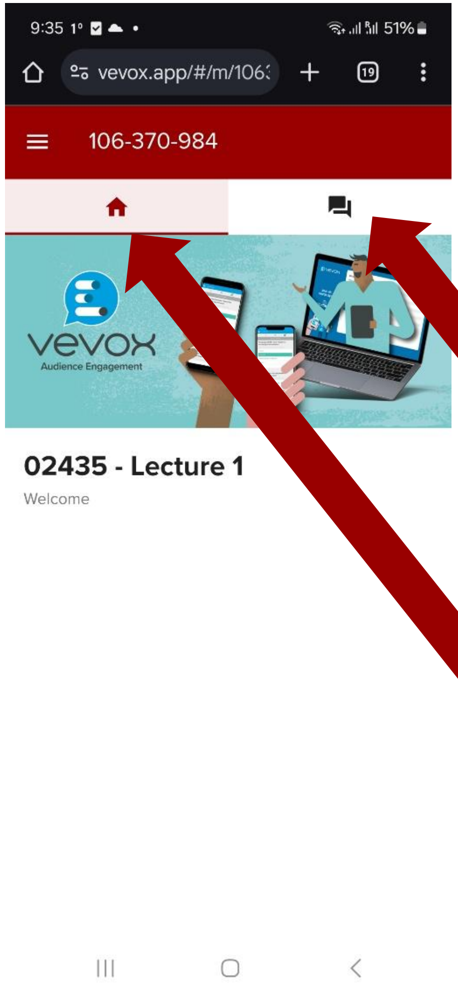
DTU

Decision Making under Uncertainty (02435)

Section for Dynamical Systems, DTU Compute.

DTU Compute
Department of Applied Mathematics and Computer Science

Scan me:



Anonymous Survey (at the end)

Anonymous Questions
(during or after the lecture)

Quizzes

Plan

→ ~~Task 0~~

→ ~~Task 1~~
~~Building an evaluation framework for sequential decision-making methods~~

→ ~~Task 2~~
~~Stochastic Programming policy (2-stage)~~
~~+ Expected Value policy a.k.a. MPC~~

→ ~~Task 2~~

~~Multi-stage Stochastic Programming + caveats~~

→ ~~Week 5: Assignment Work for Task 2 and Q&A~~

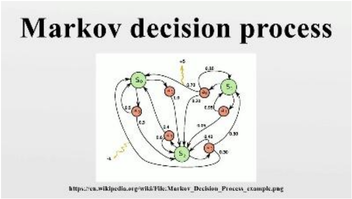
→ Weeks 6-7: Task 3
Approximate Dynamic Programming

→ Week 8: Assignment Work for Task 3 and Q&A

→ Weeks 9-11: Assignment B
Robust Optimization

Task 4 is about reporting the results from Tasks 2 and 3

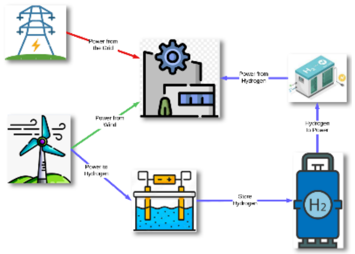
The process of designing “Decision-making” frameworks



Define → Design → Validate

This Week

Coding a simulation Environment to evaluate *any* decision-making policy



The work vs study problem

In each stage of your life (e.g. every 3 years), you need to decide whether you work or study for the next three years.
Working brings you money (salary).
Studying raises your education level, which means a higher salary when you eventually work.
Your goal is to maximize the amount of money you accrue in a given horizon (e.g. life).

1. **Stages:** $t \in T$
2. **Actions**: (work or study)
3. **State**: Education Level ε_t and base-salary level b_t
4. **Transition**:
   $\varepsilon_t = \varepsilon_{t-1} + study_{t-1} * \rho$, where ρ is the education rate
   $b_{t+1} \sim P(b_t)$, e.g. normally distributed around b_t
5. **Reward** = $work_t * b_t * \left(1 + \frac{\varepsilon_t}{2}\right)$
   if you work ($work_t = 1$) you make a salary (higher salary for higher education level)

$$\max_{u_t, x_t} \left\{ \sum_t \mathrm{E}[\mathrm{Reward}(u_t, x_t)] \right\}$$

s.t. the Transition Function, $\forall\, t$

weather we work or we study next 3 years

work give money, but studying we reaise level and we ca get more money afterwards

salary is stochastic, based on economy

Stochastic Programming for the work vs study problem

1. **Actions**: (work, study)
2. **State**: Education Level ε_t and base-salary level b_t
3. **Transition**:
   $\varepsilon_{t+1} = \varepsilon_t + study_t * \rho$, where ρ is the education rate
   $b_{t+1} \sim P(b_t)$

4. **Reward** = $work_t * b_t * \left(1 + \frac{\varepsilon_t}{2}\right)$

$$\max_{u_t, x_t} \left\{ \sum_t \mathrm{E}[\mathrm{Reward}(u_t, x_t)] \right\}$$

s.t. the Transition Function, $\forall\, t$

state variables are
Education level is endogenous - it dependes on my actions, base-salary level is exogenous - follow a random exogenous process

Stochastic Lookahead Policy:

for the base-salary level

1. Create Scenarios for the exogenous uncertainty b_t
   we can sample to create scnaraios to the exogenous uncetainities
2. Solve a multistage stochastic program:

$$\max_{u_{t,s}, x_{t,s}} \left\{ \sum_{t \in L} \sum_{s \in S} work_{t,s} * b_{t,s} * \left(1 + \frac{\varepsilon_{t,s}}{2}\right) \right\}$$

s.t. $\varepsilon_{t+1,s} = \varepsilon_{t,s} + study_{t,s} * \rho, \forall\, t, s$

non-anticipativity constraints

,scnarios are about decisions, unceerttain parameters

enfdogenous we include as variables

How to handle computational complexity

do we like short or wide?

1. **Use Decomposition**
2. **Use a small number of scenarios**
3. **Use a small lookahead horizon (e.g. 2-stage)**

Stochastic Lookahead Policy:

1. Create Scenarios for the exogenous uncertainty b_t
2. Solve a multistage stochastic program:

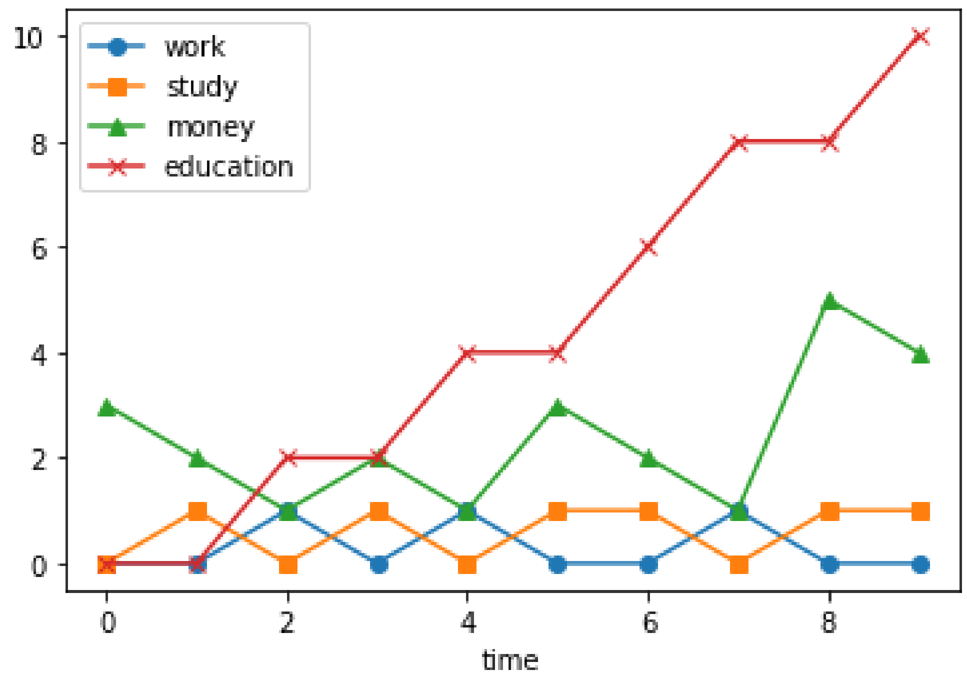
$$\max_{u_{t,s},x_{t,s}} \left\{ \sum_{t \in L} \sum_{s \in S} \left[\text{Reward}\left(u_{t,s}, x_{t,s}\right) \right] \right\}$$

s.t. $\varepsilon_{t+1,s} = \varepsilon_{t,s} + study_{t,s} * \rho, \forall\, t, s$

non-anticipativity constraints

How to think about an optimal policy

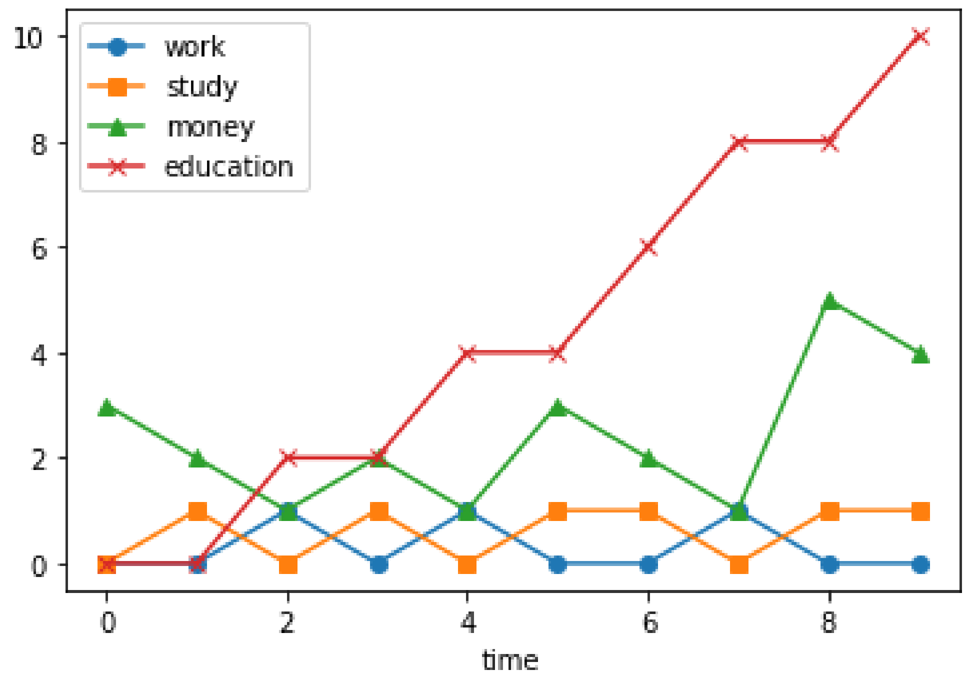
1. From the result, do you notice something that is obviously not optimal?



in the last time it is stupid to educate becasue we cannot capitalize ir further

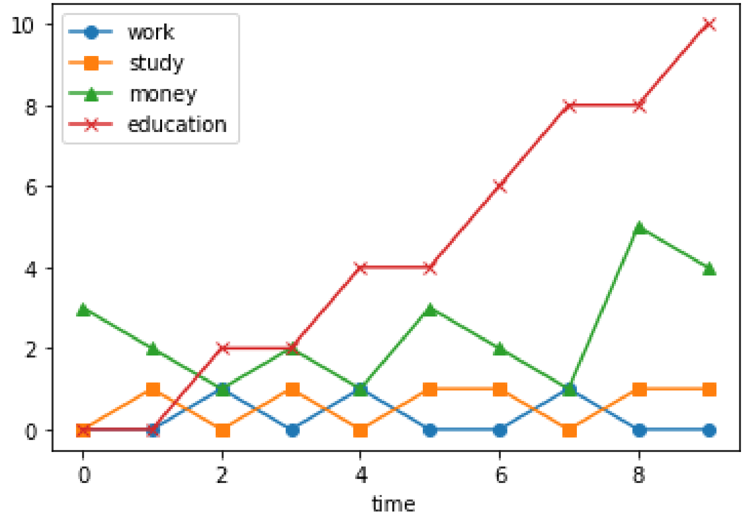
How to think about an optimal policy

1. From the result, do you notice something that is obviously not optimal?
2. Start from the end
3. Work backwards



How to think about an optimal policy

1. From the result, do you notice something that is obviously not optimal?
2. Start from the end
3. Work backwards



Let’s assume (for now) that the salary base level b_t is fixed (not uncertain) and equal to 1. So, the state is only ε_t.

In T, it is optimal to work, no matter the state.

What should I do in T-1?

Dynamic Programming

Stage T: $\max_{u_T}\left\{work_T * \left(1+\frac{\varepsilon_T}{2}\right)\right\}$

MaxReward from stage T $=\left(1+\frac{\varepsilon_T}{2}\right)=V(\varepsilon_T)$

Dynamic Programming

Stage T: $\max_{u_T} \left\{ work_T * \left(1 + \frac{\varepsilon_T}{2}\right) \right\}$

MaxReward from stage T $= \left(1 + \frac{\varepsilon_T}{2}\right) = V(\varepsilon_T)$

Stage T-1: $\max_{u_{T-1}} \left\{ work_{T-1} * \left(1 + \frac{\varepsilon_{T-1}}{2}\right) + V(\varepsilon_T) \right\}$

$$\max_{u_{T-1}} \left\{ work_{T-1} * \left(1 + \frac{\varepsilon_{T-1}}{2}\right) + \left(1 + \frac{\varepsilon_T}{2}\right) \right\}$$

Dynamic Programming

$$\varepsilon_{t+1} = \varepsilon_t + study_t * \rho$$

Stage T: $\max_{u_T}\left\{work_T * \left(1+\frac{\varepsilon_T}{2}\right)\right\}$

MaxReward from stage T $= \left(1+\frac{\varepsilon_T}{2}\right) = V(\varepsilon_T)$

Stage T-1: $\max_{u_{T-1}}\left\{work_{T-1} * \left(1+\frac{\varepsilon_{T-1}}{2}\right) + V(\varepsilon_T)\right\}$

$$\max_{u_{T-1}}\left\{work_{T-1} * \left(1+\frac{\varepsilon_{T-1}}{2}\right) + \left(1+\frac{\varepsilon_T}{2}\right)\right\}$$

function based on my current decison and current state

If work $\rightarrow$ $R_W = \left(1+\frac{\varepsilon_{T-1}}{2}\right) + \left(1+\frac{\varepsilon_{T-1}}{2}\right)$

If study $\rightarrow$ $R_S = 0 + \left(1+\frac{\varepsilon_{T-1}+\rho}{2}\right)$

ro is the factor that eincrese my salary becasue i am more educated

Dynamic Programming

$$\varepsilon_{t+1} = \varepsilon_t + study_t * \rho$$

Stage T: $\max_{u_T} \left\{ work_T * \left(1 + \frac{\varepsilon_T}{2}\right) \right\}$

MaxReward from stage T $= \left(1 + \frac{\varepsilon_T}{2}\right) = V(\varepsilon_T)$

Stage T-1: $\max_{u_{T-1}} \left\{ work_{T-1} * \left(1 + \frac{\varepsilon_{T-1}}{2}\right) + V(\varepsilon_T) \right\}$

$$\max_{u_{T-1}} \left\{ work_{T-1} * \left(1 + \frac{\varepsilon_{T-1}}{2}\right) + \left(1 + \frac{\varepsilon_T}{2}\right) \right\}$$

If work: $R_W = \left(1 + \frac{\varepsilon_{T-1}}{2}\right) + \left(1 + \frac{\varepsilon_{T-1}}{2}\right)$

If study: $R_S = 0 + \left(1 + \frac{\varepsilon_{T-1} + \rho}{2}\right)$

If $R_W > R_S$ → work
Else → study

whatever is rw or rs it is a function of e_t-1

MaxReward from stages T-1 & T $= \max\{R_w, R_S\} = V(\varepsilon_{T-1})$

Stage T-2: $\max_{u_{T-2}} \left\{ work_{T-2} * \left(1 + \frac{\varepsilon_{T-2}}{2}\right) + V(\varepsilon_{T-1}) \right\}$

Dynamic Programming

$$\varepsilon_{t+1} = \varepsilon_t + study_t * \rho$$

Stage T: $\max_{u_T}\left\{work_T * \left(1+\frac{\varepsilon_T}{2}\right)\right\}$

MaxReward from stage T $= \left(1+\frac{\varepsilon_T}{2}\right) = V(\varepsilon_T)$

Stage T-1: $\max_{u_{T-1}}\left\{work_{T-1} * \left(1+\frac{\varepsilon_{T-1}}{2}\right) + V(\varepsilon_T)\right\}$

$$\max_{u_{T-1}}\left\{work_{T-1} * \left(1+\frac{\varepsilon_{T-1}}{2}\right) + \left(1+\frac{\varepsilon_T}{2}\right)\right\}$$

If work → $R_W = \left(1+\frac{\varepsilon_{T-1}}{2}\right) + \left(1+\frac{\varepsilon_{T-1}}{2}\right)$

If study → $R_S = 0 + \left(1+\frac{\varepsilon_{T-1}+\rho}{2}\right)$

If $R_W > R_S$ → work
Else → study

MaxReward from stages T-1 & T $= \max\{R_w, R_S\} = V(\varepsilon_{T-1})$

Stage T-2: $\max_{u_{T-2}}\left\{work_{T-2} * \left(1+\frac{\varepsilon_{T-2}}{2}\right) + V(\varepsilon_{T-1})\right\}$

what i earn for the current decisons + what i can earn the next stage, this function compress all the future into one v(e_T)

Optimal Decision at stage t: $\max_{u_t}\{Reward(x_t,u_t) + V(\varepsilon_{t+1})\} = \max_{u_t}\{Reward(x_t,u_t) + V(x_t,u_t)\}$

Depends on the immediate reward plus the *value* of the next state that I will transition to.

Depends on the current state and current decision

the reward now, imediate reawerd,, and how good is the position i will land into, value fucntion of the next state
I have colapsed the future into this value function

The Value Function

1. The value function $V^{\pi}(x)$ represents the expected reward that can be achieved from a given state x onwards, when a policy π is applied
2. The optimal value function $V(x_t)$ represents the expected reward from state x onwards, when we apply an optimal policy thereafter.

transition function probability

colaps the future into one function

$$V(x_t) = \max_{u_t}\left\{R(x_t, u_t) + \gamma * \sum_{x_{t+1}} P(x_{t+1}|x_t, u_t) * V(x_{t+1})\right\}$$

it can have a good reward now but tomorrow a terrible tomorrow

watch alfa go

The Value Function

1. The value function $V^{\pi}(x)$ represents the expected reward that can be achieved from a given state x onwards, when a policy π is applied
2. The optimal value function $V(x_t)$ represents the expected reward from state x onwards, when we apply an optimal policy thereafter.

$$V(x_t) = \max_{u_t} \left\{ R(x_t, u_t) + \gamma * \sum_{x_{t+1}} P(x_{t+1} | x_t, u_t) * V(x_{t+1}) \right\}$$

- If we have the Optimal Value for each and every state, can we derive the optimal policy? yes,
- How can we calculate the values?

The Value Function

1. The value function $V^{\pi}(x)$ represents the expected reward that can be achieved from a given state x onwards, when a policy π is applied
2. The optimal value function $V(x_t)$ represents the expected reward from state x onwards, when we apply an optimal policy thereafter.

$$V(x_t) = \max_{u_t}\left\{R(x_t, u_t) + \gamma * \sum_{x_{t+1}} P(x_{t+1}|x_t, u_t) * V(x_{t+1})\right\}$$

discounted value of the next state

- If we have the Optimal Value for each and every state, can we derive the optimal policy?

$$\max_{u_t}\left\{R(x_t, u_t) + \gamma * \sum_{x_{t+1}} P(x_{t+1}|x_t, u_t) * V(x_{t+1})\right\}$$

$$\max_{u_t}\{Q(x_t, u_t)\}$$

- How can we calculate the values?

even if it os not convex i can do brute force checking, because it is very small

if we have theh values everythong is easy, how ddo we get the vlues, straung from the end until the beginig, dynamic programmming

Dynamic Program

$$V(x_t) = \max_{u_t}\left\{R(x_t, u_t) + \gamma * \sum_{x_{t+1}} P(x_{t+1}|x_t, u_t) * V(x_{t+1})\right\}$$

Backward Induction:

Calculate the value of the final stage $V_T = \max_{\mathrm{u}} R(x, u)$, for all possible states x_T

we can do becasye it is a deterministic problem

Backward pass:

Use the Bellman equation to calculate the value of T-1, for all possible states x_{T-1},
Use the Bellman equation to calculate the value of T-2, for all possible states x_{T-2} etc…

Forward pass:

$$\max_{u_t}\{R(x_t, u_t) + \gamma\, V(x_{t+1})\}$$
$$s.t.\, x_{t+1} = f(x_t, u_t)$$

Value Iteration

easy form then before

transitioction function

the last update just a number

$$V(x_t) = \max_{u_t} \left\{ R(x_t, u_t) + \gamma \ * \sum_{x_{t+1}} P(x_{t+1} | x_t, u_t) * \ V(x_{t+1}) \right\}$$

2 step algorthm

1. Initialize the Values $V(x_t)$ to random values
2. For each x_t:

   Update $V(x_t)$ using the Bellman equation
3. Repeat step 2 until convergence

   it converges to the optimal values

recursive equantion, we have 3 states and we initiate randomly
i am going to each and we update the state

v1=, v2=0, v3=0

v1= R + v

Achieves the same as the dynamic program (computing the optimal value for each state), but by updating values in parallel instead of going backwards through stages.

Stochastic Programming vs Dynamic Programming

we dont have discrete spaces, in engineering sapces, we have temperatures and storage levels that are continuous because we cannot iterate by all the state becasue they are infinite

Instead of looking L stages into the future:

$$\max_{u_{t,s},x_{t,s}} \left\{ \sum_{t\in L} \sum_{s\in S} \left[\text{Reward}\left(u_{t,s}, x_{t,s}\right)\right] \right\}$$

$$\text{s.t. } \varepsilon_{t+1,s} = \varepsilon_{t,s} + study_{t,s} * \rho, \forall\, t, s$$

non-anticipativity constraints

We look only at the current stage and the value of the expected state we land in:

no look ahead horizon

$$\max_{u_t} \left\{ R(x_t, u_t) + \gamma \frac{1}{|N|} \sum_{n\in N} \tilde{V}\left(\varepsilon_{t+1}, b_{t+1,n}; \theta\right) \right\}$$

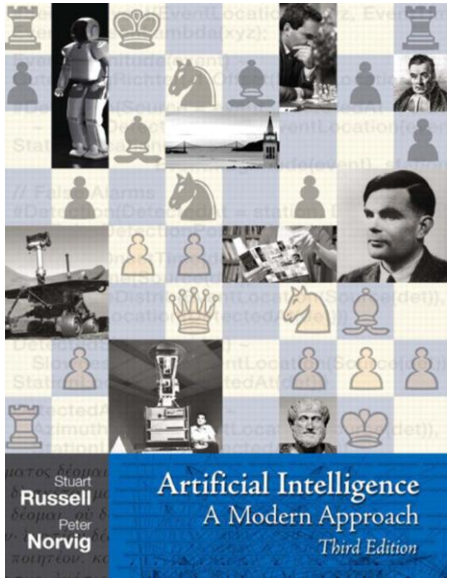
$$s.t.\, \varepsilon_{t+1} = \varepsilon_t + study_t * \rho$$

We can compute the value of every state if there are finitely many states (and not too many).
What if the state space is continuous?

Next week:
We impose a parametric form to the Value Function $\tilde{V}(x) = f(x\,; \theta)$
How can we tune θ such that $\tilde{V}(x;\theta)$ is a good approximation of $V(x)$?

Homework

1. Finish with Tasks 1 & 2
2. Study the new concepts:
   a. Dynamic Programming
   b. Bellman Equation
   c. Value Function
   d. Value Iteration



Chapters
17.1 “Sequential Decision Problems”
17.2 “Value Iteration”

https://www.youtube.com/watch?v=4LW3H_Jinr4&list=PLsOUugYMBBJENfZ3XAToMsg44W7LeUVhF&index=8
https://www.youtube.com/watch?v=JAado8hvJI0
Berkeley “Introduction to Artificial Intelligence” course.

Bertsekas: “Dynamic Programming and Optimal Control”
Sutton & Barto: “Reinforcement Learning”

Next week we will cover Value Function Approximation
a.k.a. Approximate Dynamic Programming,
as a policy for Task 3 of your Assignment

Questions and Survey

Game Quiz