MC Methods HW 2

Kylie Taylor

2/20/2019

1.

See attachment.

2.

See attachment for first half.

Along with finding the eigenvector that gives us the stationary density of $\pi = [\frac{1}{2}\frac{1}{2}]$, I included another version of finding the stationary density, which is by multiplying the transition density many times by itself (in this cas 30 times). The matrix has rows that are all equal, which reveals that the transition matirx has converged to the stationary density of $\pi = [\frac{1}{2}\frac{1}{2}]$. This means that regardless of the first state, the probability of ending up in the second state is the same.

The stationary density of $\pi = [0.5\ 0.5]$ tells that if I am intersted in being in state 2, or the state after X_n realizes, then the $Pr(X_{n+1} = 1|X_n = 1) = 0.5$. I can confirm this by generating the sequence X_n that starts with a $P(X_0 = 1) = 0.5$. I sequenced over 10,000 trials of to form X_n. I found that the probability of ending up at the 10,000 state, given the 9,999 state realized is equal to 0.5, verifying the stationary density.

```
s <- c("Xn", "Xn1")
XT <- matrix(c(2/3,  1/3,
               +2/3,  1/3), nrow = 2, byrow = T, dimnames = list(s, s))
mc <- new("markovchain", states = s, byrow = T,
                transitionMatrix = XT, name = "Problem 2")
summary(mc)
```

```
## Problem 2  Markov chain that is composed by:
## Closed classes:
## Xn Xn1
## Recurrent classes:
## {Xn,Xn1}
## Transient classes:
## NONE
## The Markov chain is irreducible
## The absorbing states are: NONE
```

```
mc^100
```

```
## Problem 2^100
##  A  2 - dimensional discrete Markov Chain defined by the following states:
##  Xn, Xn1
##  The transition matrix  (by rows)  is defined as follows:
##          Xn       Xn1
## Xn  0.6666667 0.3333333
## Xn1 0.6666667 0.3333333
```

```
set.seed(12345)
X <- matrix(rbinom(10000, 1, 0.5), nrow=10000, ncol=1)
apply(X, 2, mean)
```

```
## [1] 0.5041
```

3.

In this problem, we are asked to estimate the integral

$I = \int_{-\infty}^{\infty} \frac{1}{1-x^4} e^{-0.5x^2} dx$

using a Markov chain sample, X_n given by $X_{n+1} \sim N(\rho X_n, 1-\rho^2)$. We then need to show that the variance of the estimate of the integreal increases as ρ gets closer to 1. Paralleling notes in class, the goal is to integrate $I = \int l(x) f(x) dx$ yet sampling from f(x) is too hard so we use a transition density, $f(y) = \int p(y|x) f(x) dx$ where f(y) is the stationary density of p. This means we sample an X_0 from some $q_0(x)$, $X_1 from p(X_1|X_0)$, $X_2 from p(X_2|X_1)$ and so on. In the setting of this problem, $p(N_{n+1}|X_N) = N(\rho X_n, 1-\rho^2)$. We are able to estimate $\hat{I} = \frac{1}{N} \sum l(X_i)$.

The steps I took where:

1) Simulate an initial X_0 from a $N(0,1)$ distribution to start the chain.
2) Sample $X'_{n+1}s$ from the Markov chain sample given as $l(x) = N(\rho X_n, 1-\rho^2)$, or a normal distribution with mean $\rho * X_n$ and variance $1-\rho^2$. Then include this finding in an set.
3) Estimate $\hat{I}$ through the use of importance sampling from the set I just generated, by $\frac{1}{n} \sum_{i=1}^{n} l(x_i)$.

I used a function in R to directly integrate I, which revealed that the integral evaluates to approximately to 1.69. This means that my goal is to have the Markov chain evaluate to about the same value.

I estimated the integral with ρ values varying from 1, 0.8, 0.6, 0.5, 0.4, 0.2 and 0. The estimated values were most accurate for smaller ρ values. This is also verified by the plot that is included. The following plot reveals that estimations with larger ρ values take a longer time to converge than the estimations with small ρ values.

```r
set.seed(12)
I <- function(x) {(1/(1+x^4))*exp(-.5*x^2)}
integrate(I, -1000, 1000)
```

```
## 1.696393 with absolute error < 1.4e-05
```

```r
set.seed(87200)
gx <- function (xn) {sqrt(2*pi)/(1+(xn)^4)}
rg <- function(n) {rnorm(n, 0, 1)}
mc<- function(gx, rg, rho){
    X <- rg(1)
    samples <- {}
    K <- 0
    cEst <- {}
    i <- 1
    num <- 1000
    while (i < num) {
      K <- K + gx(X)
      samples[i] <- X
      X <- rho * X + sqrt(1-rho^2) * rnorm(1, mean = 0, sd=1)
      cEst[i] <- K/i
        i <- i+1
    }
    Estimate <- K/num
    print(Estimate)
   return(list(cEst, samples))
```

```r
}
v <- mc(gx, rg, 0.5)
```

```
## [1] 1.725765
```

```r
v0 <- mc(gx, rg, 1)
```

```
## [1] 1.498438
```

```r
v1 <- mc(gx, rg, 0.8)
```

```
## [1] 1.638165
```

```r
v2 <- mc(gx, rg, 0.6)
```

```
## [1] 1.655795
```

```r
v3 <- mc(gx, rg, 0.4)
```

```
## [1] 1.668567
```

```r
v4 <- mc(gx, rg, 0.2)
```

```
## [1] 1.692369
```

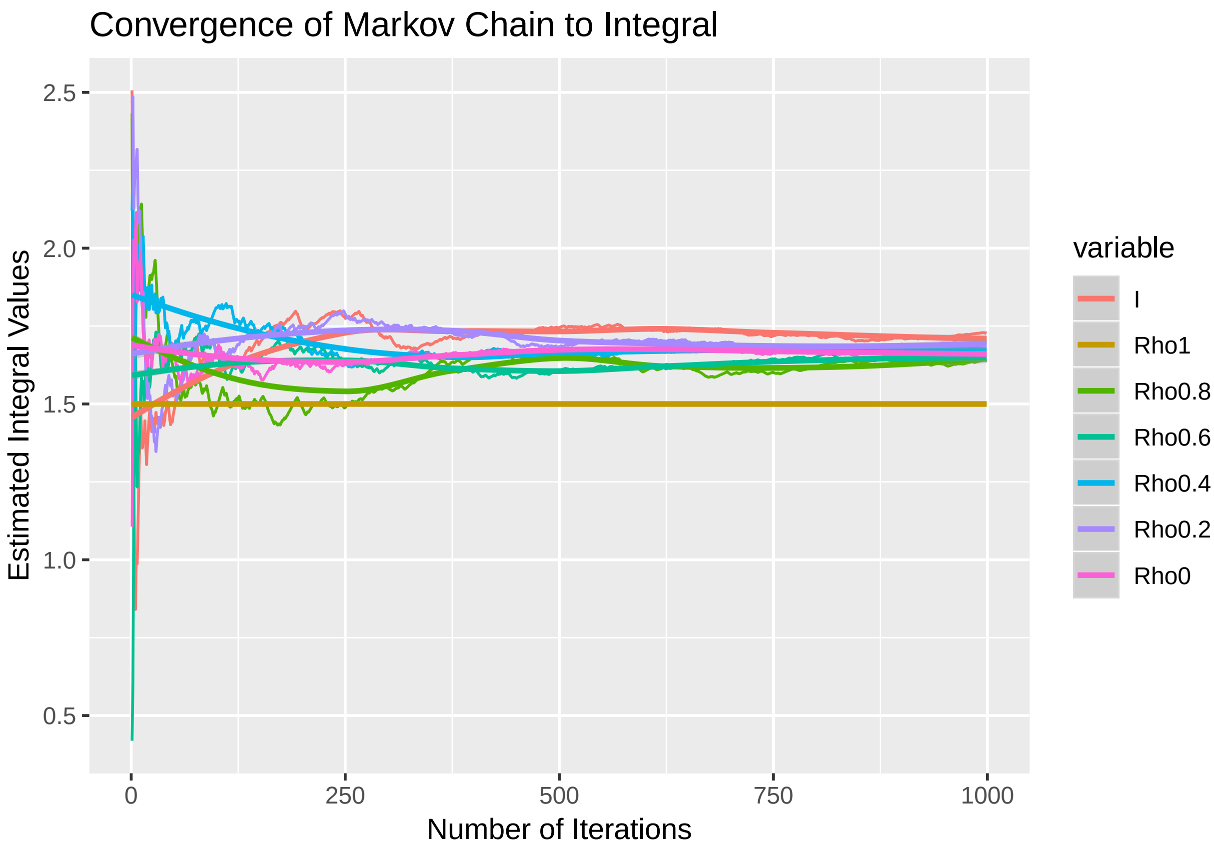
```r
v5 <- mc(gx, rg, 0)
```

```
## [1] 1.668123
```

```r
All.iterations <- as.data.frame(cbind(iterations, iterations0, iterations1, iterations2, iterations3, i

AI <- All.iterations[,-c(3,5,7,9,11,13)]
d <- melt(AI, id.vars="Iterations")
ggplot(d, aes(Iterations, value, col=variable)) +
  geom_line() +
  stat_smooth() +
  labs(x="Number of Iterations", y="Estimated Integral Values", title="Convergence of Markov Chain to I
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```


Convergence of Markov Chain to Integral
Estimated Integral Values
2.5
2.0
1.5
1.0
0.5
0
250
500
750
1000
Number of Iterations
variable
I
Rho1
Rho0.8
Rho0.6
Rho0.4
Rho0.2
Rho0