Course work 1: Wireless transmission

Background

A computer company is developing a new wireless transmission technology and have asked for your help in studying two models of operation. The first relates to one way communication and the second relates to two way communication.

One way communication

Here a single transmitter is trying to deliver a message, made of k parts, to a receiver. At each time step the transmitter attempts to transmit one part of the message to the receiver and is successful with probability p, independent of all previous transmissions. The transmitter is always knows if the transmission was successful or not. In addition at each time step the receiver might give up and leave with probability q, independent of all previous actions. So at each time step:

- With probability q the receiver leaves and never receives the message.
- With probability $p(1-q)$ the receiver does not leave and remaining number of parts is reduced by 1.
- With probability $(1-p)(1-q)$ the receiver does not leave and remaining number of parts stays the same.

We can then model this as a Markov chain with state space $\{0,...,k\}\cup\mathcal{X}$ where the states $\{0,...,k\}$ represent the number of parts left to send and the special state $\mathcal{X}$ represents the receiver having given up.

Tasks

1. For $k=4$ write down the transition matrix for this Markov chain and identify the absorbing states. *(2 marks)*

   Matrix for Question 1, writing states in the order $0,1,2,3,4,\mathcal{X}$:

$$P=\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ p(1-q) & (1-p)(1-q) & 0 & 0 & 0 & q \\ 0 & p(1-q) & (1-p)(1-q) & 0 & 0 & q \\ 0 & 0 & p(1-q) & (1-p)(1-q) & 0 & q \\ 0 & 0 & 0 & p(1-q) & (1-p)(1-q) & q \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

   Two closed classes $\{0\}$ and $\{\mathcal{X}\}$ and four open classes $\{1\}$, $\{2\}$, $\{3\}$, $\{4\}$.

2. Produce code to simulate this Markov chain for $k=4$ and use this code to estimate the probability that the receiver receives the complete message for $p=0.5$ and $q=0.1$. Use the transition matrix to calculate the exact probability of the receiver receiving the message and compare this to your estimate. *(5 marks)*

```
trials<-1000
hitone<-0 #Counter for number of times state 1 is hit before state 6 (the left state)
for( j in 1:trials){
X<-5# initial state of the Markov Chain
P <- matrix( c(1 , 0 , 0 , 0 , 0 , 0 ,
0.5*(1-0.1) , (1-0.5)*(1-0.1) , 0 , 0 , 0 , 0.1 ,
0 , 0.5*(1-0.1) , (1-0.5)*(1-0.1) , 0 , 0 , 0.1 ,
```

```
0 , 0 , 0.5*(1-0.1) , (1-0.5)*(1-0.1) , 0 , 0.1 ,
0 , 0 , 0 , 0.5*(1-0.1) , (1-0.5)*(1-0.1) , 0.1 ,
0 , 0 , 0 , 0 , 0 , 1 ),
nrow=6, ncol=6, byrow = TRUE) # P matrix
i<-1 # Number of steps
while(X[i]>1 && X[i]<6){
Y<-runif(1) # uniform sample
p<-P[X[i],] # Calculate the p values
p<-cumsum(p)
# update the chain
if(Y<=p[1]){
X[i+1]= 1
}else if(Y<=p[2]){
X[i+1]=2
}else if (Y<=p[3]){
X[i+1] = 3
}else if (Y<=p[4]){
X[i+1] = 4
}else if (Y<=p[5]){
X[i+1] = 5
}else if (Y<=p[6]){
X[i+1] = 6
}
i<-i+1
}
if(X[i]==1){
hitone<-hitone+1
}else{
hitone<- hitone+0
}
}
probest<-hitone/trials
```

Alternatively simulating directly we have

```
trials=10000

#system parameters
k=4
p=0.5
q=0.1

hitzero<-0 #Counter for number of times state 1 is hit before state 6 (the left state)
for( j in 1:trials){
X= k #number of message parts left to send
i=1
while(X[i]>0 && X[i]<k+1){
```

```
test<-runif(2)
if(test[1]<q){#The reciever leaves
X[i+1]=k+1
}else{
if(test[2]<p){#succesful transmission
X[i+1]=X[i]-1
}else{
X[i+1]=X[i]
}
}
i=i+1
}
if(X[i]==0){
hitzero<-hitzero+1
}else{
hitzero<- hitzero+0
}
}
probest<-hitzero/trials
```

We now want to calculate exactly the hitting probability. By looking at the one step behaviour we have:

$$\pi_0 = 1$$
$$\pi_i = p(1-q)\pi_{i-1} + (1-p)(1-q)\pi_i + q\pi_{\mathcal{X}} \text{ for } i = 1,...k$$
$$\pi_{\mathcal{X}} = 0$$

Rearranging these we have

$$\pi_i = \frac{p(1-q)}{1-(1-p)(1-p)}\pi_{i-1} = \left(\frac{p(1-q)}{1-(1-p)(1-p)}\right)^i.$$

So for $p = 0.5$ and $q = 0.1$ we have

$$\pi_4 = 0.448$$

.

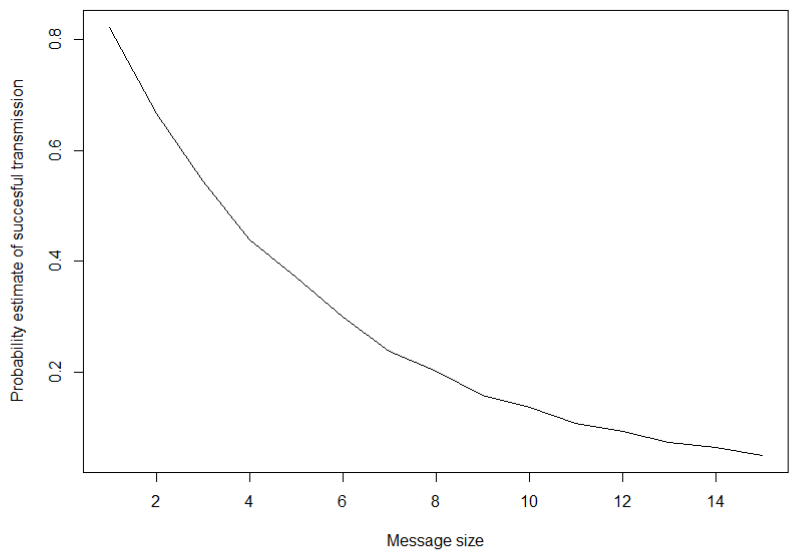
3. The company is interested in the effect of message length on the successful transmission of the message. Again for $p = 0.5$ and $q = 0.1$, produce code to estimate the probability that the receiver receives the complete message for $k = 1,...,15$. Produce a plot of the estimates against k. *(3 marks)*

```
probest=0
for(k in 1:15){
trials=10000
#system parameters
p=0.5
q=0.1

hitzero<-0 #Counter for number of times state 1 is hit before state 6 (the left state)
for( j in 1:trials){
X= k #number of message parts left to send
```

```
i=1
while(X[i]>0 && X[i]<k+1){
test<-runif(2)
if(test[1]<q){#The reciever leaves
X[i+1]=k+1
}else{
if(test[2]<p){#succesful transmission
X[i+1]=X[i]-1
}else{
X[i+1]=X[i]
}
}
i=i+1
}
if(X[i]==0){
hitzero<-hitzero+1
}else{
hitzero<- hitzero+0
}
}
probest[k]<-hitzero/trials
}
plot(1:15,probest, type='l', xlab = 'Message size',
ylab='Probability estimate of succesful transmission')
```

This produces the following plot



Two way transmission

We now consider a model for two way communication. Here we have two transmitters who are both trying to send information to each other. At each time step they randomly decide whether to attempt to broadcast or not. If they both broadcast or neither broadcast there is not a successful a transmission in that time step but if only one attempts to broadcast there will be a successful transmission. If a transmitter attempted to broadcast last time step it broadcasts with probability p_s but if they did not try to broadcast last time step they broadcast with probability p_f. This is modelled as a 3 state Markov chain with transition matrix:

$$P = \begin{bmatrix} (1-p_f)^2 & 2p_f(1-p_f) & p_f^2 \\ (1-p_s)(1-p_f) & p_f(1-p_s)+p_s(1-p_f) & p_s p_f \\ (1-p_s)^2 & 2p_s(1-p_s) & p_s^2 \end{bmatrix},$$

where state 1 is no one broadcasting, state 2 is just one broadcasting, and state 3 is both of the broadcasting. Note there is no successful transmission in state 1 or 3.

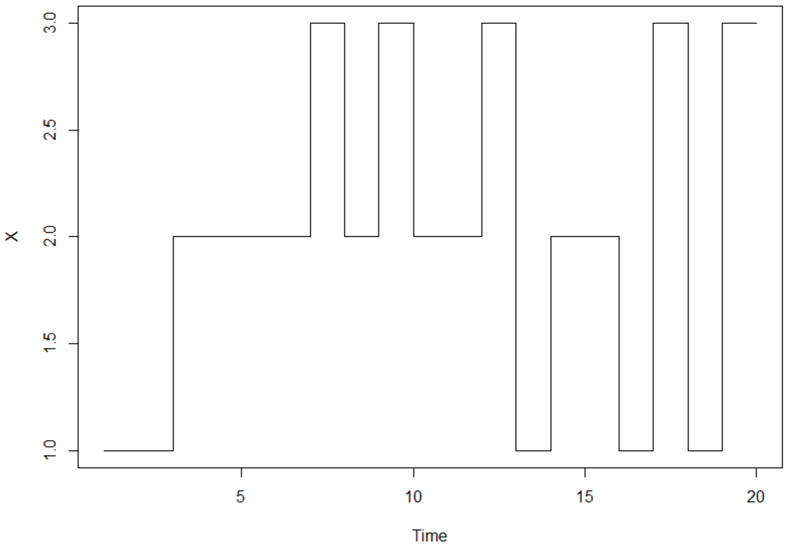
Tasks

4. Produce code to simulate the state of the system with $p_f = 0.6$ and $p_s = 0.4$. Use your code to plot a realisation of the process starting in state 1 for 20 time steps. *(2 Marks)*

```
X<-1 # initial state of the Markov Chain
n<-20 # number of steps to simulate
pf=0.6 #system parameters
ps=0.4 #system parameters
P <- matrix( c((1-pf)^2, 2*pf*(1-pf), pf^2, (1-ps)*(1-pf), pf*(1-ps)+ps*(1-pf), ps*pf, (1-p
nrow=3, ncol=3, byrow = TRUE) # P matrix

for( i in 1:(n-1)){
Y<-runif(1) # uniform sample
p<-P[X[i],] # Calculate the p values
p<-cumsum(p)
# update the chain
if(Y<=p[1]){
X[i+1]= 1
}else if(Y<=p[2]){
X[i+1]=2
}else {
X[i+1] = 3}
}
plot(1:20,X, type='s', xlab='Time')
```

An example realisation is



5. Use the simulations to estimate the long run average number of periods within which a successful transmission occurs, the proportion of time the chain is in state 2, for $p_f = 0.6$ and $p_s = 0.4$ and report this estimate in your report. Use the transition matrix to calculate the exact long run average number of periods within which a successful transmission occurs for $p_f = 0.6$ and $p_s = 0.4$.*(4 Marks)*

```
X<-1 # initial state of the Markov Chain
n<-100000 # number of steps to simulate
pf=0.6 #system parameters
ps=0.4 #system parameters
P <- matrix( c((1-pf)^2, 2*pf*(1-pf), pf^2,
 (1-ps)*(1-pf), pf*(1-ps)+ps*(1-pf),
 ps*pf, (1-ps)^2,2*ps*(1-ps),ps^2),
nrow=3, ncol=3, byrow = TRUE) # P matrix

for( i in 1:(n-1)){
Y<-runif(1) # uniform sample
p<-P[X[i],] # Calculate the p values
p<-cumsum(p)
# update the chain
if(Y<=p[1]){
X[i+1]= 1
}else if(Y<=p[2]){
X[i+1]=2
}else {
X[i+1] = 3}
}

proportion_state2<-sum(X==2)/n
```

Now we want to calculate this exactly by finding the stationary distribution for this chain. So we want to solve

$$\pi = \pi P$$

So in this case we get

$$\pi_1 = (1-p_f)^2\pi_1 + (1-ps)(1-pf)\pi_2 + (1-ps)^2\pi_3 = 0.16\pi_1 + 0.24\pi_2 + 0.36\pi_3$$
$$\pi_2 = 2p_f(1-p_f)\pi_1 + (p_s(1-p_f) + p_f(1-p_s))\pi_2 + 2p_s(1-p_s)\pi_3 = 0.48\pi_1 + 0.52\pi_2 + 0.48\pi_3$$
$$\pi_3 = p_f^2\pi_1 + p_f p_s\pi_2 + p_s^2\pi_3 = 0.36\pi_1 + 0.24\pi_2 + 0.16\pi_3$$
$$1 = \pi_1 + \pi_2 + \pi_3$$

So solving this gives

$$\pi_1 = 0.25$$
$$\pi_2 = 0.5$$
$$\pi_3 = 0.25$$

6. Use your simulation to produce a plot of the effect of varying p_s from 0 to 1 while $p_f = 0.6$ on the long run average number of periods within which a successful transmission occurs, the proportion of time the chain is in state 2. *(3 Marks)*

```
proportion_state2=0
for(k in 0:100){
X<-1 # initial state of the Markov Chain
n<-10000 # number of steps to simulate
pf=0.6 #system parameters
ps=k/100 #system parameters
P <- matrix( c((1-pf)^2, 2*pf*(1-pf), pf^2,
(1-ps)*(1-pf), pf*(1-ps)+ps*(1-pf),
ps*pf, (1-ps)^2,2*ps*(1-ps),ps^2),
nrow=3, ncol=3, byrow = TRUE) # P matrix

for( i in 1:(n-1)){
Y<-runif(1) # uniform sample
p<-P[X[i],] # Calculate the p values
p<-cumsum(p)
# update the chain
if(Y<=p[1]){
X[i+1]= 1
}else if(Y<=p[2]){
X[i+1]=2
}else {
X[i+1] = 3}
}

proportion_state2[k+1]<-sum(X==2)/n
}
plot((0:100)/100,proportion_state2, type='l', xlab='p_s', ylab='Proportion of succesful tra
```

This then produces a plot like the following: