Homework 3
MATH 191 Topics in Data Science
Posted: Saturday, October 31, 2015
Due in class, Friday, November 13, 2015

1 Problem 1 (20 points)

Assume that $x_1, x_2, \ldots, x_n$ are independent identically distributed instances of a p-dimensional random variable X with mean μ (of size $p \times 1$) and covariance C (of size $p \times p$). Show the following

(a)(5 points) If

$$\mu_n = \frac{1}{n}\sum_{k=1}^{n} x_k$$

show that $\mathbb{E}[\mu_n] = \mu$.

(b)(15 points) If

$$H = \frac{1}{n-1}\sum_{k=1}^{n}(x_k - \mu_n)(x_k - \mu_n)^T,$$

show that $\mathbb{E}[H] = C$. In other words, show that the expected value of H_{ij} is given by $C_{i,j}$. You may use the following result: $\mathbf{Var}[\mu_n] = \frac{1}{n}C$.

Hint: try to decompose H into

$$H = \frac{1}{n-1}\left(\sum_{i=1}^{n}(x_i - \mu)(x_i - \mu)^T\right) \quad + \quad \frac{1}{n-1}Q$$

and then consider the expected value of each of the two terms in the summation.

2 Problem 2 (35 points)

Principal Component Analysis. This programming problem is a remake of the programming assignment from Homework 2, but with added steps that perform PCA. We expand the universe and add two names which were left out in HW 2, and use a total of 10 instruments

{'SPY','^VIX', '^TNX', 'OIL','GLD','GOLD','^N225','^FTSE', 'SPXS', 'SPXL'}

(a) Write a function called run_PCA() which takes as input a matrix of size $n \times p$, performs PCA, and returns the loading of the first 5 eigenvectors. You can use the prcomp library: *myPCA = prcomp(X_train, center = TRUE, scale. = TRUE, retx=TRUE);* and then extract the loadings using myPCA$x. How much of the variance in the data do the top 5 components capture? Other useful commands to be aware of are *print(summary(w));* and *w$sdev;*.

(b) Perform an in-sample analysis of the same data set, but instead of using for X_{train} the entire data set, replace X_{train} with the top $k = 5$ principal components. Record the average daily pnl, yearly pnl, total pnl and sharpe ratio (like in Homework 2). It is fine if you choose to take a print screen of the table from the R console, as opposed to typing in all the values in the table. Also compute the average of all the above statistics (across the instruments)(Hint: just use the *colMeans* command for this).

(c) Perform an out-sample analysis of the same data set, using the same sliding window approach, but instead of using the X_{train} and X_{test} used in the last homework, do the following

- first combine X_{train} and X_{test} (using the rbind command). Denote this new matrix by Q (which is of size 101 by 10).
- perform PCA on Q, and extract the top $k = 5$ components, and denote the resulting array by $\tilde{Q}$ of size 101 by 5.
- let $\tilde{X}_{train}$ and $\tilde{X}_{test}$ denote the first 100 rows, respectively the last row, in $\tilde{Q}$.
- run a regression of y_{train} against $\tilde{X}_{train}$, obtain the betas, apply them to $\tilde{X}_{test}$, and obtain your forecast (for next day).
- repeat the above procedures across days and instruments.

Record the same statistics as in part (b), and compare the results. What can you conclude?

(d) Replace the above step in (c), replace the usual multiple regression with a knn-regression. You should install the following package

install.packages('FNN')

and then make sure you load the library

library('FNN')

To call the knn regression package use the command *knn.reg* and fix the number of k-nearest-neighbors to be $k = 10$

y_hat = knn.reg(train = X_train, test = X_test, y = y_train, k=10);
y_hat = y_hat$pred;

Record the same statistics as in part (c), and compare to the previous results. What may you conclude overall?

(e) For part (a), when performing a PCA analysis on the entire data set, also capture the top three principal components, using the command

pcaRot = myPCA$rotation

which will return a matrix of size 10×10, whose columns give the principal components. Show the plots you obtain from plotting, in pairs of two, the top 3 principal components. Here is an example of how you can nicely do this in R

plot(pcaRot[,1], pcaRot[,2] , type='p', pch=20, cex=1, col='red', xlab = 'PC1', ylab='PC2')
text(pcaRot[,1], pcaRot[,2], labels = rownames(pcaRot))

3 Problem 3 (35 points)

Applications of random matrix theory. Use the provided data set which contain the daily returns of 472 stock for a period of 561 days. You can load the data using the command *load(file = 'CorrMtx_2012_2015_SP500.data');*, and the returns are stored in variable RETS.

(a) Compute and plot a histogram of the eigenvalues of the empirical covariace matrix. What do you observe? Plot the same spectrum, but leave out the largest eigenvalues.

(b) Randomly shuffle the entries in the matrix of returns (RETS), and compute the eigenvalues. Repeat this experiment 50 times, and record the average value of the obtained eigenvalues. (You can use the sample function to randomly permute entries RETSRAND = matrix(sample(c(RETS)) , nrow = dim(RETS)[1], ncol = dim(RETS)[2])). Plot a histogram of the resulting averaged eigenvalues (You could set the "breaks" parameter to 50 in your histogram, for better visualization).

(c) Compare the largest eigenvalue obtained in (a) with the largest eigenvalue obtained in (b)

(d) Leaving out the largest eigenvalue in (a), compute a Q-Q plot of the two distributions from (a) and (b). In other words, compute the quantiles of the eigenvalues obtained in (b) (You could use the R command quantile(eigvals, probs = seq(0, 1, 0.05), na.rm = TRUE);) and then do similarly for the eigenvalues from (a) but ignoring the largest eigenvalue, and plot the two quantiles against each other. What can you conclude? Are the two distributions the same?

(e) Compute a scatter plot of the top 20 largest eigenvalues in (a) excluding the largest eigenvalue (on the x-axis), versus the top 20 largest eigenvalues in (b) (on the y-axis). Overlap on this plot the line $x = y$, i.e. use *lines(eigvalsActual[2:21], eigvalsActual[2:21], type='b')*, if eigvalsActual denotes the vector of eigenvalues from part (a). How do the the two sets of eigenvalues compare. What could you conclude?