Computational Economics Lecture 3: Introduction to Dynamics

Min Fang

University of Florida

Spring 2025

Outline

1. **Motivation**
2. **States/Markov Chains**
3. **Distributions**
4. **Simulation**
5. **Assignment**
6. **Appendix**

Motivation

- Many economics models are naturally dynamics (at least with t and $t+1$)
- So numerically, we need to know how changes happen over time
  - A state S_t transit to a state S_{t+1}
  - A distribution D_t transit to a distribution D_{t+1}
- Some transitions are exogenous, either non-stochastic or stochastic
  - Non-stochastic: Deterministic rules, i.e., $x_{t+1} = x_t + 1$
  - Stochastic: i.e., Markov Chains, often used: AR(1)
- Some transitions are endogenous: i.e., $x_{t+1} = x_t + z_t$, where z_t is a control
- Finally, we often need to simulate decisions and distributions over time
- Today, we will cover them all briefly

States/Markov Chains

- State space is the basics: Chapter 24. Linear State Space Models
- Let S be a finite set with n elements $\{x_1, \ldots, x_n\}$
- The set S is called the *state space* and $x_1, \ldots, x_n$ are the *state values*
- Markov chains are one of the most useful classes of stochastic processes:
  - simple, flexible, and supported by many elegant theoretical results
  - valuable for building intuition about random dynamic models
  - central to quantitative modeling in their own right
- Finite Markov Chains: Chapter 23. Finite Markov Chains
- Continuous State Markov Chains: Chapter 18. Continuous State Markov Chains
- Review some of the theory of Markov chains
- Introduce some of the high-quality routines for working with Markov chains

Finite Markov Chains

- A Markov chain $\{X_t\}$ on S is a sequence of random variables on S with *Markov property*
- This means that, for any date t and any state $y \in S$,

$$\mathbb{P}\{X_{t+1} = y \mid X_t\} = \mathbb{P}\{X_{t+1} = y \mid X_t, X_{t-1}, \ldots\}$$

**Knowing the current state is enough to know probabilities for future states*

- The dynamics of a Markov chain are fully determined by the set of values

$$P(x, y) := \mathbb{P}\{X_{t+1} = y \mid X_t = x\} \qquad (x, y \in S)$$

- We can view P as a stochastic matrix where

$$P_{ij} = P(x_i, x_j) \qquad 1 \leq i, j \leq n$$

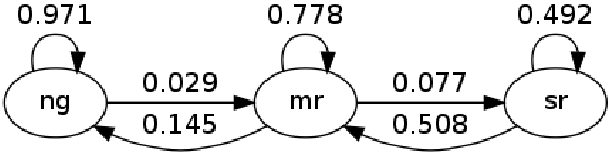
- Finally, to generate a Markov chain $\{X_t\}$, we just need S, P, and initial draw X_0

Finite Markov Chains: Simple Example

- The stochastic matrix of economic conditions: (Monthly Frequency)

$$P = \begin{pmatrix} 0.971 & 0.029 & 0 \\ 0.145 & 0.778 & 0.077 \\ 0 & 0.508 & 0.492 \end{pmatrix}$$

where states are: "normal growth", "mild recession", "severe recession"

- 

Finite Markov Chains: AR(1) with (Tauchen, 1986)

- It is convenient to replace a continuous AR(1) with a discrete Markov chains
- Discrete approximations to AR(1) processes of the form

$$y_{t+1} = \rho y_t + u_{t+1}, \text{ where } u_t \text{ is assumed to be i.i.d. and follow } N(0, \sigma_u^2)$$

- The variance of the stationary probability distribution of $\{y_t\}$ is

$$\sigma_y^2 = \frac{\sigma_u^2}{1-\rho^2}$$

- Tauchen's method is the most used for approximating this continuous state process with a finite state Markov chain; Other methods are also used, i.e., Rouwenhorst (95)
- Let's write our own Tauchen method!
- Step 1: Choose discretization objects:
  - n, the number of states for the discrete approximation
  - m, an integer that parameterizes the width of the state space
- Step 2: Choose state space $S = \{x_0, \ldots, x_{n-1}\} \subset \mathbb{R}$
  - $x_0 = -m\,\sigma_y$
  - $x_{n-1} = m\,\sigma_y$
  - $x_{i+1} = x_i + s$ where step $s = (x_{n-1} - x_0)/(n-1)$

Finite Markov Chains: AR(1) with (Tauchen, 1986)

- Step 3: Choose a stochastic $n \times n$ matrix P (most important part)
  - Let F be the cumulative distribution function of the normal distribution $N(0, \sigma_u^2)$
  - The values $P(x_i, x_j)$ are computed to approximate the AR(1) process
  - If $j = 0$, then set
  $$P(x_i, x_j) = P(x_i, x_0) = F(x_0 - \rho x_i + s/2)$$
  - If $j = n - 1$, then set
  $$P(x_i, x_j) = P(x_i, x_{n-1}) = 1 - F(x_{n-1} - \rho x_i - s/2)$$
  - Otherwise, set
  $$P(x_i, x_j) = F(x_j - \rho x_i + s/2) - F(x_j - \rho x_i - s/2)$$
- With both S and P in hand, we successfully replaced a continuous AR(1)!
- So, how good is our approximation? [Will find out in the assignment]
- How about other methods, say Rouwenhorst, 1995? [Will find out in the assignment]

Finite Markov Chains: AR(1) Extension - Uncertainty Shocks

- A very popular second-moment shock is uncertainty shock (Bloom, 2009)
- Basically, a shock to the second moment σ_u in the above AR(1) process
- How to construct an uncertainty shock? i.e., $\hat{\sigma}_u > \sigma_u$
- One answer is to manipulate the stochastic $n \times n$ matrix $\hat{P}$
    - Step 1&2: The same progress as in normal AR(1)
    - Step 3-1: use $\hat{F}$ as the cumulative distribution function of the normal distribution $N(0, \hat{\sigma}_u^2)$
    - Step 3-2: construct $P(x_i, x_j)$ with the same three cases
- When you hit the firms with uncertainty shock, force them to use the new matrix $\hat{P}$
- So, how good is this approximation? [Will find out in the assignment]
- How about other methods, say "change state space S"? [Will find out in the assignment]

Continuous State Markov Chains: Simple Introduction

- Continuous state Markov chains is a more general case of what we just studied
- The best usage is to use on "nonlinear distributions" in economic models
- We focus on the "Density Case" where state S is a bounded interval (a, b)
- Formally, a stochastic kernel on S is a function $p\colon S \times S \to \mathbb{R}$ with the property that:
  1. $p(x, y) \geq 0$ for all $x, y \in S$
  2. $\int p(x, y)dy = 1$ for all $x \in S$
- For example, let $S = \mathbb{R}$ and consider the particular stochastic kernel p_w defined by

$$p_w(x, y) := \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{(y-x)^2}{2}\right\}$$

- This is the (normally distributed) random walk

$$X_{t+1} = X_t + \xi_{t+1} \quad \text{where} \quad \{\xi_t\} \overset{\text{IID}}{\sim} N(0, 1)$$

- Applications: See QuantEcon 18.2.6. example of usage for the stochastic growth model

Distributions

- Now we formally talk about transition of distributions D_t to D_{t+1}
- Different from the very simple distributions in Continuous State Markov Chains
- We want to have distributions that are both nonlinear and endogenous
- See Chapter 25. Wealth Distribution Dynamics for an example of households' wealth
- Why do we want distribution dynamics in the model?
  - modeling heterogeneous firms or households
  - measures of inequality (households) or misallocation (firms) of the economy
  - how distribution matters for the economy, etc...
- Finally, the transition of distribution is just a collection of transitions between states
- We will play with simple one- and two-dimension distributions

Distributions: One-dimension: Model

- Let's start with QuantEcon 25.3. A Model of Wealth Dynamics
- This and the next pages describe the economic model
- The model we will study is

$$w_{t+1} = (1 + r_{t+1})s(w_t) + y_{t+1}$$

  - w_t is wealth at time t for a given household,
  - r_t is the rate of return of financial assets,
  - y_t is current non-financial (e.g., labor) income and
  - $s(w_t)$ is current wealth net of consumption
- Letting $\{z_t\}$ be a correlated state process of the form

$$z_{t+1} = az_t + b + \sigma_z \epsilon_{t+1}$$

Distributions: Model

- We'll assume that

$$1 + r_t = c_r \exp(z_t) + \exp(\mu_r + \sigma_r \xi_t)$$

$$y_t = c_y \exp(z_t) + \exp(\mu_y + \sigma_y \zeta_t)$$

- Here $\{(\epsilon_t, \xi_t, \zeta_t)\}$ is IID and standard normal in $\mathbb{R}^3$.
- The value of c_r is close to zero since the return on assets does not exhibit large trends.
- When we simulate a population of households, assume all shocks are idiosyncratic (i.e., specific to individual households and independent across them).
- Regarding the savings function s, our default model will be

$$s(w) = s_0 w \cdot \mathbf{1}\{w \geq \hat{w}\}, \quad \text{where } s_0 \text{ is a positive constant.}$$

- Thus, for $w < \hat{w}$, the household saves nothing.
- For $w \geq \bar{w}$, the household saves a fraction s_0 of their wealth.

Distributions: Implementation

```
function wealth_dynamics_model(; # all named arguments
                               w_hat = 1.0, # savings parameter
                               s_0 = 0.75, # savings parameter
                               c_y = 1.0, # labor income parameter
                               mu_y = 1.0, # labor income parameter
                               sigma_y = 0.2, # labor income parameter
                               c_r = 0.05, # rate of return parameter
                               mu_r = 0.1, # rate of return parameter
                               sigma_r = 0.5, # rate of return parameter
                               a = 0.5, # aggregate shock parameter
                               b = 0.0, # aggregate shock parameter
                               sigma_z = 0.1)
    z_mean = b / (1 - a)
    z_var = sigma_z^2 / (1 - a^2)
    exp_z_mean = exp(z_mean + z_var / 2)
    R_mean = c_r * exp_z_mean + exp(mu_r + sigma_r^2 / 2)
    y_mean = c_y * exp_z_mean + exp(mu_y + sigma_y^2 / 2)
    alpha = R_mean * s_0

    # Distributions
    z_stationary_dist = Normal(z_mean, sqrt(z_var))

    @assert alpha <= 1 # check stability condition that wealth does not diverge
    return (; w_hat, s_0, c_y, mu_y, sigma_y, c_r, mu_r, sigma_r, a, b, sigma_z,
            z_mean, z_var, z_stationary_dist, exp_z_mean, R_mean, y_mean, alpha)
end
```

```
wealth_dynamics_model (generic function with 1 method)
```

Distributions: Simulation

```julia
function simulate_wealth_dynamics(w_0, z_0, T, params)
    (; w_hat, s_0, c_y, mu_y, sigma_y, c_r, mu_r, sigma_r, a, b, sigma_z) = params # ur
    w = zeros(T + 1)
    z = zeros(T + 1)
    w[1] = w_0
    z[1] = z_0

    for t in 2:(T + 1)
        z[t] = a * z[t - 1] + b + sigma_z * randn()
        y = c_y * exp(z[t]) + exp(mu_y + sigma_y * randn())
        w[t] = y # income goes to next periods wealth
        if w[t - 1] >= w_hat # if above minimum wealth level, add savings
            R = c_r * exp(z[t]) + exp(mu_r + sigma_r * randn())
            w[t] += R * s_0 * w[t - 1]
        end
    end
    return w, z
end
```

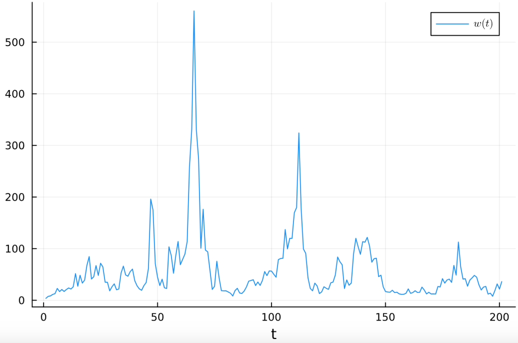
```
simulate_wealth_dynamics (generic function with 1 method)
```

Distributions: Results in One-dimension

Let's look at the wealth dynamics of an individual household.

```
p = wealth_dynamics_model()  # all defaults
y_0 = p.y_mean
z_0 = rand(p.z_stationary_dist)
T = 200
w, z = simulate_wealth_dynamics(y_0, z_0, T, p)

plot(w, caption = "Wealth simulation", xlabel = "t", label = L"w(t)")
```



Distributions: Results in One-dimension

- Assuming the distribution is stationary (is it?), we could:
- Plot the stationary distribution in one-dimension
- Generate the statistics of the stationary distribution in one-dimension
- Also, we could calculate the inequality measures like the Gini index
- [Will find out in the assignment]

Distributions: Extension in Two-dimension

- However, remember that z_{t+1} is a persistent AR(1) progress
- Actually the distribution is in two dimensions $D_t(z_t, w_t)$ instead of $D_t(w_t)$
- Plot the stationary distribution in two-dimension (z, w)
- Generate the statistics of the stationary distribution in two-dimension
- Also, we could calculate the inequality measures like the Gini index
- [Will find out in the assignment]

Simulation

- Finally, let's talk about simulation of distributions
- We have already seen stochastic simulation through random number generators

```julia
function simulate_wealth_dynamics(w_0, z_0, T, params)
    (; w_hat, s_0, c_y, mu_y, sigma_y, c_r, mu_r, sigma_r, a, b, sigma_z) = params # ur
    w = zeros(T + 1)
    z = zeros(T + 1)
    w[1] = w_0
    z[1] = z_0

    for t in 2:(T + 1)
        z[t] = a * z[t - 1] + b + sigma_z * randn()
        y = c_y * exp(z[t]) + exp(mu_y + sigma_y * randn())
        w[t] = y # income goes to next periods wealth
        if w[t - 1] >= w_hat # if above minimum wealth level, add savings
            R = c_r * exp(z[t]) + exp(mu_r + sigma_r * randn())
            w[t] += R * s_0 * w[t - 1]
        end
    end
    return w, z
end
```

```
simulate_wealth_dynamics (generic function with 1 method)
```

- Could we simulate stochastic progress without stochastic simulation?
- Yes, we can! Eric Young, 2010 provides a fantastic non-stochastic simulation method!

Simulation: Stochastic

- First, let's redo the stochastic simulation to be using discrete grids
- To make your life easier, we kill two stochastic progress $\{(\xi_t = 0, \zeta_t = 0)\}$
- For z state, use the Tauchen method: let's choose 11 states
- For w state, choose 100 grid points from $w_{low} = 0$ to $w_{cap} = 200$ (0,2,4,6,...,200)
- You can modify $w_{t+1} = (1 + r_{t+1})s(w_t) + y_{t+1}$ to enforce artificially $w_{t+1} \leq 200$
- How about the w_t that is between grid points, i.e., $w_t = 2.5 \in [2, 4]$?
- Use interpolation that you learned in the last lecture to assign mess onto 2 and 4
- So, how good is our approximation? [Will find out in the assignment]

Simulation: Non-Stochastic

- Now, let's finally do the non-stochastic simulation (Young, 2010)
- We use exactly the same Tauchen grids and wealth grids as above
- But from time t to $t+1$, we do not use random number generators!
- Instead, we could generate a distribution of possible w_{t+1} given (z_t, w_t)
- For instance, for a grid point today, $z_t = z_i$ and $w_t = w_p$
- There is a w_{t+1} value $w(z_j, w_t) \in [w_k, w_{k+1}]$ if $z_{t+1} = z_j$ with $p(z_i, z_j)$
- We could just assign the mess on w_k and w_{k+1} by distance from $w(z_j, w_t)$

$$
\begin{aligned}
D[z_j, w_k] \quad &+= \quad D[z_i, w_p] * p(z_i, z_j) * \frac{w(z_j, w_t) - w_k}{w_{k+1} - w_k} \\
D[z_j, w_{k+1}] \quad &+= \quad D[z_i, w_p] * p(z_i, z_j) * \frac{w_{k+1} - w(z_j, w_t)}{w_{k+1} - w_k}
\end{aligned}
$$

- Brilliant! So, how good is our approximation? [Will find out in the assignment]

Assignment 1

- Individual assignment, but you can discuss with classmates
- Make presentable slides of your results (maybe some codes if you wish)
- Push codes and slides of the results to your GitHub (shared with me)
- Everyone will have 8 minutes to present the results on **Feb.12th (DDL)**
- You will be graded depending on:
    - If you could deliver the correct results (40)
    - How clean and organized are your codes written (30)
    - How well you could deliver your results in a presentation (30)

Task 1: AR(1) on Page 7&8

- Suggested Parameters: $\{\rho = 0.9, \sigma_u = 0.1, n = 11, m = 3\}$
  (Choose your own if not working)
- All the outputs of tasks below is a plot!
- Task 1-1:
  Solve Tauchen (86) approximation and simulate the stationary distribution $D^T(y)$
- Task 1-2:
  Compare $D^T(y)$ to true distribution $D(y)$ (Continuous State Markov Chains)
- Task 1-3:
  Solve Rouwenhorst (95) approximation and simulate the stationary distribution $D^R(y)$
- Task 1-4:
  Compare $D^R(y)$ to true distribution $D(y)$ (Continuous State Markov Chains)

Task 2: AR(1) Extension of Uncertainty Shocks on Page 9

- Suggested Parameters: $\{\rho = 0.9, \sigma_u = 0.1, n = 11, m = 3, \hat{\sigma}_u = 0.2\}$ | Method: Tauchen (86)
- Task 2-1:
  Solve the stochastic matrix $\hat{P}$ of the uncertainty shock
- Task 2-2:
  Simulate 1000 individuals indexed by i for 21 periods $t = 0, 1, \ldots, 20$, starting with $y_{i,0} = 0$ that receive an uncertainty shock only at time $t = 11$. Plot the time path of $y_{i,t}$ of all 1000 individuals over t.
- Task 2-3: [Bonus]
  How about other methods, say "change state space S"? For instance, an uncertainty shock pushes individuals to have larger upper bounds and smaller lower bounds.

Task 3: Distributions on Page 17&18

- Same economic parameters as in the codes
- Simulate N=10,000 individuals with a random start until the distribution is stationary
- Task 3-1:
  Plot the stationary distribution in one-dimension
- Task 3-2:
  Simulate and plot the stationary distribution in two-dimension
- Task 3-3:
  Generate the statistics of the stationary distribution in two-dimension: i.e., Corr(z,w)

Task 4: Distributions on Page 20&21

- Same economic parameters as in the codes
- Simulate N=10,000 individuals with a random start until the distribution is stationary
- For z state, use the Tauchen method: let's choose 5 states
- For w state, choose 50 grid points from $w_{low} = 0$ to $w_{cap} = 200$ (0,4,8,...,200)
- Task 4-1:
  Compare the stochastic simulation using discrete grids to the original stochastic simulation in Task 3. Plot the stationary distribution in one- and two-dimension.
- Task 4-2:
  Compare the non-stochastic simulation to the stochastic simulation using discrete grids. Plot the stationary distribution in one- and two-dimension.
- Task 4-3:
  Compare the computing time for all three simulations for T=1,000 period

Appendix

References

Bloom, Nicholas (2009). "The impact of uncertainty shocks". In: *econometrica* 77.3, pp. 623–685.

Rouwenhorst, K Geert (1995). "Asset pricing implications of equilibrium business cycle models". In: *Frontiers of business cycle research* 1, pp. 294–330.

Tauchen, George (1986). "Finite state markov-chain approximations to univariate and vector autoregressions". In: *Economics letters* 20.2, pp. 177–181.

Young, Eric R (2010). "Solving the incomplete markets model with aggregate uncertainty using the Krusell–Smith algorithm and non-stochastic simulations". In: *Journal of Economic Dynamics and Control* 34.1, pp. 36–41.