Read Me File for "Demand Estimation Under Incomplete Product Availability"

By Christopher T. Conlon and Julie Holland Mortimer

This file describes the datasets and programs used in the empirical analysis of the paper.

I. Datasets

The data used in the paper are proprietary. They were provided to us by North County Vending. In September 2011, North County Vending was sold to Compass Group's Canteen Vending Division. Researchers wishing to obtain the data for the purposes of replicating our results should contact Compass Group's Canteen Vending Division.

II. Descriptive Stata Code

Two Stata files are provided to generate tables 1-3, the descriptive regressions and summary statistics in the paper

setup.do Reads the raw data into Stata

tables.do Runs a variety of regressions and outputs LATEXtables corresponding to tables 1-3.

III. Programs

There are three sets of programs that we include in the code package. One estimates the random coefficients model via maximum simulated likelihood with many fixed effects in C++. Another estimates the nested logit model via maximum likelihood AMPL with many fixed effects. The third uses the output of both maximum likelihood estimators. It imputes the missing data in the E-Step using MATLAB, and also generates the counterfactual tables in the paper.

A. CPP Code MSL (Estimation of the Random Coefficients Specification)

This code is written in C++ with 64-bit Linux systems in mind[1] and makes use of the following external packages:

[1] It also works under Mac OS 10.6

- KNITRO 7.x: A commercial optimization package developed by Ziena Optimization LLC, and distributed by Artelys.
- Armadillo 3.x: An open source C++ template library for matrix computation.
- OpenMP 3.0: an open source package for parallel computing

Both KNITRO and Armadillo perform substantially better if a machine-specific BLAS is provided during their installation.

Provided with the code is a single *Makefile* that produces all of the required executables.

mslestimate This is the main estimation routine that implements MSL estimation with market-specific fixed effects.

lleval.mexa64 This is a 64-bit MEX (Matlab) file built in C++ that provides function evaluations and gradients to MATLAB. It is used in computing standard errors, and in the E-step.

choiceprob.mexa64 This is a 64-bit MEX (Matlab) file built in C++ that computes choice probabilities at arbitrary availability vectors for a given set of parameters. This is required in the E-Step and is an important part of the counterfactuals.

test This is a C++ program that inspects the data structures and evaluates the function and its derivatives both analytically and by finite differences.

mslconstr This is an alternative to **mslestimate** (not reported in the final results of the paper), which provides a different normalization of the fixed effects – setting the sum to be zero rather than the first fixed effect to be zero. This can be more numerically stable for some datasets.

The requisite .CPP and .H files, and the decencies are all managed by the Makefile and are not described here. The loglikelihood function and its derivatives are implemented in **msllcpp.cpp**.

Example usage for **mslestimate** is:

```
mslestimate ds-full-f.mat  [outfile.mat] >& out-full-constr3.txt
```

Where `[outfile.mat]` is an optional argument for the parameter vector produced by the estimation routine. If that is not provided, the default `knitroresults.mat` is used instead. The same usage would apply to **test** or **mslconstr**.

B. NL Code AMPL (Estimation of the Nested Logit Specification)

Example usage is provided in the AMPL command file `VendingNL.cmd`. This describes how to load the model and the data as well as how to output the results to ASCII text files (that can be read in by MATLAB for E-Steps or counterfactual experiments.

```
# Choose a model file
model VendingNL.mod;
# Chose data corresponding to specification
data vending_data_full.dat

# Solve and Record output tables to ASCII files
solve;
display sum{m in M} xi[m];

table Utils2 OUT "utils_full.tab" :  [J], dj;
write table Utils2;
table Lambda2 OUT "lambda_full.tab" : [G], LAMBDA;
write table Lambda2;
table xi2 OUT "xi_full.tab" : [M], xi;
write table xi2;
table share2 OUT "share_full.tab": [ M, J], {(m,j) in AVAILABLE} pjt[m,j];
write table share2;

# Compute the Sparse Hessian and output that (for Standard Errors)
option solver gjh;
option gjh_options 'sparse';
write "hessian_full", solve;
```

We provide two examples of the Nested Logit code. One for the standard FIML nested logit estimator of McFadden (1979), `VendingNL.mod`, and the other for the EM-adjusted case, `VendingNL-EM.mod`. The primary difference is that in the standard case each observation/market has its own fixed effect. In the EM case, fixed effects for observations before and after the stockout are constrained to be identical. Both sets of code are capable of estimating a single nesting parameter (instead of category-dependent ones) by uncommenting the final constraint and forcing all nesting parameters to be equal.

C. Matlab Code

This code is organized into several task dependent folders:

Helpers These are generic MATLAB functions designed to do various matrix operations like taking the outer product of all rows in a matrix, performing clockwise inverses, or taking logs of only positive elements in a vector. *This needs to be in the MATLAB path.*

Creating Dataset This folder handles the creation and manipulation of the special dataset structure that we use in our programs. This dataset structure contains prices, quantities, availabilities, and product characteristics for all of the files. Some functions are able to collapse dataset structures by sufficient statistics, or split up the dataset according to some mask. The main datasets (one for each specification) are created by running `setup.m`. *This needs to be in the MATLAB path.*

AMPL Translation This folder contains code that allows the MATLAB `ds3` dataset structure to be exported for use in AMPL for the Nested Logit estimation. The main file is `dstoampl2.m` which takes two arguments, one for the `ds3` structure, and the other a string containing the filename of the AMPL dataset to be output. *This is called at the end of each E-Step.*

Standard Errors This folder contains code that allows MATLAB to compute standard errors for the Nested Logit Model. The Random Coefficients Standard Errors are computed by the `secompute` C++ program. There are two scripts. The file `hess_script.m` reads in and computes a sparse inverse of the Hessian matrix from the AMPL output; the file `se_compute.m` computes the standard errors for each of the NL specifications.

EStep This is the largest and most complicated folder. The main files `runem.m` and `runemn.m` compute the E-Step for the Random Coefficients and Nested Logit models respectively. The primary difference between the two is the number of parameters and which choice probability function is called. Each of the E-step functions calls a series of helper functions to deal with the missing data from the stock-out events. One, two, and three simultaneous stock-out events are handled via special case with specific files because these formulations are better for memory usage. Finally, `nmultpdf.m` provides the p.m.f. of the negative multinomial distribution which is not included in MATLAB.

There are a few other key files not in those folders:

read_results.m processes all the results files and reads them into a MATLAB data structure (from both the Nested-logit and Random-coefficients specifications).

makegraphs.m produces all of the figures included in the paper.

counterfactualnew.m produces all of the counterfactual tables included in the paper and the Appendix. Only Table 5 appears in the text of the paper.

loglik.m, choiceprobN.m produce the log likelihood and nested-logit choice probabilities as needed for the counterfactual tables.