1 Model with persistence in unobservables ("unobserved state variables")

Up to now, we consider models satisfying Rust's "conditional independence" assumption on the ϵ's. This rules out persistence in unobservables, which can be economically meaningful.

1.1 Example: Pakes (1986) patent renewal model

Pakes (1986). How much are patents worth? This question is important because it inform public policy as to optimal patent length and design. Are patents a sufficient means of rewarding innovation?

- Q_A: value of patent at age A
- Goal of paper is to estimate Q_A using data on their renewal. Q_A is inferred from patent renewal process via a *structural model* of optimal patent renewal behavior.
- Treat patent renewal system as exogenous (only in Europe)
- For $a = 1, \dots, L$, a patent can be renewed by paying the fee c_a
- Timing:
  - At age $a = 1$, patent holder obtains period revenue r_1 from patent
  - Decides whether or not to renew. If renew, then pay c_1, and proceed to age $a = 2$.
  - At age $a = 2$, patent holder obtains period revenue r_2 from patent
  - Decides whether or not to renew. If renew, then pay c_2, and proceed to age $a = 3$. And so on...

- Let V_a denote the value of patent at age a.

$$V_a \equiv \max_{t\in[a,L]} \sum_{a'=1}^{L-a} \beta^{a'} R(a+a'), \text{ where}$$
$$R(a) = \begin{cases} r_a - c_a & \text{if } t \geq a \quad \text{(when you hold onto patent)} \\ 0 & \text{if } t < a \quad \text{(after you allow patent to expire)} \end{cases} \tag{1}$$

  t above denotes the age at which the agent allows the patent to expire, and is the agent's choice variable in this problem. This type of problem is called an "optimal stopping" problem.

  $R(a)$ denotes the profits from the patent during the a-th year. The sequence $R(1), R(2), \ldots$ is a "controlled" stochastic process: it is inherently random, but also affect by agent's actions.

- Since the maximal age L is finite, this is a finite-horizon (nonstationary) dynamic optimization problem.

- The state variable of this DO problem is r_a, the single-period revenue.

- Finite-horizon DO problems are solved via *backward recursion.* The value functions $\{V_1(\cdot), V_2(\cdot), \ldots, V_a(\cdot), \ldots, V_L(\cdot)\}$ are recursively related via Bellman's equation:

$$V_a(r_a) = \max\left\{0,\ \underbrace{r_a + \beta E\left[V_{a+1}(r_{a+1})|\Omega_a\right]}_{\equiv Q_a \text{ value of age } a \text{ patent}} - c_a\right\}.$$

  RHS means you will choose to renew the patent iff $Q_a - c_a > 0$.

  Ω_a: history up to age a, $= \{r_1, r_2, \ldots, r_a\}$

  Expectation is over $r_{a+1}|\Omega_a$. The sequence of conditional distributions $G_a \equiv F(r_{a+1}|\Omega_a)$, $a = 1, 2, \ldots$, is an important component of the model specification. Pakes' assumptions are given in Eq. (7) of the paper:

$$r_{a+1} = \begin{cases} 0 & \text{with prob. } \exp(-\theta r_a) \\ \max(\delta r_a, z) & \text{with prob. } 1 - \exp(-\theta r_a) \end{cases} \tag{2}$$

  where density of z is $q_a = \frac{1}{\sigma_a} \exp\left[-(\gamma + z)/\sigma_a\right]$ and $\sigma_a = \phi^{a-1}\sigma,\ a = 1, \ldots, L-1$.

  δ, θ, γ, ϕ, and σ are the important structural parameters of the model.

- So break down maximization problem into period-by-period problem, where each period agent decides whether or not to incur cost c_a and gain the value of the patent $Q_a = r_a +$ "option value". Option value captures the value in keeping patent alive in order to make a choice tomorrow.

Implications of model seen graphically:

- Drop out at age a if $c_a > Q_a$
- Optimal decision characterized by "cutoff points":

$$Q_a > c_a \Leftrightarrow r_a > \bar{r}_a$$

  This feature due to assumption A3.3, which ensures that Q_a is increasing in r_a (so that Q_a and c_a only cross once)

- The sequence of cutoff points $\bar{r}_a < \bar{r}_{a+1} < \cdots < \bar{r}_{L-1}$: ensured by assumption A3.4.

1.2 Estimation: likelihood function and simulation

In this section, we consider estimation of the Pakes patent renewal model. For ease of comparison with the Rust model from before, we use ϵ to denote the unobserved state variable, which in the Pakes model correpsonds to the patent revenue r_t. Furthermore, we use i_t to denote the choice (control) variable; it is equal to zero if patent is renewed, and equal to one if patent expires.

Consider one patent. Let $\tilde{T}$ denote the age at which the patent is allowed to expire. Due to the setup of the problem, $\tilde{T} \leq L$, the maximal age of the patent. Let $T = \min(L-1, \tilde{T})$ denote the number of periods in which the agent makes an active patent renewal decision. We model the agent's decisions in periods $t = 1, \ldots, T$.

ϵ evolves as a first-order Markov process, evolving according to: $F(\epsilon'|\epsilon)$. We denote the (age-specific) policy function by $i_t^*(\epsilon)$.

Now the likelihood function for this patent is:

$$\begin{aligned} & l\left(i_t, \dots, i_T | \epsilon_0, i_0; \theta\right) \\ = & \prod_{t=1}^{T} Prob\left(i_t | i_0, \dots, i_{t-1}; \epsilon_0, \theta\right) \end{aligned} \tag{3}$$

Note that because of the serially correlated ϵ's, there is still dependence between (say) i_t and i_{t-2}, even after conditioning on (i_{t-1}): compare the likelihood function in the Rust lecture notes and Eq. (3). In other words, the *joint process* $\{i_t, \epsilon_t\}$ is first-order Markov, but not the *marginal* process $\{i_t\}$ is not first-order Markov Also, because of serially correlation in the ϵ's, the $Prob\left(i_t | i_0, \dots, i_{t-1}; \theta\right)$ no longer has a closed form. Thus, we consider simulating the likelihood function.

Note that simulation is part of the "outer loop" of nested fixed point estimation routine. So at the point when we simulate, we already know the policy functions $i_t^*(\epsilon; \theta)$. (How would you compute this?)

1.3 "Crude" frequency simulator: naive approach

In this section, we describe a naive approach to simulating the likelihood of this model. *This is not something we want to do in practice, but we describe it here in order to contrast it with the particle-filtering (importance sampling) approach, which we describe in the next section.*

For simulation purposes, it is most convenient to go back to the full likelihood function (the first line of Eq. (3):

$$l(i_1, \dots, i_T | i_0, \epsilon_0; \theta).$$

Note that because the ϵ's are serially correlated, we also need to condition on an initial value ϵ_0 (which, for simplicity, we assume to be known). Pakes does something slightly more complicated– he assumes that the *distribution* of ϵ_0 is known.

Because i is discrete, the likelihood is the joint probability

$$Pr(i_t^*(\epsilon_t; \theta) = i_t, \ \forall t = 1, \dots, T)$$

where the i_t's denote the observed sequence of choices. The probability is taken over the distribution of $(\epsilon_1, \dots, \epsilon_T | \epsilon_0)$.

Let $F(\epsilon_{t+1}|\epsilon_t;\theta)$. Then the above probability can be expressed as the integral:

$$\int \cdots \int \prod_t \mathbf{1}(i_t^*(\epsilon_t;\theta) = i_t) \prod_t dF(\epsilon_t|\epsilon_{t-1};\theta).$$

We can simulate this integral by drawing sequences of (ϵ_t). For each simulation draw $s = 1, \ldots, S$, we take as initial values i_0, ϵ_0. Then:

- Generate (ϵ_1^s, i_1^s):
    1. Generate $\epsilon_1^s \sim F(\epsilon_1|\epsilon_0)$
    2. Compute $i_1^s = i_1^*(\epsilon_1^s;\theta)$
- Generate (ϵ_2^s, i_2^s):
    1. Generate $\epsilon_2^s \sim F(\epsilon_2|\epsilon_1^s)$
    2. Subsequently compute $i_2^s = i_2^*(\epsilon_2^s;\theta)$

... and so on, up to (ϵ_T^s, i_T^s)

Then, for the case where (i, x) are both discrete (which is the case in Rust's paper), we can approximate

$$l\left(i_t, \ldots, i_T|\epsilon_0, i_0;\theta\right) \approx \frac{1}{S}\sum_s \prod_{t=1}^T \mathbf{1}(i_t^s = i_t).$$

That is, the simulated likelihood is the frequency of the simulated sequences which match the observed sequence.

This is a "crude" frequency simulator. Clearly, if T is long, or S is modest, the simulated likelihood is likely to be zero. What is commonly done in practice is to smooth the indicator functions in this simulated likelihood.

1.4 Importance sampling approach: Particle filtering

Another approach is to employ importance sampling in simulating the likelihood function. This is not straightforward, given the across-time dependence between

(i_t, ϵ_t). Here, we consider a new simulation approach, called *particle filtering.* It is a recursive approach to simulate dependent sequences of random variables. The presentation here draws from Fernandez-Villaverde and Rubio-Ramirez (2007) (see also Flury and Shephard (2008)). This is also called "non-Gaussian Kalman filtering".

We need to introduce some notation, and be more specific about features of the model. Let:

- $y_t \equiv \{i_t\}$. $y^t \equiv \{y_t, \ldots, y_t\}$. These are the *observed sequences* in the data.
- Evolution of utility shocks: $\epsilon_t | \epsilon_{t-1} \sim f(\epsilon' | \epsilon)$. (Ignore dependence of distribution of ϵ on age t for convenience.)
- As before, the policy function is $i_t = i^*(\epsilon_t)$.
- Let $\epsilon^t \equiv \{\epsilon_1, \ldots, \epsilon_t\}$.
- The initial values y_0 and ϵ_0 are known.

Go back to the factorized likelihood:

$$\begin{aligned} l(y^T | y_0, \epsilon_0) &= \prod_{t=1}^{T} l(y_t | y^{t-1}, y_0, \epsilon_0) \\ &= \prod_t \int l(y_t | \epsilon^t, y^{t-1}) p(\epsilon^t | y^{t-1}) d\epsilon^t \\ &\approx \prod_t \frac{1}{S} \sum_s l(y_t | \epsilon^{t|t-1,s}, y^{t-1}) \end{aligned} \tag{4}$$

where in the second to last line, we omit conditioning on (y_0, ϵ_0) for convenience. In the last line, $\epsilon^{t|t-1,s}$ denotes simulated draws of ϵ^t from $p(\epsilon^t | y^{t-1})$.

Consider the two terms in the second to last line:

- The first term $l(y_t | \epsilon^t, y^{t-1})$:

$$\begin{aligned} l(y_t | \epsilon^t, y^{t-1}) &= p(i_t | \epsilon^t, y^{t-1}) \\ &= p(i_t | \epsilon_t) = \mathbf{1}(i(\epsilon_t) = i_t). \end{aligned} \tag{5}$$

  Clearly, this term can be explicitly calculated, for a given value of ϵ_t.

- The second term $p(\epsilon^t|y^{t-1})$ is, generally, not obtainable in closed form. So numerical integration not feasible. The particle filtering algorithm permits us to draw sequences of ϵ^t from $p(\epsilon^t|y^{t-1})$, for every period t. Hence, the second-to-last line of (4) can be approximated by simulation, as shown in the last line.

Particle filtering proposes a recursive approach to draw sequences from $p(\epsilon^t|y^{t-1})$, for every t. Easiest way to proceed is just to describe the algorithm.

First period, $t = 1$: In order to simulate the integral corresponding to the first period, we need to draw from $p(\epsilon^1|y^0, \epsilon_0)$. This is easy. We draw $\left\{\epsilon^{1|0,s}\right\}_{s=1}^{S}$, according to $f(\epsilon'|\epsilon_0)$. The notation $\epsilon^{1|0,s}$ makes explicitly that the ϵ is a draw from $p(\epsilon^1|y^0, \epsilon_0)$. Using these S draws, we can evaluate the simulated likelihood for period 1, in Eq. (4). We are done for period $t = 1$.

Second period, $t = 2$: We need to draw from $p(\epsilon^2|y^1)$. Factorize this as:

$$p(\epsilon^2|y^1) = p(\epsilon^1|y^1) \cdot p(\epsilon_2|\epsilon^1). \tag{6}$$

Recall our notation that $\epsilon^2 \equiv \{\epsilon_1, \epsilon_2\}$. Consider simulating from each term separately:

- Getting a draw from $p(\epsilon^1|y^1)$, given that we already have draws $\left\{\epsilon^{1|0,s}\right\}$ from $p(\epsilon^1|y_0)$, from the previous period $t = 1$, is the heart of particle filtering.

  We use the principle of importance sampling: by Bayes' Rule,

  $$p(\epsilon^1|y^1) \propto p(y_1|\epsilon^1, y^0) \cdot p(\epsilon^1|y^0)$$

  Hence, if our desired sampling density is $p(\epsilon^1|y^1)$, but we actually have draws $\left\{\epsilon^{1|0,s}\right\}$ from $p(\epsilon^1|y^0)$, then the importance sampling weight for the draw $\epsilon^{1|0,s}$ is proportional to

  $$\tau_1^s \equiv p(y_1|\epsilon^{1|0,s}, y^0).$$

  Note that this coincides with the likelihood contribution for period 1, evaluated at the shock $\epsilon^{1|0,s}$.

  The SIR (Sampling/Importance Resampling) algorithm in Rubin (1988) proposes that, making S draws with replacement from the samples $\left\{\epsilon^{1|0,s}\right\}_{s=1}^{S}$, using weights proportional to τ_1^s, yields draws from the desired density $p(\epsilon^1|y^1)$, which we denote $\{\epsilon^{1,s}\}_{s=1}^{S}$. This is the **filtering** step.

- For the second term in Eq. (6): we simply draw one ϵ_2^s from $f(\epsilon'|\epsilon^{1,s})$, for each draw $\epsilon^{1,s}$ from the filtering step. This is the **prediction** step.

By combining the draws from these two terms, we have $\left\{\epsilon^{2|1,s}\right\}_{s=1}^S$, which is S drawn sequences from $p(\epsilon^2|y^1)$. Using these S draws, we can evaluate the simulated likelihood for period 2, in Eq. (4).

Third period, $t=3$: start again by factoring

$$p(\epsilon^3|y^2) = p(\epsilon^2|y^2) \cdot p(\epsilon_3|\epsilon^2). \tag{7}$$

As above, drawing from the first term requires filtering the draws $\left\{\epsilon^{2|1,s}\right\}_{s=1}^S$, from the previous period $t=2$, to obtain draws $\{\epsilon^{2,s}\}_{s=1}^S$. Given these draws, draw $\epsilon_3^s \sim f(\epsilon'|\epsilon^{2,s})$ for each s.

And so on. By the last period $t=T$, you have

$$\left\{\left\{\epsilon^{t|t-1,s}\right\}_{s=1}^S\right\}_{t=1}^T.$$

Hence, the factorized likelihood in Eq. (4) can be approximated by simulation as

$$\prod_t \frac{1}{S}\sum_s l(y_t|\epsilon^{t|t-1,s}, y^{t-1}).$$

As noted above, the likelihood term $l(y_t|\epsilon^{t|t-1,s}, y^{t-1})$ coincides with the simulation weight τ_t^s. Hence, the simulated likelihood can also be constructed as

$$\log l(y^T|y_0, \epsilon_0) = \sum_t \log\left\{\frac{1}{S}\sum_s \tau_t^s\right\}.$$

Summary of particle filter simulator:

1. Start by drawing $\left\{\epsilon^{1|0,s}\right\}_{s=1}^S$ from $p(\epsilon^1|y^0, \epsilon_0)$.
2. In period t, we start with $\left\{\epsilon^{t-1|t-2,s}\right\}_{s=1}^S$, draws from $p(\epsilon^{t-1}|y^{t-2}, \epsilon_0)$.
   (a) **Filter step:** Calculate proportion weights $\tau_{t-1}^s \equiv p(y_{t-1}|\epsilon^{t-1|t-2,s}, y^{t-2})$ using Eq. (5). Draw $\left\{\epsilon^{t-1|t-1,s}\right\}_{s=1}^S$ by resampling from $\left\{\epsilon^{t-1|t-2,s}\right\}_{s=1}^S$ with weights τ_{t-1}^s.

(b) **Prediction step:** Draw ϵ_t^s from $p(\epsilon_t|\epsilon^{t-1|t-1,s})$, for $s = 1, \dots, S$. Combine to get $\left\{\epsilon^{t|t-1,s}\right\}_{s=1}^{S}$.

3. Set $t = t + 1$, and go back to step 2. Stop when $t = T + 1$.

Note the difference between this recursive simulator, and the crude simulator described previously. The crude simulator draws S sequences, and essentially assigns zero weight to all sequences which do not match the observed sequence in the data. In contrast, in particle filtering, in each period t, we just keep sequences where predicted choices match observed choice *that period*. This will lead to more accurate evaluation of the likelihood. Note that S should be large enough (relative to the sequence length T) so that the filtering step does not end up assigning almost all weight to one particular sequence $\epsilon^{t|t-1,s}$ in any period t.

References

Fernandez-Villaverde, J., and J. Rubio-Ramirez (2007): "Estimating Macroeconomic Models: A Likelihood Approach," *Review of Economic Studies*, 74, 1059–1087.

Flury, T., and N. Shephard (2008): "Bayesian inference based only on simulated likelihood: particle filter analysis of dynamic economic models," manuscript, Oxford University.

Pakes, A. (1986): "Patents as Options: Some Estimates of the Value of Holding European Patent Stocks," *Econometrica*, 54(4), 755–84.

Rubin, D. (1988): "Using the SIR Algorithm to Simulate Posterior Distributions," in *Bayesian Statistics 3*, ed. by J. Bernardo, M. DeGroot, D. Lindley, and A. Smith. Oxford University Press.