Summary of the work during Summer 2020

Jorge Valdebenito

October 31, 2020

This document presents the summary for the summer of 2020 work on the political uncertainty project.

1. Literature review:

   I updated the literature of the 2016 paper "Lobbying and Legislative Uncertainty" by Buzard and Saiegh (2016). This review contained the latest literature on ideal point models estimators and recent work on political uncertainty models using roll call data. An important filter for ideal point models was to focus on research that employs a multidimensional approach. Most of the papers that take a unidimensional approach were discarded, unless they specifically focus on uncertainty or efficiency of the unidimensional model. The Literature review was based on:

   - Bafumi, Gelman, Park, Kaplan (2005)
   - Clinton, Jackman, Rivers (2004)
   - Crespin and Rohde (2010)

   The document is called "Literature review" and it's on the Github repository.

2. The code:

   There are 2 approaches mixed in the G drive and in the Github repository.

   The first one is from the early stages of the project and the developer is Yusuf, his work is very well detailed and documented in the "order of the files" word document. This word document explains the steps, data sets and order of how the program should be run. We discovered a problem with the code when we were running these models. This approach uses the command "destring" to merge the data sets, the problem was that it is ignoring the suffix on each law, therefore if 2 laws have the same number, but different suffix, the code will consider them

as the same. For example, let's take the laws H 2 and HR 2. These are different laws/actions, but the "destring" command removes the suffix and both laws are now called 2. Which is not correct. Another issue with this approach is that it calls datasets and other pieces of code that were on Yusuf's computer, not on the G drive. It is important to consider also, that this work generates several output files that are called by different pieces of code later on, making them difficult to track and are located in different folders.

The second approach is an update of Yusuf's work and the developer is Yimin. His program is documented in the "Replication" file located in the Github repository. This approach consists of running 3 different pieces of code. We encountered a problem with the third of these files called `RStan_model.R`. This piece of code presents some errors when running the rstan command in R. This code avoids using the "destring" command and it should perform faster than Yusuf's code. Its also makes the merging more efficient by doing a better job at cleaning the data set.

I encountered the following error when running line 104 in RStan_model.R

"Error in file(con, "w") : cannot open the connection

In addition: Warning message: In file(con, "w") : cannot open file 'C:\Users\javaldeb\AppData\Local\Temp\Rtmp2veS6l\file4dbc4163331': No such file or directory"

Troubleshooting this I found different solutions, but non have worked yet. The most possible one is that I do not have writting access. The literature recommed to set the working directory again, but it didn't fixed the problem. Another proposed solution was to restarted RStudio and reinstall the packages. After trying this, it also didn't worked.

After checking the documentation for STAN command, I found no mistakes in the code, particularly line 37 (I meant it is not commented out, its just the syntax) and its generating a vector of values that it is called back in line 105. Which is where I get the error described before.

The other 2 programs `Generate_vote_data_for_R.do` and `Generate_group_dummy_by_action.do` run perfectly and their output is located in the Combined files folder in the G drive.

One detail that brings confusion is that Yusuf's and Yimin's work are located in the same folders. Yusuf's work is structure inside the folders `build` and `analysis`. Both folders have the subfolders `input`, `output`

and `code` from where he calls (input) and export (output) different files and datasets. The step by step description and structure of his work is detailed in the word document "order of the files.docx".

Yimin's work is located in build/code (`Generate_vote_data_for_-R.do` and `Generate_group_dummy_by_action.do`) and analysis/code `RStan_model.R`. His work is describe in "Replication", located in the political-uncertainty Github repository.

Therefore, future work will be to separate them and organize the files from these two different approaches, which will entail changing the output locations in all the individual R and Stata programs.

3. Data

Yusuf's approach uses 3 different data sets, 2 from Maplight database and one from Poole's (Voteview.com).

- The data set maplight_bill_position.dta"

  This data set can be downloaded from `https://maplight.org/data\_guide`. A key can be requested from their website (at the end of the document there is the PAI I received) to download the newest version of the dataset. The most updated data set (according to the webpage) is the 114th session of the United States Congress (2015-2016).

  The parameters on this data set are: legislative_session, bill_number, bill_topic, bill_description, motion, action_id, vote_roll, date, position, org_name, sector, industry, business, OS_catcode, maplight_url

  The dataset that Yusuf used is located in `build\input\maplight_bill_positions.dta` in the Github folder.

- Second data set is maplight_votes.dta

  This was emailed to professor Buzard at the time, according to one of the earlier doc files in the onedrive and the data set is located in `build\input\maplight_votes.dta` in the Github folder. The parameters are: action_id, politician_id, fullname, party, vote

- Poole's data set

  Poole's data set can be downloaded from `https://voteview.com/data`

  There is a clear description of the data in the help link. The most updates congress data is for 116th (2019 to 2021) "This data includes every vote taken by every member in the selected

congresses and chambers along with the probability we assign to the member taking the position they did. Members are indexed by their ICPSR ID number."

The parameters are: congress, chamber, rollnumber, icpsr, cast_code, prob

The website voteview.com also have the following datasets:

- Congressional Votes: "This data includes the result and ideological parameters of every vote taken in the selected congresses and chambers. This is information about the vote itself, not individual members positions."

- Members Ideology: "This data includes basic biographical information (state, district, party, name) and ideological scores for members of the selected congresses."

- Congressional parties: "This data includes ideological mean and median scores for congressional parties in the selected congresses and chambers."

Yimin's program only use maplight_bill_position.dta and maplight_votes.dta

Both of this data sets are merged by action_id.

The API key I requested is below.

The API key is: 8d6d8a79f2f3df97ef128dbfea969a82 and the link is: `https://maplight.org/data\_guide/bulk-data-sets-and-apis/`

4. To conclude

The most updated work comes from the file "REPLICATION", instead of the word document "Order of the Files".

"Order of the Files" details the initial steps of the proccess, while 'REPLICATION" uses the RSTAN command to increase efficiency. It also solves the problem of the "destring" command in the earlier version.

The "destring" command made the mistake of deleting the prefix of bills (H, HR, ...) therefore, it was treating different bills as if they were one.

Example, bill numbers H2 and HR2. The "destring" command called both bills as 2, treating them as the same for merging with the votes.dta data set. The previous problem is fixed with RSTAN, given that it is not using "destring".

I then run the first two programs using the "REPLICATION" file. As discussed before, the file RStan_model.R couldn't be run.

The names and locations of these programs are:

Generate_vote_data_for_R.do

Generate_group_dummy_by_action.do

Both are in the github rep and the G disk inside the folder: `political-uncertainty/build/code/`

The RStan_model.R is located also in the github rep and the G disk in the folder: `political-uncertainty/analysis/code/`