Endogenous Vertical Differentiation, Variety, and the Unequal Gains from Trade

README FILE

September 16, 2021

1 General Comments

The majority of files set the working directory to "D:/data_replication". This folder needs to be adjusted depending on the user's directory. The codes for this project were written in R (Version 4.1.0), Matlab (Version R2019a), and Stata (Version 14). In terms of software, the R packages Rcpp and RcppArmadillo are required as well as the optimizer ipopt (or an alternative, such as Knitro). Realistically, one also needs access to a server given both the estimation time and memory requirements.

The structure of the codes is as follows: *data* contains all data used during the estimation, whereas *estimation* contains all codes necessary to replicate the demand estimation, the supply side inference, as well as the counterfactuals. The extensions and robustness checks described in Section 6 in the paper are covered in the folder *robustness*. Lastly, *statistics_and_graphs* creates summary statistics, especially those shown in Figure 1 and Table 1 in the paper.

Codes always begin with A followed by a number and should be run in numerical order when they are located in the same folder, i.e. A1 should be run before A2, and so on.

2 Estimation

2.1 Data Format (estimation/1_data_format)

The main data on trade and production is located in folder data/trade and data/production. These datasets are merged and crosswalked using the files located in folder 1_data_format. "A1_Merge_trade_and production.do" first merges the trade and production data quarter by

quarter. A2_Merge_years.do then crosswalks the data by quarter over time, using a time-consistent classification (pc8plus). After this step, A3_Create_dataset.do merges in distance and demographics used in the estimation, creates the baseline instruments, and addresses outliers as described in the paper. A4_import_R.R then splits the data into multiple R Datasets that are later used for the estimation. This avoids having to read in the entire dataset during the estimation in R.

2.2 Aggregate data (estimation/2_product_list & 3_aggregate_data)

The codes in the folder "2_product_list" aggregate the data up whenever a certain 8-digit category is not produced. A1_Aggregate_categories.do does so for each country and creates a final list of all product categories. A2_Merge.do and A3_Create_summary_file.do then merge these country-specific files into one list (product_id_declarant.dta).
The codes in the folder "3_aggregate_data" then use this classification to create the final dataset used for the estimation. A1_Create_dataset.do merges in distance and demographics used in the estimation, creates the baseline instruments, and addresses outliers as described in the paper for each product category used. A2_import_R.R then splits the data into multiple R Datasets that are later used for the estimation. This avoids having to read in the entire dataset during the estimation in R. A3_Create_basefile.do collects all estimation data into 1 file which is later used to create summary statistics.

2.3 Demand Estimation (estimation/4_demand_estimation)

The main files for the demand estimation are located in folder codes_server. A1_BLP_full.R uses 3 inputs: i_group, which specifies the R-file it accesses, kk, which specifies the product within that file, and reps, which chooses a particular starting value (out of 5). A1_BLP_full.R estimates demand for the full sample, whereas A2_BLP_sub.R restricts the sample by removing small market shares.
Demand for all products and starting values would need to be estimated and the results of each is saved as resultsr_p_r, where p denotes a product category and r a starting value. Since this involves more than 20,000 instances, this part realistically requires the use of a server. In terms of software, the packages Rcpp and RcppArmadillo are required as well as the optimizer ipopt (or an alternative such as e.g. Knitro).
Once the demand estimation has finished, the codes in the 4_demand_estimation folder A1_Create_Baseline.do and A2_import_matlab.m collect the results, format them, and then create the file main_data_supply.mat, which will be the main estimation file on the supply side. As described in the main text, the results of A1_BLP_full.R are only replaced by those

generated with A2_BLP_sub.R if no starting value converged. Output files for both codes in an example sector are posted in the folders 1_results_subsample and 2_results_full.
The demand estimation folder also contains how the Cobb-Douglas Weights are inferred (folder 4_cobb_douglas_weights). These codes would need to be run after the Matlab file main_data_supply.mat is created since some of these programs build on this file.

2.4 Supply Side (estimation/5_supply_side)

These codes infer the supply side of the model, i.e. the cost parameters as well as entry and exit. The file A1_cost_estimation.m first infers the fixed and marginal cost and then computes counterfactual adjustments when moving to autarky. Since this involves more than 4,000 instances, this part realistically requires the use of a server. The output files for each sector are saved in folder cost_estimates. These files are numbered such that good 1 is that with the lowest price coefficient in the dataset (i.e. the most price-elastic one), while the last good has the highest. An output file for one example sector is posted in the folder cost_estimates.
For some products, fsolve() can struggle to find a solution for the fixed cost in an individual market, especially when this market features a very large factual number of firms as well as when the price coefficients are extremely high or low (i.e. for low or high i_k). These cases can either be resolved by using a more sophisticated optimizer, such as Knitro (see Code) or Ipopt, or by also accepting output flag 2 and 3 instead of just 1. Since the fixed cost in practice has little impact on the outcome of the counterfactuals, the chosen approach has little impact on the results overall.
Based on the adjustments computed via A1_cost_estimation.m, A2a_price_index_no_adj.m, A2b_price_index_entry.m, and A2c_price_index_full_adj.m determine factual and counterfactual price indexes in each sector in the no-adjustment and the entry-only, as well as in the full model. Codes A3a - A3c collect these sector-specific results and create the results and plots for Table 2. Codes A4a - A4c create the final ideal prices indexes over all sectors and create the results and plots for Table 3. An output file for one example sector is posted in the folder price_index/product_specific.

3 Robustness

3.1 Price Adjustments only (robustness/price_adjustments_only)

The codes in this folder evaluate the case in which only prices and the number of firms can adjust in a counterfactual move to autarky, i.e. they create the results presented in

Column (5) of Table 2. The structure is very similar to that in the baseline specification (estimation/5_supply_side): A1_cost_estimation.m infers the cost parameters as well as counterfactual entry and price adjustments. Codes A2 and A3 then evaluate both factual and counterfactual price indexes.

3.2 Aggregation Levels (robustness/PC2 & PC4 & PC6)

The codes in these folders perform the analysis described in the estimation part on the 2-, 4-, and 6-digit aggregation level. The structure is similar to that in the baseline case: 1_data_format contains the files necessary to format the data and prepare it for the demand estimation. 2_demand_estimation contains the codes necessary for the estimation of demand and then saves the output into a file main_data_supply.mat, respectively. 3_supply_side then determines costs, and evaluates the counterfactuals and welfare.

3.3 Excluding Intermediates (robustness/excl_intermediates)

The codes in this folder evaluate the counterfactuals for the case in which intermediates are excluded. A1a_price_index_entry.R and A1b_price_index_full_adj evaluate the average extent of inequality in the gains from trade across sectors while A2a_Price_index_agg_entry.R and A2b_Price_index_agg_full_adj.R compute the degree of inequality in the overall ideal price index.

3.4 Overall Price Index (robustness/overall_PI)

The codes in this folder evaluate the overall price index which accounts for service consumption. A1_PI_entry_overall.R computes this index at factual and counterfactual levels when firms can only enter and exit while A2_PI_full_adj_overall.R computes it for the full model.

3.5 Alternative Instruments (robustness/alt_IV)

The codes in these folders perform the analysis using the alternative instruments described in the main text. The structure is the same as that in the baseline case: 1_data_format contains the files necessary to format the data and prepare it for the demand estimation. 2_demand_estimation contains the codes necessary for the estimation of demand and then saves the output into a file main_data_supply.mat, respectively. 3_supply_side then determines costs, and evaluates the counterfactuals and welfare.

3.6 Firm Heterogeneity (robustness/firm_heterogeneity)

The codes in these folders relate to the extension described in Section 6 regarding firm productivity. Codes A1a_mc_estimation.m and A1b_fixed_cost_estimation first infer the fixed and marginal cost and create sector-specific factual datasets with all firms and their corresponding cost parameters. A1c_counterfactual_entry.m and A1d_counterfactual_full_adj.m then evaluate changes in prices, quality and entry counterfactually. These codes can be quite time-intensive and also have to be performed for each of about 4,000 sectors. This part therefore again realistically requires the use of a server.
Based on the estimates obtained in the previous part, the codes A2b_price_index_entry.m and A2c_price_index_full_adj.m, compute ideal prices indexes for each sector for the entry-only and full model. The A3 and A4 codes then collect these estimates and evaluate the extent of inequality in the gains from trade on average across sectors and overall.

3.7 Upper Tier CES (robustness/nested_CES)

The codes in this folder relate to the extension described in Section 6, in which preferences over sectors are CES. The codes A1a_extract_data_entry.m and A1b_extract_data_full_adj.m first extract the necessary information on price indexes and income from the baseline results for the entry-only case and the full model. The Matlab codes A2a_format_entry.m and A2b_format_full_adj.m then format the data and address outliers, while after that step, A3a_price_index_entry.do and A3c_price_index_full_adj.do evaluate the extent of inequality in the gains from trade. Lastly A3b_reg_entry.R and A3d_plot_full_adj.R create the plots shown in Appendix Table B3.