Replication instructions for “Audience Costs and the Dynamics of War and Peace”

Casey Crisman-Cox and Michael Gibilisco

21 December 2017

Replication package contents

If you downloaded and extracted this replication archive from the tar.gz file then the file and folder structure will be correct (as it is listed below). However, if you downloaded the files as they are from the dataverse page you will first need to create the folder structure outlined below and place the files in the correct folders.

File structure

Main folder:

- `readme.txt`: plain text version of this pdf
- `readme.md`: plain text version of this pdf
- `readme.pdf`: This document
- `xubuntu-14.04-desktop-amd64.iso`: CD image of Xubuntu 14.04 (Trusty Tahir), the operating system used in the estimation.
- `fullReplication.sh`: a bash script file that installs all necessary software and runs all the replication code. Once Xubuntu 14.04 is running, this file will setup all software, compile the dataset and replicate all results
- `Data`: a folder containing the following
  - `DyadicDataCreate.R`: File to combine all the data from the Sources folder (below) and create the dataset used throughout the paper. Creates the file DyadicMIDS_Rdata.rdata
  - `DyadicMIDS_Rdata.rdata`: Rdata file containing the data used in the estimation
  - `CBIRMPEC.pdf`: Codebook for the variables included in the dataset
  - `dataSources.pdf`: List of complete references for all data used in this archive
  - `Sources`: A folder containing all data sources used to create the data set

        - `MIDIP_4.01.csv`: Militarized Interstate Dispute Incident Profiles data from Correlates of War (COW)
        - `alliance1993_2007.csv`: COW alliance data
        - `NMC_v4_0.csv`: National Materials and Capabilities Data (COW)
        - `p4v2013.csv`: Polity IV data
        - `dyadic_trade_3.0.csv`: COW trade data
        - `israelTradeSupplement.csv`: Additional world bank data on trade with Israel. Used to fill in missing values
        - `trickTrade.csv`: A list of remaining dyads with some missing data on trade
        - `directedDyads.csv`: COW list of directed dyads
        - `states2011.csv`: States of the world list (COW)
        - `distance.csv`: Distance between countries (COW)
        - `Press_FH.csv`: Freedom of the press data (Freedom House)
        - `jcr_event1_democracy.dta`: Data from Quan Li's 2005 *JCR* piece on terrorism. Used to fill in missing values for free press
        - `pwt8_gdp.csv`: Penn World Table GDP data
        - `world-gdp.csv`: World Bank GDP data
        - `world-growth.csv`: World bank growth data
        - `world-gdp-growth.csv`: Additional World Bank growth data to supplement the above
    - `Additional Functions`: A folder containing
        - `ISO_to_COW.R: R` code to convert ISO country codes to COW country codes
        - `stringr_to_cow.R`: R code to convert country names to COW country codes
- `Ubuntu Optimization`: A folder with two sub folders
    - `UbuntuSetup`: A folder containing one file
        - `UbuntuSetup.sh`: A bash script to be run once the live session of Xubuntu 14.04 is booted. This will install all the necessary outside software to replicate the results. (Internet connection is required)
    - `UbuntuOptimization`: A folder containing the Python code used to solve the constrained optimization problem in the paper (estimate the main model)
        - `runEstimation.sh`: A bash script to convert the data from R to Python formats, run the estimation, and convert the results into R format. It does this by running the python scripts in this folder.
        - `convertData_R2py.py`: Python code for converting the R version of the data, DyadicMIDS_Rdata.rdata (from the Data folder, above), into the python readable file, `mainDataSet.p`. This script also drops non-conflictual dyads as stated in the manuscript.
        - `replicationInput.p`: A pickle file containing starting values. For the purposes of this package, the estimates are used as starting values. This is to save time and computing effort. We originally

used draws from a uniform distribution. Doing that requires 3-9 weeks of computing time. The procedure verifies that the final solution is a local solution to the constrained maximization problem.

    - `mainDataSet.p`: A pickle file containing the main dataset
    - `mainModelEstimation.py`: The python script that reads in replicationInput.p, creates and solves the constrained optimization problem, and exports the results to the Paper Results, below, as `ReplicationOutput.rdata`
    - `estFunctions.py`: Python code containing functions to evaluate the log-likelihood and the equilibrium constraints
    - `genDyadGiven.py`: Python code to convert the data into a format usable to the likelihood and constraint functions
    - `kappaDyad.py`: Python code to aid with the constraint evaluations
    - `UsaDyadParam.py`: Python code to evaluate the utility function
- `Paper Results`: Folder containing data and R code to replicate the tables and figures in the manuscript
  - `conductAnalysis.sh`: A bash script to run all the R files in this directory and its subdirectories. This produces all the tables and figures in the manuscript
  - `analyticalSE_code.r`: Reads in the estimation results and produces the standard errors. Outputs `mainModelResults.rdata`
  - `mainModelResults.rdata`: Rdata file containing the model estimates and standard errors
  - `ReplicationOutput.rdata`: Rdata file containing just the model estimates, this is generated from `mainModelEstimation.py` (above)
  - `AnalysisExtraDatasets`: Folder containing extra datasets that are only used in the post-estimation analysis
    - `bdm2s2_nation_year_data_may2002.dta`: Replication dataset from Bueno de Mesquita, et al. (2002) "The Logic of Political Survival"
    - `countryIndexV2.csv`: List of COW country codes to match with the 1, 2, … system used in the code
    - `COW State list.csv`: COW list of countries of the world
    - `institutions-data-11-16-11.dta`: Institutions and elections data (Regan, et al. 2009)
    - `thompsonRivals.csv`: Data on interstate rivals from Thompson and Dreyer (2012)
    - `UGSreplication.dta`: Replication data from Uzonyi, et al (2012)
  - `Additional Functions`:
    - `choiceProb.r`: R function to compute equilibrium choice probabilities from expected utility parameters
    - `compStat2.R`: R function to compute comparative statics on parameters of interest (audience costs)
    - `DPhiTheta.R`: R function for evaluating the derivative of the

equilibrium constrain with respect to the structural parameters

  - `dyadPhiQRE.r`: R function to evaluate the equilibrium constraint
  - `genDyadGiven.r`: R function for converting data into a usable format for the other functions.
  - `gradLL.R`: R function to evaluate the gradient of the log-likelihood
  - `invarDist.R`: R function to compute the invariant distribution over outcomes
  - `JMPECct.R`: R function to calculate the Jacobian of the equilibrium constraint
  - `kappaDyad.r`: R function to help with utility function evaluation
  - `prepDyadID.r`: R function to convert COW country codes into a unique dyadic identifier
  - `UsaDyad.R`: R function to evaluate the utility function
  - `UsaDyadParams.R`: R function to evaluate the utility function with the estimates
  - `v_PhiDer.R`: R function to evaluate the derivative of the equilibrium constraint with respect to the expected utility parameters

- `Section4`: A folder containing replication code for section 4
  - `Table1.r`: R file containing code to recreate Table 1 from the paper. Table contents are printed to screen
- `Section5`: A folder containing replication code for section 5
  - `Section5.0`: A folder containing replication code for section 5.0
    - `replication5.0.r`: R code to produce Figure 1. Figure 1 is saved in PDF format.
    - `Figure1.pdf`: Output of replication5.0.r, this PDF contains Figure 1 from the manuscript.
  - `Section5.1`: A folder containing replication code for section 5.1
    - `Table2.r`: R code to produce Table 2. Output is printed to the screen.
  - `Section5.2`: A folder containing replication code for section 5.2
    - `replication5.2.R`: R code to produce the information in Figure 2 (regression tree) and Table 3. Regression tree is saved to pdf, while information from Table 3 is printed to the screen. The actual Figure 2 is created in the manuscript using TikZ to reproduce the PDF output in a more customized way.
    - `Figure2.pdf`: Regression tree output (Figure 2 in the manuscript is a TikZ reproduction of this PDF so it looks different; the information presented in the plots is identical)
    - `Rplot.pdf`: Extra graphical output from fitting the regression and pruning the regression tree
  - `Section5.3`: A folder containing replication code for section 5.3
    - `replication5.3.R`: R code to produce Figures 3 and 4.
    - `Figure3.pdf`: Figure 3 in the manuscript
    - `Figure4.pdf`: Figure 4 in the manuscript
    - `Rplot.pdf`: Extra graphical output from computing the ef-

fects in Figures 3 and 4

Instructions for running Xubuntu 14.04 Live and installing necessary software

The estimation was conducted using a “Live” version of Xubuntu 14.04 (Trusty Tahir). The estimation code provided in this replication archive is designed only to work with that system. To setup a live version of Xubuntu 14.04 please follow the below steps. We recommend printing this section out before beginning.

1. On a Windows PC download the Rufus tool to create a “bootable” flash drive and download the CD image (.iso file) associated with Xubuntu 14.04
   - Rufus Link: Rufus disk utility link
   - The iso file is found in the main replication folder. We also provide a direct link: iso link
2. Run Rufus with the Xubuntu disk image
   - Follow Dell’s instructions for creating a bootable USB provided here: link
   - Select your USB drive for Device and the Xubuntu 14.04 iso file for “Create a bootable disk using:” option
   - All other options can remain at their default
3. Insert bootable flash drive into a powered down computer and turn the machine on.
   - As soon as you power on the machine repeatedly press whatever key takes you to the machine’s boot menu

   - The exact key will vary with machine make and model, common keys include F12 (DELL, Most ACERs), F8 (Most ASUS), and ESC (HP and some ASUS). See this link for more information
4. Once in the boot menu, select the option to boot from USB
5. At this point, you should be given the option to either install Xubuntu or Try Xubuntu. Select “Try.” This will take you to the desktop
6. Download the replication package (the web browser can be found by clicking on the “whisker menu” –small blue circle with a white mouse– in the upper left-hand corner of the screen. The first thing listed will be “Web Browser”)
7. Extract the replication package to the home folder (`/home/xubuntu`)
8. At this point you can run fullReplication.sh which will install all the necessary software and then run all the replication files

- To do this navigate to the replication folder (`/home/xubuntu/Replication`) using the file manager (also found in the whisker menu), right click on some white space, select "Open Terminal Here", and run the command

  ```
  bash fullReplication.sh
  ```

9. Alternatively, if you prefer to run the estimation and analysis files individually, you can just install all necessary software at this point by navigating to `/home/xubuntu/Replication/Ubuntu Optimization/UbuntuSetup`, opening the terminal (again by right clicking and selecting "Open Terminal Here" and running the command

   ```
   bash UbuntuSetup.sh
   ```

   This step may take up to 30 minutes depending on computer and internet speed.

- You will be required to press "Enter" at some point (roughly 20 minutes) into the process.
- As the software updates, other background options may occur (screen resolution may adjust, printers may be discovered, other drives may be mounted, etc.)
- Errors may appear during this process, but so long as the code does not stop, they can be safely ignored
- When `UbuntuSetup.sh` finishes, the line "`Xubuntu Setup Complete`" will print to the screen

Instructions for compiling the dataset

To recreate our dataset from its source components simply source the file `/home/xubuntu/Replication/Data/dyadicDataCreate.R` either from within an R environment (installed when running `UbuntuSetup.sh`, above) or by opening a terminal in this location and running

```
Rscript dyadicDataCreate.R
```

To convert the data from an rdata file to a pickle file (python readable), run `convertData_R2py.py` by navigating to `/home/xubuntu/Replication/Ubuntu Optimization/UbuntuOptimization/`, opening a terminal in that location, and running the command

```
python convertData_R2py.py
```

Instructions for replicating the coefficients

To run the estimation procedure navigate to the folder `/home/xubuntu/Replication/Ubuntu Optimization/UbuntuOptimization`. Open a terminal and run

```
bash runEstimation.sh
```

This script will convert the data (by running `convertData_R2py.py`) and run the estimation (by running `mainModelEstimation.py`). To just run the estimation, open a terminal and run

```
python mainModelEstimationCode.py
```

In addition to estimating the coefficients, `mainModelEstimation.py` also exports the coefficients to `/home/xubuntu/Replication/Paper Results/ReplicationOutput.rdata`.

Instructions for replicating the tables and figures

To replicate the tables and figures, navigate to `/home/xubuntu/Replication/Paper Results/`. These are all R files and can be run within any standard R environment or through the terminal.

- `analyticalSE_code.r` reads in the model coefficients and exports the coefficients and standard errors to `mainModelResults.rdata`. Source this file in R or run

  ```
  Rscript analyticalSE_code.r
  ```

  in the terminal

- `Section4/Table1.r` replicates Table 1. This file can be sourced in R or by opening a terminal in the `Section4` folder and running

  ```
  Rscript Table1.r
  ```

- `Section5/Section5.0/replication5.0.R` produces and Figure 1. This file can be sourced in R or by opening a terminal in the `Section5.0` folder and running

  ```
  Rscript replication5.0.R
  ```

- `Section5/Section5.1/Table2.R` replicates Table 2. This file can be sourced in R or by opening a terminal in the `Section5.1` folder and running

  ```
  Rscript Table3.R
  ```

- `Section5/Section5.2/replication5.2.R` generates the information used to create Figure 2. As discussed above, we draw the regression tree in LaTeX using the TikZ to produce a more customized figure, `Figure2.pdf` looks different from this image, but the information is identical. It also produces the information for Table 3 and prints it to the screen. This file can be sourced in R or by opening a terminal in the `Section5.2` folder and running.

```
Rscript replication5.2.R
```

- `Section5/Section5.3/replication5.3.R` replicates Figures 3 and 4. This file can be sourced in R or by opening a terminal in the `Section5.3` folder and running

  ```
  Rscript replication5.3.R
  ```

- `conductAnalysis.sh` runs all the above files on this list and prints the output to screen. Run this file by opening a opening a terminal in this folder and running

  ```
  bash conductAnalysis.sh
  ```