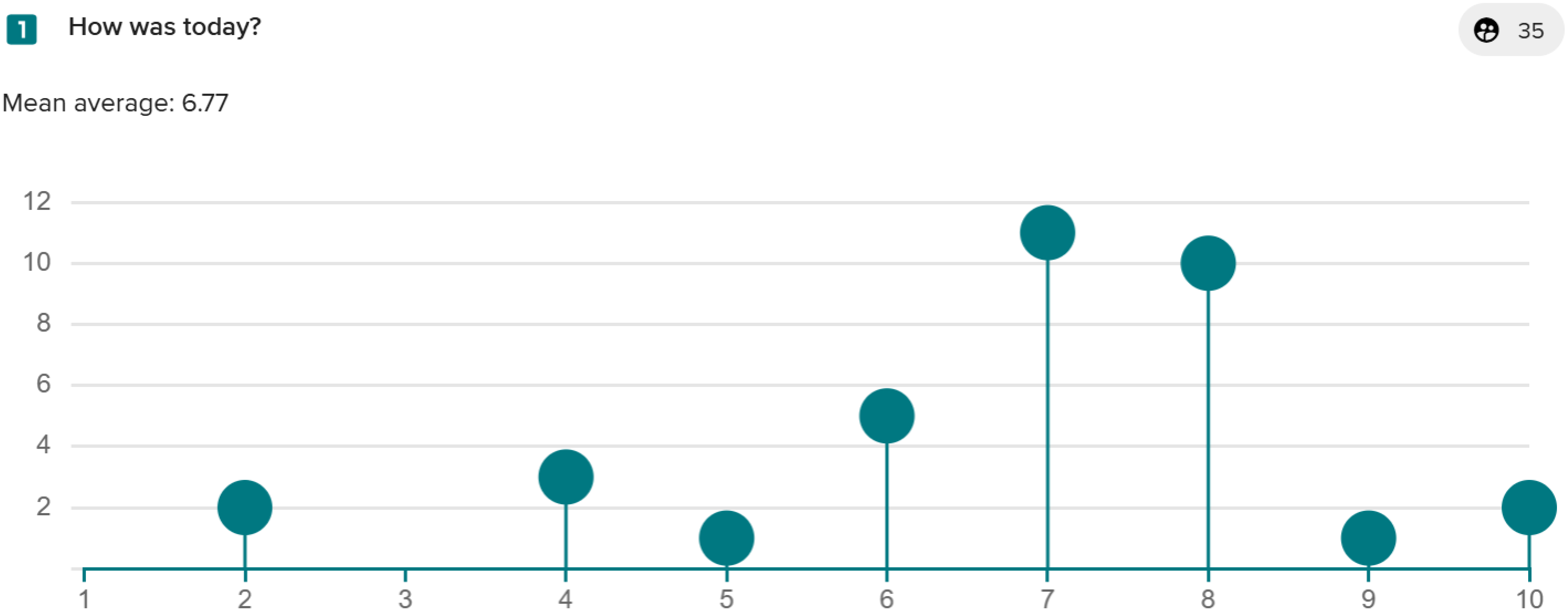
DTU

Decision Making under Uncertainty (02435)

Section for Dynamical Systems, DTU Compute.

DTU Compute
Department of Applied Mathematics and Computer Science

Feedback & Follow-up



What you liked

The flow	Very interesting stuff	Clearer than last week
Nice presentation	There wasn't anything particular that I didn't like. I would really lile some material to practice on, like some exercises as well as coding exercises.	I like that the lecture is focusing on what's relevant for the report. It gives a clear and structured way of thinking when working on the tasks.
Recap of previous policies and purpose of approximate dynamic programming	It was a short lecture	Good description of material
Quick lecture	Theory	As always: very motivational presentation.
Going over example	-	The algorithm for task 3
Nice explanations	Nice and short	It wa nice. Clear structure and well explained
Short duration of lecture	Interesting topic	Everything

What could be improved

There wasn't a recap of the precious week

The explanation for this amount of complexity was not completely satisfactory. I had a hard time grasping everything. Perhaps you could consider including visual...

A longer course with more explanations and examples

More specific examples with numbers

More exercises

Sometimes it did take a little side track for stuff not relevant for the assignment. But this is minor.

No example for task 3 in the assignment

Everything was confusing and very fast, I am not really sure what I am supposed to be learning. Because I know I will be spending 90% of the time and energy figure out ho...

Need for more examples as well as more python guidelines

Show more examples of how to apply Dynamic Programming in the project.

why can we only give integer ratings? Had to go down 0.5 since you don't allow 8.5

I don't know

In would have liked more examples along the explanations.

Some more numerical results with the example could be provided. In general, we could spend more time during the lectures to explain the materiał more/cover more...

-

It was a complex topic to understand, perhaps bringing an (easy) example focused on the value function would've been great.

More time could be used to analyze the concepts in depth

A bit quick walkthrough, not very concise example of how to actually approximate the value function

I would like some actual code, the pseudo code does not make it easy to visualize for me

The concepts could use an example or two that are more into the topic than study/education

Too theoretical, an example with data would be great with helping understand

Too short short. I would have loved more details.

More theory regarding the approximations we are assuming

I would have liked more detail on model-based RL

Really hard to understand how to use it for our assignment

AI

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

VFA/ADP policy for the Electrolyzer Problem

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 $t = T, T-1, \dots, \tau$:

1. Sample representative state pairs

$$\{(\mathbf{z}_t^i, \mathbf{y}_t^i)\}_{i=1}^I$$

2. For each sample $(\mathbf{z}_t^i, \mathbf{y}_t^i)$:
   1. Sample K next exogenous states:

$$\{\mathbf{z}_{t+1,k}^i\}_{k=1}^K \sim P(\mathbf{z}_{t+1} \mid \mathbf{z}_t^i)$$

   2. Compute target value:

$$V_t^{\text{target},i} = \max_{\mathbf{u}_t} \left[r(\mathbf{z}_t^i, \mathbf{y}_t^i, \mathbf{u}_t) + \frac{\gamma}{K} \sum_{k=1}^K \tilde{V}(\mathbf{z}_{t+1,k}^i, \mathbf{y}_{t+1}; \boldsymbol{\theta}_{t+1}) \right]$$

subject to:

$$\mathbf{y}_{t+1} = f(\mathbf{y}_t^i, \mathbf{u}_t)$$

3. Update the parameter $\boldsymbol{\theta}_t$ by minimizing squared error:

$$\boldsymbol{\theta}_t \leftarrow \arg\min_{\boldsymbol{\theta}_t} \sum_{i=1}^I \left(\tilde{V}(\mathbf{z}_t^i, \mathbf{y}_t^i; \boldsymbol{\theta}_t) - V_t^{\text{target},i} \right)^2$$

Step 2: Policy Execution

1. At time τ, given current state $(\mathbf{z}_\tau, \mathbf{y}_\tau)$, compute optimal action $\mathbf{u}_\tau$:
   - Sample $|S|$ next exogenous states:

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

   - Compute:

$$\mathbf{u}_\tau = \arg\max_{\mathbf{u}_\tau} \left[r(\mathbf{z}_\tau, \mathbf{y}_\tau, \mathbf{u}_\tau) + \gamma \frac{1}{|S|} \sum_{s=1}^S \tilde{V}(\mathbf{z}_{\tau+1,s}, \mathbf{y}_{\tau+1}; \boldsymbol{\theta}_{\tau+1}) \right]$$

subject to:

$$\mathbf{y}_{\tau+1} = f(\mathbf{y}_\tau, \mathbf{u}_\tau)$$

Output: current decisions u_τ

$$min_{u_t} \left\{ c_t(u_t) + \sum_{x_{t+1}} P(x_{t+1} | x_t, u_t) \, V(x_{t+1}) \right\}$$

VFA/ADP policy for the Electrolyzer Problem

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

1. **Sample representative state pairs**

$$\{(\mathbf{z}_t^i, \mathbf{y}_t^i)\}_{i=1}^I$$

2. **For each sample $(\mathbf{z}_t^i, \mathbf{y}_t^i)$:**

   1. **Sample K next exogenous states:**

$$\{\mathbf{z}_{t+1,k}^i\}_{k=1}^K \sim P(\mathbf{z}_{t+1} \mid \mathbf{z}_t^i)$$

   2. **Compute target value:**

$$V_t^{\text{target},i} = \max_{\mathbf{u}_t} \left[r(\mathbf{z}_t^i, \mathbf{y}_t^i, \mathbf{u}_t) + \frac{\gamma}{K} \sum_{k=1}^K \tilde{V}(\mathbf{z}_{t+1,k}^i, \mathbf{y}_{t+1}; \boldsymbol{\theta}_{t+1}) \right]$$

   subject to:

$$\mathbf{y}_{t+1} = f(\mathbf{y}_t^i, \mathbf{u}_t)$$

3. **Update the parameter $\boldsymbol{\theta}_t$ by minimizing squared error:**

$$\boldsymbol{\theta}_t \leftarrow \arg\min_{\boldsymbol{\theta}_t} \sum_{i=1}^I \left(\tilde{V}(\mathbf{z}_t^i, \mathbf{y}_t^i; \boldsymbol{\theta}_t) - V_t^{\text{target},i} \right)^2$$

Step 2: Policy Execution

1. **At time τ, given current state $(\mathbf{z}_\tau, \mathbf{y}_\tau)$, compute optimal action $\mathbf{u}_\tau$:**

   - Sample $|S|$ next exogenous states:

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

   - Compute:

$$\mathbf{u}_\tau = \arg\max_{\mathbf{u}_\tau} \left[r(\mathbf{z}_\tau, \mathbf{y}_\tau, \mathbf{u}_\tau) + \boxed{\gamma \frac{1}{|S|} \sum_{s=1}^S} \boxed{\tilde{V}(\mathbf{z}_{\tau+1,s}, \mathbf{y}_{\tau+1}; \boldsymbol{\theta}_{\tau+1})} \right]$$

   subject to:

$$\boxed{\mathbf{y}_{\tau+1} = f(\mathbf{y}_\tau, \mathbf{u}_\tau)}$$

Output: current decisions u_τ

$$min_{u_t} \left\{ c_t(u_t) + \sum_{x_{t+1}} P(x_{t+1} | x_t, u_t)\, V(x_{t+1}) \right\}$$

VFA/ADP policy for the Electrolyzer Problem

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

1. Sample representative state pairs

$$\{(\mathbf{z}_t^i, \mathbf{y}_t^i)\}_{i=1}^I$$

2. For each sample $(\mathbf{z}_t^i, \mathbf{y}_t^i)$:

   1. Sample K next exogenous states:

$$\{\mathbf{z}_{t+1,k}^i\}_{k=1}^K \sim P(\mathbf{z}_{t+1} \mid \mathbf{z}_t^i)$$

   2. Compute target value:

$$V_t^{\text{target},i} = \max_{\mathbf{u}_t} \left[r(\mathbf{z}_t^i, \mathbf{y}_t^i, \mathbf{u}_t) + \frac{\gamma}{K} \sum_{k=1}^K \tilde{V}(\mathbf{z}_{t+1,k}^i, \mathbf{y}_{t+1}; \boldsymbol{\theta}_{t+1}) \right]$$

   subject to:

$$\mathbf{y}_{t+1} = f(\mathbf{y}_t^i, \mathbf{u}_t)$$

3. Update the parameter $\boldsymbol{\theta}_t$ by minimizing squared error:

$$\boldsymbol{\theta}_t \leftarrow \arg\min_{\boldsymbol{\theta}_t} \sum_{i=1}^I \left(\tilde{V}(\mathbf{z}_t^i, \mathbf{y}_t^i; \boldsymbol{\theta}_t) - V_t^{\text{target},i} \right)^2$$

Step 2: Policy Execution

1. At time τ, given current state $(\mathbf{z}_\tau, \mathbf{y}_\tau)$, compute optimal action $\mathbf{u}_\tau$:

- Sample $|S|$ next exogenous states:

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

- Compute:

$$\mathbf{u}_\tau = \arg\max_{\mathbf{u}_\tau} \left[r(\mathbf{z}_\tau, \mathbf{y}_\tau, \mathbf{u}_\tau) + \boxed{\gamma \frac{1}{|S|} \sum_{s=1}^S} \boxed{\tilde{V}(\mathbf{z}_{\tau+1,s}, \mathbf{y}_{\tau+1}; \boldsymbol{\theta}_{\tau+1})} \right]$$

subject to:

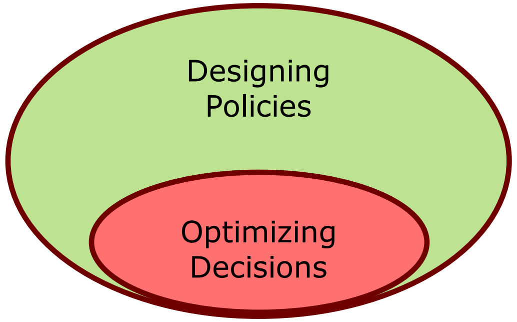
$$\boxed{\mathbf{y}_{\tau+1} = f(\mathbf{y}_\tau, \mathbf{u}_\tau)}$$

Output: current decisions u_τ

$$min_{u_t} \left\{ c_t(u_t) + \sum_{x_{t+1}} P(x_{t+1} | x_t, u_t)\, V(x_{t+1}) \right\}$$

Extras: what if the uncertainty is decision-dependent? e.g. epidemics control, strategic decisions, economic policies, maintenance decisions, etc.

Policy Classes



Policy Class	Description	Examples
Direct Lookahead	Optimize Decisions over a Lookahead horizon	MPC, Stochastic Optimization $\boldsymbol{u} = (u_\tau)_{\tau \in \mathcal{H}}$ $u_t = argmin_{\boldsymbol{u}}\{\sum_{\tau \in \mathcal{H}} c_\tau\,(u_\tau, x_\tau)\}$
Value Function Approximation	Optimize the current reward + the Value of the next state	Dynamic Programming (Bellman) $u_t = argmin_{u_t}\left\{c_t(u_t) + \sum_{x_{t+1}} P(x_{t+1} \mid x_t, u_t)\, V(x_{t+1})\right\}$
Cost Function Approximation	Greedy Deterministic Optimization but imposing a heuristic Slack, or a Penalty for aggressive actions	$u_t = argmin_{u_t}\{c_t(u_t)\}$ s.t. $\theta_1 \leq \sum_{x_{t+1}} P(x_{t+1} \mid x_t, u_t)\, x_{t+1} \leq \theta_2$
Policy Function Approximation	Parameterized Policy Function	LQR: $u_t = -Kx_t$ Rule-based: $if\ x_t > x_a, then\ u_t = u_b$ Neural Networks: $u_t = NN(x_t)$

Considerations for choosing a policy class

- How accurately can the transition dynamics be modeled?
- Are the states exogenous or endogenous?
- Do they evolve stochastically or deterministically?
- Are the state and action spaces low or high dimensional?
- How much time is there to make a decision (online)?
- Does the MDP have delayed rewards or temporally-coupled (critical) constraints?
- Is there access to expert knowledge?
- How important is policy interpretability?

The ultimate test is evaluating and comparing policies in a good simulation environment.

Resources for Approximate Dynamic Programming

1) Warren Powell, Approximate Dynamic Programming: Solving the curses of dimensionality
2) Warren Powell, Reinforcement Learning and Stochastic Optimization
3) Mykel Kochenderfer, Algorithms for Decision Making
4) Sutton & Barto, Reinforcement Learning
5) Dimitri Bertsekas, Dynamic Programming and Optimal Control