soepy Documentation

Release 1.0

Development Team

Jan 30, 2019

CONTENTS:

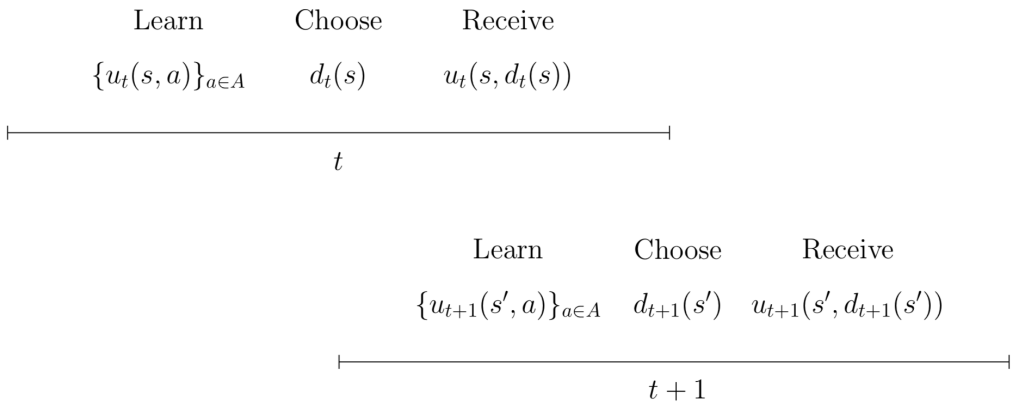
CHAPTER

ONE

ECONOMIC MODEL

This part of the documentation contains detailed information on the economic model. In particular, it discusses all assumptions on primitives we base our computational model on, e.g., preference structure, attitutes towards risk, beliefs regarding key variables, etc.

At time $t = 1, ..., T$ each individual observes the state of the economic environment $s \in S$ and chooses an action a from the set of admissible actions A. The decision has two consequences: an individual receives an immediate utility $u_t(s, a)$ and the economy evolves to a new state s'. The transition from s to s' is affected by the action. Individuals are forward-looking, thus they do not simply choose the alternative with the highest immediate utility. Instead, they take the future consequences of their current action into account.



The above figure depicts the timing of events in the model for two generic time periods. At the beginning of each period t an individual fully learns about the immediate utility of each alternative, chooses one of them, and receives its immediate utility. Then the state evolves from s to s' and the process is repeated in $t + 1$.

A decision rule d_t specifies the action at a particular time t for any possible state. A policy $\pi = (d_1, ..., d_T)$ provides the individual with a prescription for choosing an action in any possible future state. It is a sequence of decision rules and its implementation generates a sequence of utilities. The evolution of states over time is at least partly unknown as future utilities depend on, for example, shocks to preferences. Let X_t denote the random variable for the state at time t.

In this life cycle model, individuals make their decisions facing risk. Agents have rational expectations and thus rational choices are expressed by expected utility preferences. In a static setting, individuals in state :math:s rank each action :math:a according to the criterion formalized in the equation below:

$$v_{T-1}(s, a) = \int_S u_T(s', d_T(s'))\, dp_{T-1}(s, a)$$

In a dynamic setting, individuals maximize their total doscounted utility. A constant discount factor ensures dynamic consistency of preferences as the individual's future actions agree with the planned-for contingencies.

$$v_1^{\pi^*}(s) = \max_{\pi \in \Pi} E_s^{\pi} \left[\sum_{t=1}^{T} \delta^{t-1} u_t(X_t, d_t(X_t)) \right]$$

The above equation provides the formal representation of the individual's objective. Given an initial state s, individuals seek to implement the optimal policy π^* from the set of all possible policies Π that maximizes the expected total discounted utility over all T.

CHAPTER

TWO

MATHEMATICAL MODEL

This part of the documentation addresses the mathematical model we apply to the economic problem. In particular, it presents the details of the Markov decision process which we use to analyse agents' life-cycle employment decisions.

The life cycle model of human capital investment subject to the analysis at hand is set up as a standard MDP where there is a unique transition probability distribution $p_t(s, a)$ associated with each state and action.

Let the sequence of previous states and decisions up to time t be denoted by $h_t \equiv (s_1, a_1, ..., s_{t-1}, a_{t-1}, s_t)$ and $\mathcal{H}_T$ be the set of all possible histories over the T decision periods.

When making sequential decisions in this model, the task is to determine the optimal policy π^* with the largest expected total discounted utility $v_1^{\pi^*}$. In principle, this requires to evaluate the performance of all policies based on all possible sequences of utilities and the probability that each occurs. Fortunately, however, the multistage problem can be solved by a sequence of simpler inductively defined single-stage problems.

Let $v_t^\pi(s)$ denote the expected total discounted utility under π from period t onwards:

$$v_t^\pi(s) = E_s^\pi \left[\sum_{\tau=t}^{T} \delta^{\tau-t} u_\tau(X_\tau, d_\tau(X_\tau)) \right].$$

Then $v_1^\pi(s)$ can be determined for any policy by recursively evaluating the equation:

$$v_t^\pi(s) = u_t(s, d_t(s)) + \delta E_s^p \left[v_{t+1}^\pi(X_{t+1}) \right].$$

The above equation expresses the utility $v_t^\pi(s)$ of adopting policy π going forward as the sum of its immediate utility and all expected discounted future utilities.

The principle of optimality (see Bellman, 1957) allows to construct the optimal policy π^* by solving the optimality equations for all s and t in equation below recursively:

$$v_t^{\pi^*}(s) = \max_{a \in A} \left\{ u_t(s, a) + \delta E_s^p \left[v_{t+1}^{\pi^*}(X_{t+1}) \right] \right\}.$$

The value function $v_t^{\pi^*}$ is the expected discounted utility in t over the remaining time horizon assuming the optimal policy is implemented going forward.

The algorithm presented below allows to solve the MDP by a simple backward induction procedure. In the final period T, there is no future to take into account and so the optimal decision is simply to choose the alternative with the highest immediate utility in each state. With the results for the final period at hand, the other optimal decisions can be determined recursively as the calculation of their expected future utility is straightforward given the relevant transition probability distribution.

Algorithm 1 Backward Induction Algorithm for MDP

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

if t == T **then**

$$v_T^{\pi^*}(s) = \max_{a \in A} \left\{ u_T(s, a) \right\} \qquad \forall\, s \in S$$

else

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

$$v_t^{\pi^*}(s) = \max_{a \in A} \left\{ u_t(s, a) + \delta \mathrm{E}_s^p \left[v_{t+1}^{\pi^*}(X_{t+1}) \right] \right\}$$

and set

$$d_t^{\pi^*}(s) = \arg\max_{a \in A} \left\{ u_t(s, a) + \delta \mathrm{E}_s^p \left[v_{t+1}^{\pi^*}(X_{t+1}) \right] \right\}.$$

end if

end for

CHAPTER

THREE

COMPUTATIONAL MODEL

This part of the documentation specifies the functional forms, types of shocks and exogenous processes we bring together with the developed economic model to perform simulation and estimation. We discuss the functional form of the utility function, the laws of motion for the agents' budget constraints, the chosen formulation of the wage and the experience accumulation processes, etc. in greater detail.

3.1 Toy model

This section lays out the computational model corresponding to the current version of the code.

3.1.1 Structural equations

In the toy model, the individual's utility is given by:

$$u(c_t, l_t; \theta) = \frac{c_t^{\mu}}{\mu} exp \begin{cases} 0, & \text{if } l_t = O, \\ \theta_P, & \text{if } l_t = P, \\ \theta_F, & \text{if } l_t = F, \end{cases}$$

In assuming this form of utility, the toy model abstracts away from some components we would like to include in the enhanced model. First, the above utility function does not include any covariates and corresponding coefficients in $U(.)$ apart from the choice specific constant θ_l. Second, it abstracts from weighting consumption by an equivalence scale.

The budget constraint in the toy model is given by $c_t = h_t w_t$.

The wage equation is given by:

$$\begin{aligned} ln\, w_t^m &= \gamma_{s,0} + \gamma_{s,1} ln(e_t + 1) + \xi_t, \\ ln\, w_t &= ln\, w_t^m - \xi_t, \\ e_t &= (e_P * g_{sP} + e_F)(1 - \delta_s), \end{aligned}$$

where e_P and e_F measure the total years in part-time and full-time experience accumulated up to period t.

3.1.2 Transformatons

To make the calculation of the flow utilities and value functions more explicit we substitute the above equation in one another. As a result we arrive at a single term that represents $u(c_t, l_t, \theta)$.

First we substitute the experience term in the first line of the wage equation,

$$ln\, w_t^m = \gamma_{s,0} + \gamma_{s,1} ln[(e_P * g_{sP} + e_F)(1 - \delta_s) + 1] + \xi_t,$$

and take the exponent of both sides of the equation:

$$w_t^m = exp\, \gamma_{s,0} * \gamma_{s,1}[(e_P * g_{sP} + e_F)(1 - \delta_s) + 1] * exp\, \xi_t,$$

We then substitute the wage in the budget constraint:

$$c_t = h_t * \{exp\, \gamma_{s,0} * \gamma_{s,1}[(e_P * g_{sP} + e_F)(1 - \delta_s) + 1] * exp\, \xi_t\},$$

And we arrive at the final expression by substituting consumption in the utility function:

$$u(c_t, l_t; \theta) = \frac{h_t^\mu * \{exp\, \gamma_{s,0} * \gamma_{s,1}[(e_P * g_{sP} + e_F)(1 - \delta_s) + 1] * exp\, \xi_t\}^\mu}{\mu} *$$
$$* \; exp \begin{cases} 0, & \text{if } l_t = O, \\ \theta_P, & \text{if } l_t = P, \\ \theta_F, & \text{if } l_t = F, \end{cases}$$

Finally, the distribution of the error term is assumed to be multinomial normal with zero means and covariances. Standard diviations are set to 1, 2, and 2.5 for non-employment, the part-time, and the full-time wage respectively.

3.2 Enhanced model

The goal of development efforts is directed to gradually extending the toy model towards the enhanced model presented below.

3.2.1 Outline of model components

In this framework women are modelled between the age when they start working after having completed education and 60 years of age. They retire at the age of 60 and live for additional 10 years using their accumulated savings. No re-entry in education is possible in the model. Having completed education, in each period (year) of their life, women make consumption and labor supply choices. They choose between nonemployment (N), part-time (P), or full-time employment (F). Female workers face labor-market frictions.

In the first period of observation, each woman draws a random preference for work, consisting of a utility cost of part-time work (θ_P) and a utility-cost of full-time work (θ_F). The utility cost parameters, θ_F and θ_P, can take on two (or more) values each, i.e., there are two types: high - type I, and low - type II. The values of the type II coefficients are normalised to zero. Both parameter values, θ_F and θ_P for type I, as well as the frequency of type I individuals in the data are estimated alongside the other free parameters of the model.

The tax and welfare system is year specific reflecting actual real world changes. It defines disposable income for each employment option. Households are credit constrained, i.e., they cannot borrow. Finally, the following elements enter the computational model as exogenous processes: childbirth, marriage, divorce and the male wage process.

3.2.2 Structural equations

Instantaneous utility

The individuals' flow utility is given by:

$$u(c_t, l_t; \theta, Z_t) = \frac{(c_t/n_t)^\mu}{\mu} exp\{U(l_t, \theta, Z_t)\}$$

$$U(l_t, \theta, Z_t) = \begin{cases} 0, & \text{if } l_t = O, \\ \theta_l + Z_t' \alpha(l_t), & \text{if } l_t = P \text{ or } F, \end{cases}$$

where $\alpha(l_t) = \alpha_F + \alpha_P \cdot \mathbf{1}\ (l_t = P)$.

The CRRA utility depends on consumption per adult equivalent, female labor supply l, characteristics Z, and {diw preference for work θ}. Z can contain information on marital status, presence of children, their interaction, dummies for children in different age groups, an indicator wether or not the partner is working, etc. $U(.)$ of not working is normalised to zero; β is set to 0.98;

There are several implications of the choice of this particular form of the utility function. Given the above form, instantaneous utility is non-separable in consumption and leisure. Total (lifetime) utility is the sum of CRRA functions, i.e., it is additively separable intertemporaneously. μ is the curvature parameter that governs risk-aversion and the elasticity of intertemporal substitution. The choice of $\mu < 0$ means that the utility $u(.)$ is always negative (bounded by zero from above, i.e., for $c \to \infty$), and the higher the argument U in the exponential, the lower the overall utility. A positive utility, $U(.)$, for $l = P/F$ implies that working reduces the utility of consumption and that consumption and labor supply are complements.

Budget constraint

In a more involved case, the value function is maximised subject to the following budget constraint:

$$\begin{cases} a_{t+1} = (1+r)a_t + h_t w_t + m_t \tilde{h_t} \tilde{w_t} - T(l_t, X_t) - Q(t^k, h_t, \tilde{h_t}, m_t) - c_t, \\ a_{t+1} = \underline{a_s}, \end{cases}$$

with initial and terminal conditions $a_0 = 0$ and $a_{\tilde{t}+1} \geq 0$.

Notation is to be read as follows:

- r - risk free interest rate
- $(w, \tilde{w})$ - hourly rates of wife and husband
- $(h, \tilde{h})$ - working hours of wife and husband
- $\underline{a_s}$ - borrowing limit, which is either zero, or equal to the amount of student loan borrowed (negative number)
- T - tax and welfare transfer system, nonconcave, nonsmooth, and often discontinuous
- Q - childcare costs

In the current simplified version of the model, the budget constrained is given by $c_t = h_t w_t + m_t \tilde{h_t} \tilde{w_t} - T(l_t, X_t) - Q(t^k, h_t, \tilde{h_t}, m_t)$.

Female wage equation

The baseline specification of the female wage process is summarized in the following equations:

$$\begin{aligned} ln\, w_t^m &= \gamma_{s,0} + \gamma_{s,1} ln(e_t + 1) + \xi_t, \\ ln\, w_t &= ln\, w_t^m - \xi_t, \\ e_t &= e_{t-1}(1 - \delta_s) + g_s(l_{t-1}), \end{aligned}$$

where

- $ln\, w_t^m$ - observed hourly wage rate
- ξ_t - i.i.d. normal measurement error
- e_t - experience measured in years
- δ_s - per period depreciation rate
- g_s - per period rate of experience accumulation: $g_s(F) = 1$

3.2.3 To be implemented

The goal of this project is to develop a computational model similar to the one used in Blundell et. al. (2017). Features of the model that are still missing in the current implementation include:

- budget constraint:
  - male wages
  - tax function which varies by year
  - childcare costs
  - savings
- female wage equation:
  - individual AR1 peoductivity process
  - beliefs
- exogenous processes
  - male wage equation
  - prpbability of child arriving
  - probability of partner arriving
  - probability of partner leaving

Furthermore, we plan to include model features that go beyond the application in Blundell et. al. (2017):

- beliefs in the female wage equation
- labor market frictions

CHAPTER

FOUR

DEVELOPER NOTES

In this part, we provide detailed explanation on how certain elements of the computational model are implemented in the current version of the code. There can be multiple ways of casting a given computation model in code. Here we specify programming choices made when implementing certain aspects of the model and provide logic and specification.

4.1 pyth_create_state_space

4.1.1 Functionality

The function creates state space related objects given state space components.

4.1.2 Inputs

The inputs of the function come from the model specification. They consist of necessary information on the initial conditions and on the dimension of the state space components. In the soepy toy model, the initial condition are the completed years of education. The function requires the range of years of education completed from the (simulated) sample as input. It further requires the number of periods in the model and the number of discrete choices individuals face each period.

4.1.3 Outputs

The tate space related outputs of the function are collected in the array "args". "args" contains: * states_all: an array of tuples of state space components that constitute an admissible state space point each and taken together represent the entirity of admissible states across all periods in the model * states_number_period: a one dimensional array (i.e., a vector) that collects the number of admissible state space points for each period * mapping_state_index: an array which maps each state recorded in states all to a unique integer identifier * max_states_period: a scalar value corresponding to the maximum of the number of states per period over all periods

4.1.4 Code content

In the following the way the function is programmed to work is described in more detail.

First, we create array containers for the states_all and mapping_state_index arrays. In the loop executed further below as a part of this function, we want to populate these arrays with the state space components. The components of our toy model state space are:

- the time variable: period

- the initial conditions: education
- the history of choices: choice_lagged
- the accumulated experience in part-time employment: exp_p
- the accumulated experience in full-time employment: exp_f

Initially, the containers are filled with missing values. The important part in initializing the containers is to specify the correct dimensions.

Let us briefly discuss the dimension of the mapping_state_index array. Each dimension component corresponds to one state space component we would like to keep record of. In the funtion loop, a unique tuple of state space component values is assigned a unique integer identifier k. The dimensions correspond to:

- dimension 1 num_periods to the period number
- dimension 2 educ_range to the years of education
- dimension 3 num_choices to the choice of the individual
- dimension 4 num_periods to years of experience in part-time
- dimension 5 num_periods to years of experience in full-time

Let us briefly discuss the dimention of the states_all array:

- the 1st dimension is determined by the number of periods in our model
- the 2nd dimension is related to the maximum number of state space points ever feasible / possibly reachable in one of the periods.
- the 3rd dimension is equal to the number of remaining state space components (except period number) that we want to record: educ_years + educ_min, choice_lagged, exp_p, exp_f

We note that we set the size of dimension to equal to the highest possible number of state space points ever reachable in a period. This number is calculated as the maximum number of combinantions of the state space componenents given no restrictions.

Now we can loop through all admissible state space points and fill up the constructed arrays with necessary information.

In the following, we note how education is introduced as an initial condition in the loop:

If we want to record only admissible state space points, we have to account for the fact that individuals in the model start making labor supply choices only after they have completed education. As a reminder, we note that we model individuals from age 16 (legally binding mininum level of education) until age 60 (typical retirement entry age for women), i.e., for 45 periods, where the 1st period corresponds to age 16, the second period to age 17, etc. The current specification of starting values is equivalent to the observation / simulated reality that individuals in the sample have completed something between 10 and 14 years of edcuation. In our loop we want to take into account the fact that, in the first period at age 16, only individuals who have completed 10 years of education (assuming education for everyone starts at the age of 6) will be making a labor market choice between full-time (F), part-time (P), and non-employment (N). The remaining individuals are still in education, such that a state space point where years of education equal e.g. 11 and a labor market choice of e.g. part-time is observed is not admissible and should therefore not be recorded. This is ensured by the if clause "educ_years > period".

When starting the model in the way described above, we need to account for an additional state space point which corresponds to individuals who have just entered the model. Let us consider the very first period, period 0 in the loop, as an example. In this period, only individuals with 10 years of education enter the model and make their first labor supply choice. In order to later be able to compute the instataneous utility for this individuals, we need to record their state space components. As there is no lagged choice available to record, we force set the lagged choice component to zero. In particular, for states for which the condition (period == educ_years) is true, i.e., for individuals who have just entered the model, we record a state: (educ_years + educ_min, 0, 0, 0).

Finally, we briefly repeat what has been recorded in one of the main function output objects, the states_all array:

- educ_years + educ_min: in this example, values from 10 to 14
- choice_lagged: 0, 1, 2 corresponding to N, P, and F
- exp_p: part-time experience that can range from 0 to 9
- exp_f: full-time experience that can range from 0 to 9

Note: There is a difference to respy here. In respy, the loop in experience is one iteration longer, goes to num_periods + 1 instead of to num_periods.

4.2 pyth_backward_induction

4.2.1 Functionality

Solves the dynamic discrete choice model in a backward induction procedure, in which the error terms are integrated out in a Monte Carlo simulation procedure.

4.2.2 Inputs

In a final version of soepy, the pyth_create_state_space function is called before the backwardinduction procedure. The backward induction procedure needs the outputs of the pyth_create_state_space function as inputs. It further relies on multiple inputs from the model specification: num_periods, num_choices, educ_max, educ_min, educ_range, mu, delta,o ptim_paras, num_draws_emax, seed_emax, shocks_cov.

4.2.3 Outputs

The array periods_emax containing the highest value function among the choice specific value functions for the three labor market choices at each admissible state space point in each period of the model. The array is of dimension number periods by maximum number of admissible choices over all periods.

4.2.4 Code content

The individuals in our model solve their optimization problem by making a labor supply choice in every period. They choose the option that is associated with the highest value function. The value function for each of the 3 alternatives is the sum of the current period flow utility of choosing alternative j and a continuation value.

The flow utility includes the current period wage shock, which the individual becomes aware of in the begining of the period and includes in her calculations. To obtain an estimate of the continuation value the individual has to integrate out the distribution of the future shocks. In the model implementation, we perform numerical integration via a Monte Carlo simulation.

In the function, we generate draws from the error term distribution defined by the error term distribution paramters in the model specification, (in the current model spesification - shocks_cov). For each peiod we draw as many disturbances as num_draws_emax. These draws let us numerically integrate out the error term in a Monte Carlo simulation procedure. This is necessary for computing the continuation values and, ultimately, the value functions and the model's solution. Integating out the error term, represents the process in which individuals in the model form expectations about the future. Assuming rational expectations and a known error term distribution up to its parameters, individuals take the possible realization of the error terms into account by computing the expected continuation values over the distribution of the errors. For every period, we simmulate num_draws_emax draws from the error term distribution.

In the current formulation, we assume that the wage process is subject to additive measurement error. The disturbances for non-employment, the part-time, and the full-time wage are normally distributed with mean zero. The spesification assumes no serial and also no contemporaneous correlation across the error terms.

Before we begin the backward iteration procedure, we initialize the container for the final result. It is the array periods_emax with dimensions number of periods by maximum number of admissible states (max_states_period).

The backward induction loop calls several functions defined separately. As the name suggest, we loop backwards:

- construct_covariates: determines the education level given the state space component years of education
- construct_emax: integrates out the error term by averaging the value function values over the drawn realization of the error term. In this, the value function is computed using furter nested functions.
- calculate_utilities: calculates the flow utility using the systematic wage (wage without error), the period wage (systematic wage and error), the consumption utility (first part of the utility function), and total utility (consumption utility and U(.)).
- calculate_continuation_values: recursively obtains a continuation value given period and state. The function selects the relevant element of the periods_emax array given period number and state space components. This is possible since the whole loop is executed backwards.

Note concerning calculate_consumption_utilities:

In the toy model, consumption in any period is zero if the individual chooses non-employment. This is the case because consumption is simply the product of the period wage and the hours worked, and the hours worked in the case of non-employment are equal to zero. The calculation of the 1st part of the utility function related to consumption involves taking period consumption to the negative pover mu. In the programm, this would yield -inf. To avoid this complication, here the consumption utility of non-employment is normalized to zero.

4.3 pyth_simulate

4.3.1 Functionality

Simulate a data set given model spesification.

4.3.2 Inputs

In a final version of soepy, the functions pyth_create_state_space and pyth_backward_induction are called before the backward_induction procedure. The simulation procedure requires the outputs of the former functions as inputs. It further relies on multiple inputs from the model specification. Most are the same as the ones required by the backward induction procedure: num_periods, num_choices, educ_max, educ_min, educ_range, mu, delta,o ptim_paras, num_draws_emax, seed_emax, shocks_cov. In addition, num_agents_sim and seed_sim are also required.

4.3.3 Outputs

A pandas data frame with infromation about agents experiences in each period such as the choice, wage, flow utility, etc.

First, we need to genrate draws of the error term distribution. We note that this set of draws is different to the one used in the backward induction procedure. In the simulation, we need another set of draws to represent our simulated reality. In our model, at the beginning of every new period, individuals are hit by a productivity shock. They are aware of the realization of the shock when making their labor supply choice for the period. For every period, we simmulate num_agents_sim draws of the error term distribution.

Next, we need to simulate a sample of initial conditions. In this example, we need to assing a value for the years of education to every agent whose life-cycle we want to simulate.

Finally, we loop forward through all agents and all periods to generate agent's experiences in the model and record these in a data frame.

We note that the sumulation procedure uses a slightly modified verion of the construct covariates function than the backward iteration procedure. During backward iteration, the education level is determined for all possible years of education depending on which state space point has currently been reached by the loop. During simulation, the education level needs to be determined according the simulated initial condition for the individual currently reached by the loop. We further note that the simulation procedure does need all subfunctions related to the calculation of the instantaneous utility, but it does not need the construction of the expected maximum (construct_emax) as a subfunction. The model's solution has already been computed in the backward iteration procedure. During simulation, we can access the relevant continuation value recorded in the periods_emax array given the current period number and state space components determined by the agent's experiences so far.