Logistics

- Office hours Thursdays at 16:00–17:00 in my office (see Blackboard)
- Exercise 1 will be posted later this week
  - Find other group members on Discourse if possible
  - If no partner by next Wednesday, email Guillermina
  - Remember to check out 'Learning R' page on Blackboard *before* starting project if possible

Lecture 4: Review

What have we done until now?

- Simulation from discrete probability models
  - General Algorithm
  - Some special algorithms for specific distribution
- Simulation from continuous probability models
  - Inversion Sampling
  - Use known relationships between RV
  - Change of variables
  - Ratio of uniform methods
  - Mixtures
  - Multivariate distribution
  - Rejection Sampling

Today:

- More on rejection sampling
  - Weighted resampling
  - Adaptive rejection sampling
- Monte Carlo Integration
- Importance sampling

Rejection sampling

- We want $x \sim f(x)$ (target density).
- We know how to generate realisations from a density $g(x)$
- We know a value $c > 1$, so that $\frac{f(x)}{g(x)} \leq c$ for all x where $f(x) > 0$.

Algorithm:

finished $= 0$

while (finished $= 0$)

    generate $x \sim g(x)$

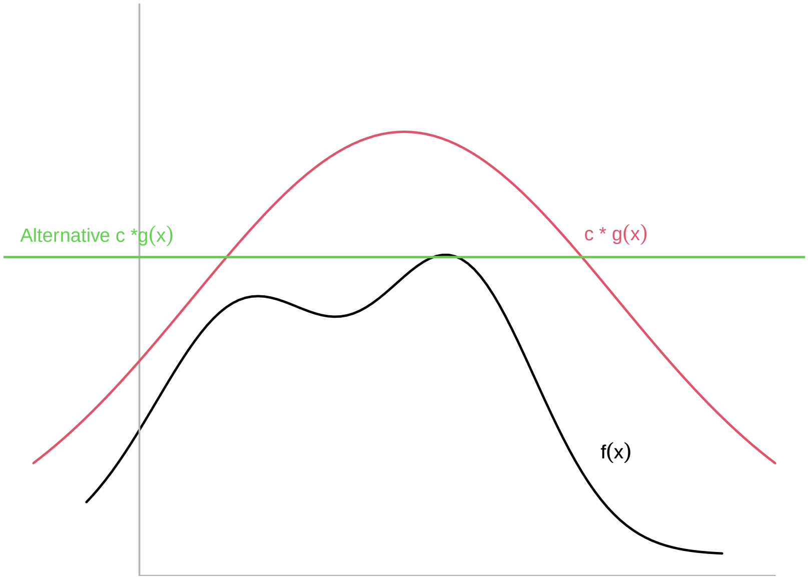
    compute $\alpha = \frac{1}{c} \cdot \frac{f(x)}{g(x)}$

    generate $u \sim U[0, 1]$

    if $u \leq \alpha$ set finished $= 1$

return x

Rejection sampling



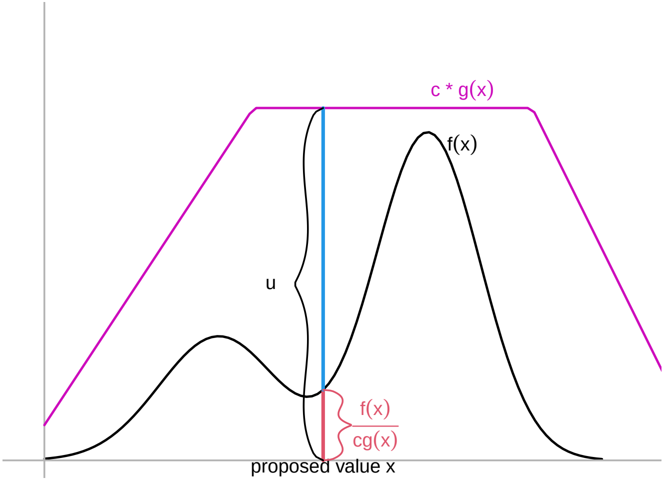
Rejection sampling

- The overall acceptance probability for the algorithm is

$$\mathrm{P}\left(U \leq \frac{1}{c} \cdot \frac{g(X)}{f(X)}\right) = \int_{-\infty}^{\infty} \frac{f(x)}{c \cdot g(x)} g(x)\, dx = \int_{-\infty}^{\infty} \frac{f(x)}{c}\, dx = c^{-1}.$$

- The expected number of trials up to the first success is c
- The smaller c the more efficient the algorithm

Rejection sampling



Example I: Sample from $N(0, 1)$ with rejection sampling

- Target distribution:

$$f(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right).$$

- Proposal distribution:

$$g(x) = \frac{\lambda}{2} \exp(-\lambda|x|), \lambda > 0$$

Example I: Sample from $N(0,1)$ with rejection sampling

- Find bound c:

$$\frac{f(x)}{g(x)} = \frac{\frac{1}{\sqrt{2\pi}}\exp(-1/2x^2)}{\frac{\lambda}{2}\exp(-\lambda|x|)} \leq \sqrt{\frac{2}{\pi}}\lambda^{-1}\exp\left(\frac{1}{2}\lambda^2\right) \equiv c(\lambda)$$

- We choose λ such that c is as small as possible

$$c(\lambda) \overset{\lambda=1}{=} \sqrt{\frac{2}{\pi}}\exp\left(\frac{1}{2}\right) \approx 1.3$$

- Then the acceptance probability is:

$$\alpha(\lambda) \overset{\lambda=1}{=} \exp\{-\frac{1}{2}x^2 + |x| - \frac{1}{2}\}$$

Example II: Standard Cauchy

Remember: Using ratio-of-uniforms method we can simulate from standard Cauchy as:

- Sample (x_1, x_2) uniformly from tht semi-unit circle
- Compute $y = \frac{x_2}{x_1}$
- y is a sample from the uniform Cauchy

How can we sample from the semi-unit circle?

Rejection sampling also works when x is a vector.

Standard Cauchy: Rejection sampling algorithm

finished $= 0$

while finished $= 0$ **do**

generate $(x_1, x_2) \sim g(x_1, x_2)$

compute

$$\alpha = \frac{1}{c}\frac{f(x_1,x_2)}{g(x_1,x_2)} = \begin{cases} \frac{1}{c} \cdot \frac{2}{\text{area}(C_f)} \overset{c=\frac{2}{\text{area}(C_f)}}{=} 1, & (x_1, x_2) \in C_f \\ 0, & \text{otherwise} \end{cases}$$

generate $u \sim \mathcal{U}(0, 1)$

if $u \leq \alpha$ **then** finished $= 1$

end if ▷ i.e. If $(x_1, x_2) \in C_f$ finished $= 1$

end while

return x_1, x_2

Standard Cauchy: Summary

Note: To do this algorithm we do not need to know the value of the normalising constant $\text{area}(C_f)$.

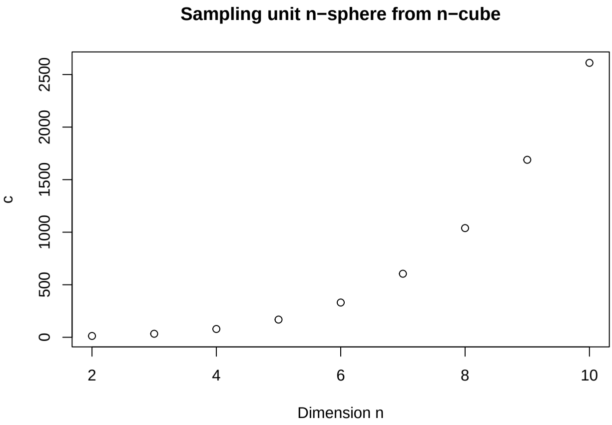
This is always true in rejection sampling.

Rejection sampling - Acceptance probability

Note: For c to be small, $g(x)$ must be similar to $f(x)$.
The art of rejection sampling is to find a g(x) that is similar to f(x) and which we know how to sample from.

Issues: c is generally large in high-dimensional spaces, and since the overall acceptance rate is $1/c$, many samples will get rejected.

Sampling uniformly from the unit n-dimensional sphere



Rejection Sampling

Difficulties when implementing rejection sampling:

- Finding the constant $c \rightarrow$ Weighted resampling
- Finding the proposal density $g(x) \rightarrow$ Adaptive rejection sampling

Weighted resampling

A problem when using rejection sampling is to find a legal value for c. An approximation to rejection sampling is the following:

Let, as before:

- $f(x)$: target distribution
- $g(x)$: proposal distribution

Algorithm

- Generate $x_1, \dots, x_n \sim g(x)$ iid
- Compute weights

$$w_i = \frac{\frac{f(x_i)}{g(x_i)}}{\sum_{j=1}^{n} \frac{f(x_j)}{g(x_j)}}$$

- Generate a second sample of size m from the discrete distribution on $\{x_1, \dots, x_n\}$ with probabilities $w_1, \dots, w_n$.

The resulting sample $\{y_1, \dots, y_m\}$ has approximate distribution $f(x)$

Weighted resampling

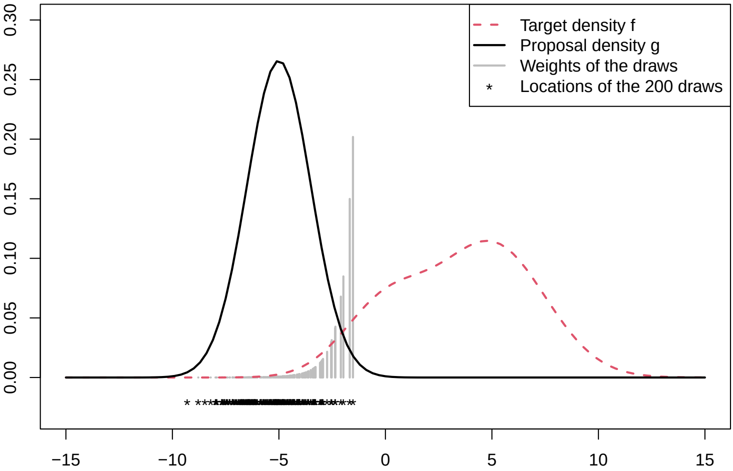
For illustration see `Lecture4.R`

Comments

- The advantage is that we do not need the constant c
- The resulting sample has approximate distribution f
- The resample can be drawn with or without replacement provided that $n >> m$, a suggestion is $n/m = 20$.
- The normalising constant is not needed.
- This approximate algorithm is sometimes called sampling importance resampling (SIR) algorithm.
- g should have tails at least as heavy as f!

Illustration

A bad choice of g will result in a bad representation of f



Adaptive rejection sampling

Remember:

Algorithm:

finished $= 0$

while (finished $= 0$)

generate $x \sim g(x)$

compute $\alpha = \frac{1}{c} \cdot \frac{f(x)}{g(x)}$

generate $u \sim U[0, 1]$

if $u \leq \alpha$ set finished $= 1$

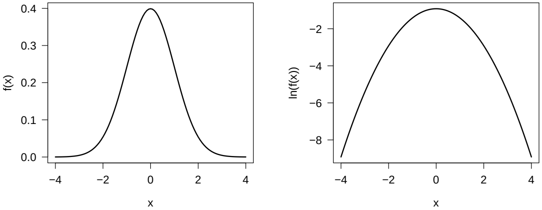
return x

- Note that the algorithm is valid even if $g(x)$ is different in every iteration
- How to find $g(x)$?

Adaptive rejection sampling

This method works only for log concave densities, i.e.

$$(\ln f)''(x) \leq 0, \quad \text{for all } x.$$



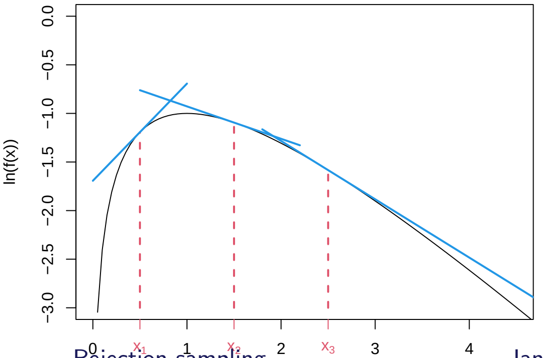
Many densities are log-concave, e.g. the normal, the gamma ($a > 1$), densities arising in GLMs with canonical link.

Adaptive rejection sampling (2)

Basic idea: Start with a proposal distribution $g_0(x)$ (with $c = c_0$).
If we propose a value from $g_0(x)$ and reject it, then we use it to construct an improved proposal $g_1(x)$ with $c_1 \leq c_0$.
Continue untill acceptance

Adaptive rejection sampling (2)

- Start with an initial grid of points $x_1, x_2, \ldots, x_m$ (with at least one x_i on each side of the maximum of $\ln(f(x))$) and construct the envelope using the tangents at $\ln(f(x_i))$, $i = 1, \ldots, m$.
- Draw a sample from the envelop function and if accepted the process is terminated. Otherwise, use it to refine the grid.



Monte Carlo integration

Assume we are interested in

$$\mu = \mathsf{E}[h(X)];\ X \sim f(x)$$

If X is continuous and scalar we have

$$\mu = \mathsf{E}[h(X)] = \int_{-\infty}^{\infty} h(x) f(x)\ dx$$

Analytical solution is the best when possible!

Monte Carlo integration

Assumption

It is *easy* to generate independent samples $x_1, \ldots, x_N$ from a distribution $f(x)$ of interest.

A Monte Carlo estimate of

$$\mu = \mathrm{E}(h(x)) = \int h(x)f(x)dx$$

is then given by

$$\hat{\mu} = \frac{1}{N}\sum_{i=1}^{N} h(x_i).$$

What is the mean and variance of this estimator?

Monte Carlo integration (II)

$\hat{\mu}$ is an unbiased estimate of μ

- $\mathrm{E}(\hat{\mu}) = \mu$
- $\widehat{\mathrm{Var}}(\hat{\mu}) = \frac{1}{N(N-1)} \sum_{i=1}^{N} (h(x_i) - \hat{\mu})^2$
- Then the strong law of large numbers says:

$$\hat{\mathrm{E}}(h(x)) = \frac{1}{N} \sum_{i=1}^{N} h(x_i) \overset{a.s}{\to} \int h(x) f(x) dx = \mathrm{E}(h(x))$$

Monte Carlo integration (III)

Monte carlo integration can be used for any function $h(\cdot)$

Examples

- Using $h(x) = x^2$ we obtain an estimate for $\mathrm{E}(x^2)$.
- An estimate for the variance follows as

$$\widehat{\mathrm{Var}}(x) = \hat{\mathrm{E}}(x^2) - \hat{\mathrm{E}}(x)^2$$

- Setting $h(x) = I(x \in A)$ we get:

$$E[h(x)] = E[I(x \in A)] = \mathrm{P}(x \in A)$$

Importance sampling

One of the principal reasons for wishing to sample from complicated probability distributions $f(z)$ is to be able to evaluate expectations with respect to some function $p(z)$:

$$\mathrm{E}(p) = \int p(z) f(z) dz$$

The technique of importance sampling provides a framework for approximating expectations directly but does not itself provide a mechanism for drawing samples from a distribution.

Importance sampling: Idea

[See blackboard]

Importance sampling

Let $x_1, \ldots, n_N \sim g(x)$ then the importance sampling estimator of $\mu = \mathrm{E}_f(h(x))$ is given by

$$\hat{\mu}_{IS} = \frac{1}{N} \sum_{i=1}^{N} \frac{h(x_i) f(x_i)}{g(x_i)} = \frac{1}{N} \sum_{i=1}^{N} h(x_i) w(x_i)$$

wih

- We need $g(x) > 0$ where $h(x)f(x) > 0$
- The quantities $w(x_i) = \frac{f(x_i)}{g(x_i)}$ are called importance weights
- $\mathrm{E}(\hat{\mu}_{IS}) = \mu$
- $\mathrm{Var}(\hat{\mu}_{IS}) = \frac{1}{N} \mathrm{Var}_g[\frac{h(x)f(x)}{g(x)}]$

Importance sampling estimators

To compute the importance sampling estimator

$$\hat{\mu}_{IS} = \frac{1}{N} \sum_{i=1}^{N} h(x_i) w(x_i)$$

we need to know the normalizing constant of f and g.

When this is not possible an alternative is a "self-normalizing" importance sampling estimator

$$\tilde{\mu}_{IS} = \frac{\sum h(x_i) w(x_i)}{\sum w(x_i)}$$

where we need that

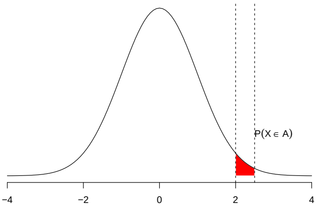
$$g(x) > 0 \text{ where } f(x) > 0$$

Importance sampling: Example

Assume we want to estimate

$$P(X \in [2, 2.5]) \text{ where } X \sim \mathcal{N}(0, 1)$$

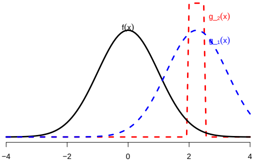
- Can use MC estimate $\rightarrow$ small efficiency
- Importance sampling can help "focus" the sampler in the correct area



`Show code`

Importance sampling: Example

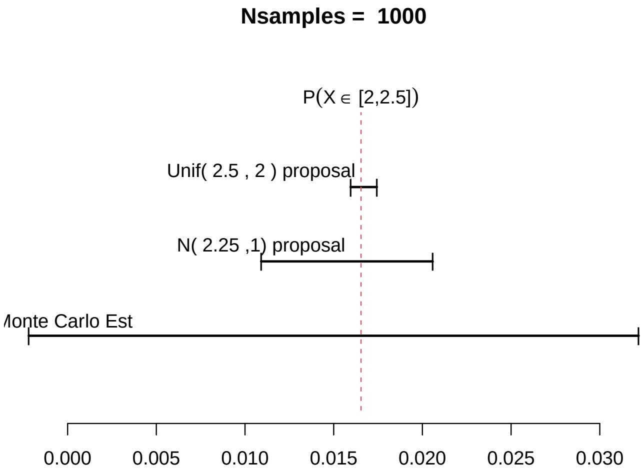
$\mu = P(X \in [2, 2.5]) = \int_{\mathcal{R}} I(x \in [2, 2.5]) f(x) dx$ with $f(x) = \mathcal{N}(0, 1)$
Three estimation schemes:



1. MC estimate
2. IS with proposal $g_1(x) = \mathcal{N}(2.25, 1)$
3. IS with proposal $g_2(x) = \mathcal{U}(2, 2.5)$

Note: in case 3) we cannot use the self-normalizing version of the IS algorithm

Importance sampling: Example



Importance sampling

We are interested in

$$\mu = E_f(h(x)) = \int h(x)f(x)dx$$

- If possible compute it analytically!
- If we can sample from $f(x)$ we can use Monte Carlo integration
- Possible alternative: Importance sampling
  - sample from auxiliary distribution $g(x)$ and re-weight
  - no need to reject samples, throwing out information
  - can be used as variance-reduction technique

Importance sampling Algorithm

Let $x_1, \dots, x_n \sim g(x)$, and let $w(x_i) = \frac{f(x_i)}{g(x_i)}$, $i = 1, \dots, n$ then

$$\hat{\mu}_{IS} = \frac{\sum h(x_i) w(x_i)}{n}$$

- Unbiased
- Consistent
- Need to know the normalizing constant

$$\tilde{\mu}_{IS} = \frac{\sum h(x_i) w(x_i)}{\sum w(x_i)}$$

- Biased for finite n
- Consistent
- Self-normalizing

Importance sampling: Summary

As with rejection sampling, the success of importance sampling depends crucially on how well the proposal distribution $g(x)$ matches the target distribution $f(x)$.