Lecture 5

Dynamics theory review

Ivan Rudik
AEM 7130

Building a dynamic economic model

We need 5 things for a dynamic economic model

Building a dynamic economic model

We need 5 things for a dynamic economic model

1. Controls: what variables are we optimizing, what decisions do the economic agents make?

Building a dynamic economic model

We need 5 things for a dynamic economic model

1. Controls: what variables are we optimizing, what decisions do the economic agents make?

2. States: What are the variables that change over time and interact with the controls?

Building a dynamic economic model

We need 5 things for a dynamic economic model

1. Controls: what variables are we optimizing, what decisions do the economic agents make?

2. States: What are the variables that change over time and interact with the controls?

3. Payoff: What is the single-period payoff function? What's our reward?

Building a dynamic economic model

We need 5 things for a dynamic economic model

1. Controls: what variables are we optimizing, what decisions do the economic agents make?

2. States: What are the variables that change over time and interact with the controls?

3. Payoff: What is the single-period payoff function? What's our reward?

4. Transition equations: How do the state variables evolve over time?

Building a dynamic economic model

We need 5 things for a dynamic economic model

1. Controls: what variables are we optimizing, what decisions do the economic agents make?

2. States: What are the variables that change over time and interact with the controls?

3. Payoff: What is the single-period payoff function? What's our reward?

4. Transition equations: How do the state variables evolve over time?

5. Planning horizon: When does our problem terminate? Never? 100 years?

Two types of solutions

Dynamic problems can generally be solved in two ways

Two types of solutions

Dynamic problems can generally be solved in two ways

Open-loop: treat the model as one optimization problem

Two types of solutions

Dynamic problems can generally be solved in two ways

Open-loop: treat the model as one optimization problem

- Transitions act like constraints, solve for optimal controls at each time
- Drawback: solutions will be just a function of time so we can't introduce uncertainty, strategic behavior, etc

Two types of solutions

Feedback: treat the model as a bunch of single-period optimization problems with the immediate payoff and the *continuation value*

Two types of solutions

Feedback: treat the model as a bunch of single-period optimization problems with the immediate payoff and the *continuation value*

- Yields a solution that is a function of states
- Permits uncertainty, game structures
- Drawback: need to solve for the continuation value function or policy function

Markov chains

Dynamic models in economic models are typically **Markovian**

Markov chains

Dynamic models in economic models are typically **Markovian**

A stochastic process $\{x_t\}$ is said to have the **Markov property** if for all $k \geq 1$ and all t

$$Prob(x_{t+1}|x_t, x_{t-1}, \ldots, x_{t-k}) = Prob(x_{t+1}|x_t)$$

Markov chains

Dynamic models in economic models are typically **Markovian**

A stochastic process $\{x_t\}$ is said to have the **Markov property** if for all $k \geq 1$ and all t

$$Prob(x_{t+1}|x_t, x_{t-1}, \ldots, x_{t-k}) = Prob(x_{t+1}|x_t)$$

The distribution of the next vector in the sequence (i.e. the distribution of next period's state) is a function of only the current vector (state)

Markov chains

Dynamic models in economic models are typically **Markovian**

A stochastic process $\{x_t\}$ is said to have the **Markov property** if for all $k \geq 1$ and all t

$$Prob(x_{t+1}|x_t, x_{t-1}, \ldots, x_{t-k}) = Prob(x_{t+1}|x_t)$$

The distribution of the next vector in the sequence (i.e. the distribution of next period's state) is a function of only the current vector (state)

The Markov property is necessary for the feedback representation

Markov chains

We characterize stochastic state transitions with **Markov chains**

Markov chains

We characterize stochastic state transitions with **Markov chains**

A Markov chain is characterized by:

1. n-dimensional state space with vectors $e_i, i = 1, \ldots, n$ where e_i is an $n \times 1$ unit vector whose ith entry is 1 and all others are 0
2. An $n \times n$ *transition matrix* P which captures the probability of transitioning from one point of the state space to another point of the state space next period
3. $n \times 1$ vector π_0 whose ith value is the probability of being in state i at time 0: $\pi_{0i} = \text{Prob}(x_0 = e_i)$

Markov chains

P is given by

$$P_{ij} = \text{Prob}(x_{t+1} = e_j | x_t = e_i)$$

Markov chains

P is given by

$$P_{ij} = \text{Prob}(x_{t+1} = e_j | x_t = e_i)$$

We need one assumption:

- For $i = 1, .., n$, $\sum_{j=1}^{n} P_{ij} = 1$ and π_0 satisfies: $\sum_{i=1}^{n} \pi_{0i} = 1$

Markov chains

Nice property of Markov chains:

We can use P to determine the probability of moving to another state in *two* periods by P^2 since

$$\begin{aligned}
&\text{Prob}(x_{t+2} = e_j | x_t = e_i) \\
&= \sum_{h=1}^{n} \text{Prob}(x_{t+2} = e_j | x_{t+1} = e_h)\text{Prob}(x_{t+1} = e_h | x_t = e_i) \\
&= \sum_{h=1}^{n} P_{ih} P_{hj} = P^2_{ij}
\end{aligned}$$

Markov chains

$$\text{Prob}(x_{t+2} = e_j | x_t = e_i) = \sum_{h=1}^{n} P_{ih} P_{hj} = P_{ij}^2$$

Markov chains

$$\text{Prob}(x_{t+2} = e_j | x_t = e_i) = \sum_{h=1}^{n} P_{ih} P_{hj} = P_{ij}^2$$

iterate on this to show that

Markov chains

$$\text{Prob}(x_{t+2} = e_j | x_t = e_i) = \sum_{h=1}^{n} P_{ih} P_{hj} = P_{ij}^2$$

iterate on this to show that

$$\text{Prob}(x_{t+k} = e_j | x_t = e_i) = P_{ij}^k$$

Dynamic programming

Start with a general sequential problem to set up the basic recursive/feedback dynamic optimization problem

Dynamic programming

Start with a general sequential problem to set up the basic recursive/feedback dynamic optimization problem

Let $\beta \in (0,1)$, the economic agent selects a sequence of controls, $\{u_t\}_{t=0}^{\infty}$ to maximize

$$\sum_{t=0}^{\infty} \beta^t r(x_t, u_t)$$

subject to $x_{t+1} = g(x_t, u_t)$ and with x_0 given

Dynamic programming

If we want to maximize the PV of total utility:

$$\max_{u_0, u_1, ..., u_n, ...} \sum_{t=0}^{\infty} \beta^t r(x_t, u_t)$$

we have a tough problem, we need to select a full sequence of u_ts

Dynamic programming

If we want to maximize the PV of total utility:

$$\max_{u_0, u_1, \ldots, u_n, \ldots} \sum_{t=0}^{\infty} \beta^t r(x_t, u_t)$$

we have a tough problem, we need to select a full sequence of u_ts

Dynamic programming makes this simpler

Dynamic programming

The idea behind dynamic programmming (like most of what we do) is to exchange solving one hard problem for solving a bunch of easier problems

Dynamic programming

The idea behind dynamic programmming (like most of what we do) is to exchange solving one hard problem for solving a bunch of easier problems

The essence of dynamic programming is summed up in **Bellman's Principle of Optimality**

Dynamic programming

The idea behind dynamic programmming (like most of what we do) is to exchange solving one hard problem for solving a bunch of easier problems

The essence of dynamic programming is summed up in **Bellman's Principle of Optimality Principle of Optimality:**

> An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision

Dynamic programming

The idea behind dynamic programmming (like most of what we do) is to exchange solving one hard problem for solving a bunch of easier problems

The essence of dynamic programming is summed up in **Bellman's Principle of Optimality Principle of Optimality:**

> An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision

In simpler terms: your current optimal decision is only dependent on the current state, not your past actions

Dynamic programming

In dynamic programming we really just solve for one object

Dynamic programming

In dynamic programming we really just solve for one object

This object is a function that tells us what is the optimal action to take given the current state

Dynamic programming

In dynamic programming we really just solve for one object

This object is a function that tells us what is the optimal action to take given the current state

The tricky thing about this, is how we actually solve for this function

Dynamic programming

Policy functions or value functions tell us how to take optimal actions

Dynamic programming

Policy functions or value functions tell us how to take optimal actions

They either tell you your optimal action given the state, or your maximized payoff given the state

Dynamic programming

Policy functions or value functions tell us how to take optimal actions

They either tell you your optimal action given the state, or your maximized payoff given the state

Once we have either of these functions we can solve for the optimal action in any given state of the world and solve our problem

Dynamic programming

Assume r is concave, continuously differentiable, and the state space is convex and compact

Dynamic programming

Assume r is concave, continuously differentiable, and the state space is convex and compact

We want to recover a *policy function* h which maps the current state x_t into the current control u_t, such that the sequence $\{u_s\}_{s=0}^{\infty}$ generated by iterating

$$
\begin{aligned}
u_t &= h(x_t) \\
x_{t+1} &= g(x_t, u_t),
\end{aligned}
$$

starting from x_0, solves our original optimization problem

Value functions

Consider a function $V(x)$, the **continuation value function** where

$$V(x_0) = \max_{\{u_s\}_{s=0}^{\infty}} \sum_{t=0}^{\infty} \beta^t r(x_t, u_t)$$

subject to the transition equation: $x_{t+1} = g(x_t, u_t)$

The value function defines the maximum value (payoff) of our problem as a function of the state

Value functions

Consider a function $V(x)$, the **continuation value function** where

$$V(x_0) = \max_{\{u_s\}_{s=0}^{\infty}} \sum_{t=0}^{\infty} \beta^t r(x_t, u_t)$$

subject to the transition equation: $x_{t+1} = g(x_t, u_t)$

The value function defines the maximum value (payoff) of our problem as a function of the state

It's the dynamic indirect utility function

Value functions

Suppose we know $V(x)$, then we can solve for the policy function h by solving for each $x \in X$

$$\max_u r(x, u) + \beta V(x')$$

where $x' = g(x, u)$ and primes on state variables indicate next period

Value functions

Suppose we know $V(x)$, then we can solve for the policy function h by solving for each $x \in X$

$$\max_u r(x, u) + \beta V(x')$$

where $x' = g(x, u)$ and primes on state variables indicate next period

Conditional on having $V(x)$ we can solve our dynamic programming problem

Value functions

Suppose we know $V(x)$, then we can solve for the policy function h by solving for each $x \in X$

$$\max_u r(x, u) + \beta V(x')$$

where $x' = g(x, u)$ and primes on state variables indicate next period

Conditional on having $V(x)$ we can solve our dynamic programming problem

Instead of solving for an infinite dimensional set of policies, we instead find the $V(x)$ and h that solves the continuum of maximization problems, where there is a unique maximization problem for each x

Bellman equations

Issue: How do we know $V(x)$ when it depends on future (optimized) actions?

Bellman equations

Issue: How do we know $V(x)$ when it depends on future (optimized) actions?

Define the **Bellman equation**

$$V(x) = \max_u r(x, u) + \beta V[g(x, u)]$$

Bellman equations

Issue: How do we know $V(x)$ when it depends on future (optimized) actions?

Define the **Bellman equation**

$$V(x) = \max_{u} r(x,u) + \beta V[g(x,u)]$$

$h(x)$ maximizes the right hand side of the Bellman

Bellman equations

The policy function satisfies

$$V(x) = r[x, h(x)] + \beta V\{g[x, h(x)]\}$$

Bellman equations

The policy function satisfies

$$V(x) = r[x, h(x)] + \beta V\{g[x, h(x)]\}$$

Solving the problem yields a solution that is a function, $V(x)$

Bellman equations

The policy function satisfies

$$V(x) = r[x, h(x)] + \beta V\{g[x, h(x)]\}$$

Solving the problem yields a solution that is a function, $V(x)$

This is a recursive problem, one of the workhorse solution methods exploits this recursion and contraction mapping properties of the Bellman operator to solve for $V(x)$

Solution properties

Under standard assumptions we have that

1. The solution to the Bellman equation, $V(x)$, is strictly concave
2. The solution is approached in the limit as $j \to \infty$ by iterations on: $V_{j+1}(x) = \max_u r(x, u) + \beta V_j(x')$, given any bounded and continuous V_0 and our transition equation
3. There exists a unique and time-invariant optimal policy function $u_t = h(x_t)$ where h maximizes the right hand side of the Bellman
4. The value function $V(x)$ is differentiable

Euler equations

Euler equations are dynamic efficiency conditions: they equalize the marginal effects of an optimal policy over time

E.g: set the current marginal benefit, energy from burning fossil fuels, with the future marginal cost, global warming

Euler equations

Euler equations are dynamic efficiency conditions: they equalize the marginal effects of an optimal policy over time

E.g: set the current marginal benefit, energy from burning fossil fuels, with the future marginal cost, global warming

1. We have a stock of capital K_t that depreciates at rate $\delta \in (0,1)$
2. We can invest to increase our future capital \$I_t\$\$
3. Per-period payoff $U(C_t)$ from consuming output C_t
4. Discount factor is $\beta \in (0,1)$

Euler equations

The Bellman equation is

$$V(K_t) = \max_{C_t} \{u(C_t) + \beta V(K_{t+1})\}$$
$$\text{subject to:} \quad K_{t+1} = (1-\delta)K_t - C_t$$

Euler equations

The Bellman equation is

$$V(K_t) = \max_{C_t} \{u(C_t) + \beta V(K_{t+1})\}$$
$$\text{subject to:} \quad K_{t+1} = (1-\delta)K_t - C_t$$

The FOC with respect to consumption is

$$u'(C_t) = \beta V_K(K_{t+1})$$

Euler equations

The Bellman equation is

$$V(K_t) = \max_{C_t} \{u(C_t) + \beta V(K_{t+1})\}$$
$$\text{subject to:} \quad K_{t+1} = (1-\delta)K_t - C_t$$

The FOC with respect to consumption is

$$u'(C_t) = \beta\, V_K(K_{t+1})$$

Envelope theorem gives us

$$V_K(K_t) = \beta\, \delta\, V_K(K_{t+1})$$

Euler equations

The FOC with respect to consumption is

$$u'(C_t) = \beta V_K(K_{t+1})$$

Envelope theorem gives us

$$V_K(K_t) = \beta (1 - \delta) V_K(K_{t+1})$$

Euler equations

The FOC with respect to consumption is

$$u'(C_t) = \beta\, V_K(K_{t+1})$$

Envelope theorem gives us

$$V_K(K_t) = \beta\,(1-\delta)\, V_K(K_{t+1})$$

Advance both by one period since they must hold for all t

$$\begin{aligned} u'(C_{t+1}) &= \beta\, V_K(K_{t+2}) \\ V_K(K_{t+1}) &= \beta\,(1-\delta)\, V_K(K_{t+2}) \end{aligned}$$

Euler equations

Substitute the time t and time $t+1$ FOCs into our time $t+1$ envelope condition

$$u'(C_t) = \beta\,(1-\delta)\,u'(C_{t+1})$$

Euler equations

Substitute the time t and time $t+1$ FOCs into our time $t+1$ envelope condition

$$u'(C_t) = \beta\,(1-\delta)\,u'(C_{t+1})$$

LHS is marginal benefit of consumption, RHS is marginal cost of consumption **along an optimal path**

Euler equations

$$u'(C_t) = \beta \, (1 - \delta) \, u'(C_{t+1})$$

LHS: marginal benefit of consumption RHS: marginal cost of lower utility from more less output because of a lower future capital stock

Euler equations: no-arbitrage

Euler equations are **no-arbitrage conditions**

Suppose we're on the optimal capital path and want to deviate by cutting back consumption

Euler equations: no-arbitrage

Euler equations are **no-arbitrage conditions**

Suppose we're on the optimal capital path and want to deviate by cutting back consumption

Yields a marginal cost today less consumption utility

Euler equations: no-arbitrage

Euler equations are **no-arbitrage conditions**

Suppose we're on the optimal capital path and want to deviate by cutting back consumption

Yields a marginal cost today less consumption utility

The benefit is that we have $1 - \delta$ units of greater capital tomorrow after depreciation which lets us increase our consumption at some utility discount rate β

Euler equations: no-arbitrage

If this deviation (or deviating by investing more today) were profitable, we would do it

$\rightarrow$ the optimal policy must have zero additional profit opportunities: this is what the Euler equation defines

Basic theory

Here we finish up the basic theory pieces we need

We will focus on deterministic problems but this easily ports to stochastic problems

Basic theory

Here we finish up the basic theory pieces we need

We will focus on deterministic problems but this easily ports to stochastic problems

Final two pieces

1. Stationarity: does not depend explicitly on time
2. Discounting: $\beta \in (0, 1)$, the future matters but not as much as today

Discounting and bounded payoffs ensures total value is bounded

Basic theory

The general problem can be written recursively as

$$V(s_0) = \max_{u_0 \in U(s_0)} r(s_t, u_t) + \beta V(s_1)$$

$$\text{subject to: } s_{t+1} = g(s_t, u_t)$$

Value function existence and uniqueness

Reformulate the problem as,

$$V(s) = \max_{s' \in \Gamma(s)} r(s, s') + \beta V(s'), \;\; \forall s \in S$$

where $\Gamma(s)$ is our set of feasible states next period

Value function existence and uniqueness

Reformulate the problem as,

$$V(s) = \max_{s' \in \Gamma(s)} r(s, s') + \beta V(s'), \quad \forall s \in S$$

where $\Gamma(s)$ is our set of feasible states next period

There exists a solution to the Bellman under a (particular) set of sufficient conditions

Value function existence and uniqueness

If the following are true:

Value function existence and uniqueness

If the following are true:

1. $r(s_t, u_t)$ is real-valued, continuous and bounded
2. $\beta \in (0, 1)$
3. the feasible set of states for next period is non-empty, compact, and continuous

Value function existence and uniqueness

If the following are true:

1. $r(s_t, u_t)$ is real-valued, continuous and bounded
2. $\beta \in (0, 1)$
3. the feasible set of states for next period is non-empty, compact, and continuous

then there exists a unique value function $V(s)$ that solves the Bellman equation

Intuitive sketch of the proof

Define an operator T as

$$T(W)(s) = \max_{s' \in \Gamma(s)} r(s, s') + \beta W(s'), \quad \forall s \in S$$

Intuitive sketch of the proof

Define an operator T as

$$T(W)(s) = \max_{s' \in \Gamma(s)} r(s, s') + \beta W(s'), \quad \forall s \in S$$

This operator takes some value function $W(s)$, maximizes it, and returns another $T(W)(s)$

Intuitive sketch of the proof

Define an operator T as

$$T(W)(s) = \max_{s' \in \Gamma(s)} r(s, s') + \beta W(s'), \ \ \forall s \in S$$

This operator takes some value function $W(s)$, maximizes it, and returns another $T(W)(s)$

It is easy to see that any $V(s)$ that satisfies $V(s) = T(V)(s) \ \ \forall s \in S$ solves the Bellman equation

Intuitive sketch of the proof

$$T(W)(s) = \max_{s' \in \Gamma(s)} r(s, s') + \beta W(s'), \quad \forall s \in S$$

We simply search for the **fixed point** of $T(W)$ to solve our dynamic problem, but how do we find the fixed point?

Intuitive sketch of the proof

$$T(W)(s) = \max_{s' \in \Gamma(s)} r(s, s') + \beta W(s'), \ \ \forall s \in S$$

We simply search for the **fixed point** of $T(W)$ to solve our dynamic problem, but how do we find the fixed point?

First we must show that a way exists by showing that $T(W)$ is a **contraction**: as we iterate using the T operator, we will get closer and closer to the fixed point

Intuitive sketch of the proof

Blackwell's sufficient conditions for a contraction are

Intuitive sketch of the proof

Blackwell's sufficient conditions for a contraction are

1. Monotonicity: if $W(s) \geq Q(s)\ \ \forall s \in S$, then $T(W)(s) \geq T(Q)(s)\ \ \forall s \in S$

Intuitive sketch of the proof

Blackwell's sufficient conditions for a contraction are

1. Monotonicity: if $W(s) \geq Q(s) \ \ \forall s \in S$, then $T(W)(s) \geq T(Q)(s) \ \ \forall s \in S$
2. Discounting: there exists a $\beta \in (0, 1)$ such that $T(W+k)(s) \leq T(W)(s) + \beta\, k$

Intuitive sketch of the proof

Blackwell's sufficient conditions for a contraction are

1. Monotonicity: if $W(s) \geq Q(s) \;\; \forall s \in S$, then $T(W)(s) \geq T(Q)(s) \;\; \forall s \in S$
2. Discounting: there exists a $\beta \in (0,1)$ such that $T(W+k)(s) \leq T(W)(s) + \beta\, k$

Monotonicity holds under our maximization

Intuitive sketch of the proof

Blackwell's sufficient conditions for a contraction are

1. Monotonicity: if $W(s) \geq Q(s) \;\; \forall s \in S$, then $T(W)(s) \geq T(Q)(s) \;\; \forall s \in S$
2. Discounting: there exists a $\beta \in (0, 1)$ such that $T(W + k)(s) \leq T(W)(s) + \beta\, k$

Monotonicity holds under our maximization

Discounting reflects that we must be discounting the future

Intuitive sketch of the proof

Blackwell's sufficient conditions for a contraction are

1. Monotonicity: if $W(s) \geq Q(s) \;\; \forall s \in S$, then $T(W)(s) \geq T(Q)(s) \;\; \forall s \in S$
2. Discounting: there exists a $\beta \in (0,1)$ such that $T(W+k)(s) \leq T(W)(s) + \beta\, k$

Monotonicity holds under our maximization

Discounting reflects that we must be discounting the future

If these two conditions hold then we have a contraction with modulus β

Intuitive sketch of the proof

Blackwell's sufficient conditions for a contraction are

1. Monotonicity: if $W(s) \geq Q(s) \;\; \forall s \in S$, then $T(W)(s) \geq T(Q)(s) \;\; \forall s \in S$
2. Discounting: there exists a $\beta \in (0,1)$ such that $T(W+k)(s) \leq T(W)(s) + \beta\, k$

Monotonicity holds under our maximization

Discounting reflects that we must be discounting the future

If these two conditions hold then we have a contraction with modulus β

Why do we care this is a contraction?

Intuitive sketch of the proof

So we can take advantage of the contraction mapping theorem which states:

Intuitive sketch of the proof

So we can take advantage of the contraction mapping theorem which states:

1. T has a unique fixed point
2. $T(V^*) = V^*$
3. We can start from any arbitrary initial function W, iterate using T and reach the fixed point

Function iteration

What this tells us is we can solve for V using a variant of function iteration

Function iteration

What this tells us is we can solve for V using a variant of function iteration

Before we were evaluating a function at value x, and then updating that value until it converges to a fixed point x^*

Function iteration

What this tells us is we can solve for V using a variant of function iteration

Before we were evaluating a function at value x, and then updating that value until it converges to a fixed point x^*

Here we are evaluating a function T of functions V, updating the function V until it converges to a fixed point V^*

Function iteration

What this tells us is we can solve for V using a variant of function iteration

Before we were evaluating a function at value x, and then updating that value until it converges to a fixed point x^*

Here we are evaluating a function T of functions V, updating the function V until it converges to a fixed point V^*

Even though it seems a bit more complicated the solution concept is exactly the same

Function iteration

What this tells us is we can solve for V using a variant of function iteration

Before we were evaluating a function at value x, and then updating that value until it converges to a fixed point x^*

Here we are evaluating a function T of functions V, updating the function V until it converges to a fixed point V^*

Even though it seems a bit more complicated the solution concept is exactly the same

Next we will start learning how to do this