Replication instructions

Saman Darougheh*

June 2024

The code here has successfully finished on a Linux machine running Ubuntu 20.04.6 LTS. The quantitative part needs at least 100GB memory to complete. If the model is to be re-run (as opposed to using downloadable cache files that contain the solutions), more memory is recommended for parallel computations.

1 Empirical part

1.1 Data

Requirement: Anaconda.To access the data, either

- Download 3 publicly available datasets from IPUMS. IPUMS-descriptions.txt contains samples, variables, and what name the final dataset should have. Store these in empirical/data_orig.
- Or use the three provided json descriptions for IPUMS download under empirical/data_orig. I have provided a python script to download these automatically from IPUMS. To use the script:
    1. If you do not have an IPUMS account, create one at https://cps.ipums.org/cps/
    2. If you do not have an IPUMS API key, create one at https://account.ipums.org/api_keys
    3. Edit main_download_data_empirical.py, line 8, to replace "X" with your API key.
    4. Install environment "ipums.yml" using Anaconda.
    5. In a command line, activate the ipums environment, change into the top level of the replication package, and run main_download_data_empirical.py. This will download and extract all needed stata files in the empirical/data_orig/ folder.

1.2 Analysis

Requirement: Anaconda, Stata 18.

*Danmarks Nationalbank. E-mail: s.darougheh@gmail.com.

1. Install environment "mismatch.yml" using Anaconda.

2. In a command line, activate the mismatch environment, change into empirical/ and run main.py.

2 Quantitative part

Requirement: Anaconda, x86-64 linux operating system.

1. Install environment "myenv3.yml" using Anaconda. Activate that environment, and, if not present:

2. pip install brotli

3. pip install git+https://github.com/uqfoundation/pathos

    - Additional information under https://stackoverflow.com/questions/27873093/installing-python-package-pathos-from-git

4. config.py controls folder for cache_folder (created during runtime), or CACHE_SHARED_FOLDER (created during runtime, but can be used to create final figures)

5. I provide the CACHE_SHARED files necessary to recreate the main figures in the .7z-zipped files in this package. If you simply want to load the final results, unzip the files into the CACHE_SHARED folder, and run "python main.py -profile default" from the quantitative/mismatch folder

    - The cache_shared.7z.001-cache_shared.7z.010 all are needed to regenerate the cache files
    - Store them in the same folder and unzip using .7z
    - On Ubuntu:
        - sudo apt-get install p7zip-full
        - 7z e cache_shared.7z.001

6. Table 3 is based on many alternative transitions, which in zipped form add up to 80GB. Please contact me directly if you are interested in obtaining these.

7. If you want to recreate the results from scratch, be aware that the code is both memory and CPU intensive.

    - CONTENT_FOLDER defines where output is to be stored.
    - N_CORES allows you to calibrate several steady states in parallel
    - N_CORES_TRANSITION allows you to run several transitions or simulations (memory-heavy) in parallel

- Each transition requires approx 80GB in memory
- After setting up these in the config, simply run from quantitative/mismatch "python main.py" to compute all of the steady states and transitions. If you're only interested in the baseline results, run "python main.py -profile default"
- If you're interested in the robustness table, after having computed all of the transitions, run "python main_comparison.py".