Computer Tutorial 3: Machine Learning Methods and VARs

Data and Matlab code for all questions are available on the course website.

Exercise 1: *SSVS*
In my handout, "Bayesian Methods for Fat Data", I carry out an empirical exercise using an SSVS prior and a data set from the economic growth literature. This data set covers $N = 72$ countries and contains $K = 41$ potential explanatory variables. The dependent variable is average per capita GDP growth for the period 1960-1992. I use a default semi-automatic prior with particular values for the constant used to define "small" and "large" prior variances. The code is called ssvs.m. Use this code to reproduce the tables in the handout. Next, investigate prior sensitivity with respect to the constant. Optional exercise: Modify the code to remove the default semi-automatric prior and replace it with subjectively chosen values for τ_{0i}^2 and τ_{1i}^2 chosen by you.

Exercise 2: *The LASSO*
In my handout, "Bayesian Methods for Fat Data", I carry out an empirical exercise using a LASSO prior for the same data set as Exercise 1. The code LASSO.m replicates the results in the handout using particular prior hyperparameter values. After replicating my results, carry out a prior sensitivity analysis. Are results sensitive to choice of prior?

Exercise 3: *VAR posteriors using analytical results, and their properties*
Use the MATLAB code BVAR_ANALYT.m that estimates the VAR model using analytical methods (i.e. no posterior simulation is done), with a choice of three available priors (Noninformative, Minnesota and natural conjugate). Load the macroeconomic dataset provided, and experiment with the prior hyperparameters of the Minnesota prior. (Note: This code does not directly print out any output to the screen. So you will have to figure out what the program is producing and how to print it out).
Take a training sample of the first 40 quarters of data and estimate a VAR model using this training sample (and a Noninformative prior. Use the posterior from the training sample VAR to determine the prior hyperparameters of a Normal-Wishart prior. Estimate a VAR using this prior and the remainder of the data. Compare your results with those of part a).

Exercise 4: *Impulse response analysis*
Perform impulse response analysis using the code BVAR_FULL.m and replicate the results of the first empirical illustration in the monograph by Koop and Korobilis. This code gives you the option to choose six between different priors. Experiment with all of them and try different prior hyperparameter choices.

Bonus Exercise 5: *BART*
This is a bonus exercise since the code is more difficult than for the other exercises and there may not be time in the computer tutorials to cover this in any detail. Furthermore, as discussed on page 70 of the Topic 3 lecture slides, several fine (computationally fast) BART packages exist in R which are very simple to use and involve minimal programming. If you wish to use BART in the future (e.g. for your empirical project, MSc or PhD dissertation) you are probably better off learning a small amount of R and using one of the them. However, I have often found that the best way to gain a deep understanding of computation in a Bayesian model is to work with the computer code myself in the language I am most familiar with. For this reason, I have provided Matlab code for BART. It consists of a main file (BARTmain.m) in which you can load in a data set and make specification choices (e.g. number of MCMC draws, the number of trees, prior hyperparameters) and a function (BART.m) which actually does MCMC for BART (it calls a function lambdafxn.m which is also provided in the package of code for Computer Tutorial 3). This is very basic code which produces MCMC draws of everything in BART (e.g. nodes, splitting rules, etc.) and nothing more.
Your first challenge is to figure out where everything is in the MCMC output. Section 2 of the Hill et al. paper (included in this package of materials) describes how BART involves trees and parameters (e.g. splitting rules, terminal nodes, etc.). Where can we find all of these in the MCMC output? I have put some comments in the code which should help with this. I recommend starting with a single tree (set m=1).
Your second challenge is to think of empirical results you might present (this can be difficult with black box methods such as BART where there is no single easy measure of the importance of a variable such as

a β coefficient in a linear regression). BART and related methods are often used as black box methods for prediction. The code does not do out-of-sample prediction. However, it does calculate the posterior mean of the fitted regression line and the Root Mean Squared Error (RMSE) based on the distance between this fitted regression line and the actual observations (this is labelled RMSE in the code and the fitted regression line is labelled yfit). One question of interest with BART is whether it is better to work with many simple trees or a small number of more complicated ones. In the code, the number of trees can be set controlled m and the depth of trees can be controlled using alpha and beta. Discuss this question by investigating how RMSE varies when change m, alpha and beta.