Read Me "Occupations and Import Competition"

Sharon Traiberman

July 8, 2019

Main Data Cleaning and Estimation (Available on Stats DK)

This code is only available through access to Statistics Denmark. Upon receiving access, one can receive copies of my code by contacting the Data Manager at Aarhus University. All of my code is freely available for any replication, or future use. This documentation outlines the files used, with some detail since not all code and data are public. The is a file `master_FINAL.do` which runs ALL files INCLUDING MATLAB (through shell). This file MUST be run as it contains PATH DECLARATIONS. Despite the length of this document, ALL FILES are called by this single file. I highly recommend running this and then troubleshooting through calls to this program. A few matlab files (detailed below) also contain path definitions. The code must be run on a machine with access to 4 cores for bootstrapping (or the parfor loop must be manually changed in Matlab). File headers and the code proper contain additional comments. Easiest access to these files are through my project at Aarhus University (accessible by anyone with access through Aarhus University). Alternatively, email me at sharon.traiberman@gmail.com for troubleshooting.

As a brief review of the main estimation routine, the first stage estimation uses the EM algorithm and makes the assumption that wage shocks occur after selection has taken place. Thus, the EM algorithm amounts to iterating on two sets of regressions:

1. Conditional on selection probabilities, wage parameters are estimated by regressing observed (log) wages on covariates:

   $$w_{it} = \beta X_{it}$$

2. Conditional on the likelihood of the observed wages, selection probabilities are updated by running a weighted LPM model:

   $$\delta_{it} = \gamma X_{it}^S$$

   where δ is a $|\mathcal{O}|$-vector with a single 1 element (corresponding to the worker's choice) and 0s elsewhere.

These objects are combined to assign type probabilities to each individual. Thus, for the first stage, the following objects must exist:

- $\{w_{it}\}$ - NT panel of wages
- $\{X_{it}\}$ - $NT \times k$ panel of covariates for the wage equation (including dummies for lagged choice)
- $\{\delta_{it}\}$ - $NT \times |\mathcal{O}|$ panel of choices
- $\{X_{it}\}$ - $NT \times k^S$ panel of covariates for the switching equation (including dummies for lagged choice)

Hence, for anyone doing replication or using this code in the future, the Matlab code in Part 3 will work as long as a version of these matrices exist (see the files proper Parts 1+2 of this ReadMe are only about the specific cleaning I do to get the Danish Administrative Data into the format above. Part 4 is about bootstrapping.

For the second stage estimation, I rely on CCPs. This requires the following objects:

1. $\{Y_{it}\}$ - the LHS CCPs as described in the main text
2. $\{X_{it}\}$ - right hand side covariates (including moving costs, fixed effects, and worker covariates)
3. $\mathbf{W}$ - an [optional] weighting matrix on the CCPs to allow for differences in how CCPs are weighted (useful for Appendix G)

Hence, given these objects, the file `lcmEstimation.m` will work. The other files in Part 5 are only about specifically converting Danish administrative data output into the format needed for estimation. Part 6 is about bootstrapping.

Finally, parts 7 and 8 are specific to the paper and only useful for replicating particular tables, figures, or appendices.

Part 0: DATA REQUIREMENTS

In order to run this code you will need access to the following registers from Statistics Denmark:

- IDA - Employee-employer panel
- IDAS - Plant/Firm level data on industry and age
- UHDI - Discretionary trade data

- VARS - Manufacturing output data (only used for offshoring instrument, not needed for main text)
- fire - Standard firm level data on revenue, wage bills, capital, etc.
- figt - More detailed firm level data on capital and balance sheets (used to calculate value-add)

With access to these *VIEWS* files, the zip folder `sasExtractionFiles` contains a SAS routine for each VIEW, that extracts the relevant variables and years for analysis.

Part 1: Cleaning

1. `sasExtraction.zip` – a zip folder of SAS files that extracts SAS views to dta files. All cleaning is done in Stata.
2. `createLEEDPanel.do` – code that reads in all raw data after converting from SAS and does cleaning WITHOUT changing the sample. Aside from reading the data, the key tasks performed are: [1] linking employees to employers and extracting industry affiliation; [2] constructing education categories; [3] concording industry classifications over time; [4] deflating all nominal values to 2000 DKK. Final product here is [1] worker panel (for bulk of analysis), [2] trade data (for calculating offshoring), [3] firm level data (for calculating labor shares in the counterfactual)
3. `newOccCleaning.do` – main code for all definitions, cleaning and sample selection. Everything this code does is contained in the data appendix. Result of this file is the final panel for analysis containing information on occupation/sector/employment, tenure, income, age, and education.
4. `makeTasksCV.do` – code that creates tasks using ONET + weights from Danish LEED data (parts are publicly available)
5. `makeMatlabFiles.do` – This file converts the stata files to matlab CSVs for doing the first stage (EM algorithm). Also cuts out rows with missing data. Finally, runs an initial FE regression to initialize the EM algorithm.

Part 2: Reduced Form/Summary Stats on Sample Frame

1. `summaryStatTables.do` – Figure 1-3, Table 1-3
2. `makeBWindices.do` – Cleans up trade data and constructs price indices from customs data (for calibration, I also construct from Prodcom data)
3. `tradeExercises.do` – Figure 4 and 6

Part 3: First Stage Estimation

1. `estimateModel.m` – main call to all auxiliary matlab files (called internally from Stata). To run, this file MUST BE EDITED for PATH DECLARATIONS. The other files DO NOT need to be edited, except for programming parameters if desired.

2. `readData.m` – reads in data from Stata

3. `pseudoEM.m` – runs the first stage EM estimator

4. `EMpooledPost.m` – writes first stage parameter estimates to CSVs for Stata

5. `recoverIncomeLevels_full.m` – recovers skill prices and outsheets to CSVs for Stata

Part 4: First Stage Results + Bootstrapping First Stage

1. `makeAfterMatlab.do` – creates smoothed EDF for counterfactuals, tests for sorting (see footnote 28)

2. `makeBootstrapSamples.do` – Creates bootstrap files. WARNING: Stata seed is set, but not sortseed. Exact numerical duplication is not guaranteed, but discrepancies are insignificant. See: https://www.stata.com/help13.cgi?sortseed

3. `estimateBootModel.m` – matlab code called from inside stata to bootstrap. Reruns Part 3 code on bootstrap samples. This file MUST BE EDITED for PATH DECLARATIONS

4. `postBootstrap.do` – Reads in bootstrap results for creating tables and for bootsrapping in second stage

Part 5: Second Stage Estimation

1. `readData.do` – formats first stage estimates for second stage

2. `makeMoments.do` – constructs the second stage estimator (see equations 17/18)

3. `lcmEstimation.m` – minimizes the second stage objective (nonlinear least squares turns out to be easier in matlab than stata, hence the outsourcing). “lcm” refers to “life cycle model. There is a publicly available monte carlo simulator of the estimator that ignores the life cycle component for ease of computation available at https://sites.google.com/site/straiberman/research.

Part 6: Second Stage Bootstrapping

1. `makeMoments_BS.do` – constructs bootstrap sample second stage from bootstrapped first stage estimates

2. `lcmEstimationParallel_BS.m` – reruns second stage estimator on bootstrapped samples. WARNING: This file assumes access to 4 cores. This is currently available on the Denmark servers. Access to more/less cores can dramatically change waiting time. Recommendation is to use max cores available.

3. `lcmEstimation_timeBS.m` – reruns second stage estimator ignoring first stage bootstrapping, and bootstrapping only on time periods.

Part 7: Exporting Tables + Robustness

1. `acmEstimation.do` – this file performs the ACM and AM estimators/specifications. columns 2-5 in Table 5; this also performs analysis for Appendix H
    - `poisson_sandbox3.do` – subroutine for PPML estimation
    - `acmSectors_sandbox.do` – subroutine for column 2 of table 5

2. `calculateConditionalSwitchingCosts` – table 6 column 2

3. `calculateSwitchingCosts` – table 6, column 1 and 3

4. `exportTables.do` – creates exportable .txt files for public release (after running this file, to have files released, you will need permission of the data manager)

5. `postEstimation.do` (PUBLICLY RELEASED) – Extracts parameter estimates and converts them into format for Stata. Also makes tex format for results tables (appendix A) and Table 5-6.

6. `makeCAtable.do` (PUBLICLY RELEASED) – Table 4 and Figure 5

Part 8: Alternative Estimators (Appendices F+G)

1. `appF_StataFiles` – contains copies of `makeMatlabFiles`, all Part 4/5/6 stata codes. All files end in "rev1". To run this code, replace relevant files with "rev1" version in the flow of code.

2. `appF_stage1MatlabFiles` – contains copies of all Part 3 matlab codes. All files end in "rev1". To run this code, replace relevant files with "rev1" version in the flow of code.

3. `appF_stage2MatlabFiles` – contains copies of all Part 5/6 matlab codes. All files end in "rev1". To run this code, replace relevant files with "rev1" version in the flow of code.

4. `jackknife_StataFiles.zip` – contains codes for calculating jack knife correction in appendix B. Contains the file `make_splitSample` which replaces `makeMatlabFiles`. Additionally, contains Part 4-5 stata codes for Appendix B. All files end in "splitSample". To run this code, replace relevant files with "splitSample" version in the flow of code.

5. `jackknife_stage1MatlabFiles.zip` – contains copies of part 3 matlab codes for calculating jack knife correction in appendix B. All files end in “splitSample”. To run this code, replace relevant files with “splitSample” version in the flow of code. NOTE: for second stage, split sample can simply use the usual second stage code, modified to run twice on each sample. Hence, no additional files.

6. `appG_stataFiles.zip` – contains copies of part 5/6 stata codes for appendix G. All files end in “rev2”. To run this code, replace relevant files with “rev2” version in the flow of code.

7. `appG_matlabFiles.zip` – contains copies of all Part 5/6 matlab codes. All files end in “rev2”. To run this code, replace relevant files with “rev1” version in the flow of code. NOTE: There is no new first stage for Appendix G, hence only one set of matlab files.

Calibration Code [Demand Side]

Run files in this order. Documentation for each file is in the header of each file. The file `allCalibration.do` contains [1] path declarations (YOU MUST RUN THIS BEFORE RUNNING THE FILES!) and [2] runs all the files in the correct order. Documentation is sparser in the Readme as these files are publicly available.

1. `makeEDF.do` – uses a smoothed (due to data censoring issues) EDF of the characteristics of the Danish population in 1996 to construct a weighted grid of initial points for simulation

2. `makeIOTable.do` – converts Danish IO tables to input-output matrices for Matlab

3. `prodcom` & `makeForeignPrices_wits` – constructs import price indices using two different sources (for comparison)

4. `demandParameters.do` & `makeIncomeLevels.do` – uses [publicly available] IO tables AND [proprietary but released] aggregated data on occupational labor shares to construct Cobb-Douglas shares, export demand, total revenues, and production coefficients.

5. `makeMatlabMatrices.do` – Takes output above and rewrites everything for ease of use by Matlab. Importantly: recodes industries to be indexed 1-N, same with occupations, etc.

Counterfactuals

Run files in this order. Documentation for each file is in the header for each file. Documentation is sparser in the Readme as these files are publicly available.

Stata:

1. `postEstimation.do` – Extracts parameter estimates and converts them into format for Matlab. Also makes all results tables (appendix A), Table 5 (column 1) and table 6. This file is previously mentioned in the cleaning/estimation section.

Matlab:
The file `runAllFiles.m` will run all files. This file ALSO runs counterfactuals for appendix I and for the Offshoring Counterfactual (both available online).

1. `steadyStateCapital.m` - Myopic steady state for initial guess

2. `steadyState_pf_cap.m` - Iterates from the initial equilibrium (1996) to a steady state assuming a fixed stock of capital and perfect foresight

3. `actualTransitionCapital.m` - Solves the transition from the initial steady state to the post-trade steady state forward assuming myopia (this is used as an initial guess for the perfect foresight equilibrium as it is very fast to compute this)

4. `cfTransitionCapital.m` - Solves the transition from the initial steady state onward for 40 more periods

5. `actualTransition_pf_cap.m` - Uses the myopic guess to solve for the transition dynamics from the initial steady state to a new equilibrium

6. `counterfactualTransition_pf_cap.m` - Solves from the initial steady state going forward assuming no change in variables[1]

7. `postCounterfactualSim.m` - Figure 7 + Table 8, also creates simulated histories for creation of Figure 8 and Table 9

Auxiliary code called:

- `readInDemandData.m` - Reads in calibrated parameters for demand side. This code MUST BE EDITED BY USERS BEFORE USING. It contains PATH DECLARATIONS.

- `readInParameters.m` - Reads in estimated parameters from supply side. This code MUST BE EDITED BY USERS BEFORE USING. It contains PATH DECLARATIONS.

[1] See paper discussing why this is necessary. Also useful for moving other parameters - productivity, capital prices, etc.

- `getContinuationValues_n` - Given a guess of wages and price of capital one can calculate from cost minimization the full path of real prices which, paired with initial conditions of labor, allows for calculation of the full path of labor supply decisions and all continuation values
- `excessDemandCapital` - Calculates labor excess demand for myopic model (period-by-period)
- `excessDemandCapital_pf` - Calculates labor excess demand for perfect foresight model (all periods simultaneously)