Institute for Applied Microeconomics
University of Bonn
Adenauerallee 24-42
53113 Bonn
Germany

August 4, 2016

Prof. Badi H. Baltagi
Department of Economics
426 Eggers Hall, Syracuse University
Syracuse, New York 13244-1020
United States

Dear Prof. Baltagi:

Thank you very much for considering my submission *Approximate Solutions to Finite-Horizon Dynamic Programming Problems: Revisiting Keane and Wolpin (1994)* for the *Replication Section* in the *Journal of Applied Econometrics (JAE)*.

I very much hope that you consider the publication of detailed replications of major structural econometric articles for the *JAE*. This could have a particularly positive effect for this line of research. It provides young researchers with a unique learning opportunity to tackle these often complex research projects in a controlled setting and increases the level of transparency regarding computational details. Of course, I would be glad to serve as a referee for such future submissions. I attach my curriculum vitae as a reference.

If you have any further questions or comments, please do not hesitate to contact me.

Yours Faithfully,

Philipp Eisenhauer

Curriculum Vitae

Philipp Eisenhauer

Contact Details

University of Bonn	Email	eisenhauer@policy-lab.org
Lennéstr. 43	Web	http://www.policy-lab.org/peisenha
53113 Bonn	Repository	https://github.com/peisenha
Germany		

Publications

Eisenhauer, Philipp and James J. Heckman, Edward Vytlacil (2015). *Generalized Roy Model and the Cost-Benefit Analysis of Social Programs*, Journal of Political Economy, 123(2):413-443

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

Education

since 11/2014	Postdoctoral Scholar at the University of Bonn Advisors: Thomas Dohmen and Armin Falk
03/2013–10/2014	Postdoctoral Scholar at the University of Chicago Advisor: James J. Heckman
09/2007–03/2013	Ph.D. in Economics at the University of Mannheim Thesis: *Essays in the Econometrics of Policy Evaluation* Advisors: Wolfgang Franz and James J. Heckman

---

Teaching

since 01/2016	*Zurich Initiative on Computational Economics*

03/2015–06/2015	Graduate *Software Engineering for Economists* at the University of Chicago
01/2013–03/2013	Graduate *Computational Econometrics* at the University of Chicago
03/2011–05/2011	Graduate *Microeconometrics* at the University of Chicago (TA)

Working Papers

Eisenhauer, Philipp (2016). *Risk and Ambiguity in Educational Choices*, research in progress

Eisenhauer, Philipp and Edward S. Sung (2012). *Optimal Treatment Reallocation*, available online

Eisenhauer, Philipp (2012). *Issues in the Economics and Econometrics of Policy Evaluation: Explorations Using a Factor Structure Model*, available online

Grants

06/2015	*AXA Research Fund* research on robust decisions over the life-cycle (EUR 120,000)
02/2015	*Microsoft Research Grant* for access to Microsoft Azure cloud computing infrastructure in support of research (USD 20,000)
01/2015	*Microsoft Educator Grant* for access to Microsoft Azure cloud computing infrastructure in support of teaching (USD 21,000)

Departmental and Professional Services

03/2015–06/2015	Co-organizer of the *Computational Economics Colloquium* at the University of Chicago
09/2009–12/2011	Co-organizer of the *Applied Economics and Econometrics Seminar* at the University of Mannheim
Refereeing	Econometrica, Journal of Political Economy, American Economic Review, The Review of Economic Studies, International Economic Review, National Science Foundation

Presentations and Summer Schools

Presentations	University of Chicago, Yale University, University of Zurich, University of Bonn, IZA Bonn, DIW Berlin, ZEW Mannheim (among others and including scheduled)
Summer Schools	*Labor Economics* of the Institute for the Study of Labor, *Computational Economics* of the Institute on Computational Economics, *Labour Economics - Theory, Empirical Methods, Current Research* of the German Research Foundation

Additional Skills

Programming	Fortran, Python, R, Stata, Linux O/S, Parallel Computing, Cluster Administration
Languages	German, English

References

James J. Heckman
University of Chicago Email jjh@uchicago.edu

Armin Falk
University of Bonn Email armin.falk@uni-bonn.de

August 4, 2016

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

Revisiting Keane and Wolpin (1994)

Philipp Eisenhauer

University of Bonn

August 4, 2016

Abstract

The estimation of finite-horizon discrete choice dynamic programming (DCDP) models pose computational challenges. This limits the realism of the models and impedes verification and validation efforts. Keane and Wolpin (1994) propose an approximation method that ameliorates the computational burden but introduces error. I describe their approach in detail, recompute their original quality diagnostics, and conduct some additional analysis.

JEL Codes: C13, C22, C53

Keywords: Discrete Choice Dynamic Programming Models, Approximation Methods

*I gratefully acknowledge support by the *AXA Research Fund*. Corresponding Address: Philipp Eisenhauer, eisenhauer@policy-lab.org, University of Bonn, Adenauerallee 24-42, 53113 Bonn, Germany. Tel.: +49 228 73-9219, 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 decisions 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 decisions.

Estimating a finite-horizon DCDP model requires the repeated solution of a dynamic programming (DP) problem under uncertainty by backward induction. The well known curse of dimensionality (Bellman, 1957) and the ensuing computational complexity is a major impediment to the application of more realistic DCDP 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 interpolate the rest. Of course, this introduces approximation error into the estimation and requires the 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.

As noted repeatedly in the original article, the analysis of reliability is highly model dependent. So, I use this note to describe their approach in detail, recompute their results, and provide some additional quality diagnostics that structural econometricians can use to verify the quality of the approximation in their setting. The rest of the paper is structured as follows. In Section 2, I present the economics of the basic model and outline the solution and estimation approach. I then turn to the results from the recomputation in Section 3 and provide additional diagnostics regarding the reliability of their approximation routine. Section 4 concludes.

[1] See Keane et al. (2011), Aguirregabiria and Mira (2010), and Arcidiacono and Ellickson (2011) for recent surveys of the literature.

2 Basic Setup

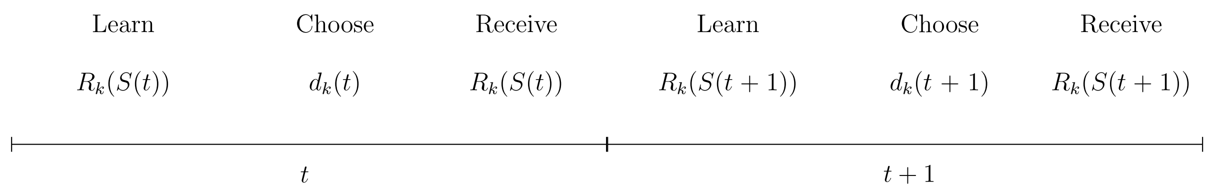
I now discuss the economics behind the prototypical DCDP model analyzed by Keane and Wolpin (1994) and present its parameterization. Then I turn to the solution approach.

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. All the information known to the agent at time t that affects immediate and future rewards is contained in the state space $S(t)$.

Figure 1 depicts the timing of events in the model for two generic time periods. At the beginning of period t the agent fully learns about the all immediate rewards, choose one of the alternatives and receives the corresponding benefits. The state space is then updated according to the agent's state experience and the process is repeated in $t + 1$.

Figure 1: Timing



Agents are forward looking. Thus, the do not simply choose the alternative with the highest immediate rewards. Instead their objective at any time $\tau = 0, ..., T$ is to maximize the expected rewards over their whole 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) consider three different parameterizations in their study. Each differs with respect to the level of unobserved variability of rewards, the returns to schooling, and transfer ability of occupation-specific human capital.

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:

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

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 (experience) 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 an indicator function equal to one if $s \geq 12$ (the agent has completed high school) and zero otherwise, β_2 is an adjustment cost associated with returning to school (if $d_3(t-1) = 0$), 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 Σ. The realizations are independent across time. I collect the parametrization of the reward functions in $\theta = \{\alpha_1, \alpha_2, \beta, \gamma, \Sigma\}$.

Given the structure of the immediate rewards, 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 the predetermined elements of the state space, $s_t, x_{1t}, x_{2t}, d_3(t-1)$, 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. I 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 the cumulative effects of different shocks.

2.2 Solution

From a mathematical perspective, this type of model boils down to a finite-horizon DP problem under uncertainty that can be solved by backward induction. For the discussion, it is useful to do 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 or the direct effect of age on rewards) 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 recursion.

$$V_k(S(t), t) = \begin{cases} R_k(S(t)) + \delta E\left[V(S(t+1), t+1) \mid S(t), d_k(t) = 1\right] & \text{if } t < T \\ R_k(S(t)) & \text{if } t = T. \end{cases} \tag{2}$$

Assuming continued optimal behavior, we must calculate expected future value of state $S(t+1)$ for all K alternatives given today's state $S(t)$ and choice $d_k(t) = 1$. We must determine $\mathbb{E}$max (S(t + 1)) for short.

$$E\max(S(t+1)) = E\left[V(S(t+1), t+1) \mid S(t), d_k(t) = 1\right]. \tag{3}$$

Future rewards are partly uncertain due to the existence of state variables at time $t+1$ that are unknown beforehand. The expected future value for state $S(t)$ is calculated as:

$$\begin{aligned} &E\max(S(t)) = \\ &\quad \int_{\epsilon_1(t)} \cdots \int_{\epsilon_K(t)} \max\{R_1(t), \ldots, R_K(t)\} f_{\boldsymbol{\epsilon}}(\epsilon_1(t), \ldots, \epsilon_K(t)) d\epsilon_1(t) \ldots d\epsilon_K(t), \end{aligned} \tag{4}$$

where $f_{\boldsymbol{\epsilon}}$ is the joint density of the uncertain component of the rewards in t not known at $t-1$.

3 Approximate Solution

The evaluation of $\mathbb{E}$max at each possible state creates a considerable computational burden. For example, even in this simplified model, it requires the repeated evaluation of the integral for the $\mathbb{E}$max at a total of 163,410 states. During an estimation, the model has to be solved repeatedly for numerous alternative parameterizations. Thus, Keane and Wolpin (1994) propose to calculate $\mathbb{E}$max only at a subset of states each period and interpolate its value for the rest. The choice of the interpolation function and the monitoring of the approximation error are key. I will now present the details of the approximation scheme and then conduct some quality diagnostics.

3.1 Operationalization

After experimentation, Keane and Wolpin (1994) settle on equation (5) as their preferred interpolation function.

$$\mathbb{E}\text{max} - \text{max}\mathbb{E} = \pi_0 + \sum_{j=1}^{4} \pi_{1j}(\text{max}\mathbb{E} - \bar{V}_j) + \sum_{j=1}^{4} \pi_{2j} \left(\text{max}\mathbb{E} - \bar{V}_j\right)^{\frac{1}{2}} \tag{5}$$

$\bar{V}_j$ is shorthand for the expected value of the alternative-specific value function and $\text{max}\mathbb{E} = \text{max}_k\{\bar{V}_j\}$ 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 state points used to fit the interpolating function is chosen at random for each period.

Let me outline the actual implementation in a bit more detail. During the backward induction procedure, starting from the second to last period, I determine each period if the total number of states in that period is larger or smaller than the number of specified interpolation points. If not, the $\mathbb{E}$max is solved at each state and we move on to the preceding period. But if the number of states is larger, then we draw a random sample of states for the interpolation points and fit equation (5) on that subset of states. Using this information, I construct the predicted values for all remaining states. Applying this interpolation scheme for a random subset of 200 states each period instead all states, reduces the number of states at which the $\mathbb{E}$max is explicitly evaluated from 163,410 to 6,930.

3.2 Original Diagnostics

Keane and Wolpin (1994) provide careful diagnostics for the quality of their proposed approximation for three alternative parameterizations θ of the reward functions. They assess the effect of alternative approximation schemes for simulated agent decisions and

Table 1: Correct Choices

$\mathbb{E}$max Draws	2000	1000	250	2000	2000
States	All	All	All	2000	500
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

also analyze the effect of using an approximation for estimation of the model. I present my recomputation of their key results for the first specification in this paper. Results for the the other two specifications are available in Appendix B.

In the following comparisons, an *exact solution* to the DP problem often serves as a benchmark. This solution is computed using 100,000 draws for the Monte Carlo integration and the $\mathbb{E}$max is computed add all states at the true parameter values. The *exact sample* refers to a set of 1,000 simulated agents based on the *exact solution*.

Simulation based on Approximate Solution. Table 1 shows the proportion of correct decisions for alternative approximation schemes. I follow each agent in the *exact sample* over time and evaluate for each period whether a decision based on an approximate solution is correct, i.e. aligns with the decision based on the *exact solution*. If I evaluate the $\mathbb{E}$max at all states with 2,000 Monte Carlo draws, then the two decisions align for 96% of the agents in the sample. This drops to 75% as the we only use 200 states each period and interpolate the rest. While always remaining above 39, the average number of correct decisions for each agent decreases as the approximation is coarsened.

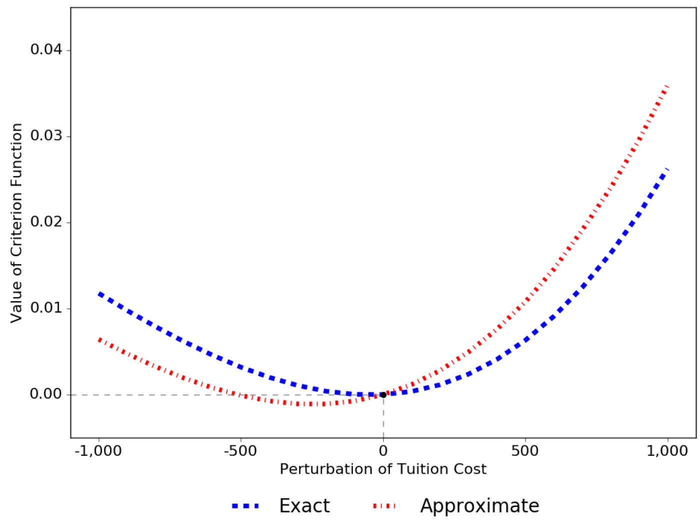
Estimation based on Approximate Solution. To assess the impact of the approximation as it ripples through an estimation, I perform a bootstrapping exercise by sampling a subset of 100 agents forty times from the *exact sample*. During estimation, I evaluate the criterion function based on *approximate* solution to the DP problem by using the interpolation routine with 200 states each period and 200 Monte Carlo draws. On average the optimizer stops after 5,262 function evaluations and 706 steps. To get an overall sense of performance, I calculate 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 an *exact solution* to the DP problem using mean parameter values across all bootstrap iterations. The RMSE is small with 0.05.[2]

However, given the experience in Eisenhauer et al. (2015) I am worried that the superb performance is partly an artifact of a noisy criterion function and using the true values at the start.

3.3 Additional Diagnostics

In Figure 2, I trace out an exact and an approximate criterion function around the true parameter values.[3] Due to its importance for the evaluation of tuition policies, I perturb β_1 around its true value in \$100 increments. For comparison, I shift both curves to have the value zero at the true parameter values. 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 -\$300.

Figure 2: Criterion Functions



I conduct a more challenging estimation exercise by constructing more challenging starting values. Instead of using the true values directly, I first estimate a misspecified static

[2] Additional results are available in Appendix B. More details about the estimation routine are available in Appendix A.2.

[3] The exact criterion function is based on the exact solution, while I evaluate the $\mathbb{E}$max at a random subset of 200 states each periods with 200 Monte Carlo draws.

Table 2: RMSE

$\mathbb{E}$max Draws	200	200	200	200	200
States	200	2000	7000	1000	All
RMSE	0.00	0.00	0.00	0.00	0.00
Evaluations	1000	1000	1000	1000	1000
Steps	1000	1000	1000	1000	1000

model ($\delta = 0.00$) on the *exact sample*.[4] Using these starting values, the initial RMSE is 0.05. Then I feed the resulting estimates back as starting values to the estimation of the correctly specified dynamic model ($\delta = 0.95$). Table 2 shows the RMSE after estimation for alternative approximation schemes.

The results show that the coarseness of the approximation method does affect estimation performance. This draws attention to Keane and Wolpin (1994) repeated statement that the performance of the approximation method need to be carefully assessed by researchers. A pragmatic approach would rely on a coarse approximation to determine and fine-tune starting values and then a finer grid for the actual estimation.

[4]I use 200 Monte Carlo draws.

4 Conclusion

I recomputed the results from Keane and Wolpin (1994) and provided some additional diagnostics about the performance of their proposed approximation method. My additional diagnostics re-emphasize the original author' claim that the performance of the approximation method need to be carefully assessed in each model. I argue for an informed pragmatism that acknowledges the risk in using relying on an approximation.

References

Aguirregabiria, V. and Mira, P. (2010). Dynamic Discrete Choice Structural Models: A Survey. *Journal of Econometrics*, 156(1):38–67.

Arcidiacono, P. and Ellickson, P. B. (2011). Practical Methods for Estimation of Dynamic Discrete Choice Models. *Annual Review of Economics*, 3(363-394):1.

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

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

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.

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.

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

Appendix

Approximate Solutions to Finite-Horizon Dynamic Programming Problems

Revisiting Keane and Wolpin (1994)

Philipp Eisenhauer
University of Bonn

August 4, 2016

Contents

A. Solution and Estimation

A.1. Likelihood Function

We estimate the model by maximum likelihood estimation. We observe a sample of agents over time whose behavior and state experiences are described by the model. While all random components are eventually observed by the agent, they remain unobserved by the econometrician. Due to the serial independence, we can compute the likelihood contribution in each period separately. The sample likelihood is then just the product overall all agents and time periods.

A key ingredient in this calculation are the choice probabilities. Decomposing the joint probability of the joint observation into the unconditional and conditional distribution, the unconditional choice probability can be written as:

$$\begin{aligned}&\text{Prob}(d_j(t) = 1 \mid \bar{S}(t)) = \\ &\int_{\epsilon_1(t)} \cdots \int_{\epsilon_K(t)} \mathrm{I}\left[V_j(S(t), t) = \max_{k \in K}\{V_k(S(t), t)\}\right] f_{\boldsymbol{\epsilon}}(\epsilon_1(t), \ldots, \epsilon_K(t)) d\epsilon_1(t) \ldots d\epsilon_K(t)\end{aligned} \tag{1}$$

In addition, at least for those agents working in the labor market, we also observe their respective wage. For this agents the we need to calculate the joint probablity of their observed wage and choice combination.

A.2. Computational Details

The model is estimate by simulated maximum likelihood (Albright et al., 1977).

Numerical Integration Integrals arise in the solution of the model and during the evaluation of the likelihood function. All are solved by Monte Carlo integration.

- For each agent in each time period, the **choice probabilities** are requires the approximated of a four-dimensional integral, see equation (1). The integral is evaluated using 200 draws.
- The calculation of the **expected future value** at each state requires the evaluation of a four dimensional integral, see equation (4). The integral is approximated using 5,000 draws unless otherwise noted.
- The same set of random draws are used for all integrals.

Numerical Optimization. I use Powell's algorithm (Powell, 1970). All tuning parameters remain set to their default values. The results are not sensitive to reasonable variations of the tuning parameters.

- With only a finite number of draws there is the risk that there are cells with zero probability, so I use kernel smoothed frequency simulation. The function that was used was the kernel smoothing function described in McFadden (1989) with a window parameter of 500. In principle, this allows for the use of gradient-based optimization algorithms. However, the simulation of choice probabilities and $\mathbb{E}$max introduces noise in the criterion function making the derivative information unreliable. Thus, I opted for a derivate-free alternative.

Approximation Methods The details for the **$\mathbb{E}$max interpolation** are already discussed in detail in the text. However, there are some additional complications.

- As the state space is truncated by ten years of education, 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 value penalty of -40,000. However, 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 dataset due to reported **collinearity problems**. These are due to the small variation in the consumption value of schooling across states. I encounter the same problems for small number of interpolation points and thus follow their example.

I am indebted several open source tools: `Python`, `Fortran`, `GNU Compiler Selection`, `SciPy` (Jones et al., 2012), `LAPACK` (Anderson et al., 1999), `statsmodels` (Seabold and Perktold, 2010), `NumPy` (van der Walt et al., 2011), `Matplotlib` (Hunter, 2007), `Pandas` (McKinney, 2010), `F2PY` (Peterson, 2009), `Vagrant` (Hashimoto, 2013).
The pseudorandom number generation is taken care of by George Marsaglia's KISS generator (Marsaglia, 1968) in Fortran.

B. Results

This section presents all recomputation results for reach of the specifications in Table B.1. All implementation details follow Keane and Wolpin (1994) directly. Please see Appendix A.2 for additional details on the computations.

For the *exact* solutions, I use 100,000 Monte Carlo draws and solve the DP problem at all states.

B.1. Specifications

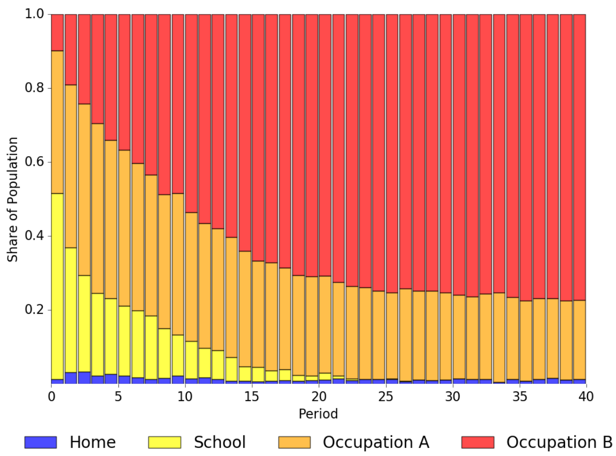
Table B.1: Specifications

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	0.3300	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	0.5500
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

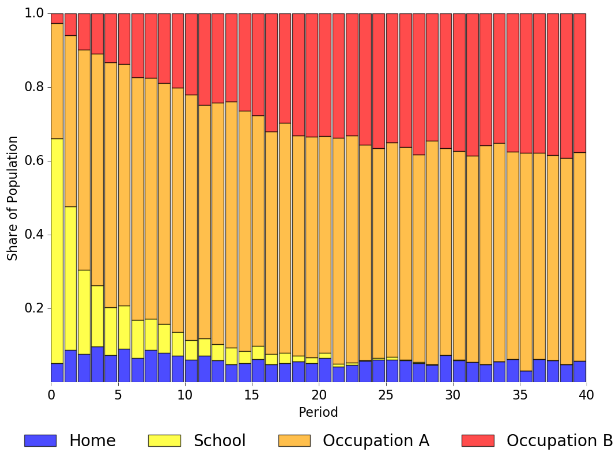
B.2. Choice Patterns

Figure B.1 shows the share of agents opting for each of the four alternatives in each period. The results are based on a simulated sample of 1,000 agents. Agent decisions are determined by the *exact* solution to the dynamic programming problem.

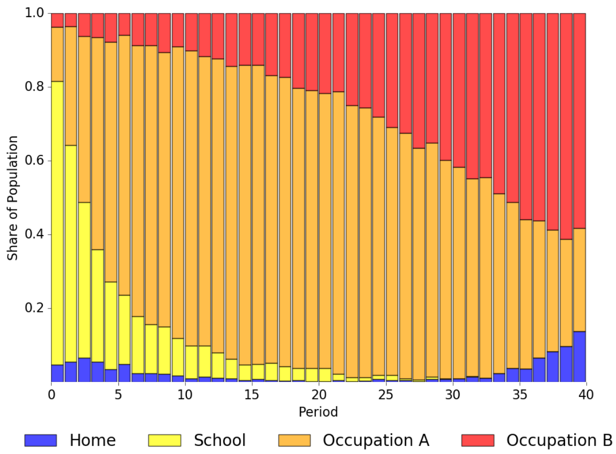
Figure B.1: Choice Patterns



(a) Data One



(b) Data Two



(c) Data Three

B.3. Approximation Performance

B.3.1. Correct Choices

Tables B.2 - B.4 show the proportion of correct choices for alternative approximation schemes. I first construct an *exact* solution to the dynamic programming problem. Then, I follow a sample of 1,000 agents through the model. At each state they visit, I compare the decision implied by the exact and an approximated solution to the dynamic programming problem. Each table presents the proportion of correct choices for alternative approximation schemes. The agent state is updated according to the exact solution.

Table B.2: Correct Choices, Dataset One

$\mathbb{E}$max Draws	2,000	1,000	250	2,000	2,000
States	All	All	All	2,000	500
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

$\mathbb{E}$max Draws	2,000	1,000	250	2,000	2,000
States	All	All	All	2,000	500
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

$\mathbb{E}$max Draws	2,000	1,000	250	2,000	2,000
States	All	All	All	2,000	500
			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: RMSE

	Data One	Data Two	Data Three
RMSE	0.00	0.00	0.00
Evaluations	1000	1000	1000
Steps	1000	1000	1000

B.3.2. Monte Carlo Exercise

Table B.5 provides a summary assessment of the estimation performance. It shows the root-mean-square error (RMSE) for the choice probabilities by comparing a simulated sample of 1,000 agents based on the *exact* solution to the DP problem at the true parameter values and the mean values across all bootstrap iterations. Also, it provides the information about the mean number of function evaluation and steps of the optimizer.

Tables B.6 - B.8 show the how the approximation affects estimation performance for each of the structural parameters. Agent decisions are determined based on the exact solution to the dynamic programming model. Keane and Wolpin (1994) subset of 100 agents from the full sample and estimate the model. For the solution step during estimation, they use 500 draws for the integration of the expected future values, 200 draws for the integration of the choice probabilities, and solve the model for 200 states each period. For each of the bootstrap iterations, the optimization routine is started at the true value of the structural parameters. Please see Appendix A.2 for more information about computational details.

Let θ denote the true value of each parameter, $\hat{\theta}$ the average estimate across all bootstrap replications, and $\hat{\theta}_j$ is the estimated parameter in iteration j. The statistics in the Table B.6 - B.8 are calculated as follows:

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

Note that the table contains the Cholesky decomposition parameters a_{ij} used to generate the covariance matrix of the errors and not directly its elements.

Table B.6: Monte Carlo Exercise, Dataset One

Parameter	True Value	Bias	t - statistic	Std. Deviation
α_{10}	9.2100	0.0000	0.000	0.000
α_{11}	0.0380	0.0000	0.000	0.000
α_{12}	0.0330	0.0000	0.000	0.000
α_{13}	0.0005	0.0000	0.000	0.000
α_{14}	0.0000	0.0000	0.000	0.000
α_{15}	0.0000	0.0000	0.000	0.000
α_{20}	8.4800	0.0000	0.000	0.000
α_{21}	0.0700	0.0000	0.000	0.000
α_{22}	0.0670	0.0000	0.000	0.000
α_{23}	0.0010	0.0000	0.000	0.000
α_{24}	0.0220	0.0000	0.000	0.000
α_{25}	0.0005	0.0000	0.000	0.000
β_0	0.0000	0.0000	0.000	0.000
β_1	0.0000	0.0000	0.000	0.000
β_2	4000.0000	0.0000	0.000	0.000
γ_0	17750.0000	0.0000	0.000	0.000
a_{11}	0.2000	0.0000	0.000	0.000
a_{21}	0.0000	0.0000	0.000	0.000
a_{22}	0.2500	0.0000	0.000	0.000
a_{31}	0.0000	0.0000	0.000	0.000
a_{32}	0.0000	0.0000	0.000	0.000
a_{33}	1500.0000	0.0000	0.000	0.000
a_{41}	0.0000	0.0000	0.000	0.000
a_{42}	0.0000	0.0000	0.000	0.000
a_{43}	0.0000	0.0000	0.000	0.000
a_{44}	1500.0000	0.0000	0.000	0.000

Notes: Std. Deviation = Standard Deviation

Table B.7: Monte Carlo Exercise, Dataset Two

Parameter	True Value	Bias	t - statistic	Std. Deviation
α_{10}	9.2100	0.0000	0.000	0.000
α_{11}	0.4000	0.0000	0.000	0.000
α_{12}	0.3300	0.0000	0.000	0.000
α_{13}	0.0005	0.0000	0.000	0.000
α_{14}	0.0000	0.0000	0.000	0.000
α_{15}	0.0000	0.0000	0.000	0.000
α_{20}	8.2000	0.0000	0.000	0.000
α_{21}	0.0800	0.0000	0.000	0.000
α_{22}	0.0670	0.0000	0.000	0.000
α_{23}	0.0010	0.0000	0.000	0.000
α_{24}	0.0220	0.0000	0.000	0.000
α_{25}	0.0005	0.0000	0.000	0.000
β_0	5000.0000	0.0000	0.000	0.000
β_1	5000.0000	0.0000	0.000	0.000
β_2	15000.0000	0.0000	0.000	0.000
γ_0	14500.0000	0.0000	0.000	0.000
a_{11}	0.4000	0.0000	0.000	0.000
a_{21}	0.0000	0.0000	0.000	0.000
a_{22}	0.5000	0.0000	0.000	0.000
a_{31}	0.0000	0.0000	0.000	0.000
a_{32}	0.0000	0.0000	0.000	0.000
a_{33}	6000.0000	0.0000	0.000	0.000
a_{41}	0.0000	0.0000	0.000	0.000
a_{42}	0.0000	0.0000	0.000	0.000
a_{43}	0.0000	0.0000	0.000	0.000
a_{44}	6000.0000	0.0000	0.000	0.000

Notes: Std. Deviation = Standard Deviation

Table B.8: Monte Carlo Exercise, Dataset Three

Parameter	True Value	Bias	t - statistic	Std. Deviation
α_{10}	8.0000	0.0000	0.000	0.000
α_{11}	0.0700	0.0000	0.000	0.000
α_{12}	0.0550	0.0000	0.000	0.000
α_{13}	0.0000	0.0000	0.000	0.000
α_{14}	0.0000	0.0000	0.000	0.000
α_{15}	0.0000	0.0000	0.000	0.000
α_{20}	7.9000	0.0000	0.000	0.000
α_{21}	0.0700	0.0000	0.000	0.000
α_{22}	0.0600	0.0000	0.000	0.000
α_{23}	0.0000	0.0000	0.000	0.000
α_{24}	0.5500	0.0000	0.000	0.000
α_{25}	0.0000	0.0000	0.000	0.000
β_0	5000.0000	0.0000	0.000	0.000
β_1	5000.0000	0.0000	0.000	0.000
β_2	20000.0000	0.0000	0.000	0.000
γ_0	21500.0000	0.0000	0.000	0.000
a_{11}	1.0000	0.0000	0.000	0.000
a_{21}	0.5000	0.0000	0.000	0.000
a_{22}	0.8660	0.0000	0.000	0.000
a_{31}	0.0000	0.0000	0.000	0.000
a_{32}	0.0000	0.0000	0.000	0.000
a_{33}	7000.0000	0.0000	0.000	0.000
a_{41}	0.0000	0.0000	0.000	0.000
a_{42}	0.0000	0.0000	0.000	0.000
a_{43}	-4250.0000	0.0000	0.000	0.000
a_{44}	7361.0000	0.0000	0.000	0.000

Notes: Std. Deviation = Standard Deviation

C. Recomputation Instructions

I ensure full recomputability of my results by providing the image of a virtual machine (VM) for download. The image provides a software-based emulation of a computer, where all the software and scripts used in our analysis are already properly installed. This makes the recomputation of our results straightforward.

Two 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 software. Please consult their websites for information about download and installation. The following instructions were tested for `VirtualBox 5.0` and `Vagrant 1.8`.

Once Vagrant and Virtualbox are installed, the VM can be downloaded and started by the following commands:

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

The VM is now ready to be accessed:

```
$ vagrant ssh
```

All scripts and required software are already properly installed on the VM, so by simply typing:

```
$ ./recompute
```

This starts all required computations and the published results will be available in the `_published` directory. Note, however, that this process takes a couple days due to the large number of bootstrap iterations for the Monte Carlo exercises.

Table C.1 provides the mapping between the figures in the results directory and the two relevant publications.

The results in this paper were created with the first release of the **respy** package (The respy Team, 2016).

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 X
table_2.2.txt	Table 2.2	Table X
table_2.3.txt	Table 2.3	Table X
	Monte Carlo Estimation	
table_4.1.txt	Table 4.1	Table X
table_4.2.txt	Table 4.2	Table X
table_4.3.txt	Table 4.3	Table X
	Choice Patterns	
choices_one.png	—	Figure X
choices_two.png	—	Figure X
choices_three.png	—	Figure X
	Noise	
noise.txt	—	Figure X
	Points	
points.txt	—	Figure X

Notes:

References

Albright, R. S., Lerman, S., and Manski, C. (1977). *Report on the Development of an Estimation Program for the Multinomial Probit Model*. Cambridge Systematics.

Anderson, E., Bai, Z., Bischof, C., Blackford, S., Demmel, J., Dongarra, J., Du Croz, J., Greenbaum, A., Hammarling, S., McKenney, A., and Sorensen, D. (1999). *LAPACK Users' Guide*. Society for Industrial and Applied Mathematics, Philadelphia, PA, third edition.

Eisenhauer, P. (2016). Approximate Solutions to Finite-Horizon Dynamic Programming Problems: 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.

Jones, E., Oliphant, T., Peterson, P., et al. (2012). SciPy: Open Source Scientific Tools for Python.

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.

Marsaglia, G. (1968). Random numbers fall mainly in the planes. *Proceedings of the National Academy of Sciences*, 61(1):25–28.

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

McKinney, W. (2010). Data structures for statistical computing in python. In van der Walt, S. and Millman, J., editors, *Proceedings of the 9th Python in Science Conference*, pages 51 – 56.

Peterson, P. (2009). F2PY: A Tool for Connecting Fortran and Python Programs. *International Journal of Computational Science and Engineering*, 4(4):296–305.

Powell, M. J. D. (1970). An Effficient Method for Finding the Minimum of a Function of Several Variables Without Calculating Derivatives. *The Computer Journal*, 7(2):155–162.

Seabold, S. and Perktold, J. (2010). Statsmodels: Econometric and statistical modeling with python. In *9th Python in Science Conference*.

The respy Team (2016). respy 1.0. http://respy.readthedocs.io.

van der Walt, S., Colbert, S. C., and Varoquaux, G. (2011). The NumPy Array: A Structure for Efficient Numerical Computation. *Computing in Science & Engineering*, 13(2):22–30.