M5MS04 - Computational Statistics; Coursework 4 - Autumn 2018

Question 1

This question asked us to assess the truth of the following statements:

- *The primary purpose of the bootstrap is to improve estimators* - False. While the bootstrap can be used to improve an estimator (for example, by estimating its bias then forming a new estimator that has the bootstrap estimate for the bias subtracted), the primary purpose of the bootstrap is to allow the distributional properties of an estimator (or, more broadly, a function of the data) to be understood.

- *Bootstrap confidence intervals are guaranteed to have the exact coverage probability* - False. The bootstrap relies on an approximation of the data's distribution function and, in practice, we can only generate a finite number of bootstrap samples. Both of these factors mean that bootstrap confidence intervals will have a coverage probability that is not exactly equal to their 'target' coverage probability.

- *In a Metropolis Hastings algorithm, the value of the Markov chain changes at every step* - False. Consider being at a particular location in the chain just after a proposal has been generated. If the proposal is rejected, the value of the chain at the next location will, by definition of the Metropolis-Hastings algorithm, be equal to its value at the current location. Alternatively, a proposal may be accepted, but may in fact be equal to the chain's current value anyway. In either case, the value of the Markov chain will not change.

- *It is possible that a Markov chain gives independent samples* - True. By definition, a Markov chain $\{X_n\}$, where $n \in \mathbb{N} \cup \{0\}$, is a sequence of random variables that satisfy the Markov property:

  $$P(X_{n+1}|X_n, X_{n-1}, ..., X_0) = P(X_{n+1}|X_n) \tag{1}$$

  Dependence is therefore not required for a sequence of random variables to qualify as a Markov chain. If

  $$P(X_{n+1}|X_n, X_{n-1}, ..., X_0) = P(X_{n+1}) \tag{2}$$

  then the Markov property is satisfied and the random variables in the sequence are independent (i.e. $P(X_{n+1}, X_n) = P(X_{n+1})P(X_n)$). Since time-homogeneity is not necessary to be a Markov chain, any sequence of valid distribution functions $\{F_n\}$ that specify independent random variables produces a Markov chain. A concrete example of a chain that produces independent samples is $\{X_n\} : X_n \sim \text{Bernoulli}(p)$, where $p = \frac{1}{n}$ if $n \neq 0$, 1 otherwise.

Question 2

The Markov chain (X_t) on the discrete state space $\{1, 2, 3, 4\}$ has the transition matrix

$$P = \begin{pmatrix} 0 & 0.2 & 0.8 & 0 \\ 0.5 & 0 & 0 & 0.5 \\ 0.7 & 0 & 0 & 0.3 \\ 0 & 0.1 & 0.9 & 0 \end{pmatrix} \tag{3}$$

The transition probability $P(X_4 = 3|X_2 = 2)$ is given by $P^2_{2,3} = 0.85$.

The chain defined by this transition matrix is not ergodic. Formally, ergodicity requires that there exists $k \in \mathbb{N}$ such that P^k has strictly positive entries. No such k exists for this matrix. To see that this is the case, consider that for a 4×4 matrix of real numbers with zeros forming the pattern

$$A = \begin{pmatrix} 0 & a & b & 0 \\ c & 0 & 0 & d \\ f & 0 & 0 & g \\ 0 & h & j & 0 \end{pmatrix} \tag{4}$$

Multiplying this matrix by itself produces A^2, which has zeros in the pattern:

$$\begin{pmatrix} 0 & a & b & 0 \\ c & 0 & 0 & d \\ f & 0 & 0 & g \\ 0 & h & j & 0 \end{pmatrix}^2 = \begin{pmatrix} ac + bf & 0 & 0 & ad + bg \\ 0 & ca + dh & cb + dj & 0 \\ 0 & fa + gh & fb + gj & 0 \\ hc + jf & 0 & 0 & hd + jg \end{pmatrix} = \begin{pmatrix} k & 0 & 0 & l \\ 0 & m & n & 0 \\ 0 & p & q & 0 \\ r & 0 & 0 & s \end{pmatrix} \tag{5}$$

Premultiplying by A again returns the matrix to the form of A, in the sense that the zero and non-zero elements return to their original locations.

$$A^3 = \begin{pmatrix} 0 & a & b & 0 \\ c & 0 & 0 & d \\ f & 0 & 0 & g \\ 0 & h & j & 0 \end{pmatrix} \begin{pmatrix} k & 0 & 0 & l \\ 0 & m & n & 0 \\ 0 & p & q & 0 \\ r & 0 & 0 & s \end{pmatrix} = \begin{pmatrix} 0 & am + bp & an + bq & 0 \\ ck + dr & 0 & 0 & cl + ds \\ fk + gr & 0 & 0 & fl + gs \\ 0 & hm + jp & hn + jq & 0 \end{pmatrix} \tag{6}$$

A^4 will then be in the form of A^2, after which A^5 will be in the form of A, and so on indefinitely. In short, A^k will have at least the same zeros as A if k is odd, and at least the same zeros as A^2 if k is even. Since all natural numbers are either odd or even, we can conclude that not every entry of A^k is strictly positive for any k. Since the transition matrix P contains the same pattern of zeros as A, we can be confident that its corresponding Markov chain is not ergodic.

> *Aim to construct two different ergodic transition matrices P for (X_t) that have $\pi = (0.1, 0.6, 0.2, 0.1)$ as stationary distribution. For the first, construct it such that $P_{ij} = 0.1$ for all $j > i$. For the second, construct it such that $P_{ij} = 0.2$ for all $i > j$. Comment.*

If $P_{ij} = 0.1$ for $j > i$, then the elements above the diagonal of P_1 are equal to 0.1. Furthermore, if P_1 satisfies detailed balance with π, then π is iuts stationary distribution. The detailed balance condition is $\pi_i P_{ij} = \pi_j P_{ji}$, i.e. $(\pi^T, \pi^T, \pi^T, \pi^T) \circ P_1$ is symmetric, where $\circ$ denotes the Hadamard product.

Let a denote the upper-triangular value (in this case 0.1) and introduce variables $b, ..., k$ such that

$$(\pi^T, \pi^T, \pi^T, \pi^T) \circ P_1 = \begin{pmatrix} \pi_1 b & \pi_1 a & \pi_1 a & \pi_1 a \\ \pi_2 c & \pi_2 d & \pi_2 a & \pi_2 a \\ \pi_3 e & \pi_3 f & \pi_3 g & \pi_3 a \\ \pi_4 h & \pi_4 i & \pi_4 j & \pi_4 k \end{pmatrix} \tag{7}$$

The symmetry requirement that detailed balance enforces on this matrix enables the lower-triangular variables c, e, f, h, i, j to be expressed in terms of a (e.g. $c = \frac{\pi_1}{\pi_2} a$). This further means that the row sums of P must satisfy the equations:

$$\begin{aligned} 3a + b &= 1 \\ (\frac{\pi_1}{\pi_2} + 2)a + d &= 1 \\ (\frac{\pi_1}{\pi_3} + \frac{\pi_2}{\pi_3} + 1)a + g &= 1 \\ (\frac{\pi_1}{\pi_4} + \frac{\pi_2}{\pi_4} + \frac{\pi_3}{\pi_4})a + k &= 1 \end{aligned}$$

Denote the coefficient of a in these equations as m_i, where i denotes the row number. b, d, g, k must all be nonnegative. To satisfy this nonnegativity requirement, a must be such that $\max(m_i)a \leq 1$. In the context of this problem, $\max(m_i) = m_4 = 9$, and we have $a = 0.1$ - $\max(m_i)a = 0.1$, so the nonnegativity requirement is satisfied. The full system of equations specified so far (there are 10) can therefore be solved to obtain the transition matrix P_1. This is precisely what the following code does.

```
library(MASS)
rm(list=ls())
pi <- c(0.1, 0.6, 0.2, 0.1)
n  <- length(pi)
a  <- 0.1
m  <- c(3, pi[1]/pi[2] + 2, (pi[1]+pi[2])/pi[3] + 1, sum(pi[1:3])/pi[4])
print(max(m)*a) # Must be l.e.q. to 1

## [1] 0.9

diagP      <- 1 - m*a # solve for b,d,g,k
A          <- matrix(rep(a, n^2), nrow=4)
upperTriP  <- A*upper.tri(A)
upperTriPi <- apply(upperTriP, 2, function(col) return(col*pi))
lowerTriP  <- apply(upperTriPi, 1,
                    function(row) return(row/pi)) # solve for c,e,f,h,i,j
P_dash     <- lowerTriP + upperTriP
P          <- P_dash + diag(diagP)

print(P) # Transition matrix with a in upper tri location

##             [,1]      [,2] [,3] [,4]
## [1,] 0.70000000 0.1000000 0.10  0.1
## [2,] 0.01666667 0.7833333 0.10  0.1
## [3,] 0.05000000 0.3000000 0.55  0.1
## [4,] 0.10000000 0.6000000 0.20  0.1

print(rbind(pi %*% P, pi)) # Verify stationary distribution

##    [,1] [,2] [,3] [,4]
##     0.1  0.6  0.2  0.1
## pi  0.1  0.6  0.2  0.1
```

The second part of this question requests a transition matrix P_2 with the same stationary distribution but all elements below the diagonal equal to 0.2. The stationarity condition is

$$\pi = \pi P_2 \tag{8}$$

This implies that we require $\pi_1 = \pi_1 P_{11} + (\pi_2 + \pi_3 + \pi_4)0.2$, or, equivalently:

$$P_{11} = 1 - \frac{\pi_2 + \pi_3 + \pi_4}{\pi_1} 0.2 \tag{9}$$

Evaluating this expression yields $P_{11} = 1 - \frac{0.6+0.2+0.1}{0.1}0.2 = -0.8$. It follows that any P_2 that satisfies both the stationarity condition and has values of exclusively 0.2 below its diagonal will contain at least one negative value, precluding it from being a valid transition matrix.

An intuitive explanation for this result is as follows. Under the proposed stationary distribution, and in the stationary regime, 90% of chain states will correspond to either state 2, 3, or 4. If we stipulate that 20% of transitions from all of these states must lead to state 1, then it follows that at least 18% of chain states must be state 1. But this is a contradiction - 90% + 18% = 108%. This nonsense stems from the fact that we have simultaneously required that 90% of chain states are *not* state 1, while 20% of transitions from this 90% *lead to* state 1.

Incidentally, the smallest value that can be used below the diagonal is $0.\dot{1}$.

Question 3

We consider the density f given by

$$f(x) = k(1-x)^2, x \in [2,5] \tag{10}$$

and $f(x) = 0$ otherwise, where $k > 0$ is some normalising constant. To simulate random variables with density f using a Metropolis-Hastings sampler, we use a random walk proposal distribution

$$Y|X_{t-1} \sim \mathcal{N}(X_{t-1}, \sigma^2) \tag{11}$$

Since the density of $Y|X_{t-1}$ is equal to that of $X_{t-1}|Y$, we can omit the proposal densities when calculating the acceptance probability.

```
rm(list=ls())
set.seed(3248741)

# unnormalized density
f_unnorm <- function(x) ifelse((x >= 2 & x <= 5), (1 - x)^2, 0)

# use a random walk proposal distribution
qdensity    <- function(x, y, sd=1) dnorm(y, mean=x, sd=sd)
getProposal <- function(x, sd=1) rnorm(1, mean=x, sd=sd)

realizeChain <- function(N, X0=4.5, sd=1){
                  X <- rep(NA, N)
                  X[1] <- X0
                  for(n in 2:N){
                    Y <- getProposal(X[n-1], sd)
                    a <- f_unnorm(Y)/f_unnorm(X[n-1]) # Symmetric proposal density
                    if(runif(1) <= a) X[n] <- Y
                    else              X[n] <- X[n-1]
                  }
                  return(X)
                }

X <- realizeChain(N=10000, sd=0.4) # sd tuned by inspecting acf plots

k <- 1/integrate(f=f_unnorm, lower=2, upper=5)$value # Normalizing constant
integrate(f=function(x) k*f_unnorm(x), lower=2, upper=5)$value # Check this is 1
```

```
## [1] 1
```

```
EX2_numeric <- integrate(f=function(x) k*x^2*f_unnorm(x), lower=2, upper=5)
EX2_numeric # Estimate of E[X^2] via numerical integration
```

```
## 16.81429 with absolute error < 1.9e-13

mean(X^2) # Estimate of E[X^2] via MCMC

## [1] 16.75366

PXlt3_numeric <- integrate(f=function(x) k*f_unnorm(x), lower=2, upper=3)
PXlt3_numeric # Estimate of P(X < 3) via numerical integration

## 0.1111111 with absolute error < 1.2e-15

mean(X < 3) # Estimate of P(X < 3) via MCMC

## [1] 0.1154
```

Increasing the chain length N caused the Markov chain Monte Carlo (MCMC) estimate of $\mathrm{E}[X^2]$ to approach the value obtained by numerical integration. The same is true of the MCMC estimate for $P(X < 3)$. Both of these results suggested that the chain is functioning as intended. Furthermore, the trace plot of the chain (see Figure 1) revealed that the transition properties of the chain seemed to be acceptable (i.e. the chain moved regularly and explored the whole expanse of the density's support). Repeatedly realizing the chain under the same starting value yielded estimates the were slightly variable, however realizing the chain under different starting values did not seem to increase this variability. Crucially, the MCMC estimates were distributed about the estimate obtained by numerical integration (which is likely to be very close the the actual values of $\mathrm{E}[X^2]$ and $P(X < 3)$). The chain's insensitivity to its starting value provides further evidence that it is reliable.

```
EX2_MCMC   <- c()
PXlt3_MCMC <- c()
for(X0 in seq(2, 5, by=0.5)){
  EX2_MCMC   <- c(EX2_MCMC, mean(realizeChain(N=10000, sd=0.4, X0=X0)^2))
  PXlt3_MCMC <- c(PXlt3_MCMC, mean(realizeChain(N=10000, sd=0.4, X0=X0) < 3))
}
print(EX2_MCMC)   # E[X^2] is somewhat sensitive to the choice of starting value

## [1] 17.08583 16.72662 17.31892 16.72016 16.76120 16.97358 16.99531

print(PXlt3_MCMC) # P(X < 3) is also sensitive to the choice of starting value

## [1] 0.0985 0.1155 0.1172 0.1076 0.0894 0.1365 0.1028

EX2_MCMC   <- c()
PXlt3_MCMC <- c()
for(X0 in rep(4.5, 5)){
  EX2_MCMC   <- c(EX2_MCMC, mean(realizeChain(N=10000, sd=0.4, X0=X0)^2))
  PXlt3_MCMC <- c(PXlt3_MCMC, mean(realizeChain(N=10000, sd=0.4, X0=X0) < 3))
}
print(EX2_MCMC)

## [1] 16.91395 16.99770 16.63245 16.82614 16.45378

print(PXlt3_MCMC) # Both estimates seem stable under a single starting value

## [1] 0.1178 0.1027 0.1083 0.1030 0.1098
```

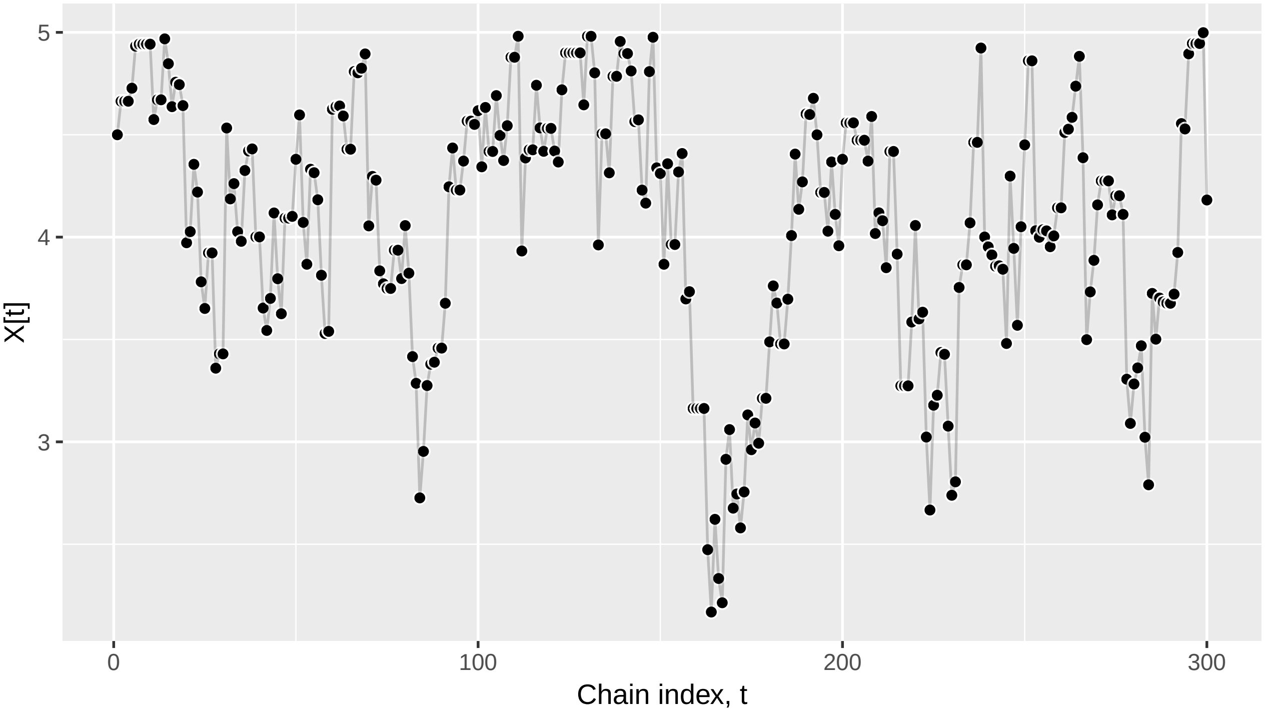

Figure 1: Trace plot of the first 300 iterations of Question 3's Markov chain. As is desirable to achieve a low autocorrelation (and therefore a low-variance MCMC estimate), the chain takes moderately-sized steps relatively regularly.

Question 4

We would like to test the null hypothesis that the first 30 and the last 20 points of the vector `X` come from the same distribution. The alternative hypothesis is that the median of the first group is larger than the median of the second group. The test statistic used is the difference in medians between a set of 30 and a set of 20 observations.

The data's distribution under the null hypothesis was approximated using the empiricial distribution of the entire sample. The test statistic was replicated $B = 9999$ times using this approximation. The distribution of these replicates indicated that the observed value of the difference in medians is slightly smaller than what would be necessary to reject the null at the $\alpha = 0.05$ significance level ($\hat{p} = 0.0589$). On these grounds, the null hypothesis was not rejected.

Figure 2 reveals the sensitivity of this conclusion to the choice of bootstrap sample size. At $B = 9$, the p-value estimator cannot return a value of lower than 0.1, meaning that the null cannot possibly be rejected. At $B = 99$, the estimate fluctuates to either side of the 0.05 decision boundary, making it unclear which test outcome should be specified. Moreover, the estimator's resolution at this value of B is only $\frac{1}{1+99} = 0.01$. The variance of the test result decreases further as B increases, resulting in a stable 'do not reject' test decision for $B \geq 9999$.

```r
set.seed(7809423)
load(file="sample.Rout") # loads data into vector X
testStatistic <- function(X) median(X[1:30]) - median(X[31:50]) # test statistic

getBootstrapSample  <- function(X) sample(X, replace=T)

getBootstrapRep     <- function(X, statistic) statistic(getBootstrapSample(X))

runBootstrapHypTest <- function(X, B=9999, statistic=testStatistic) {
```

```
                         T_tilde <- statistic(X)
                         T_star  <- rep(NA, B)
                         for(i in 1:B) T_star[i] <- getBootstrapRep(X, statistic)
                         p_hat <- (1 + sum(T_star > T_tilde))/(B+1)
                         return(p_hat)
                        }

runBootstrapHypTest(X, B=9999, statistic=testStatistic) # Null hyp cannot be rejected

## [1] 0.0589
```

The test result is borderline, so it may be the case that collecting further data from either group will allow us to conclude that the difference between the medians of either group is significantly positive at level 0.05.

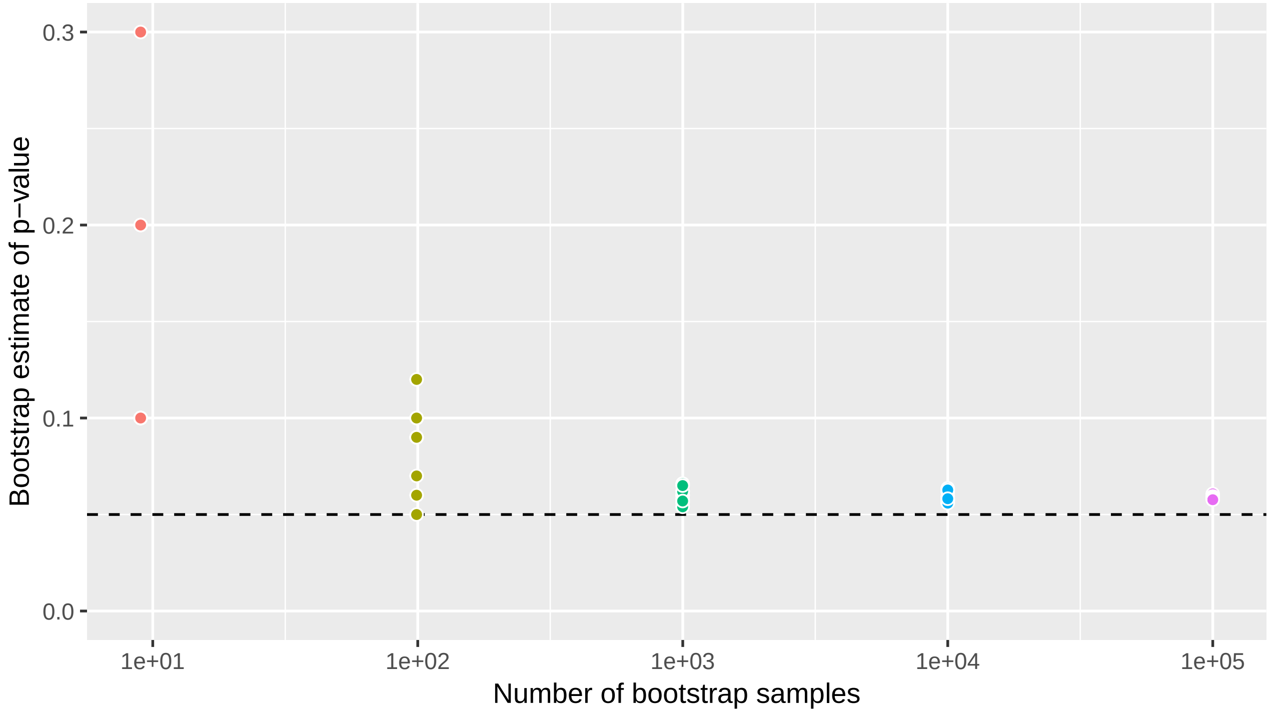

Figure 2: Bootstrap p-value estimates at $B \in \{99, 999, 9999, 99999\}$, with ten runs each. At $B = 9$, the p-value estimator cannot return a value of lower than 0.1. At $B = 99$, the estimate fluctuates to either side of the 0.05 decision boundary (dashed line), making it unclear which test outcome should be specified. The variance of the p-value estimator decreases further as B increases, resulting in an apparently stable ‘do not reject’ test decision for $B \geq 9999$.

END OF DOCUMENT