Topic: Random # Generation & Monte Carlo Methods. Ch 6 & 7.

pg143 Ch. 6 Simulation & Monte Carlo Integration

Introduction to the MC Method.

A Simple Motivating Example:

Let $h: [a,b] \to \mathbb{R}$ be a function and suppose that we want to compute

$$\mathcal{I} = \int_a^b h(x)\,dx$$

but this integral is not analytically tractable.
Now consider rewriting $\mathcal{I}$ as

$$\mathcal{I} = \int_a^b (b-a)h(x)\frac{1}{b-a}\,dx$$

$$= \int_a^b g(x) f(x)\,dx \qquad \text{where } g(x) = (b-a)h(x),\ f(x) = \frac{1}{b-a}$$

$$= E[g(x)]$$

Then our integral is eq. to finding the expectation of $g(x)$ taken w.r.t. the uniform density $f(x)$.

So, the Monte Carlo method for approx $\mathcal{I}$ is to compte the expected value of g by use of a simulated sample $X \sim \mathcal{U}(a,b)$.

More formally, the M.C. estimate is

$$\hat{I} = \frac{1}{n}\sum_{i=1}^{n} g(x_i)$$

where $x_1, x_2, \ldots, x_n$ are variated drawn from $\mathcal{U}[a,b]$.

Ex/ Use M.C. method to compute $I = \int_1^9 \frac{x}{4\sqrt{x}}\,dx$

(a) If we compute I exactly we have

$$I = \left[\frac{1}{6}x^{3/2}\right]_1^9 = \frac{13}{3} = 4.3333$$

(b) If we use M.C. method w/ $n = 5000$

$$I = \int_1^9 8\left(\frac{x}{4\sqrt{x}}\right)\frac{1}{8}\,dx$$

$$= \int_1^9 g(x)\,f(x)\,dx = E[g(x)]$$

where $g(x) = 2\sqrt{x}$ & $f(x) = 1/8$, $X \sim \mathcal{U}[1,9]$.

So we sample $x_1, \ldots, x_{5000}$ from $\mathcal{U}(1,9)$ and compute

$$\hat{I} = \frac{1}{5000}\sum_{i=1}^{5000} 2\sqrt{x_i} = 4.3339$$

3 Steps.
1. Decomp. of integrand into 2 funct. one of which is a density.
2. Reformulate problem as exp. val.
3. Sample from density & compute empirical ave.

(c) Compare: Pretty good!

In general, these ideas can be extended to solve problems in statistical inference where a quanlity of interest θ can be expressed as

$$\theta = \int_{R_X} h(x)\,dx = \int_{R_X} g(x) f(x)\,dx.$$

and $h(x) = g(x)f(x)$ & $\int_{R_X} f(x)\,dx = 1$ (w/ $f(x) \geq 0 \;\forall x \in R_X$).

We know that many quantities of interest in inferential statistical analyses can be expressed as the expectation of a function of a r.v., $E\{h(x)\}$, and hence are of this form.

1. We can estimate $\mu = E[X] = \int_{\mathcal{R}x} x f(x)\,dx$ by
$$\hat{\mu}_{mc} = \frac{1}{n}\sum_{i=1}^{n} x_i.$$
2. estimate $\mu = E[h(x)] = \int_{\mathcal{R}x} h(x) f(x)\,dx$
$$\hat{\mu}_{mc}\ \frac{1}{n}\sum_{i=1}^{n} h(x_i)$$
3. Bayesian inference: Want to compute post. exp.
$$E[h(\theta)\,|\,y] = \int_{\Theta} h(\theta)\,p(\theta|y)\,d\theta$$
can be approx. by
$$E[h(\theta)_{mc}\,|\,y] = \frac{1}{n}\sum_{i=1}^{n} h(\theta^{(i)})$$
where $\theta^{(i)}$ are drawn from $p(\theta|y)$.

Question: How to generate these random sample from our given distribution f.

Sec 6.2. Generation of Random #'s / Simulation

We know the prob dist. we want to use. How do we generate r.v.?

Commonly // Gen. a r.v. has 2 parts.

1. generate pseudo random #'s $\mathcal{U}(0,1)$
2. use these to obtain variates from $f(x)$.

(1) Your computer does this. They are "pseudo" random because they are reproducible by a mathematical algorithm, but they are considered to have passed statistical tests. Usually, they are

$$r_{n+1} = (k r_n + a) \bmod m \qquad (k, a < m)$$

w/ seed r_0. and are normalized r_{n+1}/m to $[0,1)$.

(2) Generate a particular dist. w/ CDF $F(x)$.
- many methods

This is the core of our discussion.

---

pg 145 Sec 6.2(.1) Standard parametric families.
- When your R.v. come from a standard para. family there is software to generate random deviates.

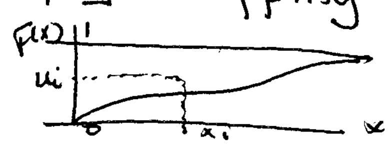
Additionally, Table 6.1 pg 146 gives a variety of methods (simple) to obtain these from $U(0,1)$.

However you can generate your own using the following methods.

Sec 6.2.2. Inverse CDF
also called: Prob. Integral transform approach.

For any continuous dist. function F_x of a r.v. X, we can define the inverse function F^{-1} on $[0,1]$ mapping into R_X.

Basic idea.



Prop

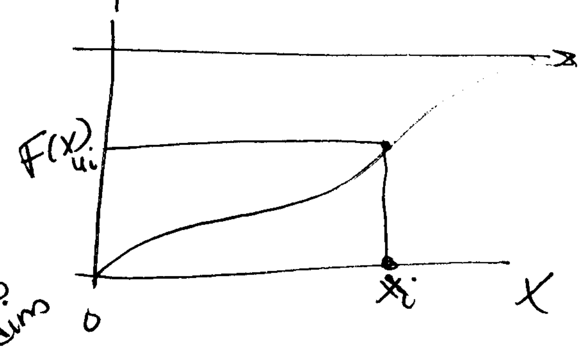
Suppose F is con't c.d.f. Let $U \sim U(0,1)$ then the R.V. $X = F^{-1}(U)$ has C.D.F. F.

Pf/ $F_X(x) = P(X \le x) = P(F^{-1}(U) \le x)$
$= P(U \le F(x))$ (F monotonic)
$= F(x)$ (since U uniform $0,1$).

So if we can acheive F^{-1} we can generate r.v. w/ CDF F by simply
1. generating $u_1, \dots, u_n$ on $U(0,1)$
2. letting $x_i = F^{-1}(u_i)$.

Comments:

1. Very simple method.
2. Can be extended to the discrete case. as follows
   1. generate $u_1, \dots,$ uniform$(0,1)$
   2. x_i is (x_i, u_i) on graph of CDF



3. Discrete method works for any CDF even if analytical results for F^{-1} not obtainable.

very crude
doesn't extend to higher dims

Example:
Generate Exp. r.v. θ
$F(x) = 1 - e^{-\theta x}$
Need to solve
$r = 1 - e^{-\theta x}$ → $e^{-\theta x} = 1 - r$

$x = \frac{-\ln(1-r)}{\theta}$ eg. $-\frac{\ln r}{\theta}$

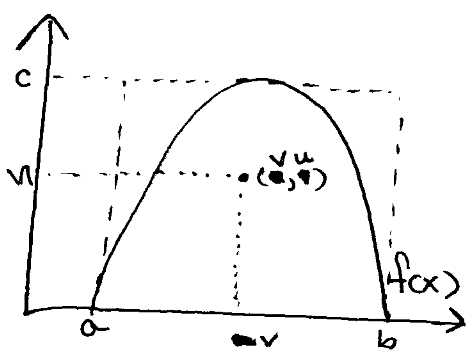
since r uniform $(0,1)$

Sec 6.2.3
pg 147

Rejection Sampling.

If $f(x)$ can be calculated, at least up to a proportionality constant, then we can use rejection sampling to obtain a random draw exactly from the target dist.

Basic idea: Suppose $a < x < b$, $f(x) \le c$



1. Create a "box" around f. $[a,b] \times [0,c]$
2. Generate $u \sim \mathcal{U}(a,b)$ and $v \sim \mathcal{U}(0,c)$ (ie (v,u))
3. if $u \le f(v)$ (ie (v,u) undr curve) accept v as your random variate from p.d.f f.
4. o.w. reject & go back to 1

Comments: ① If v is accepted, (v,u) lies below the graph of $f(x)$. $\therefore$ $P(v \le d)$ is prop to area under the graph & left of d, and thus C.D.F of v is $F(x)$.

② The prob. of rejection is

$$1 - P(\text{acceptance}) = 1 - \frac{\text{area under } f(x)}{\text{area of box}}$$
$$= 1 - \frac{1}{(b-a)\cdot c}$$

to improve eff of alg. want this value as small as possible which could be difficult if $b-a$ is large.

③ Can improve on this idea by drawing x from a known dist that contains f.

More formally: Réjection Sampling. ($f(x)$)

Let g denote another density from which we know how to sample and for which we can easily calculate $g(x)$.

Let $e(\cdot)$ denote an envelope, having the prop.
$e(x) = g(x)/\alpha \geq f(x) \quad \forall\, x \ni f(x) > 0$ & some $\alpha \leq 1$.

<u>Rejection Sampling Algorithm</u>

1. Sample $Y \sim g$
2. Sample $U \sim \mathcal{U}(0,1)$
3. If $U > f(Y)/e(Y) = f(Y)/(g(Y)/\alpha)$ reject Y and return to 1.
4. o.w. keep $X = Y$ as an element of target sample.

Prop.
The variable X in the R-S. method is dist according to f.

Pf/
$$P[X \leq y] = P\left[Y \leq y \,\middle|\, U \leq \frac{f(Y)}{e(Y)}\right]$$
$$= \frac{P(Y \leq y \text{ and } U \leq f(Y)/e(Y)]}{P(U \leq f(Y)/e(Y))}$$
$$= \frac{\int_{-\infty}^{y} \int_{0}^{f(z)/e(z)} du\, g(z)\, dz}{\int_{-\infty}^{\infty} \int_{0}^{f(z)/e(z)} du\, g(z)\, dz)}$$
$$= \frac{\int_{-\infty}^{y} \frac{f(z)}{e(z)} g(z)\, dz}{\int_{-\infty}^{\infty} \frac{f(z)}{e(z)} g(z)\, dz} = \frac{\alpha \int_{-\infty}^{y} f(z)\, dz}{\alpha \int_{-\infty}^{\infty} f(z)\, dz} = \int_{-\infty}^{y} f(z)\, dz.$$

So rejection sampling provides exact draws from f and α can be interpreted as the exp. prop of candidates accepted. Hence the eff of the alg. depends on α. $(P(\text{reject}) \sim 1-\alpha.)$ α near 1 desirable

See fig 6.1 pg 148

Now suppose that the target dist f is only known up to a prop. constant c. and we are only able to compute easily

$$g(x) = f(x)/c.$$

Sec 1.5

Ex/ Bayesian posterior dist

$$f(\theta|x) = c\, f(\theta) L(\theta|x)$$

where $c = 1/\int f(\theta) L(\theta|x) d\theta$ is the normalizing constant & can be diff to compute.

Fortunately rejection sampling still works (and hence is appealing to Bayesians).

We find an evelope e, s.t. $e(x) \geq g(x)$ $\forall x \ni g(x) > 0$. Draw $Y=y$ is rejected if $U > g(y)/e(y)$.

The sampling remains correct because the unknown c cancels out of the num. & denom in proof. when f is replaced by g.

Now the prop accepted is α/c.

Comments: ① f is called target dist.

② g is called the instrumental or proposal dist.

③ Draws are independent.

④ Can be extended to multivariate case.

Pg 150 Ex 6.2 Bayesian posterior

Assume $(8, 3, \ldots, 7)$ observed from $X_i \mid \lambda \sim \text{Poisson}(\lambda)$.

prior dist assumed log normal
$\log \lambda \sim N(4, .5^2)$. $f(\lambda)$

Suppose we know that $\hat{\lambda} = \bar{x} = 4.3$ maximizes the Likelihood $L(\lambda \mid x)$ w.r.t. λ (Poisson)

from HW

$\therefore$ we have the unnormalized posterior dist
$$g(\lambda \mid x) = f(\lambda) L(\lambda \mid x)$$
is bounded above by the envelope
$$e(\lambda) = f(\lambda) \cdot L(4.3 \mid x)$$
(picture 150)

And so we use rejection sampling by

① sampling λ_i from lognormal prior.

② sampling U_i from $U(0,1)$

③ Keeping λ_i if $U_i < g(\lambda_i \mid x) / e(\lambda_i) = \dfrac{L(\lambda_i \mid x)}{L(4.3 \mid x)}$.

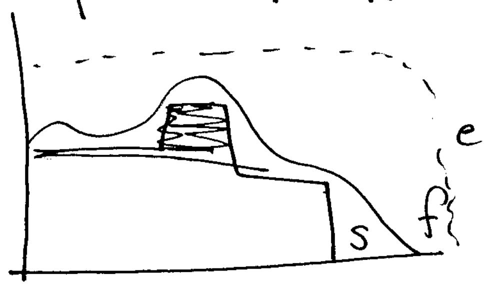
Any kept λ_i are drawn from posterior dist.

Only 30% kept. but it was easy.

Sec 6.2.3.1 Squeezed rejection sampling
pg 150

Ordinary rejection sampling requires one evaluation of f for every candidate draw Y. If eval. f is comp. expensive, we can improve speed via squeezed rej. sampling.

Basic strategy: preempt eval of f in some instances by using a nonneg squeezing function s that is easy to evaluate



want s s.t.

1. $s(x) \leq f(x) \quad \forall\ x$ in R_f
2. easy to compute.

we will also need $e(t)$.

S.R.S. Algorithm

1. Sample $Y \sim g$
2. Sample $U \sim U(0,1)$
3. If $U \leq s(Y)/e(Y)$ keep Y. Set $X = Y$ as an element of target dist. go to 6
4. O.W. check if $U \leq f(Y)/e(Y)$. if so keep Y. Set $X = Y$ go to 6
5. O.W. Reject Y and go to 1
6. Return to 1 until you have enough Xs.

Ex. pg 152. Note:
- Same results as R.S w/ fewer eval. of f.
- avoid eval of f $\int s(x)dx / \int e(x)dx$ of the time
- works even when target only known to prop. const.

Pseudo code for Squeezed Rejection Sampling.

Suppose you want to generate n random numbers from distribution f using S.R.S. w/ s, g and $e = g/\alpha$.

Pseudocode

```
- Initialize constants   n & α
- Initialize variables   Y[n]=0 & accepted=0
- While (accepted < n) {
      x = r.n. from g               (sample g)
      u = r.n. from U(0,1)          (sample U(0,1)).
      S = s(x)                      (eval s at x)
      e = g(x)/α                    (eval e at x).

   If (u ≤ s/e) {
      accepted = accepted + 1
      Y[accepted] = x
   }
   Else {
      f = f(x)
     If (u ≤ f/e) {
       accepted = accepted + 1
       Y[accepted] = x
     }
   }
  }
  Return Y
```