Replication instructions for "Estimating Crisis Signaling Games in International Relations: Problems and Solutions"

Casey Crisman-Cox and Michael Gibilisco

August 10, 2019

A note for replicators

Conducting constrained maximum likelihood estimation (CMLE) requires specialized (open source) software that we run using a Ubuntu Linux operating system. All our PL, NPL, and tML results are based solely on `R` and can be replicated without issue on any operating system. Here, we provide instructions for reproducing our results using the Windows Subsystem for Linux (WSL) for Windows 10. All the code should also work on an ordinary Ubuntu system. We provide detailed setup instructions below.

Replication package contents

Contents:

- Background information
    - `readme.md`: plain text readme for git archive
    - `readme.txt`: plain text readme for pdf
    - `readme.pdf`: PDF version of readme
- Installation files
    - `CMLE_setup.sh`: A bash script to be run once WSL is installed and setup. This will install all the necessary outside software to replicate the results. (Internet connection is required)
    - `install_Rpackages_noadmin.r`: An R script that installs all the `R` packages used here.
- Support functions
    - `signalingFunctions_main.r`: Contains objective functions and helper functions for the PL, NPL, and tML methods
    - `gradientFunctions.r`: Contains gradient functions for the PL and NPL
    - `CMLE_functions.R`: Contains objective and gradient functions for the CMLE
    - `CMLE_functions.py`: Contains objective functions and helper functions for the CMLE
    - `CMLE_MonteCarlo_meq_support.py`: Contains a function to run one Monte Carlo iteration with multiple equilibria
    - `CMLE_MonteCarlo_meq_support.r`: Helper code for the CMLE Monte Carlo with multiple equilibria
    - `CMLE_MonteCarlo_unique_support.py`: Contains a function to run one Monte Carlo iteration with a unique equilibrium
    - `CMLE_MonteCarlo_unique_support.r`: Helper code for the CMLE Monte Carlo with a unique equilibrium
    - `CMLE_estimation_support.R`: Helper code for fitting the CMLE to sanctions data
    - `parmap.py`: Helper code for running the CMLE Monte Carlos in parallel
- Monte Carlos
    - `replicateMonteCarlos.sh`: Runs `eqGraph.r`, `MonteCarloMEQ.r`, `MonteCarloUnique.r`, `CMLE_MonteCarlo_meq.py`, `CMLE_MonteCarlo_unique.py`, `AnalyzeSimulationMEQ.r`, and `AnalyzeSimulationUnique.r` and outputs Figures 2, 4, and 5. Produces the log file

`MonteCarloLog.txt`. Note that IPOPT output is suppressed here to prevent producing a log file that is several gigabytes in size.
- – `eqGraph.r`: Produces the equilibrium correspondences in Figure 2 (`figure2.pdf`)
- – `MonteCarloMEQ.r`: Runs the Monte Carlo simulations when data generating game has multiple equilibria. Outputs `MonteCarloResults_MEQ.rdata`
- – `MonteCarloUnique.r`: Runs the Monte Carlo simulations when data generating game has a unique equilibrium. Outputs `MonteCarloResults_Unique.rdata`
- – `CMLE_MonteCarlo_meq.py`: Runs the Monte Carlo simulations for the CMLE when data generating game has multiple equilibria. Outputs `CMLE_meq.rdata`
- – `CMLE_MonteCarlo_unique.py`: Runs the Monte Carlo simulations for the CMLE when data generating game has a unique equilibrium. Outputs `CMLE_unique.rdata`
- – `AnalyzeSimulationMEQ.r`: Inputs `MonteCarloResults_MEQ.rdata` and `CMLE_meq.rdata` and outputs Figure 3 (`figure3.pdf`)
- – `AnalyzeSimulationUnique.r`: Inputs `MonteCarloResults_unique.rdata` and `CMLE_unique.rdata` and outputs Figure 4 (`figure4.pdf`)
- Estimation and analysis
  - – `replicateSanctions.sh`: Bash script to run `CMLE_estimation.py`, `estimation.r`, `standardErrors.R` and `comparativeStatics.R`. Outputs Figure 6 and the log file `estimationLog.txt`.
  - – `SanctionsDataSet.rdata`: Economic sanctions data
  - – `CMLE_estimation.py`: Fits the CMLE to the sanctions data. Outputs `CMLE_estimation_output.rdata`
  - – `estimation.R`: Fits tML, PL, and NPL to sanctions data. Outputs `estimation_output.Rdata`
  - – `standardErrors.R`: Estimates standard errors for the tML, PL, NPL, and CMLE and prints the results reported in Table 3
  - – `ComparativeStatics.R`: Inputs `CMLE_estimation_output.rdata` and `estimation_output.Rdata` and produces Figure 6 (`figure6.pdf`)
- Output files
  - – `figure2.pdf`: Figure 2 in the manuscript
  - – `MonteCarloResults_MEQ.rdata`: Raw Monte Carlo results for the PL, NPL, and tML when there are multiple equilibria in the data generating game
  - – `MonteCarloResults_unique.rdata`: Raw Monte Carlo results for the PL, NPL, and tML when there is a unique equilibrium in the data generating game
  - – `CMLE_unique.rdata`: Raw Monte Carlo results for the CMLE when there are multiple equilibria in the data generating game
  - – `CMLE_meq.rdata`: Raw Monte Carlo results for the CMLE when there is a unique equilibrium in the data generating game
  - – `figure3.pdf`: Figure 3 in the manuscript
  - – `figure4.pdf`: Figure 4 in the manuscript
  - – `CMLE_estimation_output.rdata`: CMLE point estimates and model information
  - – `estimation_output.Rdata`: PL, NPL, and tML point estimates and model information
  - – `SIGMA.rdata`: Variance-covariance matrix for the first-stage estimates
  - – `figure6.pdf`: Figure 6 in the manuscript
  - – `MonteCarloLog.txt`: Log file for the Mont Carlo experiments
  - – `estimationLog.txt`: Log file for the sanctions application. Prints all the information contained in Table 3.

WSL Setup

All applications of the CMLE to either simulation or actual data was done with Ubuntu 18.04.1 (Bionic Beaver). The automatic differentiation (AD) software is only tested for Ubuntu operating systems. To allow for replication we provide these instructions for users using a Windows 10 computer with build 16215 or later (Fall Creators Update and later). The installation steps are taken from this link.

1. Enable WSL by opening the Windows PowerShell as an administrator and running the following command

```
Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Windows-Subsystem-Linux
```

2. Restart your computer

3. Open the Windows store

4. Search for "Ubuntu"

5. Select "Ubuntu 18.04 LTS" and click "Get"

6. Once it is installed, open the Ubuntu app. This may take a few minutes to load the first time.

7. You will be prompted to choose a user name and password. You will need to use this password throughout.

8. Once you are setup with the Ubuntu app, navigate to the replication folder from within the Ubuntu command line using the `cd` command. For example, if the replication folder is saved in the Windows Downloads folder you could get there by using the command

```
cd /mnt/<c>/Users/<WINDOWS USER>/Downloads/<REPLICATION FOLDER>
```

Where `<c>` refers to your main Windows drive (almost always `c`), `<WINDOWS USER>` refers to your Windows username (**not** the Ubuntu user name you selected in step 7), and `<REPLICATION FOLDER>` is the folder containing all the replication files.

9. Run the file `CMLE_setup.sh` using the command

```
bash CMLE_setup.sh
```

This step may take up to a few hours depending on network speed and you may be prompted for your Ubuntu password (chosen in step 7), to select "yes", or to press "Enter" at various points in the process. An Internet connection is required for this step.

10. Once this script has completed we are ready to reproduce all the results. If Windows is configured for automatic updating, we recommend that you disconnect from the Internet during the Monte Carlo steps as they may take a few hours or days.

Simulation results

To produce Figures 2-4 run bash script `replicateMonteCarlos.sh`. Do this by opening the Ubuntu app, navigating to the Replication folder (as in step 8 above) and running the command:

```
bash replicateMonteCarlos.sh
```

This command produces four output files (`MonteCarloResults_MEQ.rdata`, `MonteCarloResults_Unique.rdata`, `CMLE_meq.rdata`, and `CMLE_unique.rdata`), a log file (`MonteCarloLog.txt`), and three figures (`figure2.pdf`, `figure3.pdf`, and `figure4.pdf`). These three PDF files correspond to Figures 2-4 in the manuscript, respectively. Individual figures can be reproduced separately by running the commands in `replicateMonteCarlos.sh` one at a time. Note that this may take several days or longer depending on the computer power available.

Estimates, standard errors, and comparative statics

To produce the values from Table 3 and the results in Figure 6 run the bash script `replicateSanctions.sh` Do this by opening the Ubuntu app, navigating to the Replication folder (as in step 8 above) and running the command:

```
bash replicationSanctions.sh
```

This command produces three output files (`CMLE_estimation_output.rdata`, `estimation_output.Rdata`, and `SIGMA.rdata`), a log file (`estimationLog.txt`), and one figure (`figure6.pdf`). All the values reported in Table 3 are found in `estimationLog.txt`, and Figure 6 is reproduced in `figure6.pdf`. Individual aspects of the analysis can be reproduced separately by running the commands in `replicateSanctions.sh` one at a time.