Daisy Sun

README FILE

[1] Objective:

The objective of this project is to write a R script or more elaborately a R package (actually just a couple of functions…) (like those scripts for a function t.test(), mean(), st.dev(), lm(y~x) in R…etc). The advantage of such package is that a person need only type in specific arguments in the parentheses later to obtain desirable results.

For example, remember those days when you were taking econometrics and tried to find out how to use Probit model in R? The standard way is to use the following function "glm()" with arguments listed below.

>glm(formula = admit ~ gre + gpa + as.factor(rank), family = binomial(link = "probit"))
So the argument in red "family = …" actually refers to the model you want to use, that is the probit model.

Similarly to the construction of this function "glm()" that some smart guy already wrote in a R package, our aim is to construct a function script (say "knn( )") that do simulations based on different data generating processes.

[2] Background

The big picture of this project is actually to explore the bandwidth selection using locally weighted regression techniques, based on different data generating processes. By minimizing GCV scores, we hope to obtain a corresponding bandwidth. Meanwhile, we also care about the parameter estimators and their distributions, GCV value and SCORE value.

[3]Two data generating processes

As we discussed last term, we wanted to extend the number of variables in the data generating processes with some (if not all) coefficient parameters having a spatial-varying property. Therefore, X_1 and X_2 are important variables (randomly drawn from a known distribution) that contribute to the data generating processes.

[[1]]
Having ONE spatial-varying parameter β_1 associated with X_1 and the other NON spatial-varying (equivalent as fixed number) parameter β_2 associated with X_2;

$$Y = \beta_1 X_1 + \beta_2 X_2 + e$$

Where $\beta_1 = \mathbf{m_1\ latt + 1 - 5m_1}$; Notice, m is a parameter that needs specifying for each simulation; "latt" could be set as runif(n)*10;

One can let $\beta_2 = 1$, assuming it to be the easiest case. Of course, you can choose your favorite number, such as 10111 (a prime number).

Under this situation, I want to apply two regression models (a well-specified one and a poorly-specified one) to keep track of the estimated βs, GCV, SCORE values, and the bandwidth k where gcv is minimized. Specifically, I want to separately run $Y \sim X_1$ and $Y \sim X_1 + Z_1$ as my regression models.

knn(m1, m2, e, model = "lm($Y \sim X_1$)", β_1= " spatial varying", β_2 = " fixed value", n, loops) or

knn(m1, m2, e, model = "lm($Y \sim X_1 + X_2$)", β_1= " spatial varying ", β_2 = " fixed value", n, loops)

In both cases, function "knn( )" shall print out an array of

(1) Estimated beta hats
(2) Standard deviations of beta hats
(3) GCV values
(4) Score values
(5) The bandwidth at which GCV is minimized

[[2]]

Now, the second data generating process refers to the situation when β_1 and β_2 are both spatial- varying.

$$Y = \beta_1 X_1 + \beta_2 X_2 + e$$

For example, this could be some data values: $\beta_1 = m_1$ latt+1-5m_1, and β_2= 5m_2 long + 5-25m_2.

Again,

knn(m1, m2, e, model = "lm($Y \sim X_1$)", β_1= "spatial varying", β_2 = "spatial varying ", n, loops) or

knn(m1, m2, e, model = "lm($Y \sim X_1 + X_2$)", β_1= "spatial varying", β_2 = "spatial varying", n, loops)

[4]Conclusion

So, summarize what we have:

(1) 2 regression models – $Y \sim X_1$ or $Y \sim X_1 + Z_1$;
(2) 3 cases of data generating processes – (1) two fixed parameters (No need to use lwr and tested before. Lwr is well behaved) ; (2) one spatially-varying parameter only such as $\beta_2 = 1$; (3) two spatial-varying parameters (i.e, β_1 = 3latt+2, β_2 = 5long-10)

By inputting m_1,m_2, e, β_1, β_2 (same process of specifying data generating process) as well as specifying the regression model, I hope to find out useful information such as estimated beta hats, standard deviations, GCV and SCORE values, but most importantly, the **BANDWIDTH!**

[5]How to run this script/package (knn) in your own R ?

Step 1: Download the knn package in your local computer and save it under a disc say C:/

Step 2: Find the directory where you store your statistical software R and notice its path.

Step 3: Go to PROPERTIES under my computer . Go to Advanced System Setting. Go to Environmental Variable. Now, you will see two tables and press button "P" at the lower table (system variables). Press "Edit" Of course, you will find a list of addresses under your "variable value". Copy your R directory address in that "variable value" column at the MOST FRONT. Then click "OK" in every window.

Step 4: If you are using PC, Press now "windows+R" and find the commandline box. Type cd C"/knn. Then type R CMD INSTALL knn.

Step 5: After this, close the command line box. Go to R, and type in >library (knn) and >help(knn). You should be able to see everything there.