Lecture 1: Introduction

Dynamic Programming

Thomas Jørgensen

Outline for today

1. **Introduction** to the course
2. **Example** of what we will be doing
3. The course **plan**

Why dynamic programming?

- **Policy recommendations** are central for all social science
- Need **models** for **counter-factual analysis**
- **Realistic models** $\rightarrow$ **no *analytical* solution**
  1. dynamic
  2. multi-dimensional
  3. uncertainty
  4. heterogeneity
  5. learning
- Solving such models require **dynamic programming**
- Estimation gives you **laboratory economies**

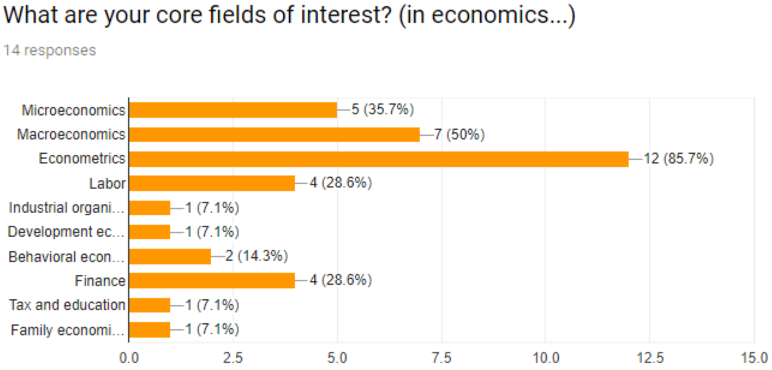
Dynamic programming

1. You should expect a **steep learning curve**
2. But it is an **extremely versatile tool**: Micro, macro, labor, industrial organization, household finance, partial and general equilibrium, games, etc. etc.
3. And **increasingly important –** better algorithms, more powerful computers

What am I using dynamic programming for?

- I am an Assistant Professor here at the Department (PhD from here too)
- My **research** is in the intersection of
  1. Household behavior
  2. Micro data
  3. Computational methods
- **Examples:**
  1. Consumption and saving over the life cycle
  2. Fertility behavior and planning
  3. Consumers information set and learning
  4. Computational methods for solving and estimating life-cycle models
     – as fast as possible

What are your interests in this course?



Organization

- **Setup:**
  - **Lectures:** Tuesday (8-10 in CSS 2-0-36) and Wednesday (10-12 in CSS 2-2-30)
  - **Classes:** Tuesday (10-12)
- **Your suggested work flow:**
  1. Prepare for the lectures (at first max 15 min.)
     slides will often be available evening before
  2. Participate actively in the lectures
  3. Re-read the slides and texts after the lecture
     and ask a question on **Padlet**
     (https://padlet.com/tjo3/DP)
  4. Read the exercises beforehand
  5. **Work on the exercises** in class by yourself in groups
     with help from Johan

Exam

- **Oral exam in group project**
    1. April 9, 8–12: Presentation of project (descriptions)
    2. April 16: **Submit project description**
    3. May 1, 9–15: 2. Supervision day
    4. May 24: **Submit project**
    5. Week 25: **Oral exam** (25 min)
- **Examples include:**
    1. **Replication** of (selected) results from some paper
    2. **Tests** (e.g. Monte Carlo) of various computational solution and estimation methods
    3. Presentation and implementation of a **new method** not taught in the course
    4. **New model, new data**
- **Formalities:** Around 8-12 pages per person (font size 12, double spacing and 3 cm margins)

Books

1. **Adda and Cooper (2003), Dynamic Economics:**
   - Chapter 2-3 ≈ lecture 1-5
   - Chapter 4 ≈ lecture 6
   - Chapter 5-10 no equivalent lectures
     – read about the topics that interests you
2. **Judd (1998), Numerical Methods:** Good for general reference and tools

Mathematical background

1. Stockey and Lucas (1989), Recursive Methods in Economics
2. Bertsekas (2005, 2012), Dynamic Programming and Optimal Control Vol. I-II
3. Puterman (2008), Markov Decision Processes: Discrete Stochastic Dynamic Programming

MATLAB

- **Download version 2016b or later**
  1. Go to **kunet.ku.dk**
  2. Under Systemadgange click on **Softwarebibliotek**
  3. Follow the instructions
- **Online MATLAB Course for Student of Economics**: See link on *Absalon*
- **Video** with Matlab essentials for this course 10mins, link on *Absalon*

The Simplest Consumption Model

- **Model:**
  1. You have M goods for consumption
  2. You have T years back of your life
  3. Your per year utility is $\sqrt{C}$
  4. $C \in \{0, 1, 2, \dots\} = \mathbb{N}$ is consumption.
  5. You discount future utility by a factor of $\beta < 1$
- **Formally:**

$$\begin{aligned} V^\star(M) &= \max_{C_1, C_2, \dots, C_T} \left\{ \sqrt{C_t} + \beta\sqrt{C_{t+1}} + \beta^2\sqrt{C_{t+2}} + \dots + \beta^T\sqrt{C_T} \right\} \\ &\text{s.t.} \\ M &= C_1 + C_2 + \dots + C_T \\ C_t &\in \mathbb{N} \end{aligned}$$

- **Solution:**
  - $T = 1$: $C_T = M$.
  - $T = 2$: Try all $C_{T-1} \leq M$ where $C_T = M - C_{T-1}$.
  - $T = 3$: Try all $C_{T-2} \leq M$ and $C_{T-1} \leq M - C_{T-2}$ where $C_T = M - C_{T-1} - C_{T-2}$.

Introduction
Introduction
Example
Plan

Algorithm 1: Algorithm for $T = 1$

input : M (goods)
output: $C_T^\star$ (optimal choice)

$V^\star = -\infty$
for $C_T = 0$ **to** M **do**
  $V = \sqrt{C_T}$
  if $V > V^\star$ **then**
    $V^\star = V$
    $C_T^\star = C_T$

$$
\begin{aligned}
V^\star(M) &= \max_{C_1, C_2, \ldots, C_T} \sqrt{C_t} + \beta\sqrt{C_{t+1}} + \beta^2\sqrt{C_{t+2}} + \cdots + \beta^T\sqrt{C_T} \\
&\text{s.t.} \\
M &= C_1 + C_2 + \cdots + C_T
\end{aligned}
$$

Algorithm 2: Algorithm for $T = 2$

input : M (goods)
output: $C_T^\star, C_{T-1}^\star$ (optimal choices)

$V^\star = -\infty$
for $C_{T-1} = 0$ **to** M **do**
  $V = \sqrt{C_{T-1}} + \beta\sqrt{M - C_{T-1}}$
  if $V > V^\star$ **then**
    $V^\star = V$
    $C_{T-1}^\star = C_{T-1}$
    $C_T^\star = M - C_{T-1}$

$$
\begin{aligned}
V^\star(M) &= \max_{C_1, C_2, \ldots, C_T} \sqrt{C_t} + \beta\sqrt{C_{t+1}} + \beta^2\sqrt{C_{t+2}} + \cdots + \beta^T\sqrt{C_T} \\
&\text{s.t.} \\
M &= C_1 + C_2 + \cdots + C_T
\end{aligned}
$$

Algorithm 3: Algorithm for $T = 3$

input : M (goods)
output: $C_T^\star, C_{T-1}^\star, C_{T-2}^\star$ (optimal choices)

$V^\star = -\infty$
for $C_{T-2} = 0$ **to** M **do**
  for $C_{T-1} = 0$ **to** $M - C_{T-2}$ **do**
    $V = \sqrt{C_{T-2}} + \beta\sqrt{C_{T-1}} + \beta^2\sqrt{M - C_{T-1} - C_{T-2}}$
    if $V > V^\star$ **then**
      $V^\star = V$
      $C_{T-1}^\star = C_{T-1}$
      $C_{T-2}^\star = C_{T-2}$
      $C_T^\star = M - C_{T-1} - C_{T-2}$

- **Write it on a poster and put on the white board for $T = 4$ (5 mins)**

Backwards induction - intuition

- **Can we do something smarter?**
- **Period T:** Value of having M goods? Consume everything

$$V_T^{\star}(M) = \sqrt{M}$$

- **Period** $T-1$: Value of having M goods?
  - today: $\sqrt{C_{T-1}}$
  - having $M - C_{T-1}$ left tomorrow: $V_T^{\star}(M - C_{T-1})$
  - $\rightarrow$seek to maximize the sum:

$$V_{T-1}^{\star}(M) = \max_{C_{T-1}} \left\{ \sqrt{C_{T-1}} + \beta V_T^{\star}(M - C_{T-1}) \right\}$$

- **Period** $T-2$: Value of having M goods?
  - today: $\sqrt{C_{T-2}}$
  - having $M - C_{T-2}$ left tomorrow: $V_{T-1}^{\star}(M - C_{T-2})$
  - $\rightarrow$seek to maximize the sum:

$$V_{T-2}^{\star}(M) = \max_{C_{T-2}} \left\{ \sqrt{C_{T-2}} + \beta V_{T-1}^{\star}(M - C_{T-2}) \right\}$$

Backwards induction - formula for all t

$$
\begin{aligned}
V_T^\star(M) &= \sqrt{M} \\
V_{T-1}^\star(M) &= \max_{C_{T-1}} \left\{ \sqrt{C_{T-1}} + \beta V_T^\star(M - C_{T-1}) \right\} \\
V_{T-2}^\star(M) &= \max_{C_{T-2}} \left\{ \sqrt{C_{T-2}} + \beta V_{T-1}^\star(M - C_{T-2}) \right\} \\
V_{T-3}^\star(M) &= \max_{C_{T-3}} \left\{ \sqrt{C_{T-3}} + \beta V_{T-2}^\star(M - C_{T-3}) \right\} \\
&\dots \\
V_t^\star(M) &= \max_{C_t} \left\{ \sqrt{C_t} + \beta V_{t+1}^\star(M - C_t) \right\}
\end{aligned}
$$

Algorithm 4: Find $V_t^\star$ given $V_{t+1}^\star$ [called "find_V" later]

input : M (goods)
$V_{t+1}^\star[\bullet]$ (value next period)
output: $V_t^\star[\bullet]$ (value of optimal choice)
$C_t^\star[\bullet]$ (optimal choice)

```
for m = 0 to M do
    i_M = m + 1
    V*_t[i_M] = -∞
    for C_t = 0 to m do
        i_next = (m - C_t) + 1
        V = sqrt(C_t) + β V*_{t+1}[i_next]
        if V > V*_t[i_M] then
            V*_t[i_M] = V
            C*_t[i_M] = C_t
```

$$V_t^\star(M) \quad = \quad \max_{C_t} \left\{ \sqrt{C_t} + \beta V_{t+1}^\star(M - C_t) \right\}$$

- On a computer, we only have discrete arrays of data:

i	M_T	$V_T^\star(M_T)$
1	0	$\sqrt{0}$
2	1	$\sqrt{1}$
3	2	$\sqrt{2}$
4	3	$\sqrt{3}$
$\vdots$	$\vdots$	$\vdots$

i	M_{T-1}	$V_{T-1}^\star(M_{T-1})$
1	0	$\max_{C_{T-1}} \sqrt{C_{T-1}} + \beta V_T^\star[0 - C_{T-1} + 1]$
2	1	$\max_{C_{T-1}} \sqrt{C_{T-1}} + \beta V_T^\star[1 - C_{T-1} + 1]$
3	2	$\max_{C_{T-1}} \sqrt{C_{T-1}} + \beta V_T^\star[2 - C_{T-1} + 1]$
4	3	$\max_{C_{T-1}} \sqrt{C_{T-1}} + \beta V_T^\star[3 - C_{T-1} + 1]$
$\vdots$	$\vdots$	$\vdots$

- On a computer, we only have discrete arrays of data:

i	M_T	$V_T^\star(M_T)$
1	0	$\sqrt{0}$
2	1	$\sqrt{1}$
3	2	$\sqrt{2}$
4	3	$\sqrt{3}$
$\vdots$	$\vdots$	$\vdots$

i	M_{T-1}	$V_{T-1}^\star(M_{T-1})$
1	0	$\max_{C_{T-1}} \sqrt{C_{T-1}} + \beta V_T^\star[0 - C_{T-1} + 1]$
2	1	$\max_{C_{T-1}} \sqrt{C_{T-1}} + \beta V_T^\star[1 - C_{T-1} + 1]$
3	2	$C_{T-2} = 0 \rightarrow \sqrt{0} + \beta V_T^\star[2 - 0 + 1]$
4	3	$\max_{C_{T-1}} \sqrt{C_{T-1}} + \beta V_T^\star[3 - C_{T-1} + 1]$
$\vdots$	$\vdots$	$\vdots$

- On a computer, we only have discrete arrays of data:

i	M_T	$V_T^\star(M_T)$
1	0	$\sqrt{0}$
2	1	$\sqrt{1}$
3	2	$\sqrt{2}$
4	3	$\sqrt{3}$
$\vdots$	$\vdots$	$\vdots$

i	M_{T-1}	$V_{T-1}^\star(M_{T-1})$
1	0	$\max_{C_{T-1}} \sqrt{C_{T-1}} + \beta V_T^\star[0 - C_{T-1} + 1]$
2	1	$\max_{C_{T-1}} \sqrt{C_{T-1}} + \beta V_T^\star[1 - C_{T-1} + 1]$
3	2	$C_{T-2} = 1 \rightarrow \sqrt{1} + \beta V_T^\star[2 - 1 + 1]$
4	3	$\max_{C_{T-1}} \sqrt{C_{T-1}} + \beta V_T^\star[3 - C_{T-1} + 1]$
$\vdots$	$\vdots$	$\vdots$

- On a computer, we only have discrete arrays of data:

i	M_T	$V_T^\star(M_T)$
1	0	$\sqrt{0}$
2	1	$\sqrt{1}$
3	2	$\sqrt{2}$
4	3	$\sqrt{3}$
$\vdots$	$\vdots$	$\vdots$

i	M_{T-1}	$V_{T-1}^\star(M_{T-1})$
1	0	$\max_{C_{T-1}} \sqrt{C_{T-1}} + \beta V_T^\star[0 - C_{T-1} + 1]$
2	1	$\max_{C_{T-1}} \sqrt{C_{T-1}} + \beta V_T^\star[1 - C_{T-1} + 1]$
3	2	$C_{T-2} = 2 \rightarrow \sqrt{2} + \beta V_T^\star[2 - 2 + 1]$
4	3	$\max_{C_{T-1}} \sqrt{C_{T-1}} + \beta V_T^\star[3 - C_{T-1} + 1]$
$\vdots$	$\vdots$	$\vdots$

Introduction
Introduction
Example
Plan

Algorithm 5: Find all $V_t^\star$

input : M (goods)
output: $V_t^\star[\bullet]$ for all t
$C_t^\star[\bullet]$ for all t

$V_T^\star[m+1] = \sqrt{m}$ for all $m = 0$ **to** M
for $t = T-1$ **to** 1 **do**
  $V_t^\star[\bullet], C_t^\star[\bullet] = \text{find\_V}(V_{t+1}^\star[\bullet])$

- **Simulate from period t with *any* M:**
  - $C_t = C_t^\star(M)$
  - $C_{t+1} = C_{t+1}^\star(M - C_t)$
  - $C_{t+3} = C_{t+2}^\star(M - C_t - C_{t+1})$
  - etc.

Vocabulary

1. *States* = number of goods, M_t
2. *Choices* = consumption, $C_t \in \mathcal{C}(M_t) = \{0, 1, 2, \ldots, M_t\}$
3. *Payoff function* = utility, $\sqrt{C_t}$
4. *Transition function* = next-period states, $M_{t+1} = \Gamma(M, C_t) = M_t - C_t$
5. *Value function* = value today, $V_t^\star(M_t)$
6. *Continuation value* = value *after* today, $\beta V_{t+1}^\star(M_{t+1}) = \beta V_{t+1}^\star(\Gamma(M_t, C_t))$
7. *Policy function* = optimal choice, $C_t^\star(M_t)$

Extensions - the rest of the course

1. Stochastic elements?
2. Continuous states and/or choices?
3. Multiple states and/or choices?
4. Infinitely many periods?
5. Estimation?
6. Multiple agents? (equilibrium and games)

Plan

1. **Theory and tools** (lecture 1-6).
2. **Consumption-saving models** (lecture 7-11, with JD as guest in week 9).
3. **Estimation of dynamic discrete choice models** (lecture 12-17)

And you should as soon as possible figure out a group and topic for a term paper!

Until next time

- **Ensure that you understand:**
  1. Algorithm 4 and **5**
  2. The vocabulary
- Go to **PadLet** and ask *or answer* a question (https://padlet.com/tjo3/DP)
- **Think about what to do if:**
  1. New goods arrived **stochastically** so that

$$M_{t+1} = \begin{cases} M_t - C_t + 1 & \text{with probability } \pi \in (0,1) \\ M_t - C_t & \text{else} \end{cases}$$

  2. Consumption was **continuous**, i.e. $C_t \in \mathbb{R}$ and not just $C_t \in \mathbb{N}$

- **Now:** go to **b.socrative.com** room name "**DynamicProgramming**" and answer a few questions