Statistical Programming:
Week 4 Lab

Gordon Ross

Question 1

Suppose that we wish to simulate observations from the Pareto distribution, which has probability density function

$$p(x) = \beta\alpha^{\beta}x^{-(\beta+1)}, \quad x > \alpha, \quad \alpha, \beta > 0$$

Derive the cumulative distribution function for x and hence implement (in R) an algorithmic procedure for sampling random variables from f(x) using the method of inversion.

Question 2

1. In R, implement an inversion sampler to draw 1000 values from the Exponential(2) distribution.
2. Now implement a rejection sampler and draw 1000 values from the Exponential(2) distribution, using a Uniform(0,5) distribution as the proposal. Make sure you choose the correct value of M!
3. Add a counter to your code which counts how many samples from the Uniform(0,5) distribution it took before 1000 were accepted, and hence modify your code to print the acceptance rate at the end (i.e. the proportion of proposed samples which were accepted).

To check that your two above simulation algorithms are correct, try plotting the empirical kernel density estimate (essentially a histogram) of your samples, superimposed over the true Exponential(2) distribution and check they match. You could do it like this:

```
y1 <-rexpInversion(1000,2)
y2 <-rexpRejection(1000,2)
y3 <-rexp(1000,2)

plot(density(y1))
lines(density(y2),col='red')
lines(density(y3),col='blue')
```

Remember that you can look up the manual page for any function you dont know by putting a question mark before it when you type it into R, e.g:

```
?density
```

A slightly more formal way to check whether your code is generating values from the correct distribution is to use a hypothesis test. A two-sample goodness-of-fit test like the Kolmogorov-Smirnov test compares two sets of observations with the null hypothesis being that they come from the same distribution, and the alternative being that they dont. Skim through the Wikipedia page for this test to get a feel for how it works. You can do this test in R using:

```
ks.test(y1,y)
```

Compare the output of both your simulation algorithms (inversion and rejection) to 1,000 samples generated from the Exponential(2) distribution using rexp(1000,2). If they are from the same distribution (i.e. if your code is correct) then the p-value should be above 0.05

If you do get a p-value below 0.05 but the above kernel density plots show that your values look correct, think carefully about what the reason for this could be.

Question 3

A random variable X is said to follow a Weibull distribution with scale parameter $\lambda > 0$ and shape parameter $\gamma > 0$, short-hand $X \sim$ Weibull(λ,γ), if its cumulative distribution function F_X is

$$F_X(x;\lambda,\gamma) = \Pr(X \leq x) = 1 - e^{-(x/\lambda)^\gamma} \qquad x > 0.$$

1) Find the inverse F_X^{-1}. How does the density look like for different values of γ?
(**Hint:** Solve $F_X(x;\lambda,\gamma) = u$.)

2) Write a function in R called `sim.weibull` that takes 3 arguments: `n` (sample size), `lambda` (scale parameter) and `gamma` (shape parameter), and returns a random sample of size n from the Weibull distribution, using inversion sampling.

Question 4

Develop a rejection sampling algorithm in R that simulates n samples from the standard normal distribution $\mathrm{N}(0,1)$ using a Laplace(λ) proposal distribution

$$g(x;\lambda) = \frac{\lambda}{2} e^{-\lambda|x|} \qquad \lambda > 0, x \in \mathbb{R}.$$

Which value of λ results in the most efficient algorithm?

(**Hint:** To find M maximise $\phi(x)/g(x;\lambda)$ over x, where ϕ denotes the density of the standard normal distribution.)

Question 5

Let (X_A, X_B) denote the goals scored by team A (home) against team B (away) in a football match. A possible model for the joint probability mass function of (X_A, X_B) is

$$f(x,y) = P(X_A = x, X_B = y) = \tau_{\lambda,\mu}(x,y)\, e^{-\lambda}\, \frac{\lambda^x}{x!}\, e^{-\mu}\, \frac{\mu^x}{y!} \qquad x, y = 0, 1, 2, \ldots$$

where $\lambda, \mu > 0$,

$$\tau_{\lambda,\mu}(x,y) = \begin{cases} 1 - \lambda\mu\rho & x = y = 0 \\ 1 + \lambda\rho & x = 0, y = 1 \\ 1 + \mu\rho & x = 1, y = 0 \\ 1 - \rho & x = y = 1 \\ 1 & \text{otherwise.} \end{cases}$$

and

$$\max(-1/\lambda, -1/\mu) \leq \rho \leq \min(1/(\lambda\mu), 1). \quad \mu, \lambda > 0$$

Develop a rejection sampling algorithm in R that generates n samples from f with proposal distribution g

$$g(x,y) = e^{-\lambda}\,\frac{\lambda^x}{x!}\,e^{-\mu}\,\frac{\mu^x}{y!} \qquad x, y = 0, 1, 2, \ldots$$

(i..e an independent Poisson distribution proposal for both X and Y).

What is the probability of accepting a sample from f? What is the role of $\tau_{\lambda,\mu}(x,y)$ function in this model?

(**Hint:** This is a two-dimensional rejection sampling question, but conceptually it works the same way as in one dimension. To find optimal M maximise the ratio $f(x,y)/g(x,y)$ over x, y for the case $\rho < 0$ and $\rho > 0$, separately. To draw from g independently simulate X^* and Y^* from Poisson(λ) and Poisson(μ), respectively.)

Note: for those who are interested, this model is called the Dixon-Coles model and has moderate success in predicting football matches. See Dixon and Coles – 'Modelling Association Football Scores and Inefficiencies in the Football Betting Market', 1997