ECON 210: Section 7

Marcelo Sena

Plan for today

- Recap on consumption-savings with uncertainty
  - finite and markovian states
- Tauchen method to approximate $AR(1)$
- Consumption-savings with uncertainty (**?**)
- Coding
- Simulation of time-series:
  - AR(1)
  - Markov Chains

Consumption-Savings with Uncertainty

Recap

- Sequential problem with uncertainty

$$\max_{\{c_t(S^t), b_t(S^t)\}_{t=0}^{\infty}} \sum_{t=0}^{\infty} \sum_{S^t} \beta^t P\left(S^t\right) u\left(c_t\left(S^t\right)\right) \tag{1}$$

$$c_t\left(S^t\right) + R^{-1} b_t\left(S^t\right) = b_{t-1}\left(S^{t-1}\right) + y_t\left(S^t\right) \tag{2}$$

$$c_t\left(S^t\right) \geq 0, b_t\left(S^t\right) \geq \underline{b} \tag{3}$$

- Assume income depends only on today's state + S_t is markovian
- Bellman equation is:

$$V(b,s) = \max_{c,b'} u(c) + \beta \sum_{s'} P\left(s'|s\right) V\left(b', s'\right) \tag{4}$$

$$c + R^{-1} b' = b + y(s), c \geq 0, b' \geq \underline{b} \tag{5}$$

- New object: $P(s'|s)$

Tauchen Method

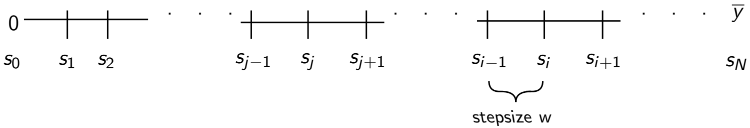
- The framework we introduced allows us to model uncertainty represented by finite states
- But in reality, the state space of uncertainty can actually be continuous
- Example: income can be \$203.43, \$12.41,...
- i.e. data from income is not discretized as in the model
- One possible model for income process is AR(1):

$$y_t = \lambda y_{t-1} + \sigma_\epsilon \varepsilon_t,\ \epsilon_t \sim N(0,1)\, iid \tag{6}$$

- How to translate this to a finite state Markov chain?
  - $\rightarrow$ Tauchen Method

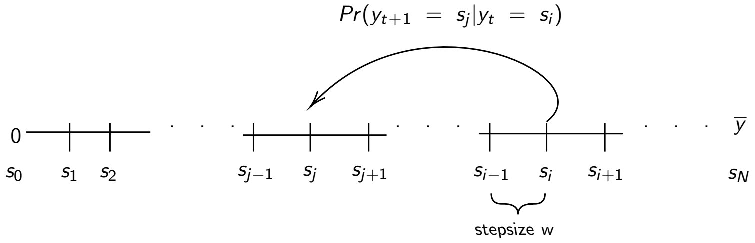
Tauchen Method

- Broad picture: discretize state space and say that the realized state is one the closest to continuous-variable



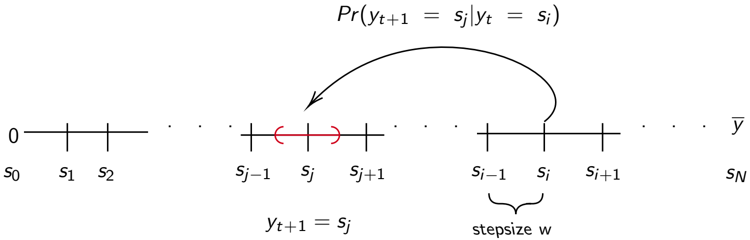
Tauchen Method

- Broad picture: discretize state space and say that the realized state is one the closest to continuous-variable



Tauchen Method

- Broad picture: discretize state space and say that the realized state is one the closest to continuous-variable



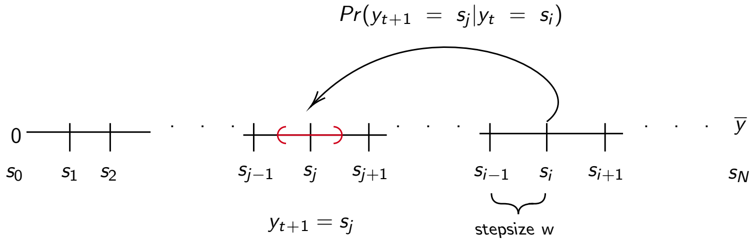
$$y_{t+1} = s_j$$

$$\Updownarrow$$

$$y_{t+1} \in \left(s_{j-1} + \frac{w}{2}, s_j + \frac{w}{2}\right)$$

Tauchen Method

- Broad picture: discretize state space and say that the realized state is one the closest to continuous-variable



$$\Updownarrow$$

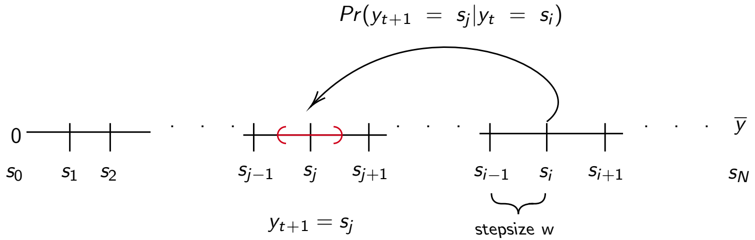
$$y_{t+1} \in \left(s_{j-1} + \frac{w}{2}, s_j + \frac{w}{2}\right)$$

- Use the fact that y_t is $AR(1)$ to calculate

$$Pr\left(y_{t+1} \in \left(s_{j-1} + \frac{w}{2}, s_j + \frac{w}{2}\right)\right)$$

Tauchen Method

- Broad picture: discretize state space and say that the realized state is one the closest to continuous-variable



$$y_{t+1} = s_j$$

$$\Updownarrow$$

$$y_{t+1} \in \left(s_{j-1} + \frac{w}{2}, s_j + \frac{w}{2}\right)$$

- Use the fact that y_t is $AR(1)$ to calculate

$$Pr\left(y_{t+1} \in \left(s_{j-1} + \frac{w}{2}, s_j + \frac{w}{2}\right)\right)$$

- What to do at the boundaries? Use only half the interval.

Tauchen Method

- Let $[\Pi_{ij}]$ be the transition matrix
- We set:

$$
\begin{aligned}
\Pi_{ij} &= Pr\left[y_{t+1} \ \in \left(s_{j-1} + \frac{w}{2}, s_j + \frac{w}{2}\right)\right] && (7)\\
&= Pr\left[s_{j-1} + \frac{w}{2} < \phi y_t + \sigma \varepsilon_{t+1} < s_j + \frac{w}{2}\right] && (8)\\
&= Pr\left[\frac{s_{j-1} + \frac{w}{2} - \phi y_t}{\sigma} < \varepsilon_{t+1} < \frac{s_j + \frac{w}{2} - \phi y_t}{\sigma}\right] && (9)\\
&= \Phi\left(\frac{s_j + \frac{w}{2} - \phi y_t}{\sigma}\right) - \Phi\left(\frac{s_{j-1} + \frac{w}{2} - \phi y_t}{\sigma}\right) && (10)\\
&= \Phi\left(\frac{s_j + \frac{w}{2} - \phi s_i}{\sigma}\right) - \Phi\left(\frac{s_{j-1} + \frac{w}{2} - \phi s_i}{\sigma}\right) && (11)
\end{aligned}
$$

where Φ is the cumulative distribution function of the standard normal

Tauchen Method

- At the boundaries:

$$\Pi_{i1} = \Phi\left(\frac{s_0 + \frac{w}{2} - \phi s_i}{\sigma}\right) \tag{12}$$

$$\Pi_{iN} = 1 - \Phi\left(\frac{s_N - \frac{w}{2} - \phi s_i}{\sigma}\right) \tag{13}$$

- How to set s_0 and s_N?
  - Support of y_t is $(-\infty, \infty)$.
  - Typically truncate at 3 standard deviations.
  - Hence,

$$s_0 = -m \times \sqrt{\frac{\sigma^2}{1-\phi^2}} \tag{14}$$

$$s_N = m \times \sqrt{\frac{\sigma^2}{1-\phi^2}} \tag{15}$$

$$m = 3 \tag{16}$$

Files

- `markovappr_section.m`: Tauchen for `Matlab`
- `markovsimul_section.m`: simulates markov chain for `Matlab`
- `tauchen.py`: Tauchen for `Python`
- `simulate_mc.py`: simulates markov chain for `Python`

Consumption-Savings with Uncertainty

Recap

- Sequential problem with uncertainty

$$\max_{\{c_t(S^t),b_t(S^t)\}_{t=0}^{\infty}} \sum_{t=0}^{\infty} \sum_{S^t} \beta^t P\left(S^t\right) u\left(c_t\left(S^t\right)\right) \tag{17}$$

$$c_t\left(S^t\right) + R^{-1} b_t\left(S^t\right) = b_{t-1}\left(S^{t-1}\right) + y_t\left(S^t\right) \tag{18}$$

$$c_t\left(S^t\right) \geq 0, b_t\left(S^t\right) \geq \underline{b} \tag{19}$$

- Assume income depends only on today's state + S_t is markovian
- Bellman equation is:

$$V(b,s) = \max_{c,b'} u(c) + \beta E_{s'}\left[V\left(b',s'\right)|s\right] \tag{20}$$

$$c + R^{-1} b' = b + y(s), c \geq 0, b' \geq \underline{b} \tag{21}$$

Consumption-Savings with Uncertainty

- Once we discretize, we have

$$V(b,s) = \max_{c,b'} u(c) + \beta \sum_{s'} P\left(s'|s\right) V\left(b', s'\right) \tag{22}$$

$$c + R^{-1} b' = b + y(s), c \geq 0, b' \geq \underline{b} \tag{23}$$

- Writing with cash-in-hand $w_t := b_t + y_t$

$$V(w,s) = \max_{b'} u\left(w - \frac{b'}{R}\right) + \beta \sum_{s'} P\left(s'|s\right) V\left(w', s'\right) \tag{24}$$

$$w' = b' + y(s'), c = w - \frac{b'}{R} \geq 0, b' \geq \underline{b} \tag{25}$$

Consumption-Savings with Uncertainty

Solution Algorithm

1. Discretize income grid and construct Markov matrix to approximate $AR(1)$ process for income.
2. Define grids for cash-on-hand and bond holdings.
3. Guess a value function $V^{(0)}(w, s)$.
4. Updating the value function:
   i) Fix a point in the state space (say (w_i, s_j)).
   ii) For each possible savings b_k, compute the right-hand side of the Bellman equation

$$T_{kij} = u\left(w_i - \frac{b_k'}{R}\right) + \beta \sum_{s'} \left[V^{(0)}\left(w', s'\right) \times P\left(s' \mid s_j\right)\right] \quad (26)$$

$$w' = b_k' + y(s') \quad (27)$$

5. Set $V^{(1)}(w_i, s_j) = \max_k T_{kij}$.
6. Having done this for all i, j, compute distance between new and guessed value functions $d = ||V^{(1)} - V^{(0)}||$.
7. If d is smaller than some pre specified threshold, claim convergence. If not, iterate with $V^{(1)}$ replacing $V^{(0)}$.

Consumption-Savings with Uncertainty

Solution Algorithm

Some reminders and points to take care:

- Note $w' = b'_k + y'$ might not be in original grid $\rightarrow$ find closest point
- Impose non-negativity constraint of consumption as usual
- Careful with bounds on the grids, check results are insensitive to these
- Remember to also save policy functions
- This problem can take longer to run on a finer grid, so try to make the implementation efficient (`Numba` and parallel code!)

Consumption-Savings with Uncertainty

Simulation

1. Simulate the markov income for T periods, yielding a sequence $\{s_0, \dots, s_{T-1}\}$.
2. Set initial bond holdings b_{-1} and compute initial cash-on-hand $w_0 = s_0 + b_{-1}$.
3. Compute optimal consumption and bond holdings from policy functions

$$c_0 = c^{pol}(w_0, s_0),\ b_0 = b^{pol}(w_0, s_0) \tag{28}$$

4. Compute next period cash-on-hand $w_1 = s_1 + b_0$.
5. Iterate until time T.

Simulation of time-series

$AR(1)$

- We define the autoregressive of order 1 ($AR(1)$) process as

$$y_{t+1} = \phi y_t + \sigma \varepsilon_{t+1} \tag{29}$$
$$\varepsilon_t \sim N(0,1) \tag{30}$$

- Simulation:
  i) Set an initial value $y_0 = y$
  ii) Simulate ϵ from $N(0,1)$
  iii) Calculate y_1 from (**??**)
  iv) Iterate until the desired horizon (say T)
- Software will typically have pre-built functions to perform the simulation of normal random variables, but we can always implement our own random-number generator if we can generate uniform random numbers

Moving Average Processes

- We can generalize autoregressive processes to an autoregressive moving-average process, which allows unobservables/shocks to affect the variable of interest with lags

$$y_t = \phi y_{t-1} + \theta_0 u_t + \theta_1 u_{t-1} \tag{31}$$

- This is an ARMA(1,1) process, since u_{t-1} has a predictive effect on y_t
- Subject to some conditions, we can map moving-average processes to “pure” AR(p) processes (and vice-versa)
- More about this in the problem set and also in time-series econometrics

Simulation of time-series

Markov Chain

- Reference: **?**, Chapter 11
- Suppose $\{s_t\}$ is a Markov chain with states $\{s_1, ..., s_N\}$ and transition matrix $\Pi = [\Pi_{ij}]$
- Start with some value s_i at date 0. We want to assign values s_t for $t = 1, 2, \dots T$
  1. compute cumulative distribution of the Markov chain Π^c

$$\Pi_{ij}^c = \sum_{k=1}^{j} \Pi_{ik} = \Pr\left(s_t \leq s_k | s_{t-1} = s_i\right)$$

  2. set initial state and simulate T random numbers from a uniform distribution over $[0, 1]$: $\{p_t\}_{t=1}^{T}$
  3. assume Markov chain was in state $i = 1$, find the index j such that

$$\Pi_{i(j-1)}^c < p_t \leq \Pi_{ij}^c \text{ for } j \geq 2$$

then s_j is the state in period t. If $p_t \leq \Pi_{i1}^c$ then s_1 is the state in period t.

Simulation

Fixing the seed

- Sometimes we may want to generate the same historical realization many times, to compare different procedures over the same series
- For example: want to compare consumption paths when the agent is more or less patient
- To do so, we have to fix the seed of the random number generator (rng) in our programs
- In `Matlab` use `rng`, in `Python` use `np.random.default_rng`

Deaton, Angus, "Saving and Liquidity Constraints," *Econometrica*, 1991, *59* (5), 1221–1248.

Miao, Jianjun, *Economic dynamics in discrete time*, MIT press, 2020.