Eckstein-Keane-Wolpin models

An invitation for transdisciplinary collaboration

Open Source Economics Bonn

Computational modeling in economics

- provide learning opportunities
- assess importance of competing mechanisms
- predict the effects of public policies

Eckstein-Keane-Wolpin (EKW) models

- understanding individual decisions
  - human capital investment
  - savings and retirement
- predicting effects of policies
  - welfare programs
  - tax schedules

Transdisciplinary components

- economic model
- mathematical formulation
- computational implementation

Cooperations

DIW BERLIN

Statistisk sentralbyrå
Statistics Norway

Institute for
Numerical Simulation

Roadmap

- Setup
- Example
- Improvements
- Extensions

Setup

Components

- economic model
- mathematical formulation
- calibration procedure

Economic model

Decision problem

$t = 1, \ldots, T$	decision period
$s_t \in S$	state
$a_t \in A$	action
$a_t(s_t)$	decision rule
$r_t(s_t, a_t)$	immediate reward

Timing of events

Learn	Choose	Receive
$\{r_t(s_t, a_t)\}_{a_t \in A}$	a_t	$r_t(s_t, a_t)$

t

Learn	Choose	Receive
$\{r_{t+1}(s_{t+1}, a_{t+1})\}_{a_{t+1} \in A}$	a_{t+1}	$r_{t+1}(s_{t+1}, a_{t+1})$

$t + 1$

$\pi = (a_1^\pi(s_1), \ldots, a_T^\pi(s_T))$	policy
δ	discount factor
$p_t(s_t, a_t)$	conditional distribution

Individual's objective

$$\max_{\pi \in \Pi} \mathrm{E}_{s_1}^{\pi} \left[\sum_{t=1}^{T} \delta^{t-1} r_t(s_t, a_t^{\pi}(s_t)) \,\middle|\, \mathcal{I}_1 \right]$$

Mathematical formulation

Policy evaluation

$$v_t^\pi(s_t) \equiv \mathrm{E}_{s_t}^\pi \left[\sum_{j=0}^{T-t} \delta^j \, r_{t+j}(s_{t+j}, a_{t+j}^\pi(s_{t+j})) \,\middle|\, \mathcal{I}_t \right]$$

Inductive scheme

$$v_t^\pi(s_t) = r_t(s_t, a_t^\pi(s_t)) + \delta \, \mathrm{E}_{s_t}^\pi \left[v_{t+1}^\pi(s_{t+1}) \,\middle|\, \mathcal{I}_t \right]$$

Optimality equations

$$v_t^{\pi^*}(s_t) = \max_{a_t \in A} \left\{ r_t(s_t, a_t) + \delta \, \mathrm{E}_{s_t}^{\pi^*} \left[v_{t+1}^{\pi^*}(s_{t+1}) \,\middle|\, \mathcal{I}_t \right] \right\}$$

Backward induction algorithm

for $t = T, \ldots, 1$ **do**

  if t == T **then**

$$v_T^{\pi^*}(s_T) = \max_{a_T \in A}\Big\{ r_T(s_T, a_T) \Big\} \qquad \forall\, s_T \in S$$

  else

    Compute $v_t^{\pi^*}(s_t)$ for each $s_t \in S$ by

$$v_t^{\pi^*}(s_t) = \max_{a_t \in A}\Big\{ r_t(s_t, a_t) + \delta\, \mathrm{E}_{s_t}^{\pi}\Big[\, v_{t+1}^{\pi^*}(s_{t+1}) \Big|\, \mathcal{I}_t \,\Big] \Big\}$$

    and set

$$a_t^{\pi^*}(s_t) = \arg\max_{a_t \in A}\Big\{ r_t(s_t, a_t) + \delta\, \mathrm{E}_{s_t}^{\pi}\Big[\, v_{t+1}^{\pi^*}(s_{t+1}) \Big|\, \mathcal{I}_t \,\Big] \Big\}$$

  end if

end for

Calibration procedure

Data

$$\mathcal{D} = \{a_{it}, x_{it}, r_{it} : i = 1, \ldots, N; t = 1, \ldots, T_i\}$$

State variables

- $s_t = (x_t, \epsilon_t)$
  - x_t observed
  - ϵ_t unobserved

Procedures

- likelihood-based

$$\hat{\theta} \equiv \arg\max_{\theta \in \Theta} \prod_{i=1}^{N} \prod_{t=1}^{T_i} p_{it}(a_{it}, r_{it} \mid x_{it}, \theta)$$

- simulation-based

$$\hat{\theta} \equiv \arg\min_{\theta \in \Theta} (M_D - M_S(\theta))' W (M_D - M_S(\theta))$$

Example

Seminal paper

- 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. *Review of Economics and Statistics*, 76(4):648–672

Model of occupational choice

- 1,000 individuals starting at age 16
- life cycle histories
  - school attendance
  - occupation-specific work status
  - wages

Labor market

$$r_t(s_t, 1) = w_{1t} = \exp\{ \underbrace{\alpha_{10}}_{\text{endowment}} + \underbrace{\alpha_{11} g_t}_{\text{schooling}} + \underbrace{\alpha_{12} e_{1t} + \alpha_{13} e_{1t}^2}_{\text{own experience}}$$
$$+ \underbrace{\alpha_{14} e_{2t} + \alpha_{15} e_{2t}^2}_{\text{other experience}} + \underbrace{\epsilon_{1t}}_{\text{shock}} \}$$

Schooling

$$r_t(s_t, 3) = \underbrace{\beta_0}_{\text{taste}} - \underbrace{\beta_1 \mathbb{I}[g_t \geq 12]}_{\text{direct cost}} - \underbrace{\beta_2 \mathbb{I}[a_{t-1} \neq 3]}_{\text{reenrollment effort}} + \underbrace{\epsilon_{3t}}_{\text{shock}}$$

Home

$$r_t(s_t, 4) = \underbrace{\gamma_0}_{\text{taste}} + \underbrace{\epsilon_{4t}}_{\text{shock}}$$

State space

$$s_t = \{g_t, e_{1t}, e_{2t}, a_{t-1}, \epsilon_{1t}, \epsilon_{2t}, \epsilon_{3t}, \epsilon_{4t}\}$$

Transitions

- observed state variables

$$
\begin{aligned}
e_{1,t+1} &= e_{1t} + \mathbb{I}[a_t = 1] \\
e_{2,t+1} &= e_{2t} + \mathbb{I}[a_t = 2] \\
g_{t+1} &= g_t \;\; + \mathbb{I}[a_t = 3]
\end{aligned}
$$

- unobserved state variables

$$
\{\epsilon_{1t}, \epsilon_{2t}, \epsilon_{3t}, \epsilon_{4t}\} \sim N(0, \Sigma)
$$

Figure: Choices over the life cycle

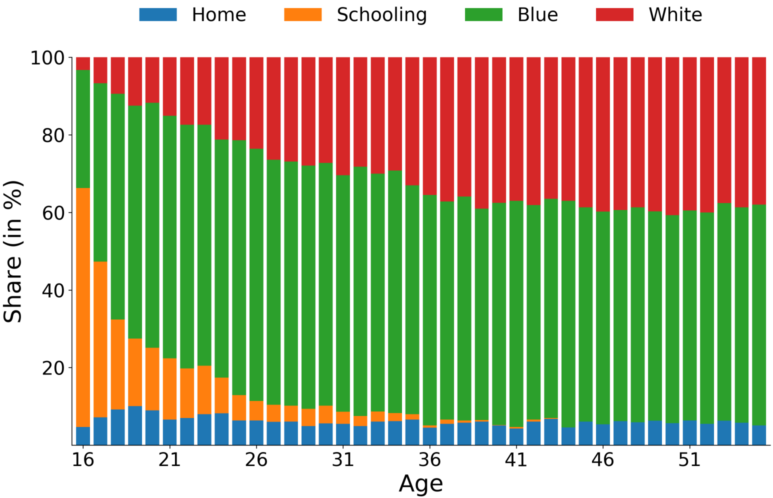
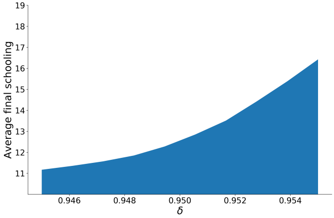
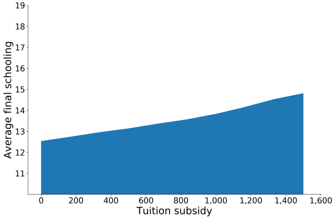


Figure: Economic mechanism and policy forecast



Time preference



Tuition subsidy

Research codes

respy

GitHub	`OpenSourceEconomics/respy`
Docs	`respy.readthedocs.io`

estimagic

GitHub	`OpenSourceEconomics/estimagic`
Docs	`estimagic.readthedocs.io`

Figure: Typical workflow

```
from estimagic.optimization.optimize import maximize
import respy as rp

# obtain model input
params, options, df = rp.get_example_model("kw_94_two")

# process model specification
crit_func = rp.get_crit_func(params, options, df)
simulate = rp.get_simulate_func(params, options)

# perform calibration
results, params_rslt = maximize(crit_func, params, "nlopt_bobyqa")

# conduct analysis
df_rslt = simulate(params_rslt)
```

Figure: Model specification

kw_94_two.csv

```
category,name,value,comment
delta,delta,0.95,discount factor
wage_a,constant,9.21,log of rental price
wage_a,exp_edu,0.04,return to an additional year of schooling
wage_a,exp_a,0.033,return to same sector experience
wage_a,exp_a_square,-0.0005,"return to same sector, quadratic experience"
wage_a,exp_b,0,return to other sector experience
wage_a,exp_b_square,0,"return to other sector, quadratic experience"
wage_b,constant,8.2,log of rental price
wage_b,exp_edu,0.08,return to an additional year of schooling
wage_b,exp_b,0.067,return to same sector experience
wage_b,exp_b_square,-0.001,"return to same sector, quadratic experience"
wage_b,exp_a,0.022,return to other sector experience
wage_b,exp_a_square,-0.0005,"return to other sector, quadratic experience"
nonpec_edu,constant,5000,constant reward for choosing education
nonpec_edu,at_least_twelve_exp_edu,-5000,"reward for going to college (tuition, etc.)"
nonpec_edu,not_edu_last_period,-15000,reward for going back to school
nonpec_home,constant,14500,constant reward of non-market alternative
shocks_sdcorr,sd_a,0.4,"Element 1,1 of standard-deviation/correlation matrix"
shocks_sdcorr,sd_b,0.5,"Element 2,2 of standard-deviation/correlation matrix"
shocks_sdcorr,sd_edu,6000,"Element 3,3 of standard-deviation/correlation matrix"
shocks_sdcorr,sd_home,6000,"Element 4,4 of standard-deviation/correlation matrix"
shocks_sdcorr,corr_b_a,0,"Element 2,1 of standard-deviation/correlation matrix"
shocks_sdcorr,corr_edu_a,0,"Element 3,1 of standard-deviation/correlation matrix"
shocks_sdcorr,corr_edu_b,0,"Element 3,2 of standard-deviation/correlation matrix"
shocks_sdcorr,corr_home_a,0,"Element 4,1 of standard-deviation/correlation matrix"
shocks_sdcorr,corr_home_b,0,"Element 4,2 of standard-deviation/correlation matrix"
shocks_sdcorr,corr_home_edu,0,"Element 4,3 of standard-deviation/correlation matrix"
lagged_choice_1_edu,probability,1,Probability that the first lagged choice is education
initial_exp_edu_10,probability,1,Probability that the initial level of education is 10
```

Parameterization

kw_94_two.yaml

```
estimation_draws: 200
estimation_seed: 500
estimation_tau: 500
interpolation_points: -1
n_periods: 40
simulation_agents: 1000
simulation_seed: 132
solution_draws: 500
solution_seed: 456
monte_carlo_sequence: random
core_state_space_filters:
  # In periods > 0, if agents accumulated experience only in one choice, lagged choice
  # cannot be different.
  - "period > 0 and exp_{i} == period and lagged_choice_1 != '{i}'"
  # In periods > 0, if agents always accumulated experience, lagged choice cannot be
  # non-experience choice.
  - "period > 0 and exp_a + exp_b + exp_edu == period and lagged_choice_1 == '{j}'"
  # In periods > 0, if agents accumulated no years of schooling, lagged choice cannot
  # be school.
  - "period > 0 and lagged_choice_1 == 'edu' and exp_edu == 0"
  # If experience in choice 0 and 1 are zero, lagged choice cannot be this choice.
  - "lagged_choice_1 == '{k}' and exp_{k} == 0"
  # In period 0, agents cannot choose occupation a or b or mil.
  - "period == 0 and lagged_choice_1 == '{k}'"
covariates:
  constant: "1"
  exp_a_square: exp_a ** 2
  exp_b_square: exp_b ** 2
  at_least_twelve_exp_edu: exp_edu >= 12
  not_edu_last_period: lagged_choice_1 != 'edu'
```

Options

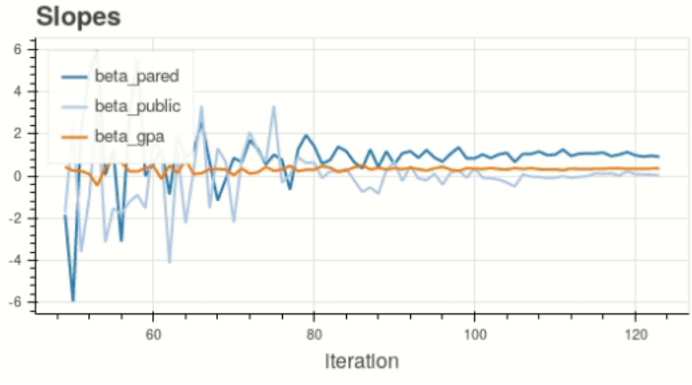


Figure: Dashboard

Roadmap

Improvements

- numerical integration
- global optimization
- function approximation
- high-performance computing

Extensions

- robust decision-making
- uncertainty quantification
- model validation
- nonstandard expectations

Join us!

GitHub	`http://bit.ly/ose-github`
Meetup	`http://bit.ly/ose-meetup`
Chat	`http://bit.ly/ose-zulip`

Appendix

Content

- Contact
- References

Contact

Philipp Eisenhauer

Mail	peisenha@uni-bonn.de
Web	`http://eisenhauer.io`
GitHub	`https://github.com/peisenha`

References

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. *Review of Economics and Statistics*, 76(4):648–672.