MAST90104: A First Course in Statistical Learning

Week 11 Lab and Workshop

1 Practical questions

1. The `pneumo` data from the `faraway` package gives the number of coal miners classified by radiological examination into one of three categories of pneumonoconiosis and by the number of years spent working at the coal face divided into eight categories.

   (a) Treating the pneumonoconiosis status as response variable as nominal, build a model for predicting the frequency of the three outcomes in terms of length of service and use it to predict the outcome for a miner with 25 years of service.

   (b) Repeat the analysis with the pneumonoconiosis status being treated as ordinal.

2. The following program performs a simulation experiment to estimate $\mathbb{E}X$ where the function `sim.X()` simulates a random value of X .

```
# seed position 1
# set.seed(7)
mu <- rep(0, 6)
for (i in 1:6) {
# seed position 2
# set.seed(7)
X <- rep(0, 1000)
for (j in 1:1000) {
# seed position 3
# set.seed(7)
X[j] <- sim.X
}
mu[i] <- mean(X)
}
spread <- max(mu) - min(mu)
mu.estimate <- mean(mu)
```

   (a) What is the value of `spread` used for?

   (b) If we uncomment the command `set.seed(7)` at seed position 3, then what is `spread`?

   (c) If we uncomment the command `set.seed(7)` at seed position 2 (only), then what is `spread`?

   (d) If we uncomment the command `set.seed(7)` at seed position 1 (only), then what is `spread`?

   (e) At which position should we set the seed?

3. For $X \sim \text{Poisson}(\lambda)$ let $F(x) = \mathbb{P}(X \leq x)$ and $p(x) = \mathbb{P}(X = x)$. Show that the probability function satisfies

$$p(x+1) = \frac{\lambda}{x+1} p(x).$$

   Using this write a function to calculate $p(0), p(1), \ldots, p(x)$ and $F(x) = p(0) + p(1) + \cdots + p(x)$.

   If X is a random variable with non-negative integer values and `F(x)` is a function in R that returns the cdf F of X, then as discussed in lectures X can be simulated using the following code:

```
F.rand <- function () {
u <- runif(1)
x <- 0
```

```
while (F(x) < u) {
x <- x + 1
}
return(x)
}
```

In the case of the Poisson distribution, this code can be made more efficient by calculating F jrecursively. By using two new variables, `p.x` and `F.x` for $p(x)$ and $F(x)$ respectively, modify this program so that instead of using the function `F(x)` it updates `p.x` and `F.x` within the `while` loop. Your code should have the form where you fill in the question marks:

```
F.rand <- function(lambda) {
u <- runif(1)
x <- 0
p.x <- ?
F.x <- ?
while (F.x < u) {
x <- x + 1
p.x <- ?
F.x <- ?
}
return(x)
}
```

Your code needs to ensure that at the start of the `while` loop you always have `p.x` equal to $p(x)$ and `F.x` equal to $F(x)$.

Check that your simulation works by choosing a parameter, generating a large number of random variables, using them to estimate the probability mass function, and comparing your estimates to the true values (which you can get using `dpois`).

4. (a) Here is some code for simulating a discrete random variable Y. What is the probability mass function (pmf) of Y?

```
Y.sim <- function() {
U <- runif(1)
Y <- 1
while (U > 1 - 1/(1+Y)) {
Y <- Y + 1
}
return(Y)
}
```

Let N be the number of times you go around the while loop when `Y.sim()` is called. What is $\mathbb{E}N$ and thus what is the expected time taken for this function to run?

(b) Here is some code for simulating a discrete random variable Z. Show that Z has the same pmf as Y

```
Z.sim <- function() {
Z <- ceiling(1/runif(1)) - 1
return(Z)
}
```

Will this function be faster or slower that `Y.sim()`?

2 Workshop questions

1. Suppose that $\mathbf{X} = (X_1, \dots, X_k) \sim \text{multinomial}(n, \pi)$ where $\boldsymbol{\pi} = (\pi_1, \dots, \pi_k)$. Since $X_i \sim \text{bin}(n, \pi_i)$, we have $\mathbb{E}X_i = n\pi_i$ and $\text{Var}\, X_i = n\pi_i(1 - \pi_i)$. Show that for $i \neq j$, $\text{Cov}\,(X_i, X_j) = -n\pi_i\pi_j$.

   Hint: just as for the binomial, we can write a multinomial(n, $\boldsymbol{\pi}$) as the sum of n independent multinomial(1, π) random variables.

   Alternative hint: $\text{Var}\,(X + Y) = \text{Var}\, X + \text{Var}\, Y + 2\text{Cov}\,(X, Y)$.

2. Suppose that $(X, Y, Z) \sim \text{multinomial}(n, (p_1, p_2, p_3))$. Show that

$$Y|\{X = x\} \sim \text{binomial}(n - x, p_2/(1 - p_1)).$$

   Hence obtain $\mathbb{E}(Y|X = x)$.

3. The Cauchy distribution with parameter α has pdf

$$f_X(x) = \frac{\alpha}{\pi(\alpha^2 + x^2)} \qquad -\infty < x < \infty.$$

   Write a program to simulate from the Cauchy distribution using the inversion method.

   Now consider using a Cauchy envelope to generate a standard normal random variable using the rejection method. Find the values for α and the scaling constant k that minimise the probability of rejection. Write an R program to implement the algorithm.

   Note: this is not a very efficient way of generating normals.

4. (a) Construct an acceptance-rejection sampling algorithm to generate a truncated exponential distribution, which has the pdf $p(z) = \frac{e^{-z}}{1-e^{-1}}$, $0 < z < 1$.

   (b) Calculate the mean and variance for the pdf $p(z)$ in (a).

   (c) Write an R program to implement the algorithm in (a) and use it to generate a sample of 1000 observations. Plot a histogram of the sample. Calculate the sample mean and variance, and compare them with the results in (b).

   (d) Show that the following algorithm also simulates from the distribution in (a).

   1° Generate U from Unif(0,1);

   2° If $U > e^{-1}$ then deliver $Z = -\ln(U)$; otherwise go to 1°.