MLRG: Basic Monte Carlo Methods

Tian Qi (Ricky) Chen

Feb 03, 2016
Overview

1. Monte Carlo Motivation
   - Law of Large Numbers

2. Generating Samples
   - Inverse Transform Sampling
   - Sampling Under the Curve
   - Rejection Sampling
     - Adaptive Rejection Sampling
   - Problems with Rejection Sampling
   - Ancestral Sampling

3. Monte Carlo Integration
   - Importance Sampling
   - Self-normalized Importance Sampling
   - Rao-Blackwellization
The Monte Carlo Method

Refers to the use of random samples to do (approximate) computations.

- Typical supervised learning $D_N = \{(x_i, y_i)\}$

$$
\text{posterior: } p(\theta|D_N) \propto p(\theta) \prod_{i=1}^{N} p(y_i|x_i, \theta)
$$

$$
\text{posterior predictive: } p(y|x, D_N) = \int p(y|x, \theta)p(\theta|D_N)d\theta
$$

- MAP:

$$
\hat{\theta} = \arg\max_{\theta} p(\theta|D_N), \quad p(y|x, D_N) \approx p(y|x, \hat{\theta})
$$

- Monte Carlo integration:

$$
\{\theta^s\}_{s=1}^{S} \overset{iid}{\sim} p(\theta|D_N), \quad p(y|x, D_N) \approx \frac{1}{S} \sum_{s=1}^{S} p(y|x, \theta^s)
$$
Theoretical Justification for Monte Carlo Integration

**Theorem (Strong Law of Large Numbers)**

If $X_1, \ldots, X_n \sim i.i.d. \pi$ with $\mathbb{E}[X_1] = \mu$, $|\mu| < \infty$ then

$$\frac{1}{n} \sum_{i=1}^{n} X_i \to \mu \text{ a.s.}$$

- Take leap of faith: $\frac{1}{n} \sum_{i=1}^{n} X_i \approx \mu$
- By definition of expectation: $\frac{1}{n} \sum_{i=1}^{n} X_i \approx \int x \pi(x) dx$
- More generally: $\frac{1}{n} \sum_{i=1}^{n} g(X_i) \approx \int g(x) \pi(x) dx$
Law of Large Numbers
Generating samples (1D)

- **Inverse Transform Sampling**
  - Want a sample $\theta \sim F$, where $F$ is the CDF.

**Inverse Transform Algorithm**

1. Sample $U \sim \text{Unif}(0,1)$.
2. Compute sample as $\theta = F^{-1}(U)$. 

![Graph showing N(0,1) CDF and N(0,1) Inverse CDF](image_url)
Generating samples (1D)

Suppose we only know the density function up to a normalizing constant.

\[ \pi(\theta) = \frac{\pi^*(\theta)}{Z} \]

e.g. \[ p(\theta|D_N) \propto p(\theta) \prod_{i=1}^{N} p(y_i|x_i, \theta) = \pi^*(\theta) \]

- Geometric interpretation of sampling: throwing darts at area under \( \pi^* \).
- Samples are generated in proportion to height of the curve.
Accept-Reject Methods

- Rejection Sampling
  - Requires a density $q$ such that $\pi^*(\theta) \leq kq(\theta)$.
  - Area under $\pi^*$ is still uniformly sampled, but must retry if the sample is above the curve.
Accept-Reject Methods

- Rejection Sampling
  - Requires a density $q$ such that $\pi^*(\theta) \leq kq(\theta)$.

Rejection Sampling Algorithm

1. Sample $Y \sim q$, $U \sim \text{Unif}(0,1)$
2. Accept $\theta = Y$ if $U \leq \pi^*(Y)/kq(Y)$
3. Otherwise, retry.
Example 1: Computing $Z$ with Rejection Sampling

Suppose we have a half-unit circle as our density.

We can get the area under the function from rejection sampling.

Fraction of samples under the curve converges to $\frac{A}{2}$, where $A = \frac{\pi}{2}$. 
Example 2: Sampling from posterior using prior

We have in supervised setting with discrete random variables:

\[ p(\theta|D_N) \propto p(\theta) \prod_{i=1}^{N} p(y_i|x_i, \theta) \leq p(\theta) \leq 1 \]

So we can do rejection sampling with

\[ \pi^* = p(\theta) \prod_{i=1}^{N} p(y_i|x_i, \theta) \]

Using \( p(\theta) \) as the upper bound.
Accept-Reject Methods

- Envelope Rejection Sampling
  - Require additional lower bound: \( g(\theta) \leq \pi^*(\theta) \leq kq(\theta) \).
  - Useful when \( g \) is easier to compute than \( \pi^* \).

Envelope Accept-Reject Algorithm

1. Sample \( Y \sim q, \; U \sim \text{Unif}(0, 1) \)
2. Accept \( \theta = Y \) if \( U \leq g(Y)/kq(Y) \);
   otherwise, accept \( \theta = Y \) if \( U \leq \pi^*(Y)/kq(Y) \)
   otherwise, retry.
Accept-Reject Methods

- Adaptive Rejection Sampling
  - Requires \( h = \log \pi^* \) to be a concave function.
  - Adaptively constructs the upper and lower bounds using only evaluations of \( \pi^* \).

Adaptive Bounds

Let \( S_n = \{x_i\}_{i=1}^n \) be a set of points in the support of \( \pi^* \) where \( x_i < x_{i+1} \).

Let \( \ell_i \) be the line through \( (x_i, h(x_i)) \) and \( (x_{i+1}, h(x_{i+1})) \).

Then \( \ell_i \) is below \( h \) in \([x_i, x_{i+1}]\) and above \( h \) outside this interval.
Accept-Reject Methods

- Adaptive Rejection Sampling
  - For $x \in [x_i, x_{i+1}]$, if we define

$$
\overline{h}_n(x) = \min\{\ell_{i-1}(x), \ell_{i+1}(x)\} \quad \text{and} \quad h_n(x) = \ell_i(x)
$$

Then the envelopes are

$$
\overline{h}_n(x) \leq h(x) \leq \overline{h}_n(x)
$$
Accept-Reject Methods

- Adaptive Rejection Sampling
  - The envelopes for the log-density are $h_n(x) \leq h(x) \leq \bar{h}_n(x)$
  - Therefore, for $f_n(\theta) := \exp(h_n(\theta))$ and $\bar{f}_n(\theta) := \exp(\bar{h}_n(\theta))$
    \[
    f_n(\theta) \leq \pi^*(\theta) \leq \bar{f}_n(x) =: Zq_n(\theta)
    \]

Where $q_n$ is a density.

- $q_n$ is piecewise exponential and can be sampled using two steps. (stratified sampling method)
  - Sample from multinomial distribution to determine a "piece".
  - Sample from the truncated exponential distribution.
Problems with Rejection Sampling

- Accept-Reject methods do not scale well with dimensions due to curse of dimensionality. (The ARS algorithm only works in 1 dimensions.)
  - Many multivariate sampling problems can be decomposed into univariate sampling steps. (e.g. acyclic belief networks)
  - Gibbs sampling (MCMC) uses only univariate sampling steps.
  - But many other Monte Carlo methods can be used to tackle the problem of “rare event simulation”, such as importance sampling.
- Accept-Reject methods require the knowledge of an upper bound $kq(\theta)$.
  - Importance Sampling has a weaker requirement.
Ancestral Sampling

Here’s a brief mention of ancestral sampling.

- Suppose we have a Bayesian network (directed acyclic).
- We can sample from the joint distribution using chain rule

\[
p(X_1, \ldots, X_n) = p(X_1)p(X_2|X_1)p(X_3|X_2, X_1) \cdots p(X_n|X_{n-1}, \ldots, X_1)
\]

\[
p(X) = \prod_i p(X_i|\text{parents}(X_i))
\]

(Not very useful if we want to condition on some observations.)
Monte Carlo Integration - Importance Sampling

- Back to the law of large numbers.
  - Using samples $X_i \overset{iid}{\sim} \pi$, we can estimate any integral by putting it in the form of $\mathbb{E}[g(X)]$ for any function $g$.

$$\frac{1}{n} \sum_{i=1}^{n} g(X_i) \approx \int g(x)\pi(x)dx$$

But $\pi(x)$ may be difficult to analyze.

- Idea: sample $Y_i$ from a different (biasing) distribution with density $f$ and add weights to the samples based on how likely this sample comes from $\pi(x)$.

$$\frac{1}{n} \sum_{i=1}^{n} g(Y_i) \frac{\pi(Y_i)}{f(Y_i)} \approx \int g(x)\frac{\pi(x)}{f(x)}f(x)dx = \int g(x)\pi(x)dx$$

- Importance Sampling only requires that $f(x) > 0$ whenever $g(x)\pi(x) \neq 0$. 
Self-normalized Importance Sampling

- What if we only know $\pi^*$?
  - Then $\frac{1}{n} \sum_{i=1}^{n} g(Y_i) \frac{\pi^*(Y_i)}{f(Y_i)} \approx Z \int g(x)\pi(x)dx$
  - We can construct an estimator for $Z$

\[
\frac{1}{n} \sum_{i=1}^{n} \frac{\pi^*(Y_i)}{f(Y_i)} \approx \int \frac{Z\pi(x)}{f(x)} f(x)dx = Z
\]

- Thus...

\[
\frac{1}{n} \sum_{i=1}^{n} g(Y_i) \frac{\pi^*(Y_i)}{f(Y_i)} \approx \int g(x)\pi(x)dx
\]

- Note: $f$ can also be un-normalized.
- Requires slightly stronger condition: $f(x) > 0$ whenever $\pi(x) > 0$.
- Cannot be said to be unbiased.
Rao-Blackwellization

- What if we only cared about $\mathbb{E}[h(X)]$ when our sampling method produces $(X, Y)$? Naive method is to throw out $Y$.
- eg. $Y$ are samples from $q$ in rejection sampling and $X$ are samples that pass the acceptance step. (note $X$ depends on $Y$ and some other r.v.’s)
- Rao-Blackwellization is a method to produce a lower-variance estimator by reducing the number of random variables that an estimator depends on.

**Theorem (Law of Total Variance)**

$\text{Var}(\delta) = \mathbb{E}[\text{Var}(\delta|Y)] + \text{Var}(\mathbb{E}[\delta|Y])$

$\implies \text{Var}(\delta) \geq \text{Var}(\mathbb{E}[\delta|Y])$

- If $\mathbb{E}[\delta]$ is the quantity we wish to approximate, then we can use $\mathbb{E}[\delta|Y]$ instead of $\delta$ to produce a better approximator.
- * If $\delta$ is a function of $Y$ plus some other random variables, then computing $\mathbb{E}[\delta|Y]$ is equivalent to marginalizing out the other random variables.
Rao-Blackwellized Accept-Reject Estimator

- Recall in the rejection sampling algorithm, if we want to accept \( m \) samples, we need to actually sample \( N \) times, satisfying

\[
m = \sum_{i=1}^{N} \mathbb{1}_{U_i \leq w_i} \quad \text{and} \quad m - 1 = \sum_{i=1}^{N-1} \mathbb{1}_{U_i \leq w_i}
\]

where \( w_i = \frac{\pi(Y_i)}{kq(Y_i)} \)

- The rejection sampling estimator can be written as

\[
\delta_1 = \frac{1}{m} \sum_{i=1}^{m} h(X_i) = \frac{1}{m} \sum_{i=1}^{N} \mathbb{1}_{U_i \leq w_i} h(Y_i)
\]

Which depends on \( N, U_1, \ldots, U_N, Y_1, \ldots, Y_N \).
Rao-Blackwellized Accept-Reject Estimator

- The rejection sampling estimator

\[ \delta_1 = \frac{1}{m} \sum_{i=1}^{N} \mathbb{1}_{U_i \leq w_i} h(Y_i) \]

- Reduction in variance can be achieved with the conditional expectation (integrate out $U_i$’s)

\[
\delta_2 = \mathbb{E} \left[ \frac{1}{m} \sum_{i=1}^{N} \mathbb{1}_{U_i \leq w_i} h(Y_i) \middle| N, Y_1, \ldots, Y_N \right]
\]

\[
= \frac{1}{m} \sum_{i=1}^{N} \mathbb{E}[\mathbb{1}_{U_i \leq w_i} | N, Y_1, \ldots, Y_N] h(Y_i)
\]

\[
= \frac{1}{m} \sum_{i=1}^{N} \rho_i h(Y_i)
\]

- Computation of $\rho_i$ is omitted but requires $O(N^2)$ complexity.
- $\delta_2$ effectively replaced $U_i, N$ with conditional expectations.
Rao-Blackwellized Accept-Reject Estimator

- The estimator $\delta_2$ is often compared to the importance sampling estimator if the random nature of $N$ and its dependence on the samples are ignored:

\[
\mathbb{E} \left[ \frac{1}{m} \sum_{i=1}^{N} \mathbb{I}_{U_i \leq w_i} h(Y_i) \right| Y_1, \ldots, Y_N ] \\
= \frac{1}{m} \sum_{i=1}^{N} \mathbb{E}[\mathbb{I}_{U_i \leq w_i} | Y_1, \ldots, Y_N] h(Y_i) \\
= \frac{1}{m} \sum_{i=1}^{N} \frac{\pi(Y_i)}{kq(Y_i)} h(Y_i) \\
\left(\text{v.s. } \frac{1}{N} \sum_{i=1}^{N} \frac{\pi(Y_i)}{q(Y_i)} h(Y_i) \right)
\]
References

- Iain Murray - NIPS Monte Carlo Tutorial 2015