"Readme" file for the estimation programs

(for "Optimal Mandates and The Welfare Cost of Asymmetric Information: Evidence from The UK Annuity Market," by Einav, Finkelstein, and Schrimpf)

This is a "readme" file that explains what program files are included in this directory, and how they are organized. To actually run these files, interested researchers would need to contact the company through us and request access to the data, which are unfortunately proprietary. Please contact us at either leinav@stanford.edu (Liran Einav), afink@mit.edu (Amy Finklstein), or paul_s@mit.edu (Paul Schrimpf) for more details about this procedure.

The included version of the code (some in Matlab, some in AMPL) is the one that produces our baseline results, which are reported in Tables III, IV, V, and VI of the paper. If you are interested in any of the other results (that are reported and discussed in the robustness section of the paper), please contact us at the e-mail addresses above. These additional programs are available upon request, but are not as nicely documented, so navigating through them may require some help from us.

Thanks for your interest!

Liran, Amy, and Paul

Overview

Conceptually, estimation proceeds in four steps:

1. Estimate lambda using mortality data
   1.a. Load data into Matlab, set various options, write data in AMPL format
   1.b. Estimate lambda in AMPL, write estimate.
2. Given lambda, compute guarantee cutoffs:
   2.a. In Matlab, read lambda estimate. Given lambda and alpha grid, solve for cutoffs. Objective function is written in a c-mex file.
   2.b. Save cutoffs.
3. Estimate distribution of alpha and beta using mortality and guarantee data:
   3.a. In Matlab, load cutoffs and data, write out in ampl format.
   3.b. Maximize likelihood using AMPL. Try grid of starting values.
4. Run simulations and create tables:
   4.a. Load results in Matlab, run simulations, create tables and graphs.

→ Maximization, steps 1 and 3, is done using AMPL. Step 2, intermediate data preparation, and analysis of the results (step 4) are done using Matlab.

Software required

- Matlab (run in version 7.8)
- AMPL (run using version 20061130)
- SNOPT (run using version 6.1-1(4))
- Perl (run using version 5.8.5)

Student versions of AMPL and SNOPT are available at http://www.ampl.com/ DOWNLOADS/ details.html and are sufficient to run the included files.

Files involved (in order of execution)

- runEstimate.m - Main script
- confPooled.m - options file
- readData.m - loads data
- loadcutoffs.m - computes (or loads if already computed) guarantee choice cutoffs
- cutoffs/findbcut.m - computes cutoffs for beta given alpha and gamma
- cutoffs/findgcut.m - computes cutoffs for gamma given alpha and beta
- cutoffs/vdiff.m - returns the difference in utility between two annuity contracts
- cutoffs/vf3.m - returns the utility of an annuity contract
- cutoffs/vfc1.c - returns utility of a given annuity contract and initial consumption value -- to use this file you will need to compile it to a Matlab "mex" file
- gaussChebyshevInt.m - computes gauss-chebyshev integration points and weights
- gaussLaguerre.m - computes gauss-laguerre integration points and weights
- writeAMPLdata.m - takes data from matlab and writes in ampl format
- mortLikePooled.ampl - AMPL script for step 1 estimation
- mortLikePooled.mod - AMPL model file for step 1 estimation
- pooled.ampl - AMPL script for step 3 estimation
- annuityLikePooled.mod - AMPL model file for step 3 estimation
- amplfunc.dll - library of distribution and integration functions for calling from AMPL. This was compiled for x86 linux. If you have a different type of computer, the source code is contained in amplfuncSource directory. The gnu scientific library (gsl) and a C
- compiler are needed to compile amplfunc.dll.
- writeEstimates.pl - perl script for parsing AMPL output
- estReport.m - makes table of estimates
- simCF.m - runs or load simulations of counterfactuals
- runCF.m - runs simulations of counterfactuals
- sampleAlphaBeta.m - samples alpha and beta for simulating
- cfTables.m - makes tables summarizing counterfactual results
- fit.m - makes tables and figures showing fit of estimates to data