Monte Carlo error

Week5-ex2, solution

R-template `ex_monte_carlo_error_template.Rmd`

The purpose of this exercise is to examine the properties of Monte Carlo approximation. We will continue from the model answers of the Markov chain exercise. Your tasks are the following:

- Sample 100 independent realizations of length 2000 chains from the Markov chain defined in exercise 3.1 (that is; $\theta^{(1)}, \dots, \theta^{(2000)}$) using each of the combinations of ϕ and σ^2 in the rows of the below table

$\mathrm{Var}[\theta^{(i)}]$	ϕ	σ^2	$\mathrm{Corr}[\theta^{(i)}, \theta^{(i+1)}]$
1	0	1	0
1	0.5	0.75	0.5
1	0.89	0.2	0.89
1	0.1	0.99	0.1

- With each of the chains approximate $E[\theta^{(i)}]$, $\Pr(\theta^{(i)} > 0.5)$ and $\Pr(\theta^{(i)} > 2)$ using Monte Carlo with the $n = 10$, $n = 100$ and $n = 1000$ last samples. Hence, you will construct 100 independent Monte Carlo approximations for the mean and two probabilities of θ corresponding to Markov chain sample sizes 10, 100 and 1000.
- Examine the calculated Monte Carlo approximations, compare them to the exact answers for $\theta \sim N(0, 1)$ (which is the distribution of $\theta^{(i)}$ in the limit of $i \to \infty$) and answer to the following questions:
  1. How does the Monte Carlo estimate of $E[\theta^{(i)}]$ behave with respect to the number of samples and with respect to the autocorrelation of the Markov chain?
  2. How does the Monte Carlo estimate of $\Pr(\theta^{(i)} > 0.5)$ behave with respect to the number of samples and with respect to the autocorrelation of the Markov chain?
  3. How does the Monte Carlo estimate of $\Pr(\theta^{(i)} > 2)$ behave with respect to the number of samples and with respect to the autocorrelation of the Markov chain?
  4. What kind of general conclusions can you make based on these results?

Solution

Let's construct a table of parameters leading to different Markov chains, each having the same marginal distribution $N(0, 1)$ at the limit of large number of samples but each also having different amount of autocorrelation between the samples.

```
varTheta = 1
sigma2.1 = 1
sigma2.3 = 0.2
phi.2 = 0.5
phi.4 = 0.1
phi.1 = sqrt(1-sigma2.1/varTheta)
phi.3 = sqrt(1-sigma2.3/varTheta)
sigma2.2 = varTheta*(1-phi.2^2)
sigma2.4 = varTheta*(1-phi.4^2)
```

```
table.entries = matrix(nrow=4, ncol=4, data=c(
  varTheta, phi.1, sigma2.1, phi.1,
  varTheta, phi.2, sigma2.2, phi.2,
  varTheta, phi.3, sigma2.3, phi.3,
  varTheta, phi.4, sigma2.4, phi.4
))
table.entries <- t(table.entries)  # take transpose since matrix fills in the elements in columnwise
colnames(table.entries) <- c("var(theta)", "phi", "sigma2","corr")
print(table.entries)
```

```
##      var(theta)       phi sigma2      corr
## [1,]          1 0.0000000   1.00 0.0000000
## [2,]          1 0.5000000   0.75 0.5000000
## [3,]          1 0.8944272   0.20 0.8944272
## [4,]          1 0.1000000   0.99 0.1000000
```

```
# write.table(table.entries, file="table.txt", row.names=FALSE, col.names=TRUE)
```

Let's then construct a function to perform Markov chain sampling

```
set.seed(123)
# let's first define a function to conduct the sampling
MarkovChain <- function(phi,sigma2,initial,m){
  theta = vector(length=m)
  theta[1] = initial
  for (i1 in seq(1,m,1)){
    theta[i1+1] = phi*theta[i1] + rnorm(1,0,sqrt(sigma2))
  }
  return(theta)
}
```

For this exercise it is handy to use multidimensional arrays to store the results (not necessary but saves some lines of code). Below an example:

```
arr = array(dim=c(3,2,5))
dim(arr)
```

```
## [1] 3 2 5
```

```
arr[1,1,] = 1
arr[1,2,] = 2
arr[3,2,] = 3
arr
```

```
## , , 1
##
##      [,1] [,2]
## [1,]    1    2
## [2,]   NA   NA
## [3,]   NA    3
##
## , , 2
##
##      [,1] [,2]
## [1,]    1    2
## [2,]   NA   NA
## [3,]   NA    3
```

```
##
## , , 3
##
##      [,1] [,2]
## [1,]    1    2
## [2,]   NA   NA
## [3,]   NA    3
##
## , , 4
##
##      [,1] [,2]
## [1,]    1    2
## [2,]   NA   NA
## [3,]   NA    3
##
## , , 5
##
##      [,1] [,2]
## [1,]    1    2
## [2,]   NA   NA
## [3,]   NA    3
```

Now we need to sample 100 independent realizations of length 2000 chains from the Markov chain defined in exercise 3.1 (that is; $\theta^{(1)}, \dots, \theta^{(2000)}$) using each of the combinations of ϕ and σ^2 in the rows of the above table.

With each of the chains we approximate $E[\theta^{(i)}]$, $\Pr(\theta^{(i)} > 0.5)$ and $\Pr(\theta^{(i)} > 2)$ using Monte Carlo with the $n = 10$, $n = 100$ and $n = 1000$ last samples. Hence, we will construct 100 independent Monte Carlo approximations for the mean and two probabilities of θ corresponding to Markov chain sample sizes 10, 100 and 1000.

For example the below rows would construct two independent Markov chains of lenght 2000 and calculate the Monte Carlo approximation for the mean with the last 10 samples

```
set.seed(123)
i1=1
m=2000
initial = 0
n=10
theta1 = MarkovChain(table.entries[i1,"phi"],table.entries[i1,"sigma2"],initial,m)  # sample a Markov c
theta2 = theta = MarkovChain(table.entries[i1,"phi"],table.entries[i1,"sigma2"],initial,m)  # sample a 
mean(theta1[(m-n+1):m])
```

```
## [1] 0.07540569
```

```
mean(theta2[(m-n+1):m])
```

```
## [1] -0.3171165
```

Now, we need to repeat the above steps 100 times, calculate the mean and asked probabilities for each of the 100 chains and then examine how these Monte Carlo estimates behave and match with the exact results as we vary the row of the table and n.

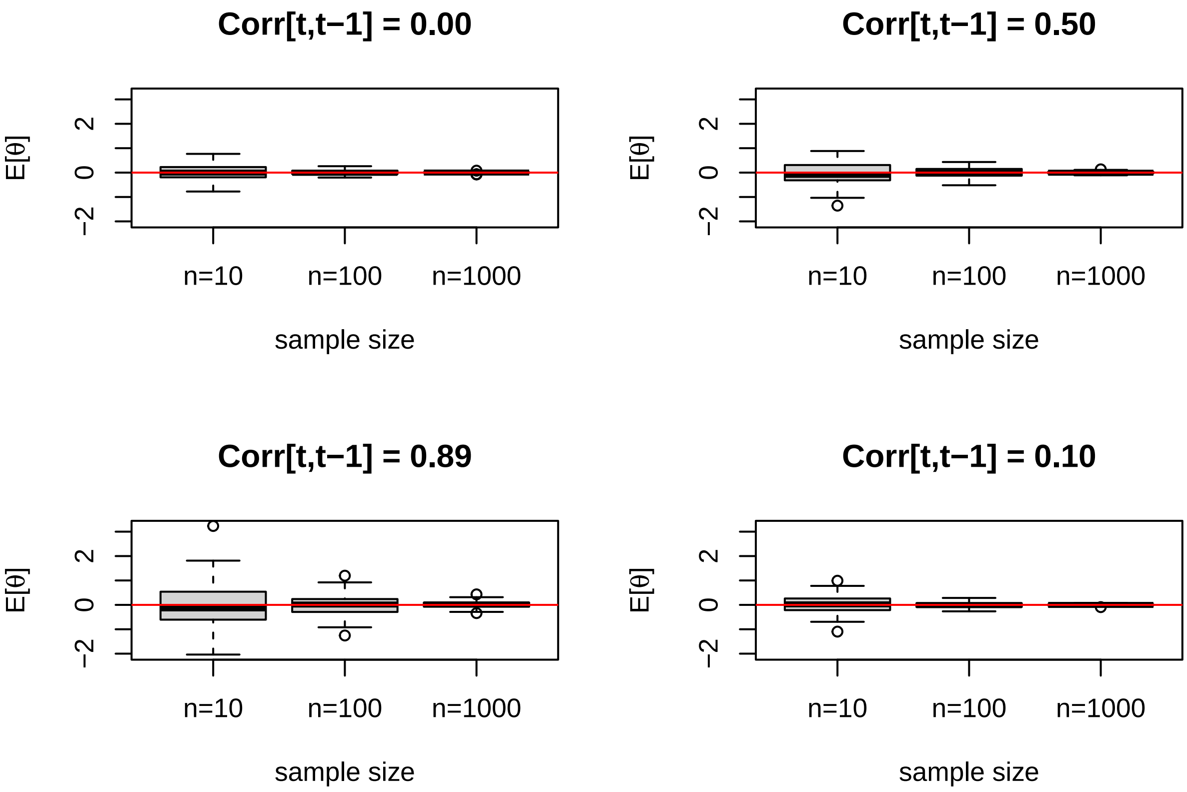
Below we calculate the 100 replicates of the Monte Carlo estimates:

```
set.seed(123)

m = 2000             # total number of samples
```

```
initial = 0          # initital value of the sample chain

# initialize arrays of size
#  alternaive parameters] x [# alternative sample sizes] x [# replicates]
# where the replicate Monte Carlo estimates are stored
summaries_mean = array(dim=c(4,3,100))
summaries_Pr05 = array(dim=c(4,3,100))
summaries_Pr2 = array(dim=c(4,3,100))
for (i1 in c(1,2,3,4)){                              # Loop through the rows of the table
  for (j1 in seq(1,100,1)){                          # Loop through the replicate Monte Carlo approximations
    theta = MarkovChain(table.entries[i1,"phi"],table.entries[i1,"sigma2"],initial,m)  # sample a Marko
    for (n in c(10,100,1000)){                       # Loop through different sizes of Monte Carlo approximat
      theta_temp = theta[(m-n+1):m]
      summaries_mean[i1,log10(n),j1] = mean(theta_temp)
      summaries_Pr05[i1,log10(n),j1] = sum(theta_temp>0.5)/length(theta_temp)
      summaries_Pr2[i1,log10(n),j1] = sum(theta_temp>2)/length(theta_temp)
    }
  }
}
```

Let't then exmine the mean estimates. In the below figures the red line denotes the true value and each box-plot shows the variation in the 100 independent Monte Carlo approximations. Based on these Figures we can answer to the questions.

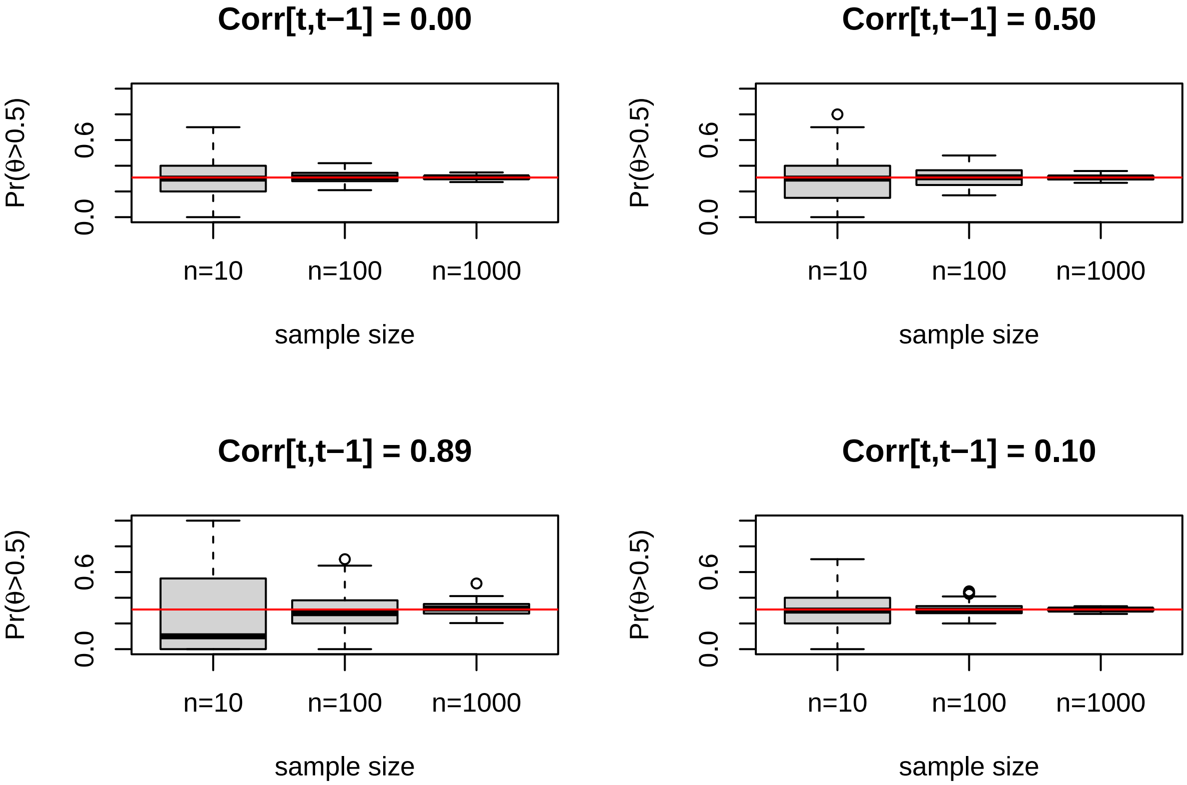
```
# Visualize variation in Monte Carlo estimates for posterio mean
par(mfrow=c(2,2))
for (i1 in c(1,2,3,4)){                              # Loop through the rows of the table
    boxplot(t(summaries_mean[i1,,]), xlab="sample size", names=c("n=10", "n=100", "n=1000"),
            main=sprintf("Corr[t,t-1] = %.2f", table.entries[i1,4]),
            ylim=c(min(summaries_mean),max(summaries_mean)), ylab=expression(paste("E[", theta, "]")))
    lines(c(0,4),c(0,0),col="red")
}
```



1 How does the Monte Carlo estimate of $E[\theta^{(i)}]$ behave with respect to the number of samples and with respect to the autocorrelation of the Markov chain?

With all sample sizes and autocorrelations the Monte Carlo estimates for the mean are centered around the true value. Hence, the estimates seem to be unbiased (in fact it can be shown that as $n \to \infty$ the Monte Carlo estimator is unbiased in for all these quantities). However, the variation in the Monte Carlo estimates increases as the sample size decreases and autocorrelation increases. For this reason, the Monte Carlo estimate is less reliable with small than large sample sizes as well as it is less reliable with chains where autocorrelation is high than in (equally long) chains where autocorrelation is small (the effective number of samples decreases with increasing autocorrelation). However, already with 100 samples the variation in Monte Carlo estimate for mean is negligible.

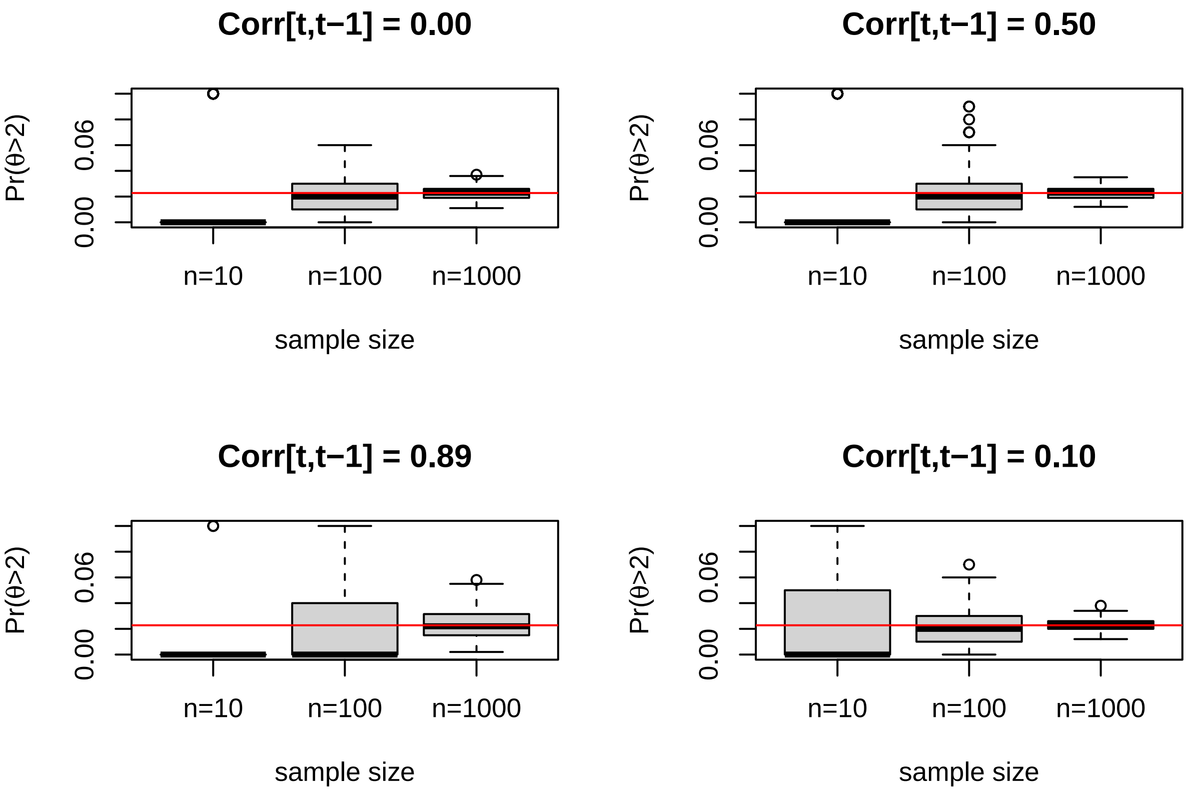
```
par(mfrow=c(2,2))
for (i1 in c(1,2,3,4)){                               # Loop through the rows of the table
    boxplot(t(summaries_Pr05[i1,,]), xlab="sample size", names=c("n=10", "n=100", "n=1000"),
            main=sprintf("Corr[t,t-1] = %.2f", table.entries[i1,4]),
            ylim=c(min(summaries_Pr05),max(summaries_Pr05)), ylab=expression(paste("Pr(", theta, ">0.5)
  lines(c(0,4),rep(1-pnorm(0.5,mean=0,sd=1),2),col="red")
}
```



2 How does the Monte Carlo estimate of $\Pr(\theta^{(i)} > 0.5)$ behave with respect to the number of samples and with respect to the autocorrelation of the Markov chain?

Here the Monte Carlo estimates behave qualitatively rather similarly as with $E[\theta^{(i)}]$. The variation in the estimates is, however, quite large relative to the true value with small sample sizes and with high autocorrelation.

```
par(mfrow=c(2,2))
for (i1 in c(1,2,3,4)){                               # Loop through the rows of the table
    boxplot(t(summaries_Pr2[i1,,]), xlab="sample size", names=c("n=10", "n=100", "n=1000"),
            main=sprintf("Corr[t,t-1] = %.2f", table.entries[i1,4]),
            ylim=c(min(summaries_Pr2),quantile(summaries_Pr2,0.95)), ylab=expression(paste("Pr(", theta
  lines(c(0,4),rep(1-pnorm(2,mean=0,sd=1),2),col="red")
}
```



3 How does the Monte Carlo estimate of Pr$(\theta^{(i)} > 2)$ behave with respect to the number of samples and with respect to the autocorrelation of the Markov chain?

The results for this case differ qualitatively from the previous two cases. When comparing $n = 100$ and $n = 1000$ the variation in Monte Carlo estimates decreases with increasing n and decreasing autocorrelation. Only with $n = 1000$ the Monte Carlo estimate are rather accurate relative to the true value. However, with $n = 10$ the variation is low in the sense that most of the Monte Carlo estimates are zero and only one or few are greater than zero. Hence, it seems that with $n = 10$ the Monte Carlo estimates are biased towards zero. Moreover, (typically) also in the case of $\text{Corr}(\theta^{(i)}, \theta^{(i)}) = 0.89$ and $n = 100$ the 100 replicates are biased towards zero. So what's going on?

The reason is that we are trying to estimate very low probabilities. Even though in the limit as $n \to \infty$ the Monte Carlo estimate $\frac{1}{n} \sum I(\theta^{(i)} > 2) \to \Pr(\theta^{(i)} > 2)$ the estimate with finite n might be very bad. Intuitively it is clear that with $n = 10$ we can approximate probabilities only in steps of $0, 0.1, \ldots, 1$. However, let's look at this more carefully. Each $I(\theta^{(i)} > 2)$ is Bernoulli distributed variable with success probability $\Pr(\theta^{(i)} > 2)$ so $\sum_{i=1}^{n} I(\theta^{(i)} > 2) \sim \text{Bin}(n, \Pr(\theta^{(i)} > 2))$. Hence, the expectation of the Monte Carlo estimate is

$E[\frac{1}{n} \sum_{i=1}^{n} I(\theta^{(i)} > 2)] = \frac{1}{n} E[\sum_{i=1}^{n} I(\theta^{(i)} > 2)] = \frac{1}{n} n \Pr(\theta^{(i)} > 2) = \Pr(\theta^{(i)} > 2)$

as expected. However, this expectation is not good summary statistics for $\frac{1}{n} \sum_{i=1}^{n} I(\theta^{(i)} > 2)]$ since the probability mass function of a Binomial distributed variable with low success probability and small sample size is very skewed. For example with $n = 10$ and $\Pr(\theta^{(i)} > 2)) = 0.025$ the probability $\Pr(\sum_{i=1}^{n} I(\theta^{(i)} > 2) = 0) = 0.78$ so that it is over three times more likely to get Monte Carlo estimate 0 than anything above zero. This leads to large relative error with large probability since estimate zero is (0.025-0)/0.025=100% smaller than the true estimate. Similarly estimate 1% would be (0.025-0.01)/0.025=60% smaller than the true estimate.

On the other hand if the true probability is large, e.g., $\Pr(\theta^{(i)} > 2)) = 1 - 0.025 = 0.975$ and $n = 10$ the probability that the estimate is 1 is $\Pr(\sum_{i=1}^{n} I(\theta^{(i)} > 2) = 10) = 0.78$ and the probability that the estimate is 0.9 is $\Pr(\sum_{i=1}^{n} I(\theta^{(i)} > 2) = 9) = 0.2$. Hence, when estimating the probability for an event that has true probability of 0.975 we make a (1-0.975)/0.975=2.5% relative error with probability 0.78 and (0.975-0.9)/0.975=7% relative error with probability 0.2. The probability for making bigger than 7% error is approximately 2%. Hence, when estimating large probabilities we get relatively accurate result already with small number of samples.

4 What kind of general conclusions can you make based on these results?

The general conclusion is that when estimating mean and variance the Monte Carlo approximation is rather reliable already with small sample sizes. When estimating probabilities that are near 0.5 or far enough from 0 or 1, the Monte Carlo approximation is rather reliable already with small sample sizes. However, the estimate gets worse if autocorrelation is large and, importantly, estimating low probabilities requires large number of Monte Carlo samples to get relatively accurate estimate.

Grading

Total 10 points The results do not need to be presented exactly similarly as in the model results as long as they are presented so that they support the claims made. Questions 1-3 give in total 3 points each so that, 1 point for a serious attempt towards right direction, 1 point more for correctly conducted experiment and 1 point more for sensible/correct discussion on the results. Question 4 gives 1 point for sensible/correct discussion on the results.