CPSC 540: Machine Learning
Rejection/Importance Sampling

Mark Schmidt

University of British Columbia

Winter 2020
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 \( f(\theta) = p(\hat{x} \mid \theta) \), 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,
     \[ p(x) \approx \frac{1}{n} \sum_{i=1}^{n} I[x^i = x]. \]
   - Turns inference into sampling.
     - For Bayesian methods, we’ll typically need to sample from posterior.
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

Want to sample from complicated target \( \gamma(x) \).
General Rejection Sampling Algorithm

We can sample from $q(x)$

Want to sample from complicated target $p(x)$. 
We can sample from \( q(x) \) \( g(x) \) times 'M' such that \( Mq(x) \geq \hat{p}(x) \) for all \( x \).

Want to sample from complicated target \( \hat{p}(x) \).
General Rejection Sampling Algorithm

We can sample from \( g(x) \)

\[ q(x) \text{ times } 'M' \text{ such that } Mq(x) \geq \hat{p}(x) \text{ for all } x. \]

Want to sample from complicated target \( \hat{p}(x) \).

\( \xrightarrow{\text{Sample from } g(x)} \)
General Rejection Sampling Algorithm

We can sample from \( g(x) \)

Accept if random sample from \([0, M g(x)]\) is less than \( \hat{p}(x) \)

\( g(x) \) times 'M' such that \( M g(x) \geq \hat{p}(x) \) for all \( x \).

Want to sample from complicated target \( \hat{p}(x) \).
General Rejection Sampling Algorithm

We can sample from \( g(x) \).

Accept if random sample from \( [0, M g(x)] \) is less than \( \tilde{p}(x) \).

\( \tilde{p}(x) \) times 'M' such that \( M g(x) \geq \tilde{p}(x) \) for all \( x \).

Want to sample from complicated target \( p(x) \).
General Rejection Sampling Algorithm

We can sample from $g(x)$.

Accept if random sample from $[0, M g(x)]$ is less than $\hat{p}(x)$.

X \rightarrow$ Sample likely to be accepted.

X \rightarrow$ Sample from $g(x)$.

Reject otherwise.

$g(x)$ times ‘$M$’ such that $M g(x) \geq \hat{p}(x)$ for all $x$.

Want to sample from complicated target $\hat{p}(x)$. 
General Rejection Sampling Algorithm

We can sample from \( g(x) \).

Sample likely to be rejected.

Accept if random sample from \([0, M g(x)]\) is less than \( \tilde{p}(x) \).

\( \tilde{p}(x) \) times 'M' such that \( M g(x) \geq \tilde{p}(x) \) for all \( x \).

Want to sample from complicated target \( \tilde{p}(x) \).

Sample likely to be accepted.
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 $\frac{\tilde{p}(x)}{q(x)}$.

Rejection sampling algorithm:

1. Sample $x$ from $q(x)$.
2. Sample $u$ from $U(0, 1)$.
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 $x$),
    \[
    \tilde{p}(\theta \mid x) = p(x \mid \theta) p(\theta) \leq 1
    \]

- 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,

  $$
  \mathbb{E}_p[f(x)] = \sum_x p(x)f(x)
  = \sum_x q(x) \frac{p(x)f(x)}{q(x)}
  = \mathbb{E}_q \left[ \frac{p(x)}{q(x)} f(x) \right],
  $$

  and similarly 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.
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.

- Back to MCMC.