Jared Yu

Computational Statistics

Problem Set 8

*NOTE: The code for the assignment is at the end under the Appendix.*

1. Posted on the course Blackboard site is the data set CVdata.txt containing 2 lists, and , each of length 100. When written as an ordered pair, , is a sample observation and , the observed value of the density at . In this problem we will use balanced half-sampling to predict the error of fitting the curve with a normal density, . As our error metric, we will use .
   1. Plot the and use the MLE to fit with a normal distribution. Clearly state the fitted function and find the apparent error, .

First needs to be calculated.

Then, the function can be written as,

The plot of with the fitted line is below in Figure 1.

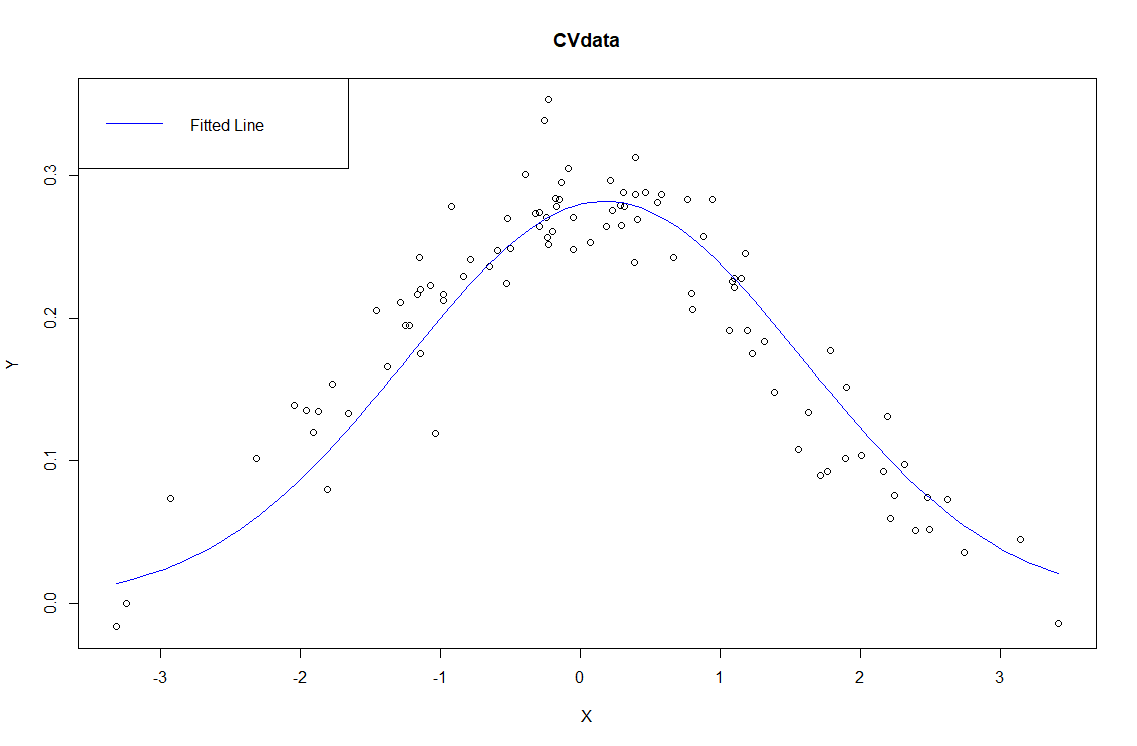


Figure 1 Plot of (X,Y) from CVdata. The blue line is the fitted line g.

Given the error metric, , then the apparent error can be written as follows,

Using R, the apparent error is roughly

* 1. Partition the data set into and and find (Eq. 12.5 in Gentle).

The formula for is as follows:

where

The values and come from the mean of and respectively. Using R, the apparent error after partitioning the dataset into two parts is roughly

* 1. Discuss your results.

In the problem we are given a dataset and a distribution that it came from. The problem with the distribution is that it has two parameters where one is known and the other is unknown. Therefore, to estimate the value of the unknown parameters, MLE is used to obtain an estimate based off the sample data. The next step is to utilize an error metric, in this case the norm, to gauge how well the fitted line is to the data itself. The difficulty is that since the same dataset was used to create the fitted line itself, there’s some contradiction in the logic of trying to understand how good the fit is.

The practical solution since there’s only one set of data is to try and partition the data using balanced half-sampling. In this case, the dataset is partitioned in to two halves and the fit is recalibrated on each of these two subsets. Then, to re-gauge the error metric, the opposite datasets will be placed into each of the other’s model fit to perform a balanced half-sampling. Then a combined type of ‘average’ of these two values is determined to be the final adjusted error metric. The trade-off of this process would be that each of the separate model fits would themselves be less accurate since there’s half of the data being used to fit an individual model.

In this case, it’s expected that the balanced half-sampling would produce a larger error in comparison to the error derived from the complete dataset. The reason being that there’s less of a case of overfit in the error metric. This result is seen in the above problems, where the error metric for the complete dataset is approximately 0.252 and the error metric from balanced half-sampling is approximately 0.0310. The results thus match what is understood from the theory and so there’s no issue.

1. (a) For , show that the jackknife variance estimate, (equation 12.11) in Gentle), can be expressed as

The following is from equation 12.11 in the text:

Then substituting and :

Then, setting , there is the following:

(b) Again, for , show that

Given that , the above can be written as follows:

This can be further simplified to:

These two equations are both similar in that they’re both a function of some value we can call . For example, they can both be rewritten as follows:

Next, to try and find the optimum of this function, the first and second derivatives will be found.

Therefore, it can be said that is the minimum of the function . From this it follows that when .

1. The statistic

is sometimes used to decide whether a least squares estimator is appropriate (otherwise, a robust method may be used).

1. What is the jackknife estimate of the standard deviation of

Letting the statistic , the variance becomes

where , , and .

Then the estimate of the standard deviation of becomes

1. Posted on the course Blackboard site is the data set Jackknife.txt containing 100 observations from a distribution. Use this sample to calculate and the jackknife estimate of the standard deviation for the cases and .

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |

1. Discuss the performance of the jackknife estimators found in (b). Be specific and use any techniques you feel are appropriate. (For example: You could draw several more samples from and use them to obtain another estimate of the standard deviation.)

In the above use of jackknife, it shows that the estimated standard deviation (standard error) of the statistic is approximately for and approximately for , while the value of itself was calculated to be approximately . The corresponding standard errors for different values of show that the statistic can be practical since the standard errors are small relative to the value of the statistic itself. If for example the standard errors were significantly larger, such as , then it could be problematic since the statistic itself is so small already. However, after doing some searching online, I couldn’t find much information about the statistic and so it’s difficult to make conclusive judgements about the statistic itself. In other words, I’m not sure if in this case a suitable value for would be anywhere from to , which would make a standard error such as still valid.

Following the example, two different new samples were created by adding random samples from the distribution in one case and random samples in another. The following table shows the calculated values of for each of the two cases. In the two cases, therefore the total sample sizes were and respectively.

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |

Furthermore, different values of were also tested. The below table shows the for the different values of . The tested values of were all chosen such that all the groups would be the same size.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

In these tests it seems apparent that from the larger number of samples, the value of the standard error will vary greatly depending on the value of . This is an interesting result, since it shows how important each of the pseudovalues possible are, since they’re calculated as if they were independent. By grouping them into larger clusters, the ability for the groups of pseudovalues to appear independent holds better. Previously, in part (b), it seemed possible that this same pattern held, but the difference it made with a sample size of was quite negligible.

It seems then that by looking at the variance of the statistic using jackknife, the statistic can be said to be appropriate for the data. When using an appropriate , the standard error of will remain within a range that appears to be relatively small enough compared to the value of itself such that the use of is not made irrelevant.

Appendix

### Problem 1  
cvdataX <- scan(file.choose()) # Load variables  
cvdataY <- scan(file.choose())  
n <- length(cvdataX)  
  
# part (a)  
# Plot the (X,Y)  
plot(cvdataX, cvdataY, main = 'CVdata',  
 xlab = 'X', ylab = 'Y') # plot data  
mu\_hat <- mean(cvdataX) # MLE mu  
x\_seq <- seq(from = range(cvdataX)[1], to = range(cvdataX)[2],  
 length.out = 100) # Generate x range  
lines(x\_seq, # Plot fitted line  
 dnorm(x = x\_seq, mean = mu\_hat, sd = sqrt(2)),  
 col = 'blue')  
legend("topleft", legend = 'Fitted Line', lty = 1, col = 'blue')  
  
g <- function(x) { # g function  
 dnorm(x = x, mean = mu\_hat, sd = sqrt(2))  
 #(1 / (2 \* sqrt(pi))) \* exp((-1 / 4) \* (x - mu\_hat)^2)  
}  
R <- function(y, g) { # R function  
 abs(y - g)  
}  
  
sum(R(y = cvdataY, g = g(cvdataX))) / n # Apparent error  
  
# part (b)  
S1 <- cvdataX[1:50]; S2 <- cvdataX[51:100] # Initialize variables  
S1\_y <- cvdataY[1:50]; S2\_y <- cvdataY[51:100]  
xbar\_1 <- mean(S1); xbar\_2 <- mean(S2)  
  
g1 <- function(x, xbar = xbar\_1) { # Create g\_1X,Y(x)  
 (1 / (2 \* sqrt(pi))) \* exp((-1 / 4) \* (x - xbar)^2)  
}  
g2 <- function(x, xbar = xbar\_2) { # Create g\_2X,Y(x)  
 (1 / (2 \* sqrt(pi))) \* exp((-1 / 4) \* (x - xbar)^2)  
}  
  
(1 / n) \* sum(R(y = S2\_y, g = g1(x = S2)) +  
 R(y = S1\_y, g = g2(x = S1))) # Partitioned apparent error  
  
### Problem 3  
# part (b)  
jackknife <- scan(file.choose()) # Load data  
b2 <- function(Y) {  
 y\_bar <- mean(Y)  
 sum((Y - y\_bar)^4) /  
 ((sum((jackknife - y\_bar)^2))^2)  
}  
b2(Y = jackknife) # 0.02669995  
  
b2\_minus\_j <- function(R\_list, j = 0) {  
 if (j == 0) { # Remove jth group, if j=0 remove none of the groups  
 # Reference: https://stackoverflow.com/questions/1335830/why-cant-rs-ifelse-statements-return-vectors  
 Z <- R\_list  
 } else {  
 # Reference: https://stackoverflow.com/questions/652136/how-can-i-remove-an-element-from-a-list  
 Z <- R\_list[-j]  
 }  
  
 # Reference: https://stackoverflow.com/questions/14924935/using-r-convert-data-frame-to-simple-vector  
 # Calculate b2 with jth group removed  
 Z <- as.vector(unlist(Z), mode = 'numeric')  
 z\_bar <- mean(Z)  
 Z\_b2 <- sum((Z - z\_bar)^4) / ((sum((Z - z\_bar)^2))^2)  
   
 return(Z\_b2)  
}  
  
J <- function(R\_j) {  
 r <- length(R\_j)  
 b2\_bar <- (1 / r) \*  
 sum(sapply(1:r,  
 function(x) { b2\_minus\_j(R\_list = R\_j, j = x) }))  
 jackknifed\_stat <- r \* b2 - (r - 1) \* b2\_bar  
 return(jackknifed\_stat)  
}  
  
se\_jack <- function(Y = jackknife, # Vector of values  
 k = 1) { # Size of each group  
 n <- length(Y) # Initialize variables  
 r <- n / k  
   
 # Reference: https://stackoverflow.com/questions/3318333/split-a-vector-into-chunks-in-r  
 # Split Y into r groups R = (r\_1, r\_2, ..., r\_r)^T  
 r\_groups <- split(Y, cut(seq\_along(Y), r, labels = FALSE))  
   
 jackknifed\_T <- J(R\_j = r\_groups) # J(T)  
 T\_stat <- b2\_minus\_j(R\_list = r\_groups, j = 0) # T  
  
 numer <- sum( # numerator  
 sapply(1:r, function(x) {  
 T\_j\_star <- r \* T\_stat - (r - 1) \*  
 b2\_minus\_j(R\_list = r\_groups, j = x)  
 (T\_j\_star - jackknifed\_T)^2  
 })  
 )  
 denom <- (r \* (r - 1)) # denominator  
 se\_JT <- sqrt(numer / denom)  
  
 return(se\_JT)  
}  
  
se\_jack(Y = jackknife, k = 1) # 0.003714044  
se\_jack(Y = jackknife, k = 5) # 0.003692711  
  
# part (c)  
se\_jack(Y = jackknife, k = 10) # 0.003362023  
se\_jack(Y = jackknife, k = 20) # 0.004758237  
  
plot(density(jackknife))  
additional\_samples <- rnorm(n = 1e4, mean = 0, sd = 1)  
jackknife\_2 <- c(jackknife, additional\_samples)  
plot(density(jackknife\_2))  
b2(jackknife\_2) # 2.517476  
  
se\_jack(Y = jackknife\_2, k = 1) # 2.653616  
se\_jack(Y = jackknife\_2, k = 5) # 1.186968  
se\_jack(Y = jackknife\_2, k = 10) # 0.8395211  
se\_jack(Y = jackknife\_2, k = 20) # 0.5939255  
  
additional\_samples2 <- rnorm(n = 1e3, mean = 0, sd = 1)  
jackknife\_3 <- c(jackknife, additional\_samples2)  
plot(density(jackknife\_3))  
b2(jackknife\_3) # 0.2688279  
  
se\_jack(Y = jackknife\_3, k = 1) # 0.7922754  
se\_jack(Y = jackknife\_3, k = 5) # 0.3549629  
se\_jack(Y = jackknife\_3, k = 10) # 0.2515717  
se\_jack(Y = jackknife\_3, k = 20) # 0.1787098