NCRM Intro to BDA 5 Briefly about computations in R

According to the Bayes rule, to compute the posterior probability of specific parameter values, we need to take the prior probabilities of those parameter values, multiply them by the likelihood associated with those parameter values, and divide this product by the sum of such products across all the possible values of the parameter. These computations may be quite easy, when we have only one parameter, and only a few possible values of the parameter. Such computations become much more complicated when we have continuous distributions of parameter values, and we have a large number of parameters. In such cases, doing calculations by hand may be prohibitively costly. And so, this is where the computational approaches to computational methods come to rescue.

One of the most typically used computational methods for computing the posterior distribution is called the Markov chain Monte Carlo approach. Without going into the details of how mC mC works, let me briefly outline the logic behind the process behind it.

To run Markov chain Monte Carlo, the computer or the user picks a point in the parameter space, a particular combination of parameter values, then the user of the computer specifies a rule according to which we should move from that particular point in the parameter space around the parameter space to a different point in that space. The computer the sampler repeats that operation many many many times and at some point, the distribution of the points in the parameter space visited by the sampler is no longer dependent on the initial values, the initial of the points where we start our journey. Now, it has been shown that the distribution of the points in the parameter space visited this way converges to the posterior distribution that we are using to determine how exactly we are moving around the space.

The outcome of running MCMC can be summarised using a table that contains a list of randomly drawn values from the posterior distribution. Here on the screen, you can see seven first rows from a hypothetical table with the randomly drawn values from the posterior. Typically this table would have much much more observations and this table would have a large number of parameters. Now, once you have such a table, you can use it to produce the summaries to produce the estimates for your parameters to capture the uncertainty that you should have about those estimates. And to compute all sorts of commodities of interest, that depend on the parameters in the model.

There are a number of different standalone programmes used for running MCMC. Here is a short list of such programmes. They include BUGS WinBUGS OpenBUGS, most recently Stan, but typically when you use such programmes, you use them in assemble with some sort of data handling programme. Typically you would start by start with R, use R to prepare your data set and to feed the data and the model into the ausstellung programmes. Once the programme is done running MCMC, it will return the draws from the posterior distribution to R. And then you can work with them using R. There are a number of packages are packages that are designed to make Bayesian analysis easier. Here is a short shorter list of those packages. These packages would allow you to specify Bayesian models common style that is without writing out the likelihood function and the priors directly.

Here I want to focus on one of these R packages specifically on the package called rstanarm this package was developed by the Stan development team and this package relies on the standard limitation of the Hamiltonian Monte Carlo for the estimation. This package includes a number of commonly used frequently used models, linear model, generalised linear model, multi level models. And in order to for you to be able to use these commands, you simply need to know the syntax for the corresponding frequencies function. If you know how to use the LM function, you know how to write a simplest version of the syntax for Stan lm, if you know how to estimate a logistic regression using GLM, you also know how to estimate a basic version of a logistic regression using standard lm.

Now, this package also has a bunch of utilities that can be used to summarise your posterior distributions and perform some predictive checks and some diagnostics on your estimates. As an exercise, I suggest that you use this package to estimate a logistic regression. Here's an outline of the steps that I want you to perform during this exercise.

As the first step, please load the file turnout forl logit.Rdata. This file contains a data frame called individual with the information from the 2008 general election in Canada. This file is based on the 2008 Canadian election study, but has much fewer observations and much fewer columns in it.

The second step, please use the glm function to estimate a conventional logistic regression, a logistic regression using maximum likelihood estimation, I would like you to use a variable called voted as your dependent variable. This is a binary indicated of turnout, I want you to use the variables called interest knowledge High School and BA degree as the independent variables. Once you have successfully run this estimation, you will have a version of syntax that can be used for running logistic regression with Emily and because of the ease of use of our scenario, we will also have a version of syntax that can be used for computing the estimates for logistic regression using the Bayesian estimation.

At Step three, please load the XML package and add Step four, please run the stan_glm function using the similar syntax to the one that we get used in glm. We can also look at the documentation for this function if you want to experiment with the priors and see what else this function can do for you. Once you have run this function, you are going to get a set of estimates.

Now, as the next step, I want you to examine the priors that we use in the analysis. And to do that you can use a prior_summary() function on the object produced at step four. The output from this function is going to be a list of the parameters that have prior distributions alongside with the prior distributions used for those parameters. Please take a look and see which parameters have prior distributions and how exactly those spreads which ones were specified.

At Step six, you can use the function called as.data.frame() to extract the random draws from the posterior distribution from the object created at step four. This should give you a table with the draws from the posterior distribution. Once you have this table, you can go ahead and take a look at this table and generate all sorts of summaries for the posterior distributions. Draw them using histograms generate the point estimates by applying the mean function or the median function to each of their of the columns. And generally look at the amount of dispersion in this table.