1 Structure Second_simulation_exercise

The folder Second_simulation_exercise contains three subfolders:

- r3, which replicates Tables C.3 and C.4;
- r4, which replicates Tables C.5 and C.6;
- r5, which replicates Tables C.7 and C.8;

Each of the three folders contains four subfolders:

- generate_data, which generates the data;
- 5, which computes the estimated identified sets under specification [5];
- 6, which computes the estimated identified sets under specification [6];
- Table, which constructs the two tables.

Gurobi and MOSEK should be downloaded and installed before running the codes. See https://www.gurobi.com/academia/academic-program-and-licenses/ and https://www.mosek.com/downloads/.

2 r3

Step 1 Open the folder generate_data. Save locally the Matlab array probs_allyears_data.mat created in the empirical application. Run the Matlab script main0.m which generates the data, the empirical match probabilities (PYX_cond.mat, PXY_cond.mat), and the Logit estimates of U and V (U_CS.mat, V_CS.mat). The Matlab script main0.m calls the ancillary scripts randsmpl.m and sup.m, which are used to generate the individual types, and equilibrium.m, which is used to simulate the equilibrium. The script randsmpl.m can be downloaded from https://uk.mathworks.com/matlabcentral/fileexchange/59724-fast-sampling-from-a-discrete-distribution?s_tid=prof_contriblnk.

Step 2 Open the folder 5. Open the folder M. Save locally the Matlab arrays PYX_cond.mat and U_CS.mat. Run the Matlab script main.m which constructs the estimated identified set of U and stores it in a matrix IdSetM.mat. The Matlab script main.m calls the ancillary scripts changecoord_3.m, Oneside.m, pairIndices.m, and useful_anyparam.m, which are used to run the linear programs.

Open the folder W. Save locally the Matlab arrays PXY_cond.mat and V_CS.mat. Run the Matlab script main.m which constructs the estimated identified set of V and stores it in a matrix IdSetW.mat. The Matlab script main.m calls the ancillary scripts changecoord_3.m, Oneside.m, pairIndices.m, and useful_anyparam.m, which are used to run the linear programs.

Open the folder `Compose` and save locally the Matlab arrays `IdSetM.mat`, `IdSetW.mat`, `U_CS.mat`, `V_CS.mat`, `PYX_cond.mat`, and `PXY_cond.mat`. Run the Matlab scripts:

- `main_M.m`, which computes the estimated identified set of U and stores it in the Matlab cell array `U_interval.mat`;
- `main_W.m`, which computes the estimated identified set of V and stores it in the Matlab cell array `V_interval.mat`;
- `main_both.m`, which computes the estimated identified set of Φ and stores it in the Matlab cell array `Phi_interval.mat`; it also computes the Logit estimates of Φ and stores them in the matrix `Phi_CS.mat`;
- `main_core.m`, which computes the estimated identified set of $D(\Phi)$ and stores it in the Matlab cell array `D_interval.mat`; it also computes the Logit estimates of $D(\Phi)$ and stores them in the matrix `D_CS.mat`;
- `main_marital.m`, which computes the estimated identified sets of $C(U)$ and $C(V)$, and stores them in the Matlab array `C_interval.mat`; it also computes the Logit estimates of $C(U)$ and $C(V)$, and stores them in the Matlab array `C_CS.mat`.

Step 3 Repeat Step 2 for the folder `6`.

Step 4 Open the folder `Table`. Run the Matlab script `gen_table` which reproduces Tables C.3 and C.4. The script calls the ancillary scripts `fill_function.m`, `fill_function_CS.m`, `fill_function_CS_part2.m`, `fill_function_part2.m` `load_function.m`, `load_function_CS.m`, `fill_function_true.m`, and `load_function_true_indep.m` which help fill the table. It also calls `matrix2latex.m` which creates a LaTex file and can be downloaded from `https://uk.mathworks.com/matlabcentral/fileexchange/4894-matrix2latex?s_tid=mwa_osa_a`. The resulting table is then split into Tables C.3 and C.4 which are slightly reformatted to fit the paper format.

3 r4

Step 1 As for `r3`.

Step 2 Open the folder `5`. Open the folder `M`. This folder contains four subfolders:

- `M_1`, which constructs the estimated identified set of $U_{1\cdot}$;
- `M_2`, which constructs the estimated identified set of $U_{2\cdot}$;
- `M_3`, which constructs the estimated identified set of $U_{3\cdot}$;
- `M_4`, which constructs the estimated identified set of $U_{4\cdot}$.

We now explain how to obtain such sets. Open the folder `M_1`. Save locally the Matlab arrays `PYX_cond.mat` and `U_CS.mat`. Submit the shell script `cg_s.sh` to a computer cluster by using the command `qsub`. This file runs sequentially the Matlab scripts `main.m` and `f_s.m`. The Matlab script `f_s.m` is parallelised as an SGE array job. Each parallel worker *id* takes some rows of the grid of parameters to be evaluated and saves the rows which belong to the estimated identified set of $U_{1\cdot}$ in the Matlab array `IdSetM.id.mat`. The Matlab script `f_s.m` calls the ancillary scripts `changecoord_4.m`, `myind2ind.m`, `Oneside.m`, `pairIndices.m`, and `useful_anyparam.m`, which are used to run the linear programs. Repeat for `M_2`, `M_3`, and `M_4`, possibly in parallel.

Open the folder `W`. This folder contains four subfolders:

- `W_1`, which constructs the estimated identified set of $V_{\cdot 1}$;
- `W_2`, which constructs the estimated identified set of $V_{\cdot 2}$;
- `W_3`, which constructs the estimated identified set of $V_{\cdot 3}$;
- `W_4`, which constructs the estimated identified set of $V_{\cdot 4}$.

We now explain how to obtain such sets. Open the folder `W_1`. Save locally the Matlab arrays `PXY_cond.mat` and `V_CS.mat`. Submit the shell script `cg_s.sh` to a computer cluster by using the command `qsub`. This file runs sequentially the Matlab scripts `main.m` and `f_s.m`. The Matlab script `f_s.m` is parallelised as an SGE array job. Each parallel worker *id* takes some rows of the grid of parameters to be evaluated and saves the rows which belong to the estimated identified set of $V_{\cdot 1}$ in the Matlab array `IdSetW.id.mat`. The Matlab script `f_s.m` calls the ancillary scripts `changecoord_4.m`, `myind2ind.m`, `Oneside.m`, `pairIndices.m`, and `useful_anyparam.m`, which are used to run the linear programs. Repeat for `W_2`, `W_3`, and `W_4`, possibly in parallel.

Go back to the folder `5`. Run the Matlab script `compose_results_HPC.m` which composes all the files created above into the Matlab arrays `IdSetM_final_x.mat` and `IdSetW_final_y.mat` for $x = 1, \ldots, 4$ and $y = 1, \ldots, 4$. For instance, `IdSetM_final_2.mat` is a matrix with 5 columns representing the estimated identified set of $U_{2\cdot}$. `IdSetW_final_4.mat` is a matrix with 5 columns representing the estimated identified set of $V_{\cdot 4}$.

Next, open the folder `Compose`. Save locally the Matlab arrays `IdSetM_final_x.mat` and `IdSetW_final_y.mat` for $x = 1, \ldots, 4$ and $y = 1, \ldots, 4$, `U_CS.mat`, `V_CS.mat`, `PYX_cond.mat`, and `PXY_cond.mat`. Run the Matlab scripts:

- `main_M.m`, which computes the estimated identified sets of U and stores it in the Matlab cell array `U_interval.mat`;
- `main_W.m`, which computes the one-dimensional projections of the estimated identified set of V and stores it in the Matlab cell array `V_interval.mat`;
- `main_both.m`, which computes the estimated identified set of Φ and stores it in the Matlab cell array `Phi_interval.mat`; it also computes the Logit estimates of Φ and stores them in the matrix `Phi_CS.mat`;

- `main_core.m`, which computes the estimated identified set of $D(\Phi)$ and stores it in the Matlab cell array `D_interval.mat`; it also computes the Logit estimates of $D(\Phi)$ and stores them in the matrix `D_CS.mat`;
- `main_marital.m`, which computes the estimated identified sets of $C(U)$ and $C(V)$, and stores them in the Matlab array `C_interval.mat`; it also computes the Logit estimates of $C(U)$ and $C(V)$, and stores them in the Matlab array `C_CS.mat`.

Step 3 Repeat Step 2 for the folder `6`.

Step 4 Open the folder `Table`. Run the Matlab script `gen_table` which reproduces Tables C.5 and C.6. The script calls the ancillary scripts `fill_function.m`, `fill_function_CS.m`, `fill_function_CS_part2.m`, `fill_function_part2.m` `load_function.m`, `load_function_CS.m`, `fill_function_true.m`, and `load_function_true_indep.m` which help fill the table. It also calls the script `matrix2latex.m` which creates a LaTex file and can be downloaded from `https://uk.mathworks.com/matlabcentral/fileexchange/4894-matrix2latex?s_tid=mwa_osa_a`. The resulting table is then split into Tables C.5 and C.6 which are slightly reformatted to fit the paper format.

4 r5

Repeat the steps outlined in Section 3 for the folder `r5`.

5 Figure_C1

Open the folder `r4/[5]/Figure_C1` which reproduces Figure C.1. This folder contains five subfolders, `M_grid1`,..., `M_grid5`, one for each of the five grid of parameter values to be evaluated.

Open the folder `M_grid1`. Save locally the Matlab arrays `PYX_cond.mat` and `U_CS.mat` created in the folder `r4/generate_data`. Submit the shell script `cg_s.sh` to a computer cluster by using the command `qsub`. This file runs sequentially the Matlab scripts `main.m` and `f_s.m`. The Matlab script `f_s.m` is parallelised as an SGE array job. Each parallel worker *id* takes some rows of the grid of parameters to be evaluated and saves the rows which belong to the estimated identified set of U_1. in the Matlab array `IdSetM.id.mat`. The Matlab script `f_s.m` calls the ancillary scripts `changecoord_4.m`, `myind2ind.m`, `Oneside.m`, `pairIndices.m`, and `useful_anyparam.m`, which are used to run the linear programs. Repeat for `M_grid2`,..., `M_grid5`, possibly in parallel.

Go back to the folder `r4/[5]/Figure_C1`. Run the Matlab script `compose_results_HPC.m` which composes all the files created above into the Matlab arrays `IdSetM_final_gridh.mat` for $h = 1, \ldots, 5$. For instance, `IdSetM_final_grid3.mat` is a matrix with 5 columns representing the estimated identified set of U_1. based on the third grid of parameter values. Finally, run the Matlab script `main_pic.m` which reproduces Figure C.1.