

Part A Simulation and Statistical Programming HT14

Lecturer: Geoff Nicholls

University of Oxford

Lecture 4: Rejection sampling

The Rejection algorithm

Here is second proof that explores the algorithmic structure of rejection. In the first proof we used the notation for a continuous scalar random variable. This time I will set it up for sampling a scalar discrete variable. This is just for simplicity. Each approach can be applied to both kinds of variables and both can be extended to the multivariate case.

Consider $\Omega = \{1, 2, 3, \dots, M\}$.

For $x \in \Omega$ let $p(x)$ be the probability mass function for X .

If $p = (p_1, p_2, p_3, \dots, p_M)$ then $p(x) = p_x$.

As before we suppose we have M so that $M \geq p(x)/q(x)$ for each $x \in \Omega$.

Rejection Algorithm simulating $X \sim p$:

[1] Simulate $y \sim q(y)$ and $u \sim U(0, 1)$.

[2] If $u < p(y)/Mq(y)$ return $X = y$ and stop, and otherwise goto [1] and try again.

Let $\Pr(X = x)$ be the probability this algorithm returns $X = x$. We want to show that $\Pr(X = x) = p(x)$.

Claim: The number of trials in the rejection algorithm is geometric with mean M

Proof: each trial at [2] succeeds independently with probability ξ say. In order to succeed we must draw a y and accept it. The first event (draw y) happens wp $q(y)$. The second event (accept y) happens wp $p(y)/q(y)M$. We don't care which y it is so...

$$\xi = \sum_{y \in \Omega} q(y) \times p(y)/(Mq(y)) \quad (1)$$

$$= 1/M \quad (2)$$

because $\sum_y p(y) = 1$. If the number of trials is N then

$$E(N) = 1/\xi = M$$

Comment: Each time we execute steps [1-2] we fail wp $1 - 1/M$

Claim: The probability the algorithm returns $X = x$ is $p(x)$.

How do we end up with $X = x$? When we enter the loop [1-2] the probability we draw x and then accept it is

$$q(x) \times p(x)/Mq(x) = p(x)/M.$$

However we might pass through [1-2] any number of times. We can draw x and then accept it at the first pass, or at the second pass, or third and so on.

$$\begin{aligned}\Pr(X = x) &= \frac{p(x)}{M} + (1 - 1/M) \frac{p(x)}{M} + (1 - 1/M)^2 \frac{p(x)}{M} + \dots \\ &= \frac{p(x)}{M} \sum_{n=1}^{\infty} (1 - 1/M)^{n-1} \\ &= p(x).\end{aligned}$$

as supposed.

First comment: Where did we use $Mq(x) \geq p(x)$? If for some y , $p(y)/q(y) \geq M$ then the probability $u \leq p(y)/Mq(y)$ at step [2] is one, not $p(y)/Mq(y)$.

Exercise: show that rejection simulates

$$X \sim \min(p(x), Mq(x))$$

(and hence $X \sim p(x)$ if $p(x) \leq Mq(x)$ for all x).

Second comment: The mean number of trials is M . If we are conservative and choose a very large M , in order to ensure the bound $M \geq p(y)/q(y)$ holds for all y , then the number of trials is very large, and the implementation runs slowly.

Example Use $Y \sim U(0, 1)$ (ie $q(x) = 1$) to simulate $X \sim p(x)$ with $p(x) = 2x$ for $0 \leq x \leq 1$. Calculate the mean number of trials per simulated X -value.

Step 1. Find M so that $p(x)/q(x) \leq M$ for all $0 \leq x \leq 1$. Here $p(x)/q(x) = 2x$. We can use $M = 2$ since $2x \leq 2$ there. We have $p(y)/q(y)M = y$ for the second step.

Step 2. Write down the algorithm.

Rejection Algorithm simulating $X \sim 2x$:

[1] Simulate $y \sim U[0, 1]$ and $u \sim U[0, 1]$.

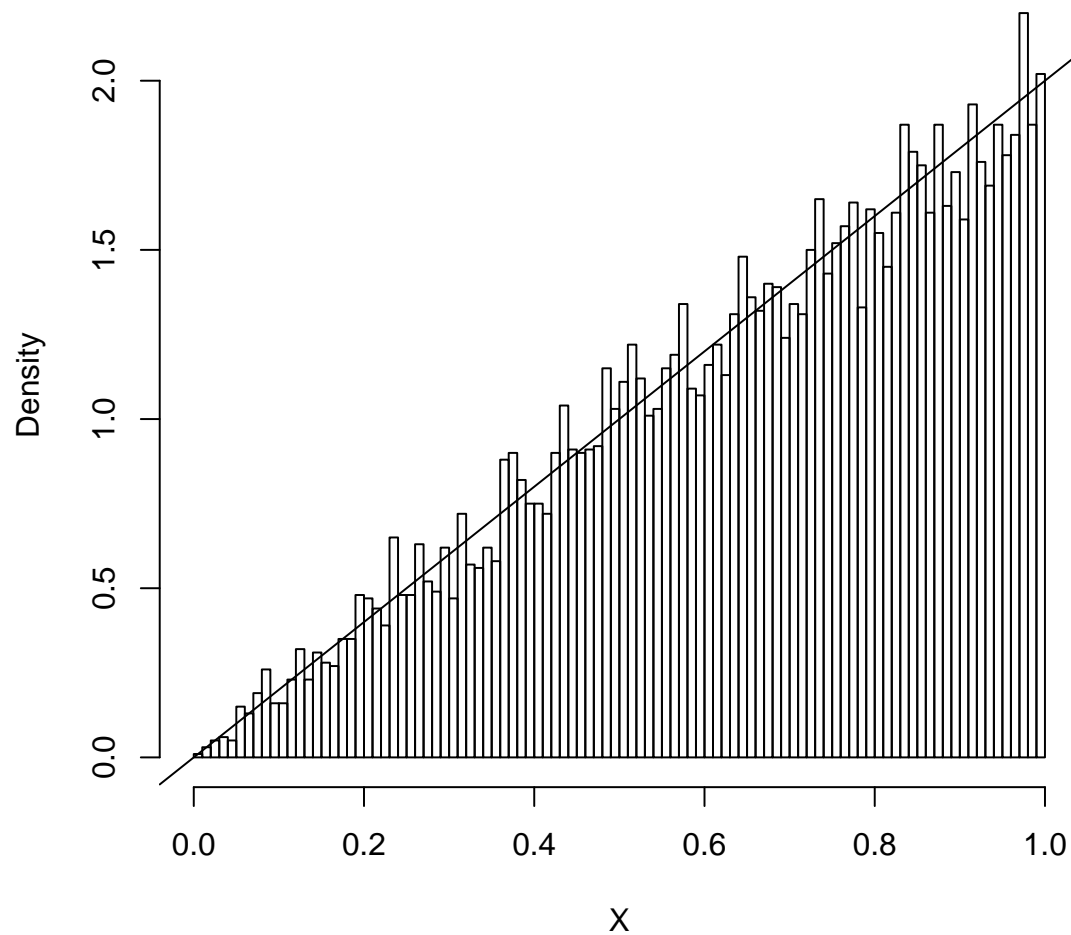
[2] If $u < y$ return $X = y$, and otherwise goto [1] and try again.

The mean number of trials is $M = 2$.

Implement this algorithm in R:

```
> while (runif(1)>(x<-runif(1))) {}; x  
[1] 0.8843435
```

Compare the histogram of 10K samples with the target density.



Normalizing constants and the rejection algorithm

A probability density $p(x)$, $x \in \Omega$ must be normalized, so that

$$\int_{\Omega} p(x) dx = 1.$$

We can specify a probability density by just giving a function $\tilde{p}(x)$ proportional to $p(x)$. If

$$p(x) \propto \tilde{p}(x)$$

then $p(x) = \tilde{p}(x)/Z_p$ with

$$Z_p = \int_{\Omega} \tilde{p}(x) dx.$$

$p(x)$ is determined from $\tilde{p}(x)$, even if we can't calculate it.

Normalizing constants are hard to calculate, so we like algorithms that simulate $X \sim p$ without our needing to calculate $p(x)$. The rejection algorithm is one of these.

Suppose (usual setup) we can simulate $Y \sim q$ and we want $X \sim p$. What if we can only calculate $\tilde{q}(x)$ and $\tilde{p}(x)$ and not Z_q and Z_p (so not q and p).

Suppose we can find M so that $M \geq \tilde{p}(x)/\tilde{q}(x)$ for all x .

Rejection Algorithm simulating $X \sim p$:

[1] Simulate $y \sim q(y)$ and $u \sim U(0, 1)$.

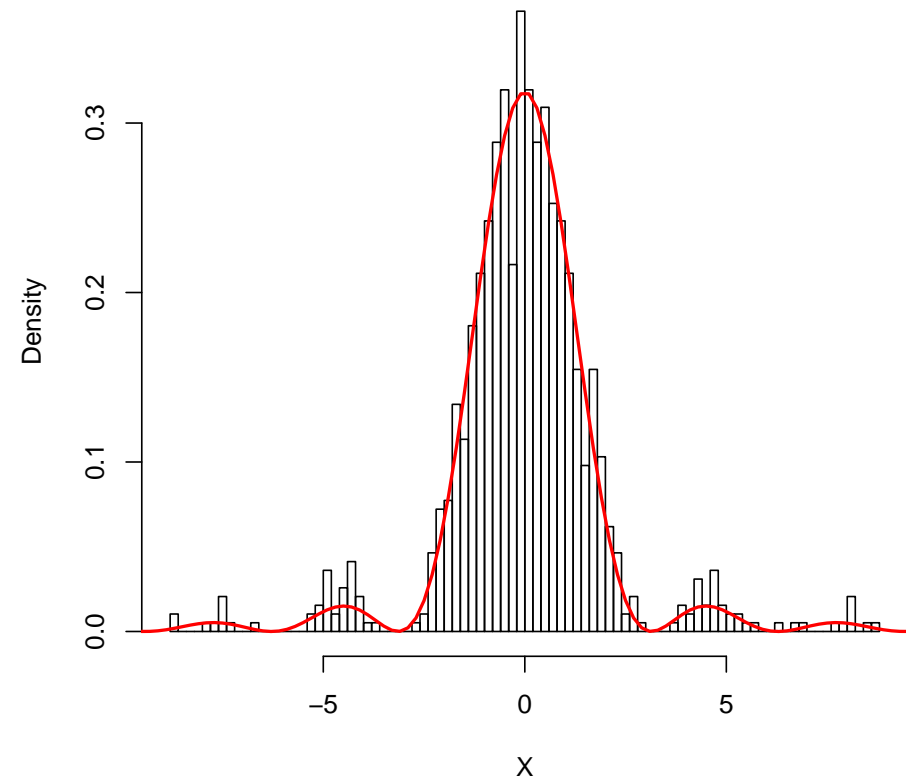
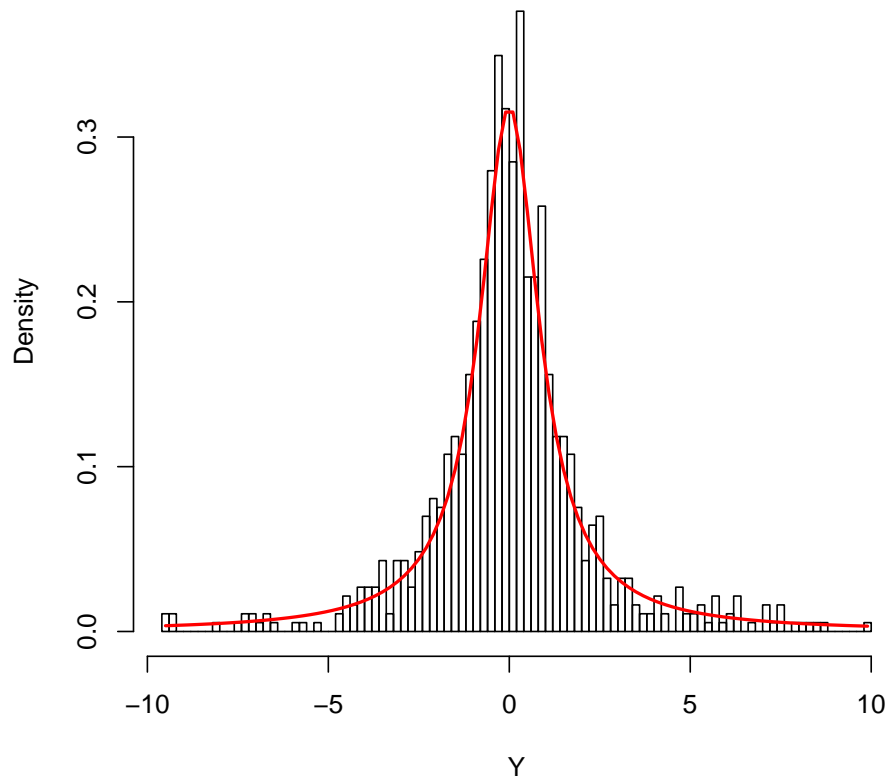
[2] If $u < \tilde{p}(y)/M\tilde{q}(y)$ return $X = y$, otherwise goto [1].

Nowhere in this algorithm did we need to compute $p(x)$ or $q(x)$.

Example: The random variable X has probability density

$$p(x) \propto \frac{\sin^2(x)}{x^2}, \quad -\infty < x < \infty$$

Give a rejection algorithm which simulates X using iid simulation of $Y \sim q$ with $q(y) \propto (1 + y^2)^{-1}$ (the Cauchy distribution).



Step 1: bound \tilde{p}/\tilde{q} .

$$\begin{aligned}\tilde{p}(x)/\tilde{q}(x) &= \frac{(1+x^2)\sin^2(x)}{x^2} \\ &= \frac{\sin^2(x)}{x^2} + \sin^2(x) \\ &\leq 2\end{aligned}$$

because $|\sin(x)/x| < 1$. We can take $M = 2$.

Step 2: we have the following algorithm for $X \sim p(x)$.

[1] Simulate $y \sim t(1)$ and $u \sim U(0, 1)$.

[2] If $u < (1+y^2)\sin^2(y)/2y^2$ return $X = y$, otherwise goto [1].

The histogram at the start of this example was produced from the implementation of this algorithm in the R-code associated with this lecture.

Comment: You will come across the t-distribution later this term in Part A Statistics. The t-distribution with one degree of freedom (called $t(1)$, or the Cauchy distribution) has density

$$q(y) \propto (1 + y^2)^{-1}.$$

Y is easy to simulate. If

$$Z_1, Z_2 \sim N(0, 1)$$

are iid standard normal and we set

$$Y = Z_1/Z_2$$

then $Y \sim t(1)$.

We can calculate the normalizing constants for both q and p using contour integrals but $F^{-1}(u)$ is very hard work for X .

Why does “unnormalized” rejection work? Let $M' = MZ_q/Z_p$.
The test

$$u < \tilde{p}(y)/M\tilde{q}(y)$$

is the exactly the same as the test

$$u < p(y)/M'q(y).$$

We cant calculate M' , p or q , but if we use M with \tilde{p} and \tilde{q} it is just as if we were using M' with p and q . However,

$$M \geq \frac{\tilde{p}(x)}{\tilde{q}(x)} \Leftrightarrow \frac{MZ_q}{Z_p} \geq \frac{\tilde{p}(x)Z_q}{\tilde{q}(x)Z_p} \Leftrightarrow M' \geq \frac{p(x)}{q(x)},$$

so the M' “implied” by our choice of M satisfies $M' \geq p(x)/q(x)$ for all x , and that means the revised algorithm is correct.

When we worked out the mean number of trials we worked with normalized densities. If we work with unnormalized densities (use M to bound \tilde{p}/\tilde{q}) the mean number of trials is $M' = MZ_q/Z_p$.

See you tomorrow and on Friday

Our next meeting is our problem class 4pm tomorrow in the building next door. Later this week (Friday 2pm) we meet in Evenlode in the OUCS facility at 13 Banbury road.

The second problem sheet will appear online shortly at

`http://www.stats.ox.ac.uk/~nicholls/PartASSP/`

It is due Wednesday 5pm of Week 5 at 1SPR.

Homework: you will need R to do problem sheet 2. If you dont have all-hours access to R by now (and need help) let me know.