MAST90104: A first course in statistical learning

Week 4 Workshop and Lab

Workshop questions

1. Suppose that X is a random variable with density function, f, given by

$$f(x) = \sum_{i=0}^{\infty} p(i)g(x;i)$$

where $p(0), p(1), \cdots$ is a discrete probability mass function on $\{0, 1, \cdots\}$ and each $g(x;i)$ is a probability density function. Suppose that $\mu(i), \sigma^2(i), M(t;i)$ are the mean, variance and moment generating function for the density $g(x;i)$. Let $M(t)$ be the moment generating function of X. Suppose also that N is a random variable with probability mass function $p(i), i = 0, 1, \cdots$. Show that

   (a) $E(X) = E(\mu(N))$
   (b) $\text{var } (X) = E(\sigma^2(N)) + \text{var } (\mu(N))$
   (c) $M(t) = E(M(t;N))$.

(Hint: You may assume that interchange of infinite sums and integrals is justified. For a random variable X with cdf F_X, the moment generating function $M_X(t) = E[e^{tX}]$)

2. Let $y_1, \ldots, y_n$ be an i.i.d. normal sample. Show that

$$\bar{y} = \tfrac{1}{n}\sum_{i=1}^{n} y_i \text{ and } s^2 = \tfrac{1}{n-1}\sum_{i=1}^{n}(y_i - \bar{y})^2$$

are independent. (*Hint*: Express them as a random "vector" and quadratic form respectively.)

3. The table below shows prices in US cents per pound received by fishermen and vessel owners for various species of fish and shellfish in 1970 and 1980. (Taken from Moore & McCabe, Introduction to the Practice of Statistics, 1989.)

Type of fish	Price (1970)	Price (1980)
Cod	13.1	27.3
Flounder	15.3	42.4
Haddock	25.8	38.7
Menhaden	1.8	4.5
Ocean perch	4.9	23.0
Salmon, chinook	55.4	166.3
Salmon, coho	39.3	109.7
Tuna, albacore	26.7	80.1
Clams, soft-shelled	47.5	150.7
Clams, blue hard-shelled	6.6	20.3
Lobsters, american	94.7	189.7
Oysters, eastern	61.1	131.3
Sea scallops	135.6	404.2
Shrimp	47.6	149.0

We will model this data using a linear model.

   (a) The linear model is of the form $\mathbf{y} = X\boldsymbol{\beta} + \boldsymbol{\varepsilon}$. Write down the matrices and vectors involved in this equation.
   (b) Using matrices, find the least squares estimators of the parameters.

(c) Calculate the sample variance s^2.

(d) A fisherman sold ocean trout for 18c/pound in 1970. Predict the price for ocean trout in 1980.

(e) Calculate the standardised residual for sea scallops.

(f) Calculate the Cook's distance for sea scallops.

(g) Does sea scallops fit the linear model? Justify your argument.

Practical exercises

Read Sections 5.1–5.3 of spuRs (which is available electronically in the library), then attempt the exercises below.

1. Last week you wrote a program to calculate $h(x, n)$, the sum of a finite geometric series. Turn this program into a *function* that takes two arguments, x and n, and returns $h(x, n)$.

   Make sure you deal with the case $x = 1$.

2. Consider the following program

```
# Program spuRs/resources/scripts/err.r

# clear the workspace
rm(list=ls())

random.sum <- function(n) {
# sum of n random numbers
x[1:n] <- ceiling(10*runif(n))
cat("x:", x[1:n], "\n")
return(sum(x))
}
```

   Below are the output of the function for $n = 10$ and $n = 5$

```
> x <- rep(100, 10)
> show(random.sum(10))
x: 6 10 7 5 8 6 5 10 9 4
[1] 70
> show(random.sum(5))
x: 8 9 4 5 10
[1] 536
```

   Explain what is going wrong and how you would fix it.

3. In this question we simulate the rolling of a die. To do this we use the function `runif(1)`, which returns a 'random' number in the range (0,1). To get a random integer in the range $\{1, 2, 3, 4, 5, 6\}$, we use `ceiling(6*runif(1))`, or if you prefer, `sample(1:6,size=1)` will do the same job.

   (a) Suppose that you are playing the gambling game of the Chevalier de Méré. That is, you are betting that you get at least one six in four throws of a die. Write a program that simulates one round of this game and prints out whether you win or lose.

   Check that your program can produce a different result each time you run it.

   (b) Turn the program that you wrote in part (a) into a function `sixes`, which returns `TRUE` if you obtain at least one six in n rolls of a fair die, and returns `FALSE` otherwise. That is, the argument is the number of rolls n, and the value returned is `TRUE` if you get at least one six and `FALSE` otherwise.

   How would you give n the default value of 4?

(c) Now write a program that uses your function `sixes` from part (b), to simulate N plays of the game (each time you bet that you get at least one six in n rolls of a fair die). Your program should then determine the proportion of times you win the bet. This proportion is an estimate of the *probability* of getting at least one six in n rolls of a fair die.

Run the program for $n = 4$ and $N = 100$, 1000, and 10000, conducting several runs for each N value. How does the *variability* of your results depend on N?

The probability of getting no 6's in n rolls of a fair die is $(5/6)^n$, so the probability of getting at least one is $1 - (5/6)^n$. Modify your program so that it calculates the theoretical probability as well as the simulation estimate and prints the difference between them. How does the *accuracy* of your results depend on N?

(d) In part (c), instead of processing the simulated runs as we go, suppose we first store the results of every game in a file, then later postprocess the results. You should read spuRs Chapter 4 to see how to read and write text files.

Write a program to write the result of all N runs to a textfile `sixes_sim.txt`, with the result of each run on a separate line. For example, the first few lines of the textfile could look like

```
TRUE
FALSE
FALSE
TRUE
FALSE
.
.
```

Now write another program to read the textfile `sixes_sim.txt` and again determine the proportion of bets won.

This method of saving simulation results to a file is particularly important when each simulation takes a very long time (hours or days), in which case it is good to have a record of your results in case of a system crash.

4. Let $\mathbf{y} = \begin{bmatrix} y_1 & y_2 \end{bmatrix}^T$ be a normal random vector with mean and variance

$$\boldsymbol{\mu} = \begin{bmatrix} 2 \\ 4 \end{bmatrix}, \quad V = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}.$$

Let

$$A = \frac{1}{4} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}.$$

.

From Theorem 3.8 we know that $\mathbf{y}^T A \mathbf{y}$ follows a χ^2 distribution with degree of freedom 1 and noncentrality parameter $\lambda = 4.5$.

(a) Generate $n = 1000$ samples $\{\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(n)}\}$ from $MVN(\boldsymbol{\mu}, V)$.

(b) Compute $\mathbf{y}^T A \mathbf{y}$ for all $\mathbf{y}^{(i)}$ that we generated in part (a).

(c) Plot the histogram of the $\mathbf{y}^T A \mathbf{y}$ values that we have computed.

(d) Now generate n samples from $\chi^2_{1,4.5}$ distribution using `rchisq()`.

(e) Plot the histogram of the generated samples on the same graph with the histogram in part (c).