- January 22, 2021 Notes: Chapter 5, Dynamic Programming

  - Partially Observable Markov Decision Process (POMDP)

    ↳ These are more common in real world

    ↳ Two notions of state

    ① Environment can be viewed as a state

    ↳ We don't know the factors driving this. We view accumulated effect.

    ↳ $S_t^{(e)}$

    ② State you need to decide on next action.

    ↳ $S_t^{(a)}$

    ↳ Previously we assumed $S_t^{(e)} = S_t^{(a)} = S_t$, but this is not realistic

    ↳ More generally, you see observations O_t and infer State $S_t^{(e)}$ based on history of observations

    ↳ Observation function:

Modeled based on understanding of business problem. Defined a priori. → $Z(s', a, o) = P[O_{t+1} = o \mid S_{t+1} = s', A_t = a]$

    ↳ You still have probabilities P_R specified in POMDP

    ↳ MDP is special case of POMDP where:

    $$O_t = S_t = S_t^{(e)} = S_t^{(a)}$$

    ↳ POMDP "guesses" S_t by maintaining Belief State

    ↳ See Slide 30 of Chapter 2 slides

↳ Belief states satisfy Markov property, but history must be recorded.

↳ Belief state of POMDP can be used as a single state of a giant MDP, which can be solved.

↳ Not practical, since state space is huge.

↳ Alternatively, treat observations as states.

↳ There are advanced techniques to solve POMDP, which lies beyond scope of class

* No assignments or exam questions on POMDP's; this is a current area of research.

- Chapter 2 Takeaways:
  - MDP Bellman Policy Equations V^π
  - MDP Bellman Optimality Equation V^*
  - There exists an optimal policy, and each optimal policy satisfies optimal value function

- Chapter 3: Dynamic Programming
  - Focus on proofs in textbook
  - DP for both prediction and control problems
    - Predict V^π
    - Predict V^*
  - Planning vs. Learning
  - Bellman referred to DP as the Principle of Optimality
    - ↳ In CS, DP is used more broadly to mean "recursive algorithms with overlapping subproblems"
    - ↳ We refer to DP as "Algorithms for prediction and control."

- We will cover 3 DP algorithms based on the concept of Fixed Point

  ↳ Fixed point of a function $f : X \to X$ is a point $x \in X$ such that $x = f(x)$

  ↳ E.g. $f(x) = \cos(x)$, fixed point: $x^* = \cos(x^*)$

    ↳ Applying a convergent function over and over again to solve this.

  ↳ <u>Banach Fixed-Point Theorem</u>

  Properties satisfied by physical distances

  X is non-empty set equipped with a complete metric $d : X \times X \to \mathbb{R}$.

  Think of d as a distance

  Let $f : X \to X$ be such that there exists an $L \in [0, 1)$ such that $d(f(x_1), f(x_2)) \leq L \cdot d(x_1, x_2)$ for all $x_1, x_2 \in X$

  Then:

  There exists a unique Fixed point $x^* \in X$

  * Proof of this theorem is beyond scope of course, but we will use this theorem

- <u>Policy Evaluation</u> (Prediction)

  - MDP with $S = \{s_1, s_2, \ldots, s_n\}$, $N = \{s_1, s_2, \ldots, s_m\}$

  - Policy π

  - Goal: find iterative algorithm to solve for π-Implied MRP value function iteratively for large data sets

    ↳ Evaluate Bellman Equation

↳ Define Bellman Policy operator that serves as L in Banach fixed-point theorem.

$$B^\pi : \mathbb{R}^m \to \mathbb{R}^m$$

↑ Vector representing a value function

$$\therefore \quad B^\pi(V) = R^\pi + \gamma P^\pi \cdot V$$

for any Value function vector $V \in \mathbb{R}^m$

↳ This can be re-expressed using V^π

$$\therefore \quad V^\pi = B^\pi(V^\pi)$$

↑ This is a fixed point!

- Metric $d : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}$ defined as L^∞ norm.

↳ Use this to prove B^π is a contraction function under L^∞ norm

$$d(X, Y) = \|X - Y\|_\infty = \max_{s \in N} |(X - Y)(s)|$$

↳ See slides 6-7 for proof. γ takes role of L in Banach fixed point theorem to prove B^π is a contraction

- <u>Policy Evaluation Algorithm:</u> Start with any Value Function $V_0 \in \mathbb{R}^m$

↳ Iterate over $i : 0, 1, 2, \ldots,$ calculate in each iteration:

$$V_{i+1} = B^\pi(V_i) = R^\pi + \gamma P^\pi \cdot V_i$$

↳ Stop within tolerance

↳ Lower γ speeds up algorithm

↳ Runtime of each iteration is $O(m^2)$

↳ Constructing MRP from MDP + policy is $O(m^2 k)$, where k is the number of actions

- **Greedy Policy**: MDP Control Problem

  ↳ "Greedy" policy improvement technique

  ↳ Greedy Policy Function:

$$G : \mathbb{R}^m \longrightarrow \underbrace{(N \to A)}_{\text{Deterministic policy}}$$

$$\underbrace{\underbrace{G(V)}_{\text{given policy}}(s)}_{\text{gives action}} = \pi'_D(s) = \underset{a \in A}{\operatorname{argmax}} \left\{ R(s,a) + \gamma \sum_{s' \in N} P(s,a,s') \cdot V(s') \right\}$$

(π'_D: Deterministic Policy)

- **Value Function Comparison**: We say $X \geq Y$ iff $X(s) \geq Y(s)$ for all s

  (values, functions)

- **Policy Improvement Theorem**:

$$V^{\pi'_D} = V^{G(V^\pi)} \geq V^\pi$$

($V^{\pi'_D}$: Value function; improved policy; $G(V^\pi)$: Greedily improved policy)

We need to prove this.

- Policy → Value Function → Greedy Policy → Value function of Greedy Policy

- Review slide 10, around 1 hr mark of lecture for proof. Prove By Induction

  ↳ End up with a tuple of slightly improved value functions that converges on V^{π_D}

  ↳ Base case, note:

$$B^{\pi_D}(V^{\pi})(s) = \max_{a \in A} \left\{ R(s,a) + \gamma \sum_{s' \in N} P(s,a,s') V^{\pi}(s') \right\}$$

(π_D: greedy policy, deterministic)

$$= \max_{a \in A} Q^{\pi}(s,a)$$

  ↳ $V^{\pi}(s)$ is a weighted average over $Q^{\pi}(s,\cdot)$, while $B^{\pi_D}(V^{\pi})(s)$ is a maximum value, which will always be greater than or equal to weighted avg.

  ↳ For induction step, use Monotonicity of B^{π} operator (for any π)

  ↳ $B^{\pi}: X \geq Y \Rightarrow B^{\pi}(X) \geq B^{\pi}(Y)$

  ↳ Prove this!

  ↳ Use monotonicity problem to show

$$(B^{\pi_D})^{i+1}(V^{\pi}) \geq (B^{\pi_D})^{i}(V^{\pi})$$

* Review lecture from slide 12, 1 hr, 10 min mark

- Policy Iteration Algorithm: Start with Value Function $V_0 \in \mathbb{R}^m$

  - Iterate over $j = 0, 1, 2, \ldots$, calculate for each iteration:

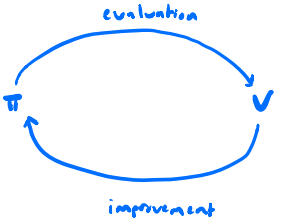
Deterministic Policy $\pi_{j+1} = G(V_j)$

Value Function $V_{j+1} = \lim_{i \to \infty} (B^{\pi_{j+1}})^i (V_j)$

- Continue until V_j converges

* Iterate between Value Function improvement and Policy Evaluation

↳ See slide 15.



↳ Running Time : $O(n^2 k)$ ⟵ How do you prove this ?

- <u>Value Function Iteration :</u>

↳ Bellman Optimality Operator

↳ Tweak definition of Greedy Policy function (argmax & max)

↳ Bellman Optimality Operator $B^* : \mathbb{R}^n \to \mathbb{R}^n$ defined as :

$$B^*(V)(s) = \max_{a \in A} \left\{ R(s,a) + \gamma \sum_{s' \in N} P(s,a,s') \cdot V(s') \right\}$$

- Review Slide 17

V^* is fixed point of B^*

↳ $V^* = B^*(V^*)$

↳ Prove that V^* is a contraction in next lecture

↳ Beginning on Slide 19