(x)/q(x)$.

- Rejection sampling algorithm:
  1. Sample $x$ from $q(x)$.
  2. Sample $u$ from $\mathcal{U}(0, 1)$.
  3. Keep the sample if $u \leq \frac{\tilde{p}(x)}{Mq(x)}$.
- The accepted samples will be from $p(x)$.

# General Rejection Sampling Algorithm

- We can use general rejection sampling for:
  - Sample from Gaussian $q$ to sample from student t.
  - Sample from prior to sample from posterior ($M = 1$ for discrete $x$),

$$\tilde{p}(\theta \mid x) = \underbrace{p(x \mid \theta)}_{\leq 1} p(\theta).$$

- Drawbacks:
  - You may reject a large number of samples.
    - Most samples are rejected for high-dimensional complex distributions.
  - You need to know $M$.

- If $-\log p(x)$ is convex and $x$ is 1D there is a fancier version:
  - Adaptive rejection sampling refines piecewise-linear $q$ after each rejection.

# Importance Sampling

- Importance sampling is a variation that accepts all samples.
  - Key idea is similar to EM analysis,

$$\begin{aligned}\mathbb{E}_p[f(x)] &= \sum_x p(x)f(x) \\ &= \sum_x q(x)\frac{p(x)f(x)}{q(x)} \\ &= \mathbb{E}_q\left[\frac{p(x)}{q(x)}f(x)\right] \approx \frac{1}{n}\sum_{i=1}^{n}\frac{p(x)}{q(x)}f(x),\end{aligned}$$

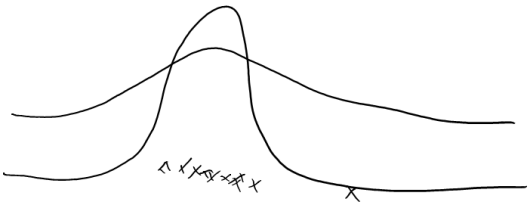
    where the Monte Carlo approximation uses samples from $q$.
    - Replace sum over $x$ with integral for continuous distributions.

- We can sample from $q$ but reweight by $p(x)/q(x)$ to sample from $p$.
- Only assumption is that $q$ is non-zero when $p$ is non-zero.
- If you only know unnormalized $\tilde{p}(x)$, a variant gives approximation of $Z$.

# Importance Sampling

- As with rejection sampling, 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, doesn't tend to work well in high dimensions.
  - Though there is room to cleverly design $q$, like using mixtures.
  - For example, $q$ could sample from mixture of Gaussians with different variances.

# Outline

# Limitations of Simple Monte Carlo Methods

- The basic ingredients of our previous sampling methods:
  - Sampling in low dimensions: Inverse CDF, rejection sampling, importance sampling.
  - Sampling in higher dimensions: ancestral sampling, Gibbs sampling.
- These work well in low dimensions or for posteriors with analytic properties.

- But we want to solve high-dimensional integration problems in other settings:
  - Deep belief networks and Boltzmann machines.
  - Bayesian graphical models and Bayesian neural networks.
  - Hierarchical Bayesian models.

- Our previous methods tend not to work in complex situations:
  - Inverse CDF may not be available.
  - Conditionals needed for ancestral/Gibbs sampling may be hard to compute.
  - Rejection sampling tends to reject almost all samples.
  - Importance sampling tends to give almost zero weight to all samples.

# Dependent-Sample Monte Carlo Methods

- We want an algorithm whose samples get better over time.
- Two main strategies for generating dependent samples:
  - Sequential Monte Carlo:
    - Importance sampling where proposal $q_t$ changes over time from simple to posterior.
    - AKA sequential importance sampling, annealed importance sampling, particle filter.
    - Usual application: Markov and HMM models with continuous non-Gaussian states.
    - "Particle Filter Explained without Equations":
      `https://www.youtube.com/watch?v=aUkBa1zMKv4`
  - Markov chain Monte Carlo (MCMC).
    - Design Markov chain whose stationary distribution is the posterior.
- These are the main tools to sample from high-dimensional distributions.

# Markov Chain Monte Carlo

- We've previously discussed Markov chain Monte Carlo (MCMC).
  1. Based on generating samples from a Markov chain $q$.
  2. Designed so stationary distribution $\pi$ of $q$ is target distribution $p$.
- If we run the chain long enough, it gives us samples from $p$.
- Gibbs sampling is an example of an MCMC method.
  - Sample $x_j$ conditioned on all other variables $x_{-j}$.
- Note that before we were sampling states according to a UGM, in Bayesian models we're sampling parameters according to the posterior.
  - But we use the same methods for both tasks.

# Limitations of Gibbs Sampling

- Gibbs sampling is nice because it has no parameters:
  - You just need to decide on the blocks and figure out the conditionals.
- But it isn't always ideal:
  - Samples can be very correlated: slow progress.
  - Conditionals may not have a nice form:
    - If Markov blanket is not conjugate, need rejection sampling (or numerical CDF).
- Generalization that can address these is Metropolis-Hastings:
  - Oldest algorithm among the "10 Best of the 20th Century".

# Warm-Up to Metropolis-Hastings: "Stupid MCMC"

- Consider finding the expected value of a fair di:
  - For a 6-sided di, the expected value is 3.5.
- Consider the following "stupid MCMC" algorithm:
  - Start with some initial value, like "4".
  - At each step, roll the di and generate a random number $u$:
    - If $u < 0.5$, "accept" the roll and take the roll as the next sample.
    - Othewise, "reject" the roll and take the old value ("4") as the next sample.

# Warm-Up to Metropolis-Hastings: "Stupid MCMC"

- Example:
  - Start with "4", so record "4".
  - Roll a "6" and generate 0.234, so record "6".
  - Roll a "3" and generate 0.612, so record "6".
  - Roll a "2" and generate 0.523, so record "6".
  - Roll a "3" and generate 0.125, so record "3".

- So our samples are 4,6,6,6,3,...
  - If you run this long enough, you will spend 1/6 of the time on each number.
  - So the dependent samples from this Markov chain could be used within Monte Carlo.

- It is "stupid" since you should just accept every sample (they are IID samples).
  - It works but it is twice as slow.

# A Simple Example of Metropolis-Hastings

- Consider a loaded di that rolls a 6 half the time (all others equally likely).
  - So $p(x = 6) = 1/2$ and $p(x = 1) = p(x = 2) = \cdots = p(x = 5) = 1/10$.

- Consider the following "less stupid" MCMC algorithm:
  - At each step, we start with an old state $x$.
  - Generate a random number $\hat{x}$ uniformly between 1 and 6 (roll a fair di), and generate a random number $u$ in the interval $[0, 1]$.
  - "Accept" this roll if
$$u < \frac{p(\hat{x})}{p(x)}.$$
  - So if we roll $\hat{x} = 6$, we accept it: $u < 1$ ("always move to higher probability").
  - If $x = 2$ and roll $\hat{x} = 1$, accept it: $u < 1$ ("always move to same probability").
  - If $x = 6$ and roll $\hat{x} = 1$, we accept it with probability $1/5$.
    - We prefer high probability states, but sometimes move to low probability states.

- This has right probabilities as the stationary distribution (not yet obvious).
  - And accepts most samples.

# Summary

- Rejection sampling: generate exact samples from complicated distributions.
  - Tends to reject too many samples in high dimensions.
- Importance sampling: reweights samples from the wrong distribution.
  - Tends to have high variance in high dimensions.
- Markov chain Monte Carlo generates a sequence of *dependent samples*:
  - But asymptotically these samples look like they come from the posterior.
- Next time: we actually start deep learning?