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

Casey Crisman-Cox and Michael Gibilisco

October 15, 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 (main text):

- 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.
  - `sigInt_0.0.0.9000.tar.gz`: R package used in Appendix I.2
- 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) and baseLinePlotDataMEQ.rdata.
  - AnalyzeSimulationUnique.r: Inputs MonteCarloResults_unique.rdata and CMLE_unique.rdata and outputs Figure 4 (figure4.pdf) and baseLinePlotDataUnique.rdata.
- 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
  - baseLinePlotDataMEQ.rdata: Processed Monte Carlo results for the PL and NPL when there are multiple equilibria; used in Appendix E.
  - baseLinePlotDataUnique.rdata: Processed Monte Carlo results for the PL and NPL when there is a unique equilibrium; used in Appendix E.
  - 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.

Contents (appendix):

- Appendix C
  - Support functions
    * CMLE_unstable_support.py: Contains a function to run one Monte Carlo iteration that

explores best response stability.
    * `CMLE_unstable_support.r`: Helper code for the CMLE Monte Carlo that explores best response stability.
    * `CMLE_sanctionsMC_support.py`: Contains a function to run one Monte Carlo iteration that is based on the economic sanctions data.
    * `CMLE_sanctionsMC_support.r`: Helper code for the CMLE Monte Carlo that is based on the economic sanctions data.
  - Monte Carlos and analysis
    * `replicateAppendixC.sh`: Runs `appendixC1.r1`, `appendixC2.r`, `appendixC3_simulation3.r`, `MonteCarloCMLE_unstable.py`, `appendixC3.R`, `appendixC4_simulation.R`, `CMLE_sanctionsMC.py`, and `appendixC4.R` and outputs Figures 8-17. Produces the log file `appendixC.txt`, which contains the information reported in Table 4.
    * `appendixC1.r`: Inputs `MonteCarloResults_MEQ.rdata` and `CMLE_meq.rdata` and outputs Figures 8-11 (`figure8.pdf`, `figure9.pdf`, `figure10.pdf`, and `figure11.pdf`).
    * `appendixC2.r`: Inputs `MonteCarloResults_unique.rdata` and `CMLE_unique.rdata` and outputs Figures 12-15 (`figure12.pdf`, `figure13.pdf`, `figure14.pdf`, and `figure15.pdf`).
    * `appendixC3_simulation.r`: Runs the Monte Carlo simulation that explores best response stability. Outputs `MonteCarloUnstable.rdata`.
    * `MonteCarloCMLE_unstable.py`: Runs the Monte Carlo simulation for the CMLE that explores best response stability. Outputs `CMLE_unstable.rdata`.
    * `appendixC3.R`: Inputs `MonteCarloUnstable.rdata` and `CMLE_unstable.rdata` and outputs Figures 16-17 (`figure16.pdf` and `figure17.pdf`).
    * `appendixC4_simulation.R`: Runs the Monte Carlo simulation that is based on the economic sanctions data. Outputs `appendixC4_results.RData`
    * `CMLE_sanctionsMC.py`: Runs the Monte Carlo simulation for the CMLE that is based on the economic sanctions data. Outputs `CMLE_sanctionsMC.rdata`.
    * `appendixC4.R`: Inputs `appendixC4_results.RData`, `CMLE_sanctionsMC.rdata`, and `CMLE_estimation_output.rdata` and prints the results in Table 4.
  - Output files
    * `figure8.pdf`: Figure 8 in the Appendix
    * `figure9.pdf`: Figure 9 in the Appendix
    * `figure10.pdf`: Figure 10 in the Appendix
    * `figure11.pdf`: Figure 11 in the Appendix
    * `figure12.pdf`: Figure 12 in the Appendix
    * `figure13.pdf`: Figure 13 in the Appendix
    * `figure14.pdf`: Figure 14 in the Appendix
    * `figure15.pdf`: Figure 15 in the Appendix
    * `figure16.pdf`: Figure 16 in the Appendix
    * `figure17.pdf`: Figure 17 in the Appendix
    * `MonteCarloUnstable.rdata`: Raw Monte Carlo results for the PL, NPL, and tML relating to best response stability
    * `CMLE_unstable.rdata`: Raw Monte Carlo results for the CMLE relating to best response stability
    * `appendixC4_results.RData`: Raw Monte Carlo results for the PL, NPL, and tML when the data are generated from the economic sanctions example
    * `CMLE_sanctionsMC.rdata`: Raw Monte Carlo results for the CMLE when the data are generated from the economic sanctions example
    * `appendixC.txt`: Log file for Appendix C. Prints all the information contained in Table 4.
- Appendix E
  - Monte Carlos and analysis
    * `replicateAppendixE.sh`: Runs `appendixE_simulation_MEQ.r`, `appendixE_simulation_Unique.r`, `appendixE_MEQ.r`, and `appendixE_Unique.r` and outputs Figures 18 and 19. Produces a log file `appendixE.txt`.
    * `appendixE_simulation_MEQ.r`: Runs the Monte Carlo simulation where the tML is started

from informative values and there are multiple equilibria. Outputs `appendixE_MEQ_out.rdata`.
    * `appendixE_simulation_Unique.r`:Runs the Monte Carlo simulation where the tML is started from informative values and there is a unique equilibrium. Outputs `appendixE_Unique_out.rdata`.
    * `appendixE_MEQ.r`: Inputs `appendixE_MEQ_out.rdata` and `baseLinePlotDataMEQ.rdata` and outputs Figure 18 (`figure18.pdf`)
    * `appendixE_Unique.r` : Inputs `appendixE_Unique_out.rdata` and `baseLinePlotDataUnique.rdata` and outputs Figure 19 (`figure19.pdf`)
  - Output Files
    * `figure18.pdf`: Figure 18 in the Appendix
    * `figure19.pdf`: Figure 19 in the Appendix
    * `appendixE_MEQ_out.rdata`: Raw Monte Carlo results for the PL and tML for the simulation where the tML is started at informative values and there are multiple equilibria.
    * `appendixE_Unique_out.rdata`: Raw Monte Carlo results for the PL and tML for the simulation where the tML is started at informative values and there is a unique equilibrium.
    * `appendixE.txt`: Log file for Appendix E (empty).
- Appendix F
  - Estimation and analysis
    * `replicateAppendixF.sh`: Runs `appendixF1.r` and `appendixF2.r` and outputs Figure 20. Produces a log file `appendixF.txt`
    * `appendixF1.r`: Considers the discontinuity problem in the tML. Produces Figure 20 (`figure20.pdf`)
    * `appendixF2.r`: Considers fitting the tML to the economic sanctions data with different implementation choices. Produces the results in Table 5.
  - Output files
    * `figure20.pdf`: Figure 20 in the Appendix
    * `appendixF.txt`: Log file for Appendix F. Prints all the information contained in Table 5.
- Appendix G
  - Analysis
    * `replicateAppendixG.sh`: Runs `appendixG.R` and outputs Figures 21-23 and log file `appendixG.txt`.
    * `appendixG.R`: Inputs `CMLE_estimation_output.rdata` and outputs Figures 21, 22, and 23 (`figure21.pdf`, `figure22.pdf`, and `figure23.pdf`)
  - Output files
    * `figure21.pdf`: Figure 21 in the Appendix
    * `figure22.pdf`: Figure 22 in the Appendix
    * `figure23.pdf`: Figure 23 in the Appendix
    * `appendixG.txt`: Log file for Appendix G.
- Appendix H
  - Analysis
    * `replicateAppendixH.sh`: Runs `appendixH.R` and outputs Figure 24 and log file `appendixH.txt`.
    * `SanctionsDataSet_1yearsT12.rdata`: Economic sanctions data with dyad-year aggregation.
    * `appendixH.R`: Outputs Figure 24 (`figure24.pdf`)
  - Output files
    * `figure24.pdf`: Figure 24 in the Appendix
    * `appendixH.txt`: Log file for Appendix H
- Appendix I
  - Support functions
    * `CMLE_sanctionsQuarters_support.R`: Helper code for fitting the model with the CMLE on quarterly data.
    * `CMLE_sanctions_5years_support.R`: Helper code for fitting the model with the CMLE on dyad-5 year data.
    * `CMLE_sanctions_1yearsT12_support.R`: Helper code for fitting the model with the CMLE

on dyad-year data.

- Estimation
  - `replicateAppendixI.sh`: runs `appendixI1_estimation.R`, `CMLE_sanctionsQuarters.py`, `appendixI1_SE.R`, `appendixI2_estimation.R`, `appendixI3_5years.R`, `CMLE_sanctions_5years.py`, `appendixI3_5yearsSE.R`, `appendixI3_1yearsT12.R`, `CMLE_sanctions_1yearsT12.py`, `appendixI3_1yearsT12_SE.R`, `appendixI3_1yearsT1.R`, and `appendixI3_1yearsT1_SE.R`. Produces a log file `appendixI.txt` that contains all the information in Tables 6-10.
  - `SanctionsDataSet_quarterly.rdata`: Economic sanctions data with actions recorded at the quarterly level.
  - `appendixI1_estimation.R`: Fits the model to the quarterly sanctions data using the PL and NPL estimators. Outputs `appendixI1_output.Rdata`.
  - `CMLE_sanctionsQuarters.py`: Fits the model to the quarterly sanctions data using the CMLE. Outputs `appendixI1_CMLEoutput.Rdata`.
  - `appendixI1_SE.R`: Estimates the standard errors for the PL, NPL, and CMLE estimates for the model fit to quarterly data. Prints out the results in Table 6.
  - `SanctionsDataSet_prelevant.rdata`: Economic sanctions data with an expanded definition of political relevance.
  - `appendixI2_estimation.R`: Fits the model to the data in `SanctionsDataSet_prelevant.rdata` using the PL and NPL. Prints out the results in Table 7.
  - `SanctionsDataSet_5years.rdata`: Economic sanctions data with dyad-5 year aggregation.
  - `appendixI3_5years.R`: Fits the model to the dyad-5 year sanctions data using the PL and NPL estimators. Outputs `appendixI3_output_5years.Rdata`.
  - `CMLE_sanctions_5years.py`: Fits the model to the dyad-5 year sanctions data using the CMLE estimator. Outputs `appendixI3_CMLEoutput_5years.Rdata`.
  - `appendixI3_5yearsSE.R`: Estimates the standard errors for the PL, NPL, and CMLE estimates for the model fit to dyad-5 year data. Prints out the results in Table 8.
  - `appendixI3_1yearsT12.R`: Fits the model to the dyad-year sanctions data using the PL and NPL estimators. Outputs `appendixI3_output_1yearsT12.Rdata`.
  - `CMLE_sanctions_1yearsT12.py`: Fits the model to the dyad-year sanctions data using the CMLE. Outputs `appendixI3_CMLEoutput_1yearsT12.Rdata`.
  - `appendixI3_1yearsT12_SE.R`: Estimates the standard errors for the PL, NPL, and CMLE estimates for the model fit to dyad-year data. Prints out the results in Table 9.
  - `SanctionsDataSet_1yearsT1.rdata`: Economic sanctions data with actions recorded at the dyad-year and dyad-year aggregation.
  - `appendixI3_1yearsT1.R`: Fits the model to data in `SanctionsDataSet_1yearsT1.rdata` using the PL and NPL estimators. Outputs `appendixI3_output_1yearsT1.Rdata`.
  - `appendixI3_1yearsT1_SE.R`: Bootstraps the PL and NPL from ‘`appendixI3_1yearsT1.R`. Prints out the results in Table 10.
- Output files
  - `appendixI1_output.Rdata`: PL and NPL estimates for quarterly data.
  - `appendixI1_CMLEoutput.Rdata`: CMLE estimates for quarterly data.
  - `SIGMA_quarters.rdata`: First stage covariance matrix for the PL estimates with quarterly data.
  - `appendixI3_output_5years.Rdata`: PL and NPL estimates for dyad-5 year data.
  - `appendixI3_CMLEoutput_5years.Rdata`: CMLE estimates for dyad-5 year data.
  - `SIGMA_5year.rdata`: First stage covariance matrix for the PL estimates with dyad-5 year data.
  - `appendixI3_output_1yearsT12.Rdata`: PL and NPL estimates for dyad-year data.
  - `appendixI3_CMLEoutput_1yearsT12.Rdata`: CMLE estimates for dyad-year data.
  - `SIGMA_1yearsT12.rdata`: First stage covariance matrix for the PL estimates with dyad-year data.
  - `appendixI3_output_1yearsT1.Rdata`: PL and NPL estimates for the dyad-year aggregation with yearly actions.
  - `appendixI3_bootstraps_1yearsT1.rdata`: Raw bootstraps produced by `appendixI3_1yearsT1_SE.R`.

* `appendixI.txt`: Log file for Appendix I. Contains all the information presented in Tables 6-10.

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.

Appendices

To replicate the appendices run the commands:

```
bash replicateAppendixC.sh
bash replicateAppendixE.sh
bash replicateAppendixF.sh
bash replicateAppendixG.sh
bash replicateAppendixH.sh
bash replicateAppendixI.sh
```

These will produce Figures 8-24 and Tables 5-10. Table 5 is found in `appendixC.txt`, while Tables 6-10 are in `appendixI.txt`. Additionally, various `rdata` files will be produced along the way.