The Approximate Solution of Finite-Horizon Discrete Choice Dynamic Programming Models*

Revisiting Keane and Wolpin (1994)

Philipp Eisenhauer
University of Bonn

December 23, 2016

Abstract

The estimation of finite-horizon discrete choice dynamic programming (DCDP) models is computationally expensive. This limits their realism and impedes verification and validation efforts. Keane and Wolpin (1994) propose an interpolation method that ameliorates the computational burden but introduces approximation error. I describe their approach in detail, successfully recompute their original quality diagnostics, and provide some additional insights that underscore the trade-off between computation time and the accuracy of estimation results.

JEL Codes: C13, C22, C53
Keywords: Discrete Choice Dynamic Programming Models, Interpolation Methods

*I gratefully acknowledge support by the AXA Research Fund. Corresponding Author: Philipp Eisenhauer, eisenhauer@policy-lab.org, Institute for Applied Microeconomics, University of Bonn, Lennéstr. 43, 53113 Bonn, Germany. Tel.: +49 228 73-9240, Fax: +49 228 73-9239.

1 Introduction

The estimation of finite-horizon discrete choice dynamic programming (DCDP) models has generated valuable insights in diverse subfields of economics such as industrial organization, labor economics, and marketing.[1] DCDP models are structural economic models as they make explicit the agents' objective as well as the informational and institutional constraints under which they operate. Thus they allow to assess the relative importance of competing economic mechanisms that guide agents' decision making and permit the ex ante evaluation of alternative policy proposals (Wolpin, 2013). In these models, economic agents make repeated choices over multiple periods. They are forward-looking and thus take the future consequences of their immediate actions into account. However, agents operate in an uncertain economic environment as at least parts of their future payoffs are unknown at the time of their decision.

Estimating a finite-horizon DCDP model poses computational challenges as it requires the repeated solution of a dynamic programming (DP) problem under uncertainty by backward induction. The well known curse of dimensionality (Bellman, 1957) is a major impediment to the application of more realistic models and their verification and validation. To alleviate the computational burden, Keane and Wolpin (1994) propose to work with an approximate solution to the dynamic programming problem instead. They suggest to solve the DP problem during the backward induction procedure at only a subset of states in each period and simply use interpolated values for all other states. This introduces approximation error and requires a careful assessment of the reliability of results. Keane and Wolpin (1994) provide some very encouraging Monte Carlo evidence for a prototypical model of occupational choice.

I will describe their approach in detail, successfully recompute their original quality diagnostics, and provide some additional insights that underscore the trade-off between computation time and the accuracy of estimation results. The rest of the paper is structured as follows. In Section 2, I present the economics of the basic model and outline the approach to its solution and estimation. I then turn to the results of my recomputation in Section 3 and provide some additional diagnostics regarding the reliability of the proposed interpolation scheme. Section 4 concludes.

[1]See Keane et al. (2011) for a recent survey of the literature.

This manuscript is supplemented by an open-source Python package for the simulation and estimation of a prototypical discrete choice dynamic programming model. The package is available at `http://respy.readthedocs.io`.

2 Basic Setup

I now discuss the economics motivating the model analyzed by Keane and Wolpin (1994) and present their assumptions about functional forms and the distributions of unobservables. Then I turn to the model solution and briefly outline the estimation approach.

2.1 Economic Model

Keane and Wolpin (1994) develop a model in which an agent decides among K possible alternatives in each of T (finite) discrete periods of time. Alternatives are defined to be mutually exclusive and $d_k(t) = 1$ indicates that alternative k is chosen at time t and $d_k(t) = 0$ indicates otherwise. Associated with each choice is an immediate reward $R_k(S(t))$ that is known to the agent at time t but partly unknown from the perspective of periods prior to t. The state space $S(t)$ encompasses all the information available to the agent at time t that affects immediate and future rewards.

At the beginning of each period t the agent fully learns about all immediate rewards, chooses one of the alternatives and receives the corresponding payoffs. The state space is then updated according to the agent's state experience and the process is repeated in $t + 1$. Agents are forward looking. Thus, they do not simply choose the alternative with the highest immediate rewards each period. Instead, their objective at any time τ is to maximize the expected rewards over the remaining time horizon:

$$\max_{\{d_k(t)\}_{k\in K}} E\left[\sum_{\tau=t}^{T} \delta^{\tau-t} \sum_{k\in K} R_k(\tau)d_k(\tau)\middle| S(t)\right]. \tag{1}$$

The discount factor $0 > \delta > 1$ captures the agent's preference for immediate over future rewards. Agents maximize equation (1) by choosing the optimal sequence of alternatives $\{d_k(t)\}_{k\in K}$ for $t = \tau, .., T$.

Within this more general model framework, Keane and Wolpin (1994) then impose additional functional form and distributional assumptions that define their prototypical model of occupational choice.

Agents live for a total of 40 periods and are risk neutral. Each period, agents choose to work in either of two occupations (k = 1,2), to attend school (k = 3), or to remain at home (k = 4). The immediate reward functions are given by:

$$
\begin{aligned}
R_1(t) &= w_{1t} = \exp\{\alpha_{10} + \alpha_{11}s_t + \alpha_{12}x_{1t} - \alpha_{13}x_{1t}^2 + \alpha_{14}x_{2t} - \alpha_{15}x_{2t}^2 + \epsilon_{1t}\} \\
R_2(t) &= w_{2t} = \exp\{\alpha_{20} + \alpha_{21}s_t + \alpha_{22}x_{2t} - \alpha_{23}x_{2t}^2 + \alpha_{24}x_{1t} - \alpha_{25}x_{1t}^2 + \epsilon_{2t}\} \\
R_3(t) &= \beta_0 - \beta_1 I(s_t \geq 12) - \beta_2(1 - d_3(t-1)) + \epsilon_{3t} \\
R_4(t) &= \gamma_0 + \epsilon_{4t},
\end{aligned}
$$

where s_t is the number of periods of schooling obtained by the beginning of period t, x_{1t} is the number of periods that the agent worked in occupation one by the beginning of period t, x_{2t} is the analogously defined level of experience in occupation two, α_1 and α_2 are parameter vectors associated with the wage functions, β_0 is the consumption value of schooling, β_1 is the post-secondary tuition cost of schooling, with I as an indicator function equal to one if the agent completed high school and zero otherwise, β_2 is an adjustment cost associated with returning to school, γ_0 is the (mean) value of the non-market alternative. The ϵ_{kt}'s are alternative-specific shocks, to occupational productivity, to the consumption value of schooling, and to the value of non-market time. The productivity and taste shocks follow a four-dimensional multivariate normal distribution with mean zero and covariance matrix $\Sigma = [\sigma_{ij}]$. They collect the parameterization of the reward functions in $\theta = \{\alpha_1, \alpha_2, \beta, \gamma, \Sigma\}$.

Given the structure of the reward functions and the agent's objective, the state space at time t is $S(t) = \{s_t, x_{1t}, x_{2t}, d_3(t-1), \epsilon_{1t}, \epsilon_{2t}, \epsilon_{3t}, \epsilon_{4t}\}$. It is convenient to denote its observable elements as $\bar{S}(t)$. The elements of $S(t)$ evolve according to:

$$
\begin{aligned}
x_{1,t+1} &= x_{1t} + d_1(t) \\
x_{2,t+1} &= x_{2t} + d_2(t) \\
s_{t+1} &= s_t \ + d_3(t) \\
f(\epsilon_{t+1} \mid S(t), d_k(t)) &= f(\epsilon_{t+1} \mid \bar{S}(t), d_k(t)),
\end{aligned}
$$

where the last equation reflects the fact that the ϵ_{kt}'s are serially independent. They set the initial conditions as $x_{1t} = x_{2t} = 0$ and $s_0 = 10$. Agents cannot attain more than ten additional years of schooling. Note that all agents start out identically, different choices over the life cycle are simply the cumulative effects of different shocks.

2.2 Solution

From a mathematical perspective, the model is a finite-horizon dynamic programming (DP) problem under uncertainty that can be solved by backward induction. For the discussion, it is useful to define the value function $V(S(t), t)$ as a shorthand for equation (1). $V(S(t), t)$ depends on the state space at t and on t itself due to the finiteness of the time horizon and can be written as

$$V(S(t), t) = \max_{k \in K}\{V_k(S(t), t)\},$$

with $V_k(S(t), t)$ as the alternative-specific value function. $V_k(S(t), t)$ obeys the Bellman equation (Bellman, 1957) and is thus amenable to a backward induction.

$$V_k(S(t), t) = \begin{cases} R_k(S(t)) + \delta E\left[V(S(t+1), t+1) \mid S(t), d_k(t) = 1\right] & \text{if } t < T \\ R_k(S(t)) & \text{if } t = T. \end{cases}$$

Assuming continued optimal behavior, the expected future value of state $S(t+1)$ for all K alternatives given today's state $S(t)$ and choice $d_k(t) = 1$, $E\max(S(t+1))$ for short, can be calculated:

$$E\max(S(t+1)) = E\left[V(S(t+1), t+1) \mid S(t), d_k(t) = 1\right].$$

This requires the evaluation of a K - dimensional integral as future rewards are partly uncertain due to the unknown realizations of the shocks:

$$E\max(S(t)) = \\ \int_{\epsilon_1(t)} \cdots \int_{\epsilon_K(t)} \max\{R_1(t), \ldots, R_K(t)\} f_{\boldsymbol{\epsilon}}(\epsilon_1(t), \ldots, \epsilon_K(t)) d\epsilon_1(t) \ldots d\epsilon_K(t),$$

where $f_{\boldsymbol{\epsilon}}$ is the joint density of the uncertain component of the rewards in t not known at $t-1$. With all ingredients at hand, the solution of the model by backward induction is straightforward.

2.3 Estimation

Estimation of the parameters of the reward functions θ is based on a sample of agents whose behavior and state experiences are described by the model. Although all shocks to the rewards are eventually known to the agent, they remain unobserved by the econometrician. So each parameterization induces a different probability distribution over the sequence of observed agent choices and their state experience. Maximum likelihood estimation appraises each candidate parameterization of the model using the likelihood function of the observed sample (Fisher, 1922). Given the serial independence of the shocks, one can compute the likelihood contribution by agent and period. The sample likelihood is then simply the product of the likelihood contributions over all agents and time periods. The agent's choice probabilities are simulated and so one ends up with a simulated maximum likelihood estimator (Manski and Lerman, 1977) minimizing the simulated negative log-likelihood of the observed sample. Additional details about the estimation routine are available in Appendix A.

3 Approximate Solution

Even in this simplified model, the evaluation of $E\max$ at all states creates a considerable computational burden. As alternative parameterizations are appraised during an estimation, it requires the repeated evaluation of a four-dimensional integral by Monte Carlo integration at a total of 163,410 states. Building on the idea of generalized polynomial approximation (Bellman et al., 1963), Keane and Wolpin (1994) propose to calculate $E\max$ only at a subset of states each period and interpolate its value for the rest. Their key choices are the interpolation function and the number of interpolation points. I will now describe their approach in detail, successfully recompute their original quality diagnostics, and provide some additional insights that underscore the trade-off between computation time and the accuracy of estimation results. I follow Keane and Wolpin (1994) in all the details of the analysis that follows unless otherwise noted. Additional results and implementation details are available in Appendix B.

3.1 Operationalization

After experimentation, Keane and Wolpin (1994) settle on equation (2) as their preferred interpolation function.

$$E\max - \max E = \pi_0 + \sum_{j=1}^{4} \pi_{1j}(\max E - \bar{V}_j) + \sum_{j=1}^{4} \pi_{2j}\left(\max E - \bar{V}_j\right)^{\frac{1}{2}} \tag{2}$$

$\bar{V}_j$ is shorthand for the expected value of the alternative-specific value function and $\max E = \max_k\{\bar{V}_j\}$ is its maximum among the choices available to the agent. The π's are time-varying as they are estimated by ordinary least squares period by period. The subset of interpolation points used to fit the interpolating function, i.e. where $E\max$ is calculated explicitly, is chosen at random for each period. The number of interpolation points remains constant across all periods.

My implementation of the backward induction procedure remains straightforward. Period by period, I determine whether the total number of states is larger than the number of interpolation points. If this is not the case, $E\max$ is evaluated at each state. Otherwise, I draw a random sample of states for the interpolation points, evaluate $E\max$ and fit equation (2). Based on the results, I construct the predicted values for all remaining states in that period. Applying this interpolation scheme with 200 interpolation points reduces the number of states at which $E\max$ is explicitly evaluated from 163,410 to 6,930 which cuts the computation time for each evaluation of the criterion function to about a twentieth.

3.2 Quality Diagnostics

Within the general setup of equation (2), Keane and Wolpin (1994) carefully analyze the impact of alternative tuning parameters such as the number of interpolation points and the number of random draws for the evaluation of $E\max$ integral on the reliability of results. They generate three Monte Carlo samples with different parameterizations of the reward functions. I focus on their first parameterization in the text but all other results are available in the Appendix.

To assess the quality of the proposed interpolation scheme, I use an *exact solution* of the model as a benchmark. This solution is computed using 100,000 random draws for the evaluation of $E\max$ at all states at the true parameter values. The *exact sample* refers to a set of 1,000 simulated agents based on the *exact solution.* As on overall measure of the approximation error, I use the root-mean-square error (RMSE) and compare the choice probabilities in the *exact sample* to the results from a newly simulated set of 1,000 agents based on the alternative parameterization of the model.

Simulation based on Approximate Solution. Table 1 shows the proportion of correct choices for alternative interpolation schemes. I vary the number of interpolation points and the number of random draws for the evaluation of $E\max$. I follow

Table 1: Correct Choices

Points	All	All	All	2000	500
$E\max$ Draws	2000	1000	250	2000	2000
Periods					
1 - 10	0.000	0.000	0.000	0.000	0.000
11 - 35	0.000	0.000	0.000	0.000	0.000
36 - 38	0.010	0.000	0.020	0.029	0.036
39	0.036	0.043	0.181	0.186	0.214
40	0.963	0.957	0.799	0.785	0.750
Average	39.962	39.957	39.777	39.752	39.710

each agent in the *exact sample* over time and evaluate for each period whether a choice based on an approximate solution is still correct, i.e. aligns with the choice based on the *exact solution.* If I evaluate $E\max$ at all states with 2,000 random draws, then the two choices align for 96% of the agents in the sample in all 40 periods. This share drops to 75% as I only use 200 states each period and interpolate the rest. While always remaining above 39 periods, the average number of correct choices for each agent decreases as I coarsen the approximation.

Estimation based on Approximate Solution. To assess the impact of the approximation as it ripples through an estimation, I perform a Monte Carlo exercise by sampling a subset of 100 agents forty times from the *exact sample.* Starting each bootstrap iteration from the true parameter values, I evaluate the criterion function based on an approximate solution to the DP problem using 200 interpolation points and 500 random draws for the evaluation of $E\max$. On average the optimizer stops after 880 function evaluations and 196 steps. The RMSE remains small with about 0.03 when simulating a new sample based on the mean parameter values across all bootstrap iterations.

So far I successfully recomputed the diagnostics provided in Keane and Wolpin (1994) and the results are encouraging. However, given the experience in Eisenhauer et al. (2015), I worry that the approximation error introduces noise in the criterion function resulting in many local minima. If so, a Monte Carlo exercise that uses the true parameters as the starting values potentially disguises the trade-off

Table 2: Interpolation Schemes

Points	200	500	1500	All
E max Draws	500	500	500	500
RMSE	0.10	0.06	0.05	0.03
Minutes	13	70	112	1995
Steps	573	2848	2307	21415
Evaluations	1838	6745	6327	41285

between computation time and reliability of results as the optimizer simply gets stuck in a local minimum close by. To assess the validity of my concern, I will now investigate the noise in the criterion function and conduct a more challenging version of the Monte Carlo exercise.

Noise in Criterion Function. I trace out the exact and approximate criterion function from the previous Monte Carlo exercise around the true parameter values. To get a sense of a possible discrepancy between the two, I perturb β_1, which captures the tuition cost of a higher education, around its true value in \$100 increments. While the exact criterion function has its minimum at the actual value this is not true for for the approximated function. The latter attains its minimum at a perturbation of -\$100. This casts doubt on the quality of the approximation.

Estimation based on Approximate Solution (revisited). I thus conduct a slightly modified version of the previous Monte Carlo exercise by choosing more challenging starting values. I first estimate a misspecified static model ($\delta = 0.00$) starting from the true parameterization of the dynamic model using the *exact sample*. I then use the estimation results as starting values for the subsequent estimation of a correctly specified dynamic model ($\delta = 0.95$) on the same dataset. Table 2 summarizes the results for alternative interpolation schemes. Focusing on the RMSE and the total computation time in minutes, it shows how the accuracy of results increases with the refinement of the interpolation scheme. However, this comes at the cost of steep increases in computation time.

4 Conclusion

I successfully recompute key results from Keane and Wolpin (1994) and provide some additional diagnostics that highlight the trade-off between computation time and reliability of results. Thus, I draw attention to the original authors repeated warnings that the performance of their proposed interpolation scheme needs to be carefully assessed by researchers in their particular setting. A pragmatic approach is to successively increase the number of interpolation points as the estimation progresses towards the final results.

References

Bellman, R. E. (1957). *Dynamic Programming.* Princeton University Press, Princeton, NJ.

Bellman, R. E., Kalaba, R., and Kotkin, B. (1963). Polynomial Approximation - A New Computational Technique in Dynamic Programming: Allocation Processes. *Mathematics of Computation*, 17(82):155–161.

Eisenhauer, P., Heckman, J. J., and Mosso, S. (2015). Estimation of Dynamic Discrete Choice Models by Maximum Likelihood and the Simulated Method of Moments. *International Economic Review*, 56(2):331–357.

Fisher, R. A. (1922). On the Mathematical Foundations of Theoretical Statistics. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 222(594-604):309–368.

Keane, M. P., Todd, P. E., and Wolpin, K. I. (2011). The Structural Estimation of Behavioral Models: Discrete Choice Dynamic Programming Methods and Applications. In Ashenfelter, O. and Card, D., editors, *Handbook of Labor Economics*, volume 4A, pages 331–461. Elsevier Science, Amsterdam, Netherlands.

Keane, M. P. and Wolpin, K. I. (1994). The Solution and Estimation of Discrete Choice Dynamic Programming Models by Simulation and Interpolation: Monte Carlo Evidence. *The Review of Economics and Statistics*, 76(4):648–672.

Manski, C. F. and Lerman, S. R. (1977). The Estimation of Choice Probabilities from Choice Based Samples. *Econometrica*, 45(8):1977–1988.

Wolpin, K. I. (2013). *Limits to Inference without Theory.* MIT University Press, Cambridge, MA.

Appendix

The Approximate Solution of Finite-Horizon Discrete Choice Dynamic Programming Models

Revisiting Keane and Wolpin (1994)

Philipp Eisenhauer

University of Bonn

December 23, 2016

Contents

A. Computational Details

The `respy` package (respy, 2016) provides the computational support for the project. Its online documentation is available at `http://respy.readthedocs.io` and thus I only outline the implementation details that are specific to this paper.

Optimization I use the NEWUOA algorithm (Powell, 2006). All tuning parameters are set to their default values. I use a diagonal scale-based preconditioner based on a gradient approximation. I set its minimum value to 0.00001.

Integration The solution and estimation of the model produces two types of integrals. I need to determine $E\max$ during the solution step and simulate the choice probabilities to construct the sample log-likelihood. I evaluate both using Monte Carlo integration. I use 200 random draws for the choice probabilities.

Differentiation The derivatives required for the preconditioning are approximated by forward finite differences with a step size of 0.0001.

Function Smoothing I simulate the choice probabilities to evaluate the sample log-likelihood. With only a finite number of draws, there is always the risk of simulating zero probability for an agent's observed decision. So I use the logit-smoothed accept-reject simulator as suggested by McFadden (1989). The scale parameter is set to 500.

Function Approximation The details for the $E\max$ interpolation are already discussed in the text. However, there are some additional complications.

- Agents are only allowed to obtain 10 additional years of education. Thus, there exist a number of inadmissible states in late periods. However, $\bar{V}_3$ is still included in the interpolation regression and assigned an ad hoc penalty of -40,000. Results are not sensitive to the exact value as only about 5% of the states in later periods are affected.

- As noted in their correspondence with the editor, Keane and Wolpin (1994) drop the linear term of V_3 from the interpolation regression for the first parameterization due to reported collinearity problems. These are due to the small variation in the consumption value of schooling across states. I encounter the same problem and thus follow their lead.

I am indebted to several other open source tools among them `matplotlib` (Hunter, 2007) and `Vagrant` (Hashimoto, 2013).

B. Additional Results

This section presents all my results for each of the parameterizations in Table B.1. The *exact solution* is constructed using 100,000 random draws for the evaluation of $E\max$ at all states at the true parameter values. The *exact sample* refers to a set of 1,000 simulated agents based on the *exact solution*. As on overall measure of the approximation error, I use the root-mean-square error (RMSE) by comparing the choice probabilities in the *exact sample* to a newly simulated set of 1,000 agents based on the relevant alternative parameterization of the model.

B.1. Parameterizations

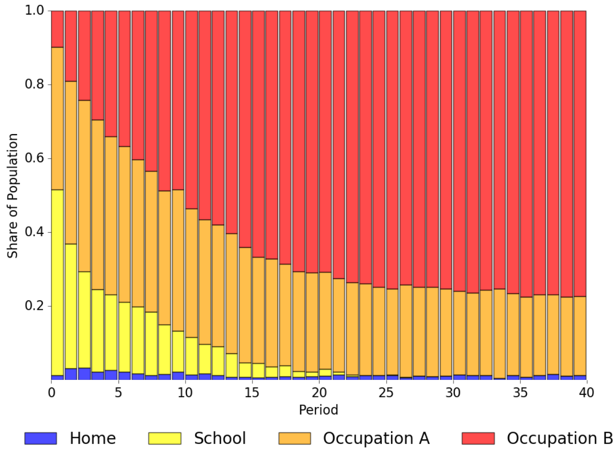
Table B.1: Parameterizations

Parameter	Data One	Data Two	Data Three
α_{10}	9.2100	9.2100	8.0000
α_{11}	0.0380	0.4000	0.0700
α_{12}	0.0330	0.0330	0.0550
α_{13}	0.0005	0.0005	0.0000
α_{14}	0.0000	0.0000	0.0000
α_{15}	0.0000	0.0000	0.0000
α_{20}	8.4800	8.2000	7.9000
α_{21}	0.0700	0.0800	0.0700
α_{22}	0.0670	0.0670	0.0600
α_{23}	0.0010	0.0010	0.0000
α_{24}	0.0220	0.0220	0.0550
α_{25}	0.0005	0.0005	0.0000
β_0	0.0000	5000.0000	5000.0000
β_1	0.0000	5000.0000	5000.0000
β_2	4000.0000	15000.0000	20000.0000
γ_0	17750.0000	14500.0000	21500.0000
$(\sigma_{11})^{1/2}$	0.2000	0.4000	1.0000
σ_{12}	0.0000	0.0000	0.5000
σ_{13}	0.0000	0.0000	0.0000
σ_{14}	0.0000	0.0000	0.0000
$(\sigma_{22})^{1/2}$	0.2500	0.5000	1.0000
σ_{23}	0.0000	0.0000	0.0000
σ_{24}	0.0000	0.0000	0.0000
$(\sigma_{33})^{1/2}$	1500.0000	6000.0000	7000.0000
σ_{34}	0.0000	0.0000	-2.975×10^7
$(\sigma_{44})^{1/2}$	1500.0000	6000.0000	8500.0000

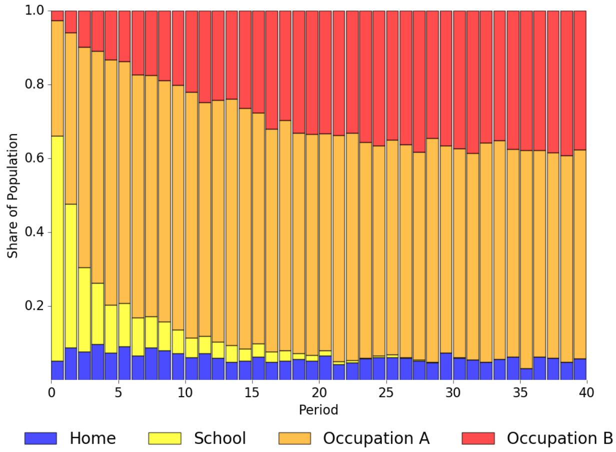
B.2. Choice Patterns

Figure B.1 shows the share of agents in the *exact sample* opting for each of the four alternatives by period.

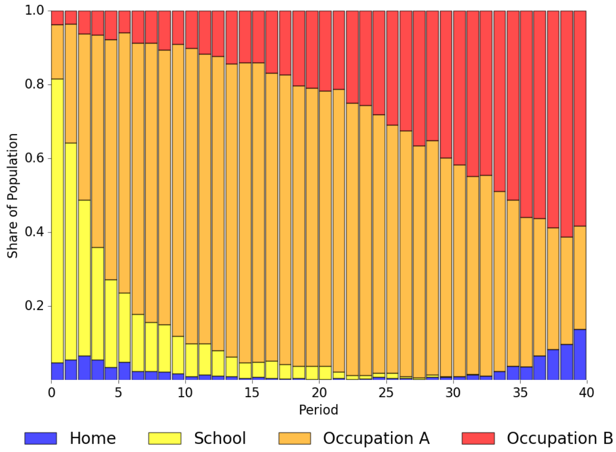
Figure B.1: Choice Patterns



(a) Data One



(b) Data Two



(c) Data Three

B.3. Correct Choices

Tables B.2 - B.4 show the proportion of correct choices for alternative interpolation schemes.

B.4. Monte Carlo Exercise

Tables B.5 - B.7 show the estimation performance for each of the model parameters during the initial Monte Carlo exercise. Let θ_i denote the true value of parameter i, $\hat{\theta}_i$ its average estimate across all bootstrap replications, and $\hat{\theta}_{ij}$ the estimated parameter in iteration j. The statistics in the Table B.5 - B.7 are calculated as follows:

Bias	$\hat{\theta}_i - \theta_i$
t - statistic	$\left(\frac{\hat{\theta}_i - \theta_i}{\sigma_{\hat{\theta}_i}}\right)\sqrt{40}$
Standard Deviation	$\left[\frac{1}{39}\sum_{j=1}^{40}(\hat{\theta}_{ij} - \hat{\theta}_i)^2\right]^{\frac{1}{2}}$

Note that the table contains the Cholesky decomposition parameters a_{ij} of the covariance matrix of the shocks to the immediate rewards. I report the RMSE, the total number of evaluations of the criterion function, and the number of steps of the optimizer as their average across all 40 bootstrap iterations.

I specify 200 interpolation points, use 500 random draws for the evaluation of $E\max$, and allow for a maximum of 1,000 evaluations of the criterion function by the optimizer during each estimation.

Table B.2: Correct Choices, Dataset One

Points	All	All	All	2,000	500
E max Draws	2,000	1,000	250	2,000	2,000
At Selected Periods					
Period					
1	1.000	0.998	0.938	0.967	0.948
10	0.990	1.000	0.989	0.989	0.983
20	1.000	1.000	1.000	0.998	0.999
30	1.000	1.000	1.000	0.994	0.998
40	1.000	1.000	1.000	1.000	1.000
Total	0.999	0.998	0.994	0.993	0.992
Number of Periods over the Lifetime					
Periods					
1 - 10	0.000	0.000	0.000	0.000	0.000
11 - 35	0.000	0.000	0.000	0.000	0.000
36 - 38	0.010	0.000	0.020	0.029	0.036
39	0.036	0.043	0.181	0.186	0.214
40	0.963	0.957	0.799	0.785	0.750
Average	39.962	39.957	39.777	39.752	39.710

Table B.3: Correct Choices, Dataset Two

Points	All	All	All	2,000	500
E max Draws	2,000	1,000	250	2,000	2,000
At Selected Periods					
Period					
1	0.998	0.994	0.993	0.996	0.988
10	1.000	0.998	0.995	0.990	0.972
20	1.000	0.997	0.994	0.979	0.961
30	0.998	1.000	0.998	0.988	0.989
40	1.000	1.000	1.000	1.000	1.000
Total	0.998	0.997	0.995	0.990	0.981
Number of Periods over the Lifetime					
Periods					
1 - 10	0.000	0.000	0.000	0.000	0.000
11 - 35	0.000	0.000	0.000	0.001	0.001
36 - 38	0.003	0.003	0.012	0.062	0.157
39	0.040	0.085	0.172	0.260	0.361
40	0.957	0.912	0.816	0.677	0.481
Average	39.954	39.909	39.804	39.600	39.265

Table B.4: Correct Choices, Dataset Three

Points	All	All	All	2,000	500
$E\max$ Draws	2,000	1,000	250	2,000	2,000
At Selected Periods					
Period					
1	0.995	0.993	0.985	0.991	0.979
10	0.995	0.995	0.982	0.975	0.931
20	0.995	0.997	0.994	0.979	0.940
30	0.994	0.999	0.989	0.974	0.972
40	1.000	1.000	1.000	1.000	1.000
Total	0.995	0.995	0.991	0.980	0.959
Number of Periods over the Lifetime					
Periods					
1 - 10	0.000	0.000	0.000	0.000	0.000
11 - 35	0.000	0.000	0.000	0.003	0.030
36 - 38	0.015	0.015	0.038	0.187	0.432
39	0.142	0.150	0.249	0.324	0.304
40	0.843	0.835	0.713	0.486	0.234
Average	39.827	39.817	39.671	39.226	38.374

Table B.5: Monte Carlo Exercise, Dataset One

Parameter	True Value	Bias	t - statistic	Std. Deviation
α_{10}	9.2100	0.0006	0.9004	0.0044
α_{11}	0.0380	0.0002	2.2596	0.0005
α_{12}	0.0330	-0.0002	-3.2184	0.0003
α_{13}	0.0005	0.0000	-9.3254	0.0000
α_{14}	0.0000	-0.0014	-5.1203	0.0018
α_{15}	0.0000	0.0000	-3.2468	0.0001
α_{20}	8.4800	0.0001	0.2152	0.0041
α_{21}	0.0700	-0.0001	-3.9652	0.0002
α_{22}	0.0670	0.0001	2.3170	0.0002
α_{23}	0.0010	0.0000	-1.6390	0.0000
α_{24}	0.0220	-0.0002	-3.0853	0.0005
α_{25}	0.0005	0.0000	-5.2772	0.0000
β_0	0.0000	-79.4140	-2.8542	175.9723
β_1	0.0000	1.0372	0.0332	197.3811
β_2	4000.0000	-9.1336	-0.3067	188.3330
γ_0	17750.0000	-43.4917	-1.6589	165.8143
a_{11}	0.2000	-0.0014	-1.9119	0.0045
a_{21}	0.0000	-0.0010	-4.1404	0.0016
a_{22}	0.2500	0.0018	3.9672	0.0029
a_{31}	0.0000	43.8599	1.3636	203.4216
a_{32}	0.0000	-70.0294	-2.5820	171.5326
a_{33}	1500.0000	-224.7522	-6.6330	214.3020
a_{41}	0.0000	-25.5905	-0.9928	163.0176
a_{42}	0.0000	-148.6872	-2.7044	347.7198
a_{43}	0.0000	17.8211	0.3382	333.3108
a_{44}	1500.0000	-232.8707	-4.3257	340.4774
Steps	196		Evaluations	880
RMSE	0.0310			

Notes: Std. Deviation = Standard Deviation.

Table B.6: Monte Carlo Exercise, Dataset Two

Parameter	True Value	Bias	t - statistic	Std. Deviation
α_{10}	9.2100	-0.0031	-5.2747	0.0037
α_{11}	0.0400	-0.0001	-1.8221	0.0003
α_{12}	0.0330	-0.0001	-1.9466	0.0003
α_{13}	0.0005	0.0000	0.6080	0.0000
α_{14}	0.0000	-0.0002	-1.1480	0.0010
α_{15}	0.0000	0.0000	-1.7886	0.0001
α_{20}	8.2000	-0.0057	-3.8997	0.0093
α_{21}	0.0800	-0.0003	-3.3396	0.0005
α_{22}	0.0670	-0.0002	-1.7560	0.0006
α_{23}	0.0010	0.0000	2.2861	0.0000
α_{24}	0.0220	0.0002	2.1251	0.0006
α_{25}	0.0005	0.0000	-4.6515	0.0000
β_0	5000.0000	-85.7904	-2.1768	249.2547
β_1	5000.0000	98.2520	1.8702	332.2710
β_2	15000.0000	-177.1514	-2.8181	397.5774
γ_0	14500.0000	-20.8554	-0.9578	137.7174
a_{11}	0.4000	-0.0409	-1.4343	0.1801
a_{21}	0.0000	0.0078	3.8322	0.0128
a_{22}	0.5000	-0.0001	-0.7259	0.0011
a_{31}	0.0000	154.8428	-2.5591	382.6844
a_{32}	0.0000	163.6551	2.5476	406.2885
a_{33}	6000.0000	92.2413	2.8473	204.8883
a_{41}	0.0000	-69.1740	-1.8808	232.6103
a_{42}	0.0000	120.6481	-2.6023	293.2152
a_{43}	0.0000	216.7521	-3.5171	389.7662
a_{44}	6000.0000	-1.0222	-0.0294	220.0363
Steps	18		Evaluations	453
RMSE	0.0212			

Notes: Std. Deviation = Standard Deviation.

Table B.7: Monte Carlo Exercise, Dataset Three

Parameter	True Value	Bias	t - statistic	Std. Deviation
α_{10}	8.0000	-0.0063	-3.7865	0.0105
α_{11}	0.0700	-0.0004	-2.2197	0.0012
α_{12}	0.0550	0.0001	0.6167	0.0014
α_{13}	0.0000	0.0000	1.2210	0.0000
α_{14}	0.0000	-0.0020	-3.6520	0.0035
α_{15}	0.0000	-0.0004	-3.6545	0.0007
α_{20}	7.9000	-0.0023	-2.4837	0.0059
α_{21}	0.0700	-0.0008	-2.7847	0.0018
α_{22}	0.0600	-0.0014	-3.7394	0.0023
α_{23}	0.0000	0.0001	3.1987	0.0002
α_{24}	0.0550	0.0002	0.7016	0.0017
α_{25}	0.0000	0.0000	-3.0081	0.0001
β_0	5000.0000	78.7045	0.8166	609.5716
β_1	5000.0000	50.7335	0.5600	572.9390
β_2	20000.0000	-210.1624	-1.2036	1104.3079
γ_0	21500.0000	23.1652	5.2334	27.9952
a_{11}	1.0000	-0.0394	-5.6275	0.0443
a_{21}	0.5000	-0.0128	-3.3182	0.0244
a_{22}	0.8660	-0.0018	-0.8715	0.0130
a_{31}	0.0000	-420.8300	-2.8233	942.7089
a_{32}	0.0000	366.3867	1.3155	1761.5370
a_{33}	7000.0000	374.6602	4.9161	482.0007
a_{41}	0.0000	225.6722	1.0027	1423.3962
a_{42}	0.0000	56.4475	0.1540	2318.6612
a_{43}	-4250.0000	-372.7514	-1.4050	1677.9763
a_{44}	7361.2159	511.8738	2.1290	1520.6238
Steps	49		Evaluations	558
RMSE	0.0133			

Notes: Std. Deviation = Standard Deviation.

Table B.8: Interpolation Schemes

Points	200	500	1500	All
E max Draws	500	500	500	500
RMSE	0.10	0.06	0.05	0.03
Minutes	13	70	112	1995
Steps	573	2848	2307	21415
Evaluations	1838	6745	6327	41285

B.5. Interpolation Schemes

Table B.8 shows the estimation results based on alternative interpolation schemes. For the static estimation, I solve the complete DP problem, use 500 random draws for the evaluation of $E\max$, and allow for a maximum of 1,000 evaluations of the criterion function by the optimizer. I use a single processor for all estimations.

C. Recomputation Instructions

I provide an image of a virtual machine (VM) for download to ensure full recomputability of my results. The image contains a software-based emulation of a computer, where all the required software is already pre-installed. This makes recomputation straightforward.

Two additional software tools are required: (1) `VirtualBox` and (2) `Vagrant`. `VirtualBox` is a virtualization software, while `Vagrant` provides a convenient wrapper around it. Both are free and open-source. Please consult their websites for installation instructions. The following instructions were tested for `VirtualBox 5.0` and `Vagrant 1.8`.

Once `VirtualBox` and `Vagrant` are available, the image can be downloaded and accessed by the following commands:

```
$ vagrant init structRecomputation/base
$ vagrant up --provider virtualbox
$ vagrant ssh
```

As all the required software is already installed, recomputation is straightforward. Simply typing the following command into the terminal produces all the results in

Table C.1: Mapping between Files and Results

File	Keane and Wolpin (1994)	Eisenhauer (2016)
	Correct Choices	
`table_2.1.txt`	Table 2.1	Table B.2
`table_2.2.txt`	Table 2.2	Table B.3
`table_2.3.txt`	Table 2.3	Table B.4
	Monte Carlo Estimation	
`table_4.1.txt`	Table 4.1	Table B.5
`table_4.2.txt`	Table 4.2	Table B.6
`table_4.3.txt`	Table 4.3	Table B.7
	Choice Patterns	
`choices_one.png`	—	Figure B.1
`choices_two.png`	—	Figure B.1
`choices_three.png`	—	Figure B.1
	Criterions	
`criterions.png`	—	Figure ??
	Interpolation Schemes	
`schemes.txt`	—	Table B.8

the paper:

```
$ ./recompute
```

The output files will be available in the `_published` subdirectory. This process takes a couple of days due to the large number of bootstrap iterations for the initial Monte Carlo exercise. There is a slight difference in the order and sign of the coefficients between the output files and the results in this paper, please see `respy`'s online documentation for details. Table C.1 provides the mapping between the output files and the results reported in the two relevant publications.

References

Eisenhauer, P. (2016). Approximate Solutions to Finite-Horizon Dynamic Programming Models: Revisiting Keane and Wolpin (1994). *Unpublished Manuscript*.

Hashimoto, M. (2013). *Vagrant: Up and Running*. O'Reilly Media, Sebastopol, CA.

Hunter, J. D. (2007). Matplotlib: A 2D Graphics Environment. *Computing in Science & Engineering*, 9(3):90–95.

Keane, M. P. and Wolpin, K. I. (1994). The Solution and Estimation of Discrete Choice Dynamic Programming Models by Simulation and Interpolation: Monte Carlo Evidence. *The Review of Economics and Statistics*, 76(4):648–672.

McFadden, D. (1989). A Method of Simulated Moments for Estimation of Discrete Response Models Without Numerical Integration. *Econometrica*, 57(5):995–1026.

Powell, M. J. D. (2006). The NEWUOA Software for Unconstrained Optimization Without Derivatives. In Pillo, Gianni, R. M., editor, *Large-Scale Nonlinear Optimization*, Nonconvex Optimization and Its Applications, pages 255–297. Springer, Boston, MA.

respy (2016). respy: A Python Package for the Simulation and Estimation of a Prototypical Discrete Choice Dynamic Programming Model, http://dx.doi.org/10.5281/zenodo.59297.