Online Appendix for "A Simple Nonparametric Approach to Estimating the Distribution of Random Coefficients in Structural Models"

Jeremy T. Fox
Rice University & NBER

Kyoo il Kim*
Michigan State University

Chenyu Yang
University of Rochester

May 2016

In our Monte Carlo study, we apply our fixed grid estimators to a capital replacement model in the spirit of Rust (1987). A dynamic program must be solved for each grid point or simulation draw, so working with a dynamic program showcases our method's advantage of requiring computation of choice probabilities only before optimization commences. We vary the number of grid points, the sample sizes and the true distributions of the random coefficients. We compare the least squares and likelihood criteria on a fixed grid (fixed grid estimators) with an alternative likelihood criterion on a flexible grid (flexible grid estimator). For the fixed grid, we optimize over only weights. For the flexible grid, we optimize over both the grid points and the weights, using a much smaller number of weights because of the computational challenges of both optimization and the need to solve the dynamic program many times. The two likelihood estimators are maximized using the EM algorithm. Our results suggest that the two fixed grid estimators are much faster than the flexible grid estimator at the cost of some statistical accuracy.

1 Heterogeneous Parameter Capital Replacement Model

We consider a capital replacement problem. A firm i uses one machine to produce output every period t. Typically, the efficiency of the machine declines with time and the firm eventually finds it profitable to replace the machine. Each period, the firm can choose to do nothing or to replace the existing machine with one of $J-1$ different types of machines. Replacing the existing machine resets firm i's machine's age $a_{i,t}$ to 0. Denote the replacement decision as $j \in \{1, 2, \ldots, J\}$, where $j = 1$ means no replacement and $j \geq 2$ means replacing the current machine with the new machine j and resetting the age $a_{i,t}$ to 0.

The observables (in addition to actions) to the researcher are the age $a_{i,t}$, a scalar shifter of the flow profits from running a machine $d_{i,t}$, the scalar current machine's loss rate from age $w_{i,t}$ in the flow profit from running a machine, the vector of all $J-1$ machines' scalar loss rates from age $v_i = (v_{2,i}, \ldots, v_{J,i})$ and the matrix of shifters of the replacement costs of the $J-1$ machines $c_i = (c_{2,i}, \ldots, c_{J,i})$, where each $c_{j,i}$ is itself a vector of length K_c. The vector of efficiency loss rates v_i and the matrix of replacement

*Corresponding authors: Jeremy Fox at jeremyfox@gmail.com and Kyoo il Kim at kyookim@msu.edu.

cost shifters c_i are specific to each firm i and are not time-varying. The value of the scalar $w_{i,t}$ is one of the elements of the vector v_i, corresponding to the loss rate for the firm's current machine. Collect the observables (other than actions) as $x_{i,t} = (a_{i,t}, d_{i,t}, w_{i,t}, v_i, c_i)$. The time-varying, observable state variables are $s_{i,t} = (a_{i,t}, d_{i,t}, w_{i,t})$.

The flow profits are also affected by a vector of firm-specific and time-invariant heterogeneous parameter β_i and a vector of firm- and time-specific unobservables $\varepsilon_{i,t} = (\varepsilon_{1,i,t}, \ldots, \varepsilon_{J,i,t})$. We discuss the components of the vector β_i below. We seek to estimate the distribution $G(\beta)$ of β_i, or how the heterogeneous parameters vary across firms. Following the setup without time invariant heterogeneous parameters in Rust (1987), the time varying unobservables in $\varepsilon_{i,t}$ are independent across firms, machines and time and identically distributed according to the extreme value type I distribution, which gives logit probabilities for the replacement decisions. As in Rust, the so-called integrated value function does not need to include $\varepsilon_{i,t}$ as a state variable because $\varepsilon_{i,t}$ is independent over time.

The flow profit for action j is $\bar{u}_i(j, s_{i,t}) + \epsilon_{j,i,t}$, where

$$\bar{u}_i(j, s_{i,t}) = \begin{cases} \beta_{i,1} + \beta_{i,2} w_{i,t}^{-a_{i,t}} + \beta_{i,3} d_{i,t}, & j = 1 \\ -\beta_{i,4}' c_{j,i} & j \geq 2. \end{cases}$$

Here the heterogeneous parameters $\beta_{i,1}$, $\beta_{i,2}$ and $\beta_{i,3}$ are scalars, the heterogeneous parameter $\beta_{i,4}$ is a vector, and $\beta_i = (\beta_{i,1}, \beta_{i,2}, \beta_{i,3}, \beta_{i,4})$ is a vector of length $K = 3 + K_c$. In addition to incorporating shifters, the main content of the flow profit equation is that profit declines in machine age $a_{i,t}$.

Recall that the time-varying state variables for firm i are $s_{i,t} = (a_{i,t}, d_{i,t}, w_{i,t})$. The scalar age $a_{i,t}$ of the machine evolves as

$$a_{i,t+1}(j, a_{i,t}) = \begin{cases} \min\{a_{i,t} + 1, 5\}, & j = 1 \\ 0, & j \geq 2, \end{cases}$$

which means that the age resets to 0 after replacement and also stops increasing after an age of 5. The current efficiency loss rate $w_{i,t}$ corresponds to the efficiency loss of firm i's current machine and so transitions as

$$w_{i,t} = \begin{cases} w_{i,t}, & j = 1 \\ v_{j,i}, & j \geq 2. \end{cases}$$

The shifter $d_{i,t}$ has a finite support and evolves according to the Markov process $\Pr(d_{i,t+1} \mid d_{i,t})$, which is independent of the firm's replacement decision and other variables.

For this model, the time invariant terms that distinguish firm i from other firms are the vector of heterogeneous parameters β_i, the matrix c_i, and the vector v_i. Therefore, the dynamic program needs to be solved for each firm i. Following Rust (1987), the integrated value function and the conditional choice probabilities, which both integrate out $\varepsilon_{i,t}$, are

$$\begin{aligned} V_i(s_{i,t}) &= 0.577\ldots + \ln\left(\sum_{j=1}^{J} \exp\left(\bar{u}_i(j, s_{i,t}) + \delta EV_i(s_{i,t+1} \mid s_{i,t}, j)\right)\right) \text{ and} \\ \bar{g}_{j,i}(s_{i,t}) &= \frac{\exp\left(\bar{u}_i(j, s_{i,t}) + \delta EV_i(s_{i,t+1} \mid s_{i,t}, j)\right)}{\sum_{j'=1}^{J} \exp\left(\bar{u}_i(j', s_{i,t}) + \delta EV_i(s_{i,t+1} \mid s_{i,t}, j')\right)}, \end{aligned}$$

where δ is the discount factor and $EV_i(s_{i,t+1} \mid s_{i,t}, j)$ is the choice specific continuation payoffs function. The expectations are non-trivially taken with respect to only $d_{i,t}$ because the transitions of $(a_{i,t}, w_{i,t})$ are deterministic given actions. Each value function V_i is stored on the computer as a vector of length equal to the number of distinct values of $s_{i,t}$.

To write the dynamic programming choice probabilities in the notation of the general mixture model (1), let $g_j(x_{i,t}, \beta_i) = \bar{g}_{j,i}(s_{i,t})$ be the probability of replacement action j given the observables $x_{i,t}$ and the time invariant, heterogeneous parameters β_i. We solve the value function, a system of nonlinear equations, using value function iteration. Using the fixed grid estimation approach, there is one value function to solve for each combination of a grid value for β_i and a value for the observable, time invariant firm characteristics (c_i, v_i).

2 Estimators

Our consistency and rate results in this paper are for cross-sectional data, not panel data. Therefore, we generate cross sectional data to test the estimators. Denote the replacement decision of firm i as $y_i \in \{1, \ldots, J\}$. The data consists of (y_i, x_i) for $i = 1, \ldots N$. Each firm is observed only once and the states and actions are statistically independent across firms, conditional on observables. For the dimension of the vector of heterogeneous parameters on the cost of replacement shifters, β_4, we consider both $K_c = 1$ and 5. Given that $K = 3 + K_c$, we estimate models with either $K = 4$ or $K = 8$ total heterogeneous parameters. In other words, the unknown distribution $F(\beta)$ is either a four- or eight-dimensional function.

We estimate using the same simulated data set using the least squares and likelihood fixed grid estimators and the likelihood flexible grid estimator. We use identical fixed grids for the least squares and likelihood criteria. The flexible grid likelihood method estimates both the vector of grid points $\mathcal{B}_R$ and the vector of weights $\left(\theta^1, \ldots, \theta^R\right)$ on each grid point. We use the MATLAB constrained linear least squares routine, `lsqlin`, for the least squares criterion. We follow the implementations of the EM algorithm for static logit models in Train (2008, Sections 4 and 6) for the fixed and flexible grid likelihood estimators. The details are in Section 6 of this appendix. Our convergence rule for the fixed grid EM algorithm is that either the maximum absolute change in the weight is less than 10^{-9} or the number of iterations is greater than 2000. In our Monte Carlo specifications, convergence is usually achieved within 100 iterations.

The fixed grid estimators are guaranteed to converge to the global optimum of the objective function from any starting value, as our optimization problems are convex. The flexible grid estimator is guaranteed to converge to only a local maximum of the likelihood function.

We apply the fixed grid estimators to sample sizes of $N = 100$, 500, and 1000 observations on independent firms. For each Monte Carlo choice of K, N, R and the number of mixture components M in the true data generating process (described below), we perform $L = 100$ replications of estimation using different fake data sets. For each replication, we use the same dataset for all three estimators. Due to the extremely long computational time of the flexible grid estimator for our dynamic programming model, we only run the flexible grid estimator for the sample size $N = 50$ and the number of grid points $R = 10$. For similar computational reasons, the flexible grid estimator runs for only 10 iterations of the EM algorithm from one starting value, so its measured statistical error is an upper bound to the

performance of the flexible grid estimator run to convergence from many starting values.

Our dynamic programming model highlights the computational savings of fixed grid estimators. For a grid vector β^r, the value function also depends on the firm specific, time-persistent observables (c_i, v_i). We solve in total $N \cdot R$ different dynamic programming problems when using a fixed grid estimator with R grid points. For each Monte Carlo replication, the dynamic programming computation is done once and applied to both the least squares and likelihood criteria. In contrast, the flexible grid estimator is much more computationally intensive because it needs to solve many dynamic programming problems every time the optimizer calls choice probabilities. Choice probabilities are usually called thousands of times during optimization in a single EM iteration, resulting in far more dynamic programs than the $N \cdot R$ from the fixed grid approaches.

3 Monte Carlo Data Generating Process

We now specify the data generating process for the Monte Carlo study. We set $J = 6$ so that there are five machines. The discount factor δ is 0.95. Given the focus on testing the estimators on cross sectional data, we draw the heterogeneous parameters β_i independently of x_i. Therefore, we do not explore statistical endogeneity, the so-called initial conditions problem in panel data.

The elements of x_i have finite, continuous uniform or normal distributions.[1] The elements of (c_i, v_i) have independent standard uniform distributions across machines. The profit shifter $d_{i,t}$ has 50 possible discrete values, drawn from a normal distribution and fixed across Monte Carlo specifications and replications. We then generate the transition matrix with elements $\Pr(d_{i,t+1} \mid d_{i,t})$ by drawing the elements from independent uniform distribution and normalizing $\sum_{\text{supp}(d_{i,t+1})} \Pr(d_{i,t+1} \mid d_{i,t})$ to be 1. The terms $\Pr(d_{i,t+1} \mid d_{i,t})$ are also fixed across Monte Carlo specifications and replications. The current $d_{i,t}$ for firm i is uniformly drawn from its support. The current age $a_{i,t}$ is uniformly drawn from the integers 0 to 5. Firm i's $w_{i,t}$ is sampled uniformly from firm i's v_i.

The true distributions $F_0(\beta)$ are mixtures of multivariate normals, each with nonzero correlations. We vary the number M of mixture components in the true distribution. There are either $M = 1$, 2 or 5 mixture components. Each of the normal mixture components has an equal weight of $1/M$.

Define five sets of means for the maximum K of eight heterogeneous coefficients

$$\begin{aligned}
\mu_1 &= [0.375, -2, 2, 2, 0.875, 0.75, 1.25, 1.875] \\
\mu_2 &= [0.25, 1, -1, 1.625, 2, 0.125, 1.25, 2] \\
\mu_3 &= [0.375, 2, -2, 0.375, 1.75, 0.625, 0.25, 0.125] \\
\mu_4 &= [0.5, -2, -2, 1.75, 0.75, 1.625, 0.875, 1.875] \\
\mu_5 &= [0.25, 0, -1, 0.625, 0.375, 0.125, 0.125, 1.25] .
\end{aligned}$$

We use these terms for the data generating process for all specifications. Denote the first K elements of the vector μ_m as $\mu_m(K)$. Say a particular Monte Carlo specification uses $K = 4$ and $M = 2$. Then the true distribution of β is a mixture of two normals, each with weight $1/2$, and the mth component has mean $\mu_m(4)$.

[1] An identification result for the distribution of heterogeneous parameters in the random coefficients logit relies on explanatory variables being continuous, as we discuss for Example 1 in Section 6 of the main text. Some of the elements of x_i have continuous distributions and others have discrete distributions, to simplify dynamic programming.

Defining the covariance matrix of each mixture component is more involved. A base matrix is a symmetric matrix that is 4.3562 on the diagonal and 0.5252 elsewhere:

$$\Sigma^{\star} = \left[\underbrace{\begin{array}{ccc} 4.3562 & 0.5252 & \ldots \\ 0.5252 & 4.3562 & \ldots \\ \vdots & \vdots & \ddots \end{array}}_{8 \text{ elements}} \right].$$

Also denote the Kth leading principal submatrix of $\Sigma^{\star}$ as $\Sigma(K)$. In the data generating process, a mixture of $1 \leq M \leq 5$ normals of dimension K is defined as $F(\beta) = \sum_{m=1}^{M} \frac{1}{M} \mathcal{N}(\mu_M(K), \Sigma_K)$, where the variance matrix of each component is

$$\begin{aligned} \Sigma_K &= \Sigma(K) + \frac{1}{5} \sum_{m=1}^{5} (\mu_m(K) - \bar{\mu}_5(K))' (\mu_m(K) - \bar{\mu}_5(K)) \\ &\quad - \frac{1}{M} \sum_{m=1}^{M} (\mu_m(K) - \bar{\mu}_M(K))' (\mu_m(K) - \bar{\mu}_M(K)), \end{aligned}$$

where $\bar{\mu}_M(K) = \frac{1}{M} \sum \mu_m(K)$. In our specifications, this somewhat recursive choice of Σ_K makes the true distributions $F(\beta)$ have the same variance matrix across choices of M, so there is not a tight link between the number of modes M and the baseline grid's coverage of the true distribution across Monte Carlo specifications.[2]

We use a quasi-random number sequence to generate a baseline grid for the two fixed grid estimators. We draw $R = 100$ or 200 values from a Halton sequence. The draws are low-discrepancy points in the box $[0,1]^K$, and we map them into a larger region by multiplying each element by 15. We shift the coefficient $\beta_{i,1}$ on the loss rate due to age by -1.5, the intercept $\beta_{i,2}$ by -7.5, the coefficient $\beta_{i,3}$ by -7.5, and the elements of the vector $\beta_{i,4}$ by 1.5.[3] We generate the grid for evaluation of measures like RMISE, defined below, in the same way, except we draw 5000 points.

4 Results on Speed

The results of various Monte Carlo specifications are in Table 1 for the fixed grid least squares estimator, Table 2 for the fixed grid likelihood estimator, and Table 3 for a comparison between all three estimators, including the flexible grid likelihood estimator. All times are for code parallelized over 12 cores.

Focus first on the speed of the fixed grid estimators in Table 1 and 2. The tables break out the time spent on computing dynamic programs before optimization and the time spent on optimization / the

[2] Note that the variance matrix of $\sum_{m=1}^{M} \frac{1}{M} \mathcal{N}(\mu_M(K), \Sigma_K)$ is $\Sigma_K + \frac{1}{M} \sum_{m=1}^{M} (\mu_m(K) - \bar{\mu}_M(K))' (\mu_m(K) - \bar{\mu}_M(K))$. With our choice of Σ_K, the variance matrix becomes $\Sigma(K) + \frac{1}{5} \sum_{m=1}^{5} (\mu_m(K) - \bar{\mu}_5(K))' (\mu_m(K) - \bar{\mu}_5(K))$ for any M.

[3] Some $\beta_{i,2}$ are negative, meaning that the machine's output actually improves with age.

EM algorithm. Table 1 shows that the overwhelming majority of time for the least squares estimator is spent on solving the $N \cdot R$ dynamic programs before optimization begins. In our largest sample size $N = 1000$ and largest estimation grid $R = 200$, the dynamic programming takes on average (across replications) 1.7 hours. The optimization command for fixed grid least squares, lsqlin in MATLAB, consumes an insignificant amount of time, under 7 seconds in all the specifications considered. Table 2, for the fixed grid likelihood criterion, does not report dynamic programming time because the dynamic programs reported in Table 1 are reused. The EM algorithm is even faster than the MATLAB command lsqlin; optimization takes under 3 seconds for all specifications in Table 2. The bottom line is that dynamic programming is far more costly than optimizing the statistical objective functions.

Table 3 compares all three estimators, including the flexible grid estimator, on a problem with $N = 50$, $R = 10$ and $K = 4$. The two fixed grid estimators require solving $N \cdot R = 500$ dynamic programs before optimization begins. For only 10 EM algorithm iterations from a single starting value, the flexible grid estimator requires solving on average (across the 100 Monte Carlo replications) over 200,000 dynamic programs for all specifications, as dynamic programs are nested inside each call to calculate choice probabilities. The average time of the 10 EM algorithm iterations for the flexible grid estimator is around 10,000 seconds, or around 2.7 hours. This compares to 12 seconds to convergence (not just 10 iterations) for the fixed grid estimators, including the dynamic programming time. Altogether, the comparison in Table 3 shows the tremendous speed benefits of fixed grid over flexible grid methods.

For the flexible grid estimator, Table 3 reports only the outcome of 10 iterations from a single starting value in order to reduce the run time of the Monte Carlo. In empirical work, more starting values should be used and the EM algorithm should be run until some notion of convergence is achieved.

5 Results on Statistical Accuracy

We measure estimation accuracy using root mean integrated squared error (RMISE) and integrated absolute error (IAE). Denote the estimated distribution of β from one Monte Carlo replication l as $\hat{F}_l$ and the true distribution as F_0. The estimated CDF is evaluated on a grid consisting of $Q = 5{,}000$ points, so that the true CDF values of these points are smoothly spread from 0 to 1. The RMISE for a Monte Carlo specification with L replications is

$$\sqrt{\frac{1}{L}\sum_{l=1}^{L}\frac{1}{Q}\sum_{q=1}^{Q}\left(\hat{F}_l\left(\beta_q\right)-F_0\left(\beta_q\right)\right)^2}.$$

The IAE for a given replication l is defined as

$$\frac{1}{Q}\sum_{q=1}^{Q}\left|\hat{F}_l\left(\beta_q\right)-F_0\left(\beta_q\right)\right|.$$

We report the mean, minimum and maximum IAE across replications for each estimator and Monte Carlo specification.

Table 1 and Table 2 report the RMISE and IAE of the least squares and likelihood fixed grid

estimators, respectively. The two estimators take as input the same grid and corresponding choice probabilities for each grid point and firm; they differ only in the statistical criterion. The tables show that both the RMISE and IAE of the least squares estimator tend to be lower than the RMISE and IAE of the fixed grid likelihood estimator. At least in our setup, the least squares criterion seems preferable because of its better statistical performance and the fact that its slower optimization speed is still trivial compared to the time spent on dynamic programming.

Tables 1 and 2 compare true data generating processes with $M = 1$, 2 and 5 multivariate normal components. Both RMISE and IAE increase with the number of modes in the true distribution generating the data, although the amounts of the increase in RMISE and IAE are not large. It appears to be harder to nonparametrically estimate distributions with more modes.

Tables 1 and 2 also vary K, the dimension of the heterogeneous parameters β. K is either 4 or 8. Not surprisingly, RMISE and mean IAE tend to increase with K. The increase is largest for the case of $M = 1$, where the true distribution is a single component multivariate normal. For higher M, changing K from 4 to 8 increases in RMISE and IAE by only a small amount. Compared to $M = 1$, the difference is poorer performance with $K = 4$, as the statistical performance of the $K = 8$ case is less sensitive to M. While there is some evidence of estimation error substantially increasing with K for $M = 1$, for other M the Monte Carlo study does not demonstrate the curse of dimensionality in K found in the upper bound of the convergence rate of the CDF supremum estimation error in Theorem 6.

Tables 1 and 2 also shed light on how grid size R affects statistical accuracy. With $K = 4$, increasing the grid size from 100 to 200 reduces RMISE and IAE. The reduction is less perceptible with higher dimensional heterogeneous parameters, $K = 8$.

Tables 1 and 2 also report the mean (across replications) number of the R grid points whose weights θ^r are estimated to be nonzero (greater than 0.001). For both fixed grid estimators, it is common to have around 10 nonzero weights, out of a grid of either $R = 100$ or $R = 200$ points. Both estimators favor sparse representations of the true distribution without including any sort of LASSO-style penalty. The number of nonzero weights varies across replications but there is no strong trend favoring more nonzero weights for either the least squares or fixed grid likelihood estimators. Both Tables 1 and 2 compare specifications by varying K and M. Not surprisingly, higher M and K tend to lead to more nonzero weights, but the effect is small relative to the grid sizes of $R = 100$ and $R = 200$.

Tables 1 and 2 report how sample size N affects accuracy. Across sample sizes of $N = 100$, 500 and 1000, we see only tiny reductions in RMISE and mean IAE with N. This is consistent with an estimator that converges slowly, as Theorem 6 indicates.

Table 3 compares all three estimators on a small sample $N = 50$ with a small grid $R = 10$. Due to the flexible estimator's prohibitively long computational time, we iterate the EM algorithm for the flexible grid estimator ten times from one starting point. We use $K = 4$ and vary M to be 1, 2 or 5, as in Tables 1 and 2. Although the flexible grid estimator has not converged and we only use one starting value, its RMISE and IAE are substantially lower than the fixed grid results. Table 3 shows that there is a large statistical advantage of being able to estimate a flexible grid instead of fixing a grid of heterogeneous parameters. Bringing in the run times in Table 3, there is a clear tradeoff between the prohibitive speed of the flexible grid estimator and the poorer statistical performances of the fixed grid estimators.

6 EM Algorithms

We use uniform random variables to generate starting points for probability weights, means and variances in the EM estimators. The flexible-grid EM estimator has the following steps (suppressing the t subscript).

1. Start with initial values β_0 and θ_0. Calculate

$$K\left(x_i, \beta_0^r\right) = \prod_{j=1}^{J} \left[g_j\left(x_i, \beta_0^r\right)\right]^{1(y_i=j)},$$

   where y_i denotes firm i's decision on replacement.
   Then calculate $h_{i,r}^0 = \dfrac{\theta_0^r K\left(x_i, \beta_0^r\right)}{\sum_{r=1}^{R} \theta_0^r K\left(x_i, \beta_0^r\right)}$.

2. Update θ_0 to θ_1: $\theta_1^r = \dfrac{\sum_i h_{i,r}^0}{\sum_i \sum_{r=1}^{R} h_{i,r}^0}$.

3. Use θ_1 to estimate β_1 by maximizing the log likelihood

$$L\left(\theta_1, \beta \middle| x\right) = \sum_i \sum_{r=1}^{R} \ln\left(\theta_1^r K\left(x_i, \beta^r\right)\right).$$

   Notice that β^r only appears in the term $\sum_i \ln\left(\theta_1^r K\left(x_i, \beta^r\right)\right)$. We only need to solve for β_1^r by maximizing $\sum_i \ln\left(\theta_1^r K\left(x_i, \beta^r\right)\right)$ separately for each r. Repeat steps 2 and 3 until convergence, letting $\beta_0 = \beta_1$ and $\theta_0 = \theta_1$. We use MATLAB's optimizer fminunc for this step.

The fixed-grid EM estimator has the following steps.

1. Start with a fixed grid $\mathcal{B}^R$ and initial weights θ_0. Calculate $K\left(x_i, \beta^r\right)$ and $h_{i,r}^0$.

2. Update θ_0 to θ_1: $\theta_1^r = \dfrac{\sum_i h_{i,r}^0}{\sum_i \sum_{r=1}^{R} h_{i,r}^0}$. Repeat step 2 until convergence, letting $\theta_0 = \theta_1$.

References

[1] Train, K. (2008), "EM Algorithms for Nonparametric Estimation of Mixing Distributions", *Journal of Choice Modeling*, 1, 1, 40–69.

Table 1: Fixed Grid Least Squares

N	R	M	K	RMISE	max IAE	mean IAE	min IAE	Time Using lsqlin (s)	Dynamic Prog Time (s)	Mean # of Positive Weights
100	100	1	4	0.23	0.38	0.18	0.1	0.46	382.03	7.59
100	100	1	8	0.3	0.37	0.28	0.23	0.46	402.89	6.07
100	200	1	4	0.21	0.35	0.16	0.07	1.55	610.64	7.91
100	200	1	8	0.3	0.34	0.27	0.21	1.57	654.59	6.52
500	100	1	4	0.21	0.27	0.18	0.13	0.79	1527.67	8.7
500	100	1	8	0.3	0.3	0.27	0.24	0.82	1603.61	7.13
500	200	1	4	0.19	0.25	0.15	0.09	4	3123.93	9.77
500	200	1	8	0.3	0.3	0.28	0.24	3.81	3276.52	7.93
1000	100	1	4	0.21	0.24	0.18	0.13	1.46	2425.95	8.94
1000	100	1	8	0.29	0.31	0.27	0.24	1.32	3149.32	7.35
1000	200	1	4	0.18	0.21	0.15	0.1	6.87	6078.88	10.72
1000	200	1	8	0.3	0.31	0.28	0.25	6.98	6379.8	8.47
100	100	2	4	0.29	0.46	0.24	0.15	0.37	314.88	7.84
100	100	2	8	0.28	0.35	0.26	0.2	0.36	326.87	6.85
100	200	2	4	0.28	0.41	0.23	0.1	1.53	613.65	8.45
100	200	2	8	0.29	0.31	0.26	0.2	1.54	647.46	7.44
500	100	2	4	0.28	0.33	0.24	0.16	0.84	1517.89	8.75
500	100	2	8	0.29	0.29	0.26	0.23	0.82	1606.72	9.04
500	200	2	4	0.26	0.31	0.23	0.14	3.86	3041.95	9.89
500	200	2	8	0.29	0.29	0.26	0.23	3.6	3199.49	9.92
1000	100	2	4	0.28	0.3	0.25	0.21	1.39	3014.08	9.14
1000	100	2	8	0.29	0.29	0.26	0.24	1.51	3243.94	9.61
1000	200	2	4	0.26	0.33	0.23	0.16	6.87	6016.28	10.56
1000	200	2	8	0.29	0.28	0.26	0.24	6.8	6241.11	11.36
100	100	5	4	0.31	0.39	0.28	0.17	0.5	398.14	7.71
100	100	5	8	0.31	0.35	0.28	0.19	0.49	416.35	7.02
100	200	5	4	0.3	0.4	0.26	0.14	1.58	617.19	8.15
100	200	5	8	0.31	0.34	0.28	0.23	1.56	658.42	7.19
500	100	5	4	0.32	0.36	0.29	0.23	0.74	1514.24	8.66
500	100	5	8	0.3	0.31	0.28	0.25	0.84	1625.02	9.53
500	200	5	4	0.29	0.35	0.26	0.18	3.78	3050.03	9.92
500	200	5	8	0.31	0.31	0.28	0.23	5.82	4010.44	10.77
1000	100	5	4	0.32	0.33	0.3	0.23	1.49	3015.38	8.33
1000	100	5	8	0.3	0.3	0.28	0.25	1.45	3174.38	11.17
1000	200	5	4	0.3	0.36	0.28	0.19	6.96	6049.14	10.49
1000	200	5	8	0.3	0.3	0.28	0.25	6.56	5122.4	12.9

Table 2: Fixed Grid Likelihood*

N	R	M	K	RMISE	max IAE	mean IAE	min IAE	Time in EM Iteration	Mean # of Positive Weights
100	100	1	4	0.26	0.4	0.22	0.1	0.49	6.81
100	100	1	8	0.39	0.46	0.35	0.26	0.45	5.92
100	200	1	4	0.24	0.4	0.18	0.08	0.81	7.26
100	200	1	8	0.38	0.43	0.35	0.29	0.66	6.55
500	100	1	4	0.25	0.31	0.23	0.14	1.06	8.49
500	100	1	8	0.38	0.39	0.35	0.32	1.23	9.04
500	200	1	4	0.19	0.3	0.15	0.1	1.73	9.14
500	200	1	8	0.38	0.39	0.35	0.31	1.36	10.86
1000	100	1	4	0.25	0.29	0.23	0.17	1.65	8.74
1000	100	1	8	0.38	0.42	0.35	0.32	1.29	10.2
1000	200	1	4	0.18	0.22	0.15	0.1	2.33	9.9
1000	200	1	8	0.38	0.38	0.35	0.32	2.3	12.89
100	100	2	4	0.31	0.47	0.27	0.16	0.43	7.07
100	100	2	8	0.35	0.42	0.32	0.22	0.43	6.5
100	200	2	4	0.3	0.4	0.26	0.12	0.78	7.95
100	200	2	8	0.35	0.41	0.32	0.22	0.78	6.92
500	100	2	4	0.31	0.34	0.28	0.23	1.3	8.31
500	100	2	8	0.35	0.37	0.33	0.27	1.17	9.74
500	200	2	4	0.28	0.33	0.25	0.15	1.54	10
500	200	2	8	0.35	0.37	0.32	0.27	1.47	10.95
1000	100	2	4	0.31	0.32	0.29	0.24	1.43	8.58
1000	100	2	8	0.35	0.37	0.32	0.29	1.63	10.8
1000	200	2	4	0.28	0.31	0.25	0.17	2.26	10.73
1000	200	2	8	0.35	0.35	0.32	0.29	2.25	13.49
100	100	5	4	0.35	0.4	0.32	0.23	0.55	6.84
100	100	5	8	0.38	0.43	0.34	0.25	0.51	6.46
100	200	5	4	0.34	0.42	0.3	0.17	0.79	7.71
100	200	5	8	0.38	0.42	0.35	0.27	0.84	6.91
500	100	5	4	0.35	0.36	0.32	0.28	1.11	8.17
500	100	5	8	0.38	0.39	0.35	0.31	1.23	9.87
500	200	5	4	0.32	0.36	0.3	0.23	1.42	9.93
500	200	5	8	0.38	0.38	0.35	0.31	1.74	11.31
1000	100	5	4	0.35	0.35	0.32	0.28	1.67	8.37
1000	100	5	8	0.38	0.37	0.35	0.32	1.58	11.1
1000	200	5	4	0.33	0.35	0.3	0.23	2.55	10.73
1000	200	5	8	0.38	0.38	0.35	0.31	2.23	13.87

*: the time to compute the dynamic programming problems is the same as in Table 1.
The two estimators use the same grid.

Table 3: N =50, R =10, K =4, All Three Estimators

Fixed Grid Least Squares

M	RMISE	max IAE	mean IAE	min IAE	Total Time (s)*	Mean # of Positive Weights	# Dynamic Prog Called
1	0.32	0.51	0.28	0.15	11.95	4.66	500
2	0.35	0.48	0.31	0.14	11.87	4.93	500
5	0.39	0.53	0.34	0.19	11.83	4.80	500

Fixed Grid Likelihood

M	RMISE	max IAE	mean IAE	min IAE	Total Time (s)*	Mean # of Positive Weights	# Dynamic Prog Called
1	0.38	0.47	0.35	0.23	12.02	4.59	500
2	0.37	0.49	0.33	0.22	11.94	4.95	500
5	0.41	0.54	0.36	0.26	11.91	4.76	500

Flexible Grid Likelihood**

M	RMISE	max IAE	mean IAE	min IAE	Total Time (s)	Mean # of Positive Weights	# Dynamic Prog Called
1	0.15	0.17	0.09	0.05	10501.86	9.94	225887
2	0.14	0.15	0.09	0.04	9488.15	9.93	206134
5	0.14	0.18	0.09	0.06	9829.64	9.95	208435

*: Combined time of dynamic programming and optimization

**: Iterate EM ten times from one starting point