Lecture 6: Monte Carlo Methods

Professor Alexander Franks

Announcements

- Reading: Chapter 10 and 11

Monte Carlo approximations of a distribution

Sampling strategies

- Inversion Sampling (works for univariate distns)
- Grid sampling (works for low dimensional problems)
- Rejection sampling
- Importance sampling
- Markov Chain Monte Carlo

Sampling strategies

Why sampling?

Probability Integral Transform

- Suppose that a random variable, Y has a continuous distribution for with CDF is F_Y.
- Then the random variable $U = F_Y(Y)$ has a uniform distribution
  - This is known as the "probability integral transform PIT"
- By taking the inverse of F_Y we have $F^{-1}(U) = Y$

Inversion Sampling

The inverse transform sampling method works as follows:

1. Generate a random number u from $\mathrm{Unif}[0, 1]$
2. Find the inverse of the desired CDF, e.g. $F_Y^{-1}(u)$.
3. Compute $y = F_Y^{-1}(u)$. y is now a sample from the desired distribution.

Inversion Sampling

Animation

Inversion Sampling

- Inversion sampling can be a fast and simple way to sample from a distribution
- Only effective if we know the inverse-CDF and can easily compute it
- This is a big challenge in practice. For example, even the normal distribution has a CDF, Φ, which cannot be expressed analytically.
  - Shifts from one hard problem (sampling) to another (computing an integral)
  - Need alternatives!

Rejection Sampling

Rejection Sampling algorithm

1. Choose a proposal density, $q(\theta)$ that we can easily sample from (e.g. uniform or normal) such that:

2.Find $M = \max \frac{p(\theta|y)}{q(\theta)}$

- If $M = \infty$ then q cannot be used as a proposal distribution
- If M is finite, $Mq(\theta)$ "envelopes" $p(\theta\ y)$

3. Draw a sample, $\theta^{(s)}$ from $q(\theta)$

4. Accept $\theta^{(s)}$ as a draw from $p(\theta \mid y)$ with probability $\frac{p(\theta^{(s)}|y)}{Mq(\theta^{(s)})}$

Rejection Sampling

Demo

Rejection Sampling

- What is the expected number of rejections?
- How can we show this algorithm is valid?

Proposal most envelope target

Sometimes its not obvious…

Target density in black, proposal density in red

The German Tank Problem

The German Tank Problem

- US and allies wanted to estimate how many tanks were produced by the Germans during WWII
- They used the serial numbers on captured or destroyed tanks to estimate tank production
- Assuming serial numbers, Z_i, are ordered from 1 to n, where n is the number of tanks produced
- How do we estimate n?

The German Tank Problem

- Assume we observe the serial number of only one tank, Z
- What is a plausible sampling model? How do we write the likelihood?
- What is a reasonable prior?

The German Tank Problem

- Assume Z is uniformly distributed between 1 and n
- Specify a Poisson prior distribution
  - Assume a priori a rate of 200 tanks / month
- Use Monte Carlo (rejection sampler) to summarize the results. What is a reasonable proposal?

The German Tank Problem, Z = 200

Acceptance rate: 0.48257

The German Tank Problem, Z = 200

The German Tank Problem, Z = 150

Acceptance rate: 0.75292

German Tank Problem, Z=150

The German Tank Problem, Z = 250

Acceptance rate: 2.749^{-4}

The German Tank Problem, Z = 250

Importance Sampling

- With rejection sampling we get actual samples from the target density
- Downside: often throw away many samples. Inefficient if rejection rate is high.
  - Main challenge is finding a valid proposal density and one that minimzes the number of rejections
- Importance sampling: use all samples from proposal to compute expected values.
- Weight the *importance* of each sample by how representative it is of the target distribution

Importance Sampling

- Examine histogram of log importance weights
- Want the largest weights to be not too large
- Effective sample size: $\frac{1}{\sum_{s=1}^{S} \tilde{w}(\theta^s)^2}$ where $\tilde{w}(\theta^s)$ are the normalized weights

Importance Sampling

- Target: a Beta(3, 4)
- Proposal: Uniform

Importance Sampling

Importance Sampling

Importance Sampling

- Target: Cauchy(0, 1)
- Proposal: Normal(0, 3)

Importance Sampling

```r
df <- 1

proposal_draws <- rnorm(1e5, sd=2)
proposal_draws <- sort(proposal_draws)

w <- dt(proposal_draws, df=df) / dnorm(proposal_draws, sd=2)
w <- w/sum(w)

q <- 0.25
idx <- match(FALSE, cumsum(w) < q)
qt(q, df=df)
```

```
[1] -1
```

```r
proposal_draws[idx]
```

```
[1] -0.860783
```

```r
q <- 0.05
idx <- match(FALSE, cumsum(w) < q)
qt(q, df=df)
```

```
[1] -6.313752
```

```r
proposal_draws[idx]
```

Importance Sampling

```
[1] "99% quantile of log10(w) is -4.773680"
[1] "max of log10(w) is -1.769665"
```

A Gamma Puzzle

- Goal: report the exact mean of a Gamma(a, b) using only one random draw
- Allowed to generate a random draw from a proposal distribution of your choosing
- You can evaluate the gamma density and the proposal density at the generated value

A Gamma Puzzle

```r
## Generate 10 draws from a gamma 5, 2
proposal_draws <- rgamma(10, shape = 5, rate=2)

## Targt a gamma 4, 2
w <-
  dgamma(proposal_draws, shape=4, rate=2) /
  dgamma(proposal_draws, shape=5, rate=2)

w*proposal_draws
```

```
[1] 2 2 2 2 2 2 2 2 2 2
```