Problem Set 1

Han Li

Due: February 07, 2024

Instructions

- Please show your work! You may lose points by simply writing in the answer. If the problem requires you to execute commands in `R`, please include the code you used to get your answers. Please also include the `.R` file that contains your code. If you are not sure if work needs to be shown for a particular problem, please ask.
- Your homework should be submitted electronically on GitHub in `.pdf` form.
- This problem set is due before 23:59 on Sunday February 11, 2024. No late assignments will be accepted.

Question 1

The Kolmogorov-Smirnov test uses cumulative distribution statistics test the similarity of the empirical distribution of some observed data and a specified PDF, and serves as a goodness of fit test. The test statistic is created by:

$$D = \max_{i=1:n} \left\{ \frac{i}{n} - F_{(i)}, F_{(i)} - \frac{i-1}{n} \right\}$$

where F is the theoretical cumulative distribution of the distribution being tested and $F_{(i)}$ is the ith ordered value. Intuitively, the statistic takes the largest absolute difference between the two distribution functions across all x values. Large values indicate dissimilarity and the rejection of the hypothesis that the empirical distribution matches the queried theoretical distribution. The p-value is calculated from the Kolmogorov- Smirnoff CDF:

$$p(D \leq d) = \frac{\sqrt{2\pi}}{d} \sum_{k=1}^{\infty} e^{-(2k-1)^2\pi^2/(8d^2)}$$

which generally requires approximation methods (see Marsaglia, Tsang, and Wang 2003). This so-called non-parametric test (this label comes from the fact that the distribution of the test statistic does not depend on the distribution of the data being tested) performs poorly in small samples, but works well in a simulation environment. Write an `R` function that implements this test where the reference distribution is normal. Using `R` generate 1,000 Cauchy random variables (`rcauchy(1000, location = 0, scale = 1)`) and perform the test (remember, use the same seed, something like `set.seed(123)`, whenever you're generating your own data).

As a hint, you can create the empirical distribution and theoretical CDF using this code:

```r
  # create empirical distribution of observed data
  ECDF <- ecdf(data)
  empiricalCDF <- ECDF(data)
  # generate test statistic
  D <- max(abs(empiricalCDF - pnorm(data)))
```

```r
set.seed(123)
# Generate 1000 Cauchy random variables
data1 <- rcauchy(1000, location = 0, scale = 1)
# create empirical distribution of observed data


# Function to approximate the KS p-value using the provided series formula
ks_p_value <- function(data) {
  ECDF <- ecdf(data)
  empiricalCDF <- ECDF(data)
  # get D
  D <- max(abs(empiricalCDF - pnorm(data)))
  # Constants for the calculation
  pi_sq <- pi^2
  upper_limit <- length(data)
  # Calculate the series sum
  series_sum <- sum(sapply(1:upper_limit, function(k) {
    exp(-(2*k - 1)^2 * pi_sq / (8 * D^2))
  }))

  # Calculate the p-value using the series sum
  p_value <- sqrt(2*pi) / D * series_sum

  # Ensure the p-value is within [0,1]
  p_value <- min(max(p_value, 0), 1)
  cat('D is ',D,' with p_value of ', p_value)
  return(p_value)
}
ks_p_value(data1)
```

And the result is:

```
D is 0.1347281  with p_value of  5.652523e-29

```

We can sanity check with the built in function and the difference is likely due to rounding:

```r
#check
ks.test(data1,'pnorm')
```

```
 Asymptotic one−sample Kolmogorov−Smirnov
 testdata:  data1
D = 0.13573, p−value = 2.22e−16
 alternative hypothesis: two−sided
```

As the pvalue is extremely small and way below 0.05, we have found evidence that supports a rejection of the null hypothesis of the two distributions match.

Question 2

Estimate an OLS regression in `R` that uses the Newton-Raphson algorithm (specifically `BFGS`, which is a quasi-Newton method), and show that you get the equivalent results to using `lm`. Use the code below to create your data.

Please see code attached with step wise comments in line

```r
######################
# Problem 2
######################

set.seed (123)
data <− data.frame(x = runif(200, 1, 10))
data$y <− 0 + 2.75*data$x + rnorm(200, 0, 1.5)
# Estimate OLS using lm for comparison
lm_model <− lm(y ~ x, data = data)

# Define the log−likelihood function for the OLS model
logLikFun <− function(theta,y,X) {
  n <−nrow(X)
  k <−ncol(X)
  beta <− theta[1:k]
  sigma <− theta[k+1]

  −sum(dnorm(y, mean =X %*% beta , sd = sigma, log = TRUE))
}

# Optimize the log−likelihood function using BFGS method

 # Starting values for the coefficients
start_values <− c(1,1,1)
bfgs_model <− optim(fn=logLikFun, par=start_values, X=cbind(1,data$x),y=data$y
    , method = "BFGS",hessian = T)

# Output the coefficients from both models
list(lm_model = coef(lm_model), bfgs_model = bfgs_model$par)
```

We can compare and see the interept and slope parameteres (first two) are almost the same if we roudn to 4 decimal. The third value in the BFGS is the sigma, which is close to what we have given in the data generating process as well.

```
$lm_model(Intercept)     x    0.1391874    2.7266985
$bfgs_model[1]  0.1391977  2.7267048  1.4394898
```