DTU

Decision Making under Uncertainty (02435)

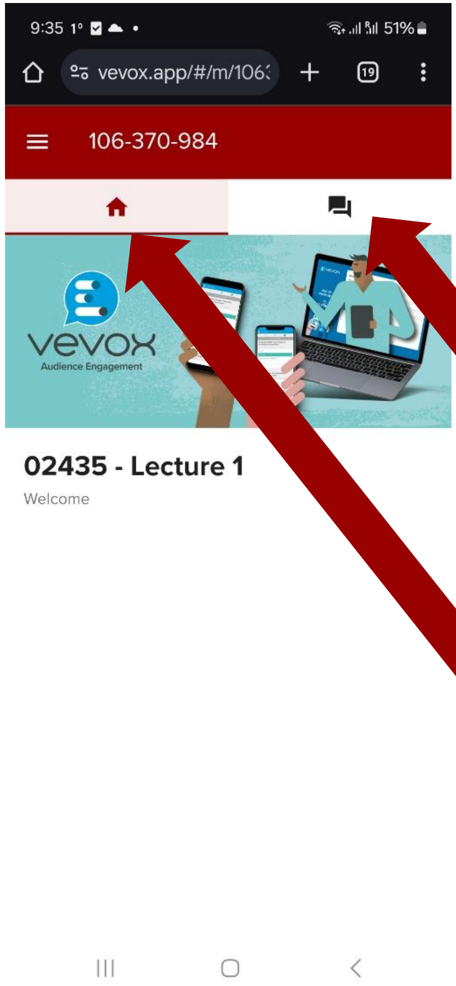
Section for Dynamical Systems, DTU Compute.

DTU Compute
Department of Applied Mathematics and Computer Science

Scan me:

Join at:
vevox.app

ID:
198-414-319



Anonymous Survey (at the end)

Anonymous Questions
(during or after the lecture)

Quizzes

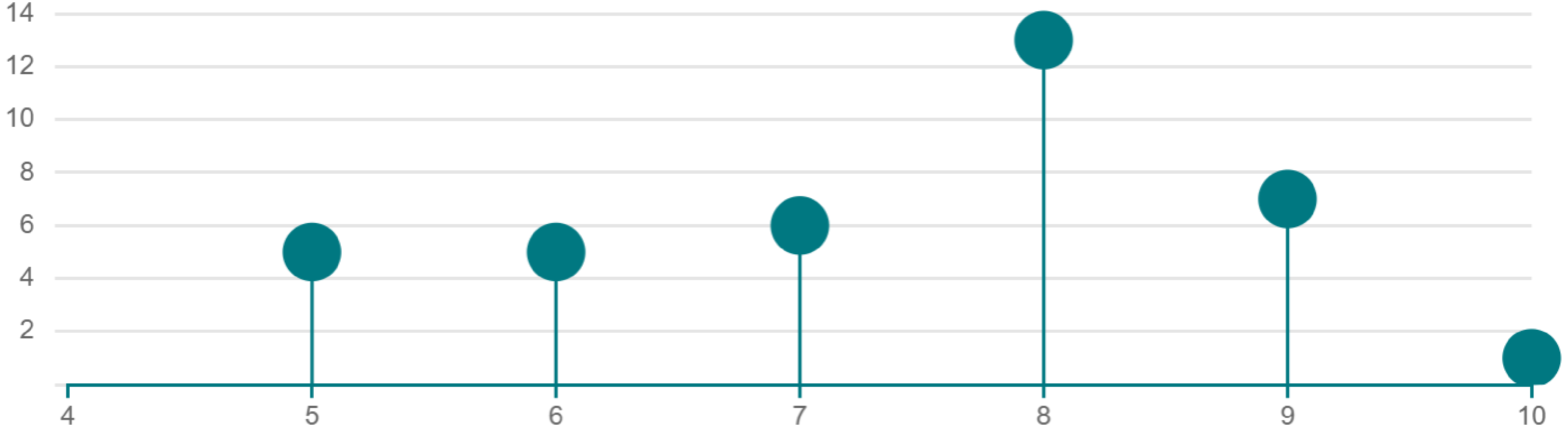
Refer to the updated slides

Feedback & Follow-up

1 How was today (scale 1-10)?

37

Mean average: 7.41



What you liked

Apart from the interesting concepts I liked the extra sources for material at the end and the "homework" section for providing some guidelines on our studying	Examples	Reference to articles where to get more information
It like the relatable examples	Topics	It was a good introduction of the recursive part of the course
Short and sweet, great analogies.	Good examples	Interesting topic
Good examples for the explained theory, very interesting topic today.	Short, nice and good	The questions during the lecture
It was explained well in terms of theory	Fine	Example quite clear
Very short. And good example with work versus study, however I did not understand how to interpret the graph in slide 10	The examples	I liked the introduction to the bellman equation and the connection to RL.
Short and precise. Nice that you mentioned what we are learning next week, so it makes sense that there are something that we may not know yet.	Well structured and informative. Good examples	Honestly i didnt understand all of it. But im probably tired
Example work study	Good, short am precise	I was short and the examples were good
Quick	Good lecture	Examples given

What you disliked

I would prefer no judgement on some of the quiz answers.. Just explain why it is wrong and move on	Nothing	Level of abstraction. Would've liled a proper example
Fewer quiz questions than in the beginning / previous lectures	Would have appreciated to have information on how to proceed with the assignment. We were told last time, that we would learn about approximate programming,...	A bit quick explanation of the bellman function
Start with intuition and then the equation would be a more nice approach	Maybe one relation between the dynamic programming for the project assignment	Lack of explanation for value iteration convergence
I lack coding skills to generate what I learnt in theory so the functions are difficult to write and create (which is generally not taught)	Work/study case more graphical if possible?	You could add more material so we could work on task 3 already
Not understood everything	Did not understand the graph in slide 10	Hard to understand some of the equations, could have helped if they were shown more visual
Pretty good today	No connections to the assignment	N/A
Maybe too brief	NA	

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~~

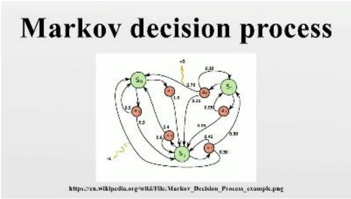
→ Weeks 6-7: Task 3
Approximate Dynamic Programming

→ 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

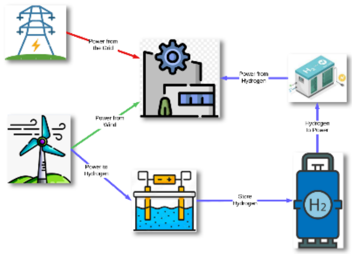
The process of designing “Decision-making” frameworks



Define → Design → Validate

This Week

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



The work vs study problem

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

4. **Reward** = $work_t * b_\mathrm{t} * \left(1 + \frac{E_t}{2}\right)$

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

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

ε_t is endogenous but evolves deterministically

b_t evolves stochastically but is exogenous

How should each be handled?

Join at:
vevox.app

ID:
198-414-319

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** = $work_t * b_\mathrm{t} * \left(1 + \frac{E_t}{2}\right)$

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

s.t. the Transition Function, ∀ t

Stochastic Programming Policy:

1. Create Scenarios for the exogenous state b_t
2. Solve a multistage stochastic program, including the (deterministic) transition dynamics of the endogenous state variables in the constraints

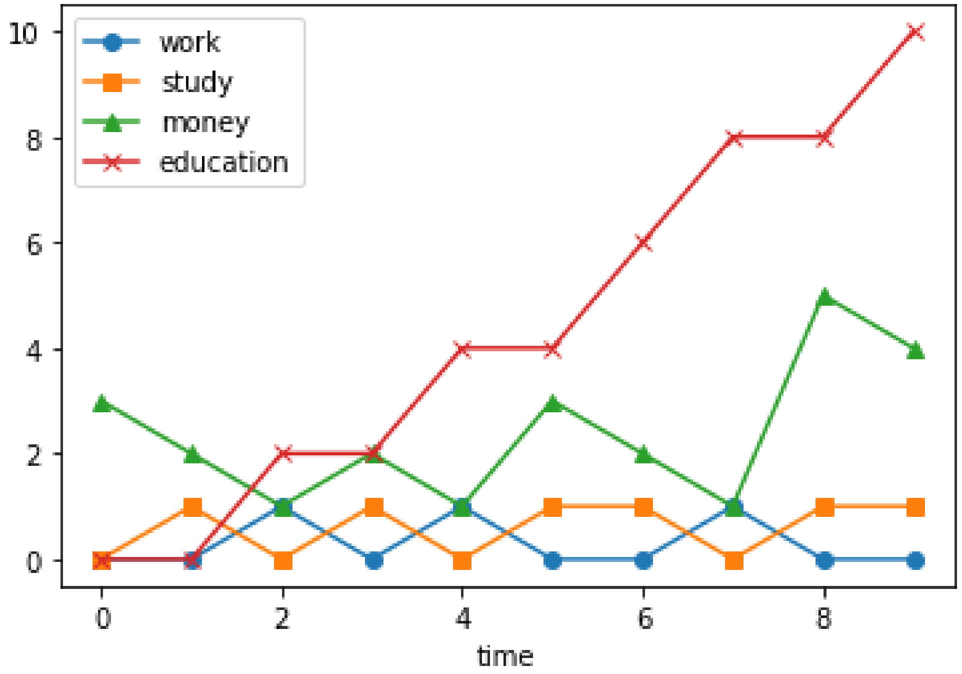
$$\max_{u_{t,s}, x_{t,s}} \left\{ \sum_{t \in L} \sum_{s \in S} \left[\mathrm{Reward}\left(u_{t,s}, x_{t,s}\right)\right] \right\}$$

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?
2. Start from the end
3. Work backwards



The Value Function

1. The optimal value function V(x) represents the maximum reward that can be achieved from a given state x onwards
2. Quantifies the future potential rewards from each state.

$$V(x) = \max_{u} \left\{ R(x,u) + \gamma * \sum_{x'} P(x'|x,u)\, V(x') \right\}$$

General form, relevant for infinite horizon MDPs

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

Time-indexed form, relevant for finite horizon MDPs

The Value Function

1. The optimal value function V(x) represents the maximum reward that can be achieved from a given state x onwards
2. Quantifies the future potential rewards from each state.

$$V(x) = \max_{u} \left\{ R(x, u) + \gamma * \sum_{x'} P(x'|x, u) \, V(x') \right\}$$

General form, relevant for infinite horizon MDPs

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

Time-indexed form, relevant for finite horizon MDPs

- If we have the Value for each and every state, can we derive the optimal policy?
- What is γ?
- How can we calculate the value?

Dynamic Program to compute the Value Function

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

Backward Induction:

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

Backward pass:

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

Forward pass:

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

Dynamic Program to compute the Value Function

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

Backward Induction:

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

Backward pass:

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

Forward pass:

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

If the state-action space is discrete (and small), then $P(x_{t+1} | x_t, u_t)$ can be explicitly represented (in a lookup table).

What if the state-action space is continuous?

Value Function Approximation (VFA)

When the state space is continuous, we cannot represent the exact value of each possible state.

We need an approximate the Value Function

In the VFA approach, we impose a parametric form to the Value Function $\tilde{V}(x) = f(x\,;\theta)$

e.g. linear: $\tilde{V}(x) = \theta^T x$

Therefore, we need to do two things:

1) Learn an approximate Value Function, i.e. impose a parametric form $\tilde{V}(x) = f(x\,;\theta)$ and tune parameters θ such that $\tilde{V}(x;\theta)$ is a good approximation of $V(x)$
2) Use the approximate Value Function to make approximately optimal decisions

Let's first address step 2…

Decision using an Approximate Value Function

Suppose we already have an approximate Value Function $\tilde{V}(x) = f(x\,;\theta)$

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

Recall: “We look only at the immediate reward plus the value of the state we land in”

How can we assess in which state we will land? Think, pair, share.

Decision using an Approximate Value Function

Suppose we already have an approximate Value Function $\tilde{V}(x) = f(x\,; \theta)$

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

Recall: “We look only at the immediate reward plus the value of the state we land in”

How can we assess in which state we will land?

- We use samples for the exogenous uncertainty b_{t+1} to estimate its expected value at $t+1$
- We treat the endogenous uncertainty as a variable and include its deterministic dynamics in the constraints

Thus, we solve:

for the indegenpur we did as varable adn explicitly we put them in the constraints

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

i can take samples from b t+1 it depends on the decison but it depends in a deterministic way

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

i can explitely say what is based on my current decisions

this problem now i have everything and i can solve

i put a deterministic constrint

So, all that’s left to do is to create a good approximate Value Function

i just need a values function, so this policy it will be equally good aproximation from the optimal policy how to have a good ~v?

Training an Approximate Value Function

The previous slide shows how to use an approximate value function to make a decision.

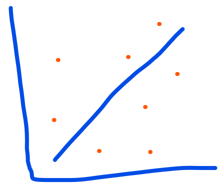
Now, we deal with the problem of training an approximate value function.

First, we impose a parametric form on the approximate value function, e.g. linear:

$$\tilde{V}(\varepsilon_t, b_{t,s}; \theta) = \theta_1 \varepsilon_t + \theta_2 b_t$$

We need to determine θ_1, θ_2

it is a superving learning program



~v is the line
the dots are my real value function for different states, i can discretize the space i and calculate

For $t = T, T-1, \ldots, \tau$:

1. Sample representative state pairs $\{(b_t^i, \varepsilon_t^i)\}_{i=1}^I$
2. Iterate N times (repeat loop):
   - For each sample (b_t^i, ε_t^i):

     take samples of states, any state will do, to evaluate function

     1. Sample K next exogenous states:

        $\{b_{t+1,k}^i\}_{k=1}^K \sim P(b_{t+1} \mid b_t^i)$

        i sample k sample of the next satge, on the very last this is not there, for every other one yes

     2. Compute target value:

        several of this ones

        i is sample i take value function

$$V_t^{\text{target},i} = \max_{u_t} \left[r(b_t^i, \varepsilon_t^i, u_t) + \frac{\gamma}{K} \sum_{k=1}^K \tilde{V}(b_{t+1,k}^i, \varepsilon_{t+1}; \theta) \right]$$

i max for each sample init, the immediate reward for that state + sample the next state for the exogenous variable and for the endogenpous I explitely put in the variable

subject to:

v target is a function of v~
with each iteraction we insert a shot, the reward, the information about the reward propagates

$$\varepsilon_{t+1} = f(\varepsilon_t^i, u_t)$$

   - Update the parameter θ by minimizing squared error:

$$\theta \leftarrow \arg\min_\theta \sum_{i=1}^I \left(\tilde{V}(b_t^i, \varepsilon_t^i; \theta) - V_t^{\text{target},i} \right)^2$$

fit the least squares to minimize the distance, to draw the line

Training an Approximate Value Function

The previous slide shows how to use an approximate value function to make a decision.

Now, we deal with the problem of training an approximate value function.

First, we impose a parametric form on the approximate value function, e.g. linear:

$$\tilde{V}\left(\varepsilon_t, b_{t,s}; \theta\right) = \theta_1 \varepsilon_t + \theta_2 b_t$$

We need to determine θ_1, θ_2

This training involves solving many optimization problems (albeit small ones). But: it can be done offline since it is not necessarily specific to a given observed state.

For $t = T, T-1, \ldots, \tau$:

1. Sample representative state pairs $\{(b_t^i, \varepsilon_t^i)\}_{i=1}^I$
2. Iterate N times (repeat loop):
   - For each sample (b_t^i, ε_t^i):
     1. Sample K next exogenous states: $\{b_{t+1,k}^i\}_{k=1}^K \sim P(b_{t+1} \mid b_t^i)$
     2. Compute target value:

$$V_t^{\text{target},i} = \max_{u_t} \left[r(b_t^i, \varepsilon_t^i, u_t) + \frac{\gamma}{K} \sum_{k=1}^{K} \tilde{V}(b_{t+1,k}^i, \varepsilon_{t+1}; \theta) \right]$$

subject to:

v tild can be initialized random, if we have some expert we can try to start in a way to cinverge quickly

$$\varepsilon_{t+1} = f(\varepsilon_t^i, u_t)$$

   - Update the parameter θ by minimizing squared error:

$$\theta \leftarrow \arg\min_{\theta} \sum_{i=1}^{I} \left(\tilde{V}(b_t^i, \varepsilon_t^i; \theta) - V_t^{\text{target},i} \right)^2$$

highl lvel picture, when we execute the policy this optimization problem is very small, computing the v tild in not easy but it can be done offline, it is not relevant for the assignment, it is not necessary

VFA policy for the Electrolyzer Problem

the reciper

Input: current state y_τ, z_τ, where y_τ are the endogenous state variables and z_τ the exogenous

Step 1: Backward Value Function Approximation

For $t = T, T-1, \ldots, \tau$:

instead of using the work vs stydu problw, I use zeta for exogenus and

1. Sample representative state pairs $\{(z_t^i, y_t^i)\}_{i=1}^I$
2. Iterate N times (repeat loop):
   - For each sample (z_t^i, y_t^i):
     1. Sample K next exogenous states: $\{z_{t+1,k}^i\}_{k=1}^K \sim P(z_{t+1} \mid z_t^i)$
     2. Compute target value:

$$V_t^{\text{target},i} = \max_{u_t} \left[r(z_t^i, y_t^i, u_t) + \frac{\gamma}{K} \sum_{k=1}^{K} \tilde{V}(z_{t+1,k}^i, y_{t+1}; \theta) \right]$$

subject to:

$$y_{t+1} = f(y_t^i, u_t)$$

   - Update the parameter θ by minimizing squared error:

$$\theta \leftarrow \arg\min_\theta \sum_{i=1}^{I} \left(\tilde{V}(z_t^i, y_t^i; \theta) - V_t^{\text{target},i} \right)^2$$

Step 2: Policy Execution

1. At time τ, given current state (z_τ, y_τ), compute optimal action u_τ:
   - Sample $|S|$ next exogenous states:

i observe current state and i sampleS, can be large number, for the nex ralization o exogenous values

$$\{z_{\tau+1,s}\}_{s=1}^S \sim P(z_{\tau+1} \mid z_\tau)$$

   - Compute:

$$u_\tau = \arg\max_{u_\tau} \left[r(z_\tau, y_\tau, u_\tau) + \gamma \frac{1}{|S|} \sum_{s=1}^{S} \tilde{V}(z_{\tau+1,s}, y_{\tau+1}; \theta) \right]$$

subject to:

solve this, relatively easy to

v tild needs to be linear to be easy to solve

$$y_{\tau+1} = f(y_\tau, u_\tau)$$

my output here and now decision

Output: current decisions u_τ

reward linear, if the val.ue function is also linear
when is a linear value fucntion good enough, non linear rewar

can you have non linear value function

Questions and Survey

Join at:
vevox.app

ID:
198-414-319