STATS 406F15 Lab 08

1 Monte-Carlo Sampling to Estimate Bias and MSE

The most common choice for evaluating estimator precision is the mean squared error,

$$MSE(\hat{\theta}) = E((\hat{\theta} - \theta)^2).$$

Example: Suppose $X_1, X_2, \ldots, X_n$ are iid $N(\theta, \theta^2)$ and we are interested in estimation of θ. Two reasonable estimators are the sample mean $\widehat{\theta}_1 = \frac{1}{n}\sum_{i=1}^n X_i$ and the sample standard deviation $\widehat{\theta}_2 = \sqrt{\frac{1}{n-1}\sum_{i=1}^n (X_i - \overline{X})^2}$. Use Monte-Carlo to compare the bias and MSE of the two estimators.

To compare these two estimators by Monte Carlo for a specific n and θ:

1. Generate $X_1, \ldots, X_n \sim N(\theta, \theta^2)$
2. Calculate $\hat{\theta}_1$ and $\hat{\theta}_2$. This gives you one realization.
3. Repeat steps 1-2 k many times.
4. Then the means of the $(\hat{\theta}_1 - \theta)^2$'s and $(\hat{\theta}_2 - \theta)^2$'s, over the k replicates, are the Monte Carlo estimators of the MSEs (likewise for Bias).

Solution:

```
## Number of replicates
num_replicate <- 1000
## Number of samples
n <- 50
## Different thetas
thetas <- seq(0.5, 10, by=0.1)
num_theta <- length(thetas)

## Initialize MSEs
MSE <- matrix(0, num_theta, 2)
## Initialize bias
bias <- matrix(0, num_theta, 2)
```

```
for (i in seq(1, num_theta))
{
    ## Generate the data for all replications, each row is a replication
    D <- matrix(rnorm(n*num_replicate, mean=thetas[i], sd=thetas[i]), n, num_replicate)

    ## Theta_hat estimted by mean
    Thetahat1 <- apply(D, 2, mean)

    ## Theta_hat estimted by standard deviation
    Thetahat2 <- apply(D, 2, sd)

    ## Record MSEs
    MSE[i, 1] <- mean((Thetahat1 - thetas[i])^2)
    MSE[i, 2] <- mean((Thetahat2 - thetas[i])^2)

    ## Record bias
    bias[i, 1] <- mean(Thetahat1 - thetas[i])
    bias[i, 2] <- mean(Thetahat2 - thetas[i])
}

## Plot MSEs
plot(thetas, MSE[, 1], xlab=quote(theta), ylab='MSE', type='l', col=1)
lines(thetas, MSE[, 2], lty=2, col=2)

## Plot bias
plot(thetas, bias[, 1], xlab=quote(theta), ylab='bias', type='l', col=1)
lines(thetas, bias[, 2], lty=2, col=2)
```

Exercise: Let $X_1, \ldots, X_n \sim Uniform(0, \theta)$. Two estimates of theta are $\hat{\theta}_1 = 2\bar{X}_n$ and $\hat{\theta}_2 = \frac{n+1}{n} \max(X_1, \ldots, X_n)$. Compare their MSEs.

2 Bootstrap

Parametric Bootstrap: Suppose $X_1, \ldots, X_n$ are i.i.d. $N(\mu, 1)$. An estimate of μ is $\bar{X}$. So a reasonable estimate of $\theta = \mu^2$ could be $\hat{\theta} = \bar{X}^2$. Write a function that estimates the bias and MSE of $\hat{\theta}$ for a choice of μ through bootstrap and for $n = 20$. Plot the bias and MSE averaged over several (say 100) realizations for different values of μ.
Algorithm: For a fixed μ

1. Generate a sample $X = \{X_1, \ldots, X_{20}\}$ from $N(\mu, 1)$.

2. Compute $\hat{\theta}(X) = \bar{X}^2$

3. For $i = 1$ up to B (say 1000) where B is the number of bootstrap samples being generated:

   (a) Generate a bootstrap sample $X^i = \{X_1^i, \dots, X_n^i\}$ from $N(\bar{X}, 1)$

   (b) Compute $\hat{\theta}^i = \bar{X^i}^2$.

4. Estimate MSE $E(\hat{\theta} - \theta)^2$ by

$$\frac{1}{B}\sum_{i=1}^{B}(\hat{\theta}^i - \hat{\theta})^2$$

   and bias by

$$\frac{1}{B}\sum_{i=1}^{B}(\hat{\theta}^i - \hat{\theta})$$

Run this for different choices of μ and over several replications report the averages.

Solution:

Exercise: Compute the MSE and Bias when $\theta = |\mu|$ is estimated using $\hat{\theta} = |\bar{X}|$.
Non-parametric Bootstrap: Suppose we observe iid values $X_1, \dots, X_n$ that are uniformly distributed on the interval (0, a), where $a > 0$ is an unknown constant. We can estimate a

using the maximum value of the sample: $\hat{a} = \max(X_1, ..., X_n)$. Since $\hat{a} < a$ it has negative bias. The relative bias is

$$\frac{E\hat{a} - a}{a}$$

It is a fact that the relative bias in this setting is $-1/(n+1)$.

```
## Number of simulation replications.
nrep = 100
## Number of bootstrap samples.
nboot = 1000
## Population value of the upper limit of the uniform distribution.
a = 2
## Sample sizes.
SS = c(5,10,20)
## Storage for the bias estimates.
Bias = NULL
## Loop over the sample sizes.
for (k in 1:3) {
## The sample size for the current iteration.
n = SS[k]
bias = NULL
for (r in 1:nrep) {
## Generate a sample from the uniform population.
X = runif(n, max=a)
## Generate non-parametric bootstrap samples.
ii = ceiling(n*runif(nboot*n))
Xboot = X[ii]
Xboot = array(Xboot, c(nboot, n))
##  Xboot = array(sample(x,n*nboot, replace= TRUE), c(nboot, n)) also works
## The bootstrap estimate of the relative bias.
MX = apply(Xboot, 1, max)
bias[r] = (mean(MX) - max(X))/max(X)
}
## The overall estimate of the relative bias.
Bias[k] = mean(bias)
}
```

3 Bootstrap confidence intervals

A major advantage of the bootstrap is that it can be applied to any estimation problem, not just estimation of the expected value. It is also very easy to apply. But it may not necessarily have good coverage properties. Here is an example where we check the coverage probabilities.

```
## Sample sizes.
N = c(10,20,40,60)
nrep = 1000 ## Number of simulation replications per sample size value.
nboot = 1000 ## The number of bootstrap data sets.
## Coverage probabilities.
CP = NULL
for (j in 1:length(N))
{
## Keep track of how many times the interval covers the true value.
nc = 0
n = N[j]
for (k in 1:nrep)
{
## Simulate a data set.
X = rnorm(n)
## Generate bootstrap data sets from X.
ii = ceiling(n*runif(n*nboot))
B = X[ii]
B = array(B, c(nboot,n))
## Get the sample mean for each bootstrap data set.
M = apply(B, 1, mean)
M = sort(M)
## Get the confidence interval lower and upper bound.
C = c(M[25], M[975])
## Check for coverage.
if ( (C[1] < 0) & (C[2] > 0) ) { nc = nc+1 }
}
CP[j] = nc/nrep
}
```