1. Introduction

This code conducts a Monte Carlo study that estimates, nonparametrically the CDF of the random coefficient distribution. The true distribution is a mixture of two, correlated, bivariate normals.

The approximating grid of points comes from a Halton sequence and is stored in gridR. The estimated values of the CDF of the random coefficients are evaluated over (another, evenly spaced) grid of points.

Each iteration of the estimator generates estimated weights, estimated values for the CDF, and 95% confidence intervals for the estimated weights and the estimated CDF. The confidence intervals are constructed using either Tikhonov regularization (using generalized cross validation to choose the perturbation) or subsampling. Subsampling may not have correct coverage for this application. The confidence intervals from regularization tend to be conservative in simulation studies.

The estimated weights are saved in thetahats. The estimated values for the CDF are saved in estimatedCDFs. Endpoints for the 95% confidence intervals for the thetahats are stored in leftEndPoints and rightEndPoints. EndPoints for the 95% confidence intervals for the estimatedCDFs are stored in leftEndPointsCDF and rightEndPointsCDF.

The true CDF is known and is stored in trueCDF. It is not obvious what the "true" theta is, since it is part of an approximation to the true CDF. For the true theta, the weights that give the best approximation to the true CDF, given the number of grid points chosen, are used. These weights are generated by running the estimator for a very large sample size (N=10,000) and are stored in trueTheta. The CDF generated by trueTheta is stored in trueCDF1.

The percentage of time in which the "true" theta and the true CDF falls within the bounds of the confidence intervals is calculated. These coverage calculations are stored in coverage for the estimated theta and coverageCDF for the estimated CDF. The coverage calculations for the CDF are then repeated using the CDF generated from trueTheta. The coverages for trueCDF1 are stored in coverageCDF1.

Comments should be sent to Jeremy Fox at fox@uchicago.edu.
An Qi contributed to this code.

2. The random coefficients logit model

In the logit, agents $i = 1,\ldots, N$ can choose between $j = 1,\ldots J$ mutually exclusive alternatives. For each agent i, the exogenous variables for choice j is stored in a $K \times 1$ vector $x_{i,j}$ which could include the product characteristics, the price of good j, and the demographics of agent i.
Let $x_i = (x'_{i,1},...,x'_{i,J})$

In the model, there are $r = 1\ldots, R$ types of consumers. The fixed preferences of type r are stored in a $1 \times$ K vector β^r. The proportion of type r in the population is θ^r, which sum to 1.

Let $\theta = (\theta^1, ..., \theta^r)$. θ will be estimated using constrained OLS in the code.

For agent *i* of type *r*, the utility derived from choosing good *j* is equal to

$$u_{i,j} = x'_{i,j}\,\beta^r + \varepsilon_{i,j}$$

There is also an outside good with utility $\varepsilon_{i,0}$. Assume that the errors are distributed as Type I extreme value.

Define choice data as

$$y_{i,j} = \begin{cases} 1 \text{ if } u_{i,j} > u_{i,j'} \text{ for all } j' \neq j \\ 0 \quad \text{otherwise} \end{cases}$$

Because the errors are extreme value, it follows that

$$\Pr(y_{i,j} = 1 \mid x_i) = \sum_{r=1}^{R} \theta^r \frac{\exp(x'_{i,j}\,\beta^r)}{1 + \sum_{j'=1}^{J} \exp(x'_{i,j}\,\beta^r)}$$

Given $x_{i,j}$, $y_{i,j}$, and β^r, the code will estimate the parameters θ^r using constrained OLS.

3. Parameters

N represents the number of agents. *J* represents the number of alternatives. *K* represents the number of product characteristics, which is hard-coded in the choice of distribution and should not be changed. *R* is the number of grid points in the approximation. *M* is the number of iterations in the Monte Carlo study. *standardErrorMethod* selects the method used to construct confidence intervals: 0 for Tikhonov regularization, 1 for subsampling.

4. True distribution and grids used for estimation

obj is a mixture of two, correlated, bivariate normals, with distribution as specified in the code. It is the distribution that is to be estimated by the model.

gridR is an R × K array of points generated using a Halton Sequence that covers [-10,10] × [-10,10]. *gridR* will be the approximating grid of points for the estimated weights *theta*. The grid of points used to evaluate the CDF is an evenly spaced grid of points covering [-10,10] × [-10,10]. This grid is stored in *combinedIndex*, a D × K array, where D is the number of grid points.

The true CDF is known and is stored in *trueCDF*. It is not obvious what the "true" *theta* is, since it is part of an approximation to the true CDF. For the true *theta*, the weights that give the best approximation to the true CDF, given the number of grid points R chosen, are used. These

weights are generated by running the estimator for a very large sample size (N=10,000) and are stored in *trueTheta*.

5. Monte Carlo Study

The estimator is iterated M times (see estimator.doc), generating M estimated *thetas*, M estimated CDFs, and M confidence intervals for the *thetas* and the CDFs.

The estimated weights are saved in *thetahats*, an R × M array. The estimated CDFs are saved in *estimatedCDFs*, a D × M array. Endpoints for the 95% confidence intervals for the *thetahats* are stored in *leftEndPoints* and *rightEndPoints*, two R × M arrays. Endpoints for the 95% confidence intervals for the *estimatedCDFs* are stored in *leftEndPointsCDF* and *rightEndPointsCDF*, two D × M arrays.

6. Coverage Calculations

coverageMat is a R × M matrix.

For each *r* and *m*, if $leftEndPoints(r,m) \le thetaMat(r,m) \le rightEndPoints(r,m)$, then *coverageMat(r,m)*=1, otherwise *coverageMat(r,m)*=0.

The entries of *coverageMat* are then summed horizontally and divided by *M*. The resulting R × 1 vector is stored in *coverage* and represents the percentage of time in which each point *r* of *trueTheta* falls within the bounds of the M confidence intervals for that point generated from the Monte Carlo study.

coverageMatCDF is a D × M matrix.

For each *d* and *m*, if $leftEndPointsCDF(d,m) \le CDFMat(d,m) \le rightEndPointsCDF(d,m)$, then *coverageMatCDF(d,m)*=1, otherwise *coverageMatCDF(d,m)*=0.

The entries of *coverageMatCDF* are then summed horizontally and divided by *M*. The resulting D × 1 vector is stored in *coverageCDF* and represents the percentage of time in which each point *d* of *trueCDF* falls within the bounds of the M confidence intervals for that point generated from the Monte Carlo study.