Lecture 16: Markov Decision Processes CS486/686 Intro to Artificial Intelligence

2023-7-04

Sriram Ganapathi Subramanian,
Vector Institute

UNIVERSITY OF WATERLOO

Instructor

- Sriram Ganapathi Subramanian
  - Postdoctoral Fellow at the Vector Institute
  - Previously: PhD Student at the University of Waterloo
  - Expertise: Multi-agent reinforcement learning, reinforcement learning, game theory, machine learning, deep learning

VECTOR INSTITUTE

UNIVERSITY OF WATERLOO

UNIVERSITY OF WATERLOO

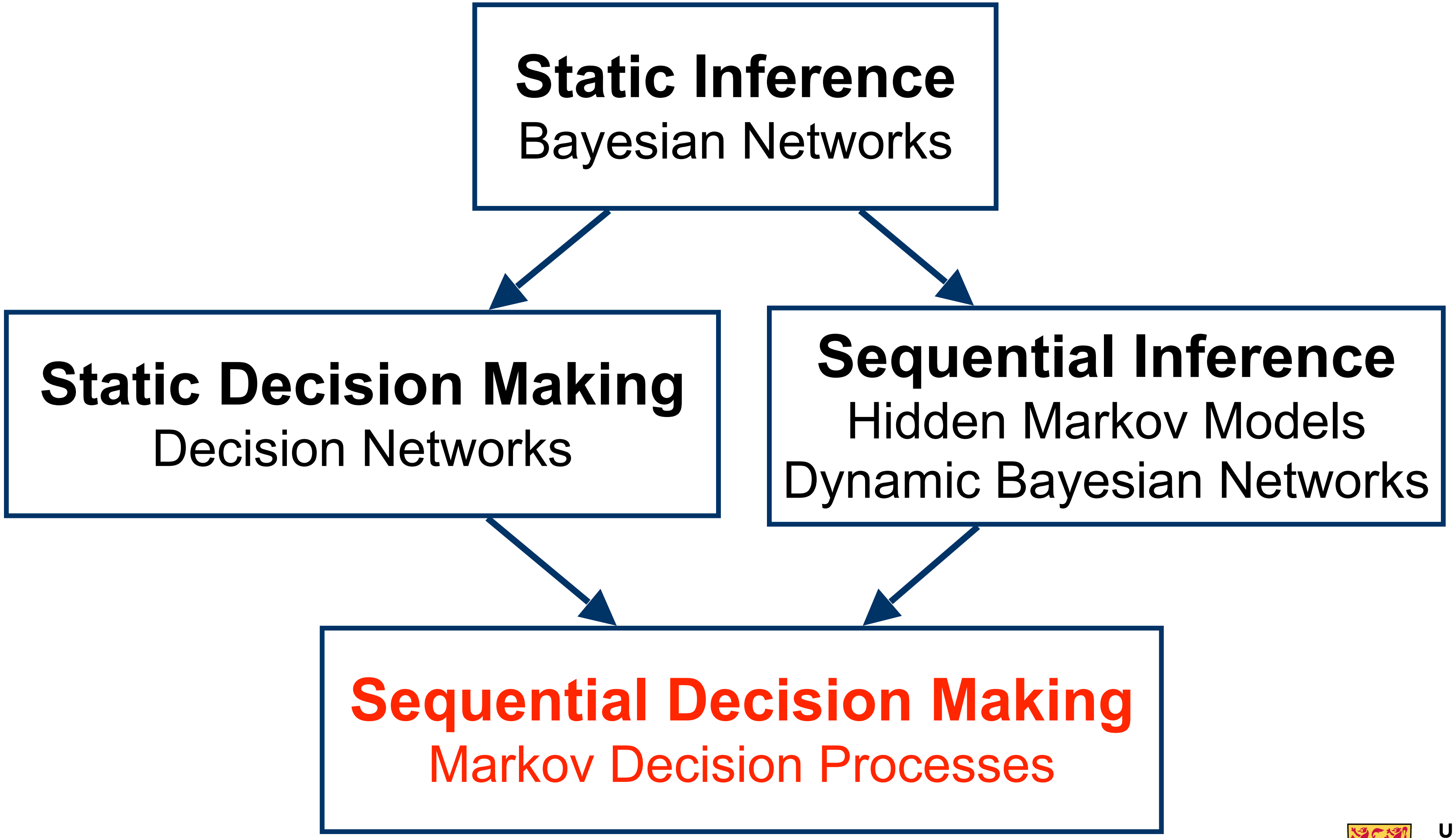
Logistics

- Private meetings:
  - Mondays 11 am — 1.30 pm
  - Appointments required
  - Online or in-person (DC-2584)
  - Book in Calendly
- Open office hours:
  - No appointments required (MC-2035)
  - Tuesdays and Thursdays 5.30 pm — 6.30 pm
- Lectures:
  - Slides available on course website
  - Recorded

Outline

- Markov Decision Processes
  - Value Iteration
  - Policy Iteration

Sequential Decision Making



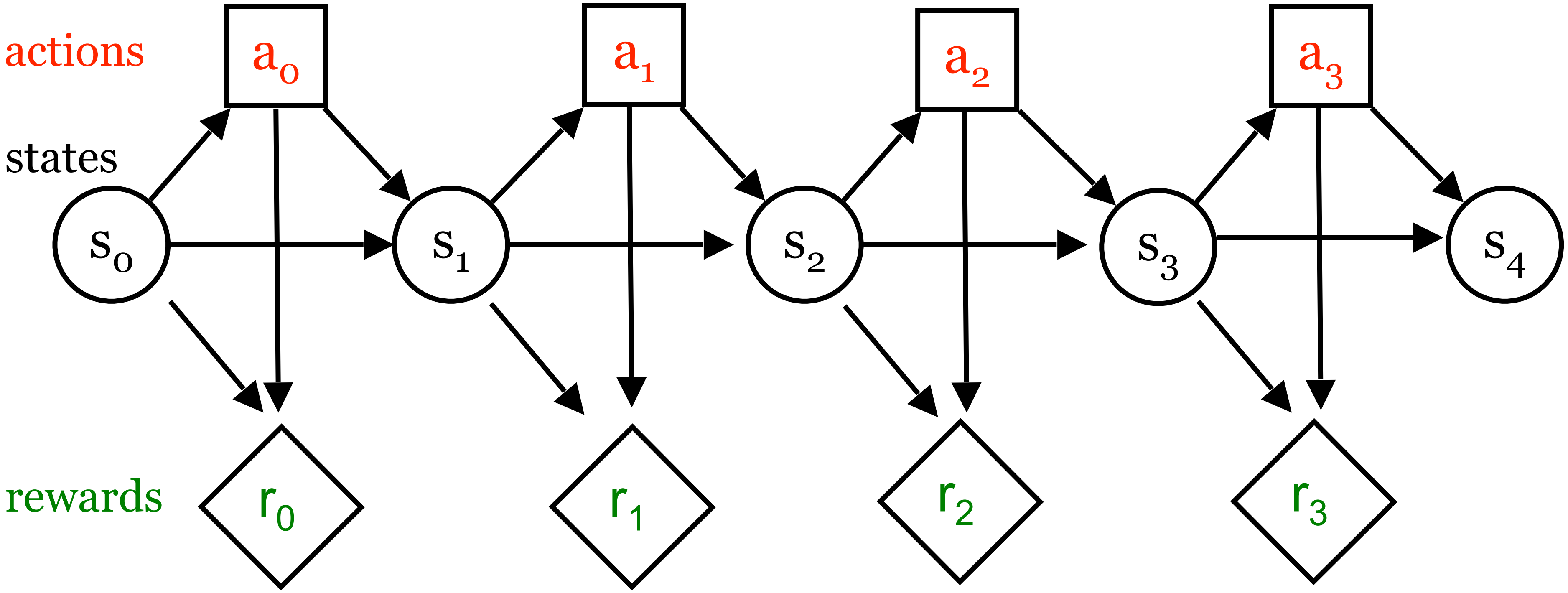
UNIVERSITY OF WATERLOO

Sequential Decision Making

- Applications
  - Robotics (e.g., control)
  - Investments (e.g., portfolio management)
  - Computational linguistics (e.g., dialogue management)
  - Operations research (e.g., inventory management, resource allocation, call admission control)
  - Assistive technologies (e.g., patient monitoring and support)

Markov Decision Processes

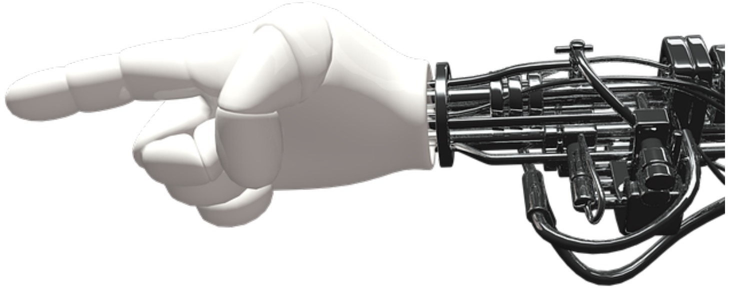
- Indefinite/Infinite Horizon Decision Networks
- Large Finite Horizon Decision Networks



UNIVERSITY OF WATERLOO

Examples

- Robotic control
  - States: $\langle x, y, z, \theta \rangle$ coordinates of joints
  - Actions: forces applied to joints
  - Rewards: - distance to goal position



- Inventory management
  - States: inventory level
  - Actions: {doNothing, orderWidgets}
  - Rewards: sales - costs - storage

Markov Decision Processes

- Formal Definition
  - States: $s \in S$
  - Actions: $a \in A$
  - Rewards: $r \in \Re$
  - Transition model: $Pr(s_t | s_{t-1}, a_{t-1})$
  - Reward model: $R(s_t, a_t)$
  - Discount factor: $0 \leq \gamma \leq 1$
    - discounted: $\gamma < 1$ undiscounted: $\gamma = 1$
  - Horizon (i.e., # of time steps): h
    - Finite horizon: $h \in \mathbb{N}$ infinite horizon: $h = \infty$

- Goal: find optimal policy

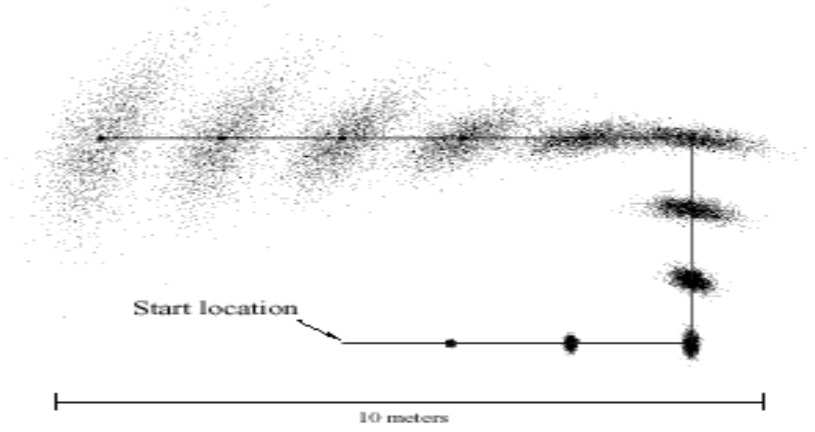
Current Assumptions

- Uncertainty: stochastic process
- Time: sequential process
- Observability: fully observable states
- No learning: complete model
- Variable type: discrete (e.g., discrete states and actions)

Transition Model

- Definition: $Pr(s_t | s_{t-1}, a_{t-1})$
  - Capture uncertainty in dynamics of the system
- Assumptions
  - Markov: $Pr(s_t | s_{t-1}, a_{t-1}, s_{t-2}, a_{t-2}, \ldots) = Pr(s_t | s_{t-1}, a_{t-1})$
  - Stationary: $Pr(s_t | s_{t-1}, a_{t-1})$ is same for given (s_t, a_{t-1}, s_{t-1}) $\forall t$

- **Mobile Robotics:**
- s_t**: position**
- a_t**: motion**



UNIVERSITY OF WATERLOO

Reward Model

- Rewards: $r_t \in \Re$
- Reward Function: $R(s_t, a_t) = r_t$
  - Mapping from state-action pairs to rewards
- Assumption: Stationary reward function
  - $R(s_t, a_t)$ is the same for a given (s, a) $\forall t$
- Exception: terminal reward is different
  - E.g., in a game: reward at each turn and +1/-1 at the end for winning/losing
- **Goal**: maximize sum of expected rewards $\sum_t R(s_t, a_t)$

UNIVERSITY OF WATERLOO

Discounted/Average Rewards

- If process infinite, isn't $\sum_t R(s_t, a_t)$ infinite?

- Solution 1: discounted rewards
  - Discount factor: $0 \leq \gamma < 1$
  - Finite utility: $\sum_t \gamma^t R(s_t, a_t)$ is a geometric sum
  - γ induces an inflation rate
  - Intuition: prefer utility sooner than later

- Solution 2: average rewards
  - More complicated computationally
  - Beyond the scope of this course

Inventory Management

- Markov Decision Process
  - States: inventory levels
  - Actions: {doNothing, orderWidgets}
  - Transition model: stochastic demand
  - Reward model: Sales – Costs - Storage
  - Discount factor: 0.999
  - Horizon: ∞

- Tradeoff: increasing supplies decreases odds of missed sales, but increases storage costs

Policy

- Choice of action at each time step

- Formally:
  - Mapping from states to actions
  - i.e., $\pi(s_t) = a_t$
  - Assumption: fully observable states
    - Allows a_t to be chosen only based on current state s_t

- Objective:

Find optimal policy such that $\pi^* = \arg\max_{\pi} \sum_{t=0}^{h} \gamma^t \mathbb{E}_{\pi}[r_t]$

UNIVERSITY OF WATERLOO

Policy Optimization

- Policy Evaluation:
  - Compute expected utility (value of following π)

$$V^{\pi}(s_0) = \sum_{t=0}^{h} \gamma^t \sum_{s_t} Pr(s_t \,|\, s_0, \pi) R(s_t, \pi(s_t))$$

- Optimal Policy:
  - Policy with highest expected utility

$$V^{\pi^*}(s_0) \geq V^{\pi}(s_0) \qquad \forall \pi$$

UNIVERSITY OF WATERLOO

Policy Optimization

- Several classes of algorithms:
  - Value iteration
  - Policy iteration
  - Linear Programming
  - Search techniques (Deterministic transition model)

- Computation may be done
  - Offline: before the process starts
  - Online: as the process evolves

Value Iteration

- Value at first time step:

$$V_0(s) = \max_a R(s, a) \qquad \forall s$$

- Value at second time step:

$$V_1(s) = \max_a R(s, a) + \gamma \sum_{s'} Pr(s'|s, a)V_0(s') \quad \forall s$$

- Value at third time step:

$$V_2(s) = \max_a R(s, a) + \gamma \sum_{s'} Pr(s'|s, a)V_1(s') \quad \forall s$$

- ....
- Bellman's equation

$$V_t(s) = \max_a R(s, a) + \gamma \sum_{s'} Pr(s'|s, a)V_{t-1}(s') \qquad \forall s$$

$$a_t = \arg\max_a R(s, a) + \gamma \sum_{s'} Pr(s'|s, a)V_{t-1}(s') \qquad \forall s$$

Value Iteration Algorithm

valueIteration(MDP)

$V_0^*(s) \leftarrow \max_a R(s,a)\ \forall s$

For $n = 1$ to h do

$$V_n^*(s) \leftarrow \max_a R(s,a) + \gamma \sum_{s'} \Pr(s' | s,a) V_{n-1}^*(s')\ \forall s$$

Return V^*

Optimal policy π^*

$t = 0$: $\pi_0^*(s) \leftarrow \text{argmax}_a R(s,a)\ \forall s$

$t > 0$: $\pi_n^*(s) \leftarrow \text{argmax}_a R(s,a) + \gamma \sum_{s'} \Pr(s' | s,a) V_{n-1}^*(s')\ \forall s$

π^* is non-stationary (i.e., time dependent)

UNIVERSITY OF WATERLOO

Value Iteration

- Matrix form:

  R^a: $|S| \times 1$ column vector of rewards for a

  V_n^*: $|S| \times 1$ column vector of state values

  T^a: $|S| \times |S|$ matrix of transition prob. for a

valueIteration(MDP)

$V_0^* \leftarrow \max_a R^a$

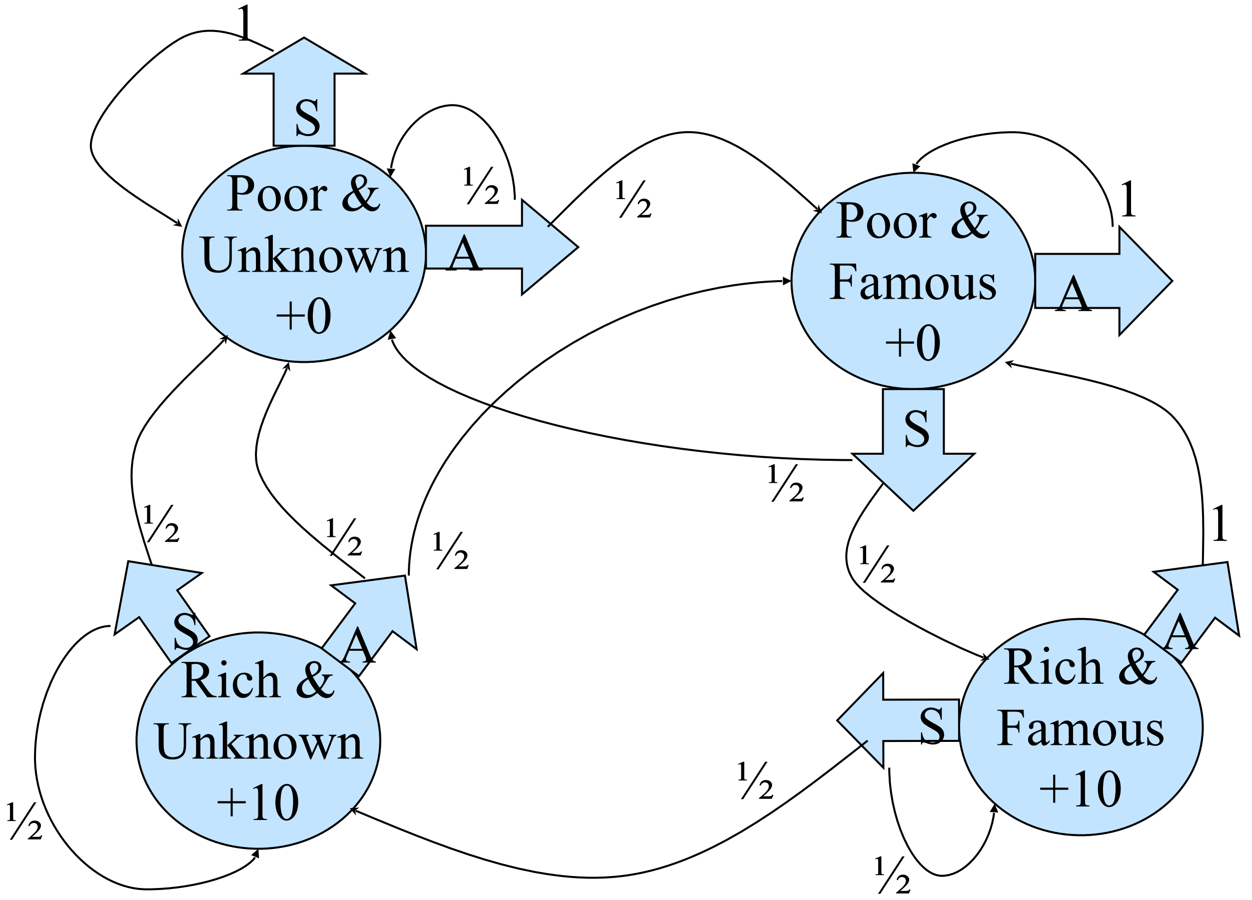
For $t = 1$ to h do

$\quad V_n^* \leftarrow \max_a R^a + \gamma T^a V_{n-1}^*$

Return V^*

UNIVERSITY OF WATERLOO

A Markov Decision Process

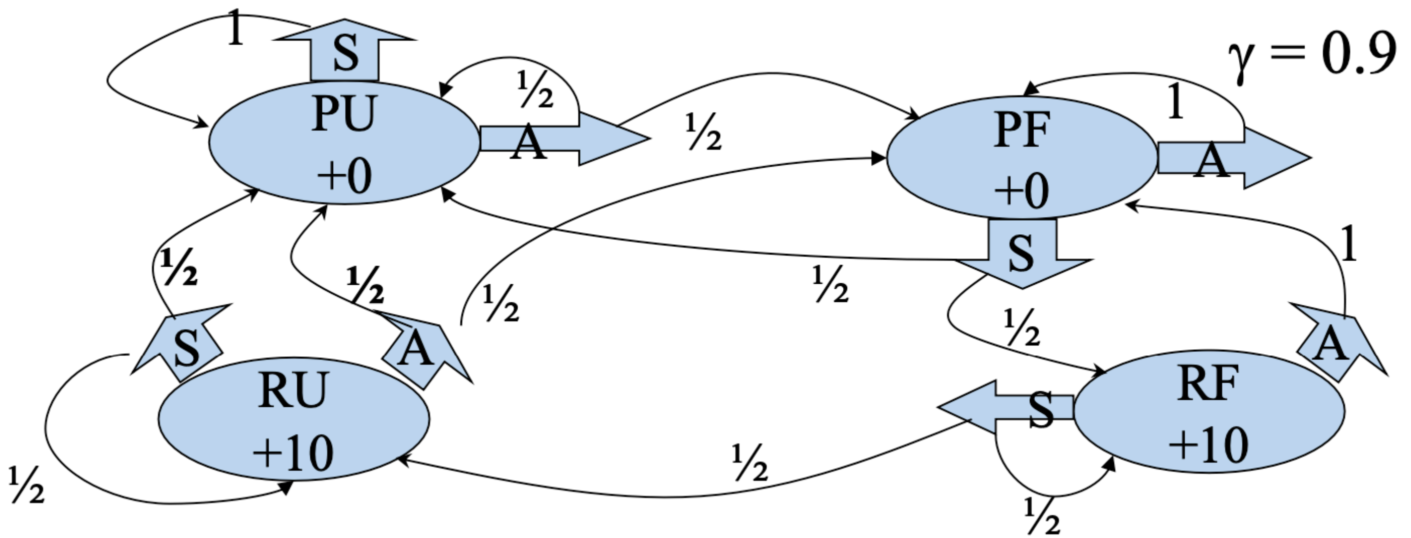


$\gamma = 0.9$

You own a company

In every state you must choose between

Saving money or **A**dvertising



n	$V(PU)$	$\pi(PU)$	$V(PF)$	$\pi(PF)$	$V(RU)$	$\pi(RU)$	$V(RF)$	$\pi(RF)$
0	0	A,S	0	A,S	10	A,S	10	A,S
1	0	A,S	4.5	S	14.5	S	19	S
2	2.03	A	8.55	S	16.53	S	25.08	S
3	4.76	A	12.20	S	18.35	S	28.72	S
4	7.63	A	15.07	S	20.40	S	31.18	S
5	10.21	A	17.46	S	22.61	S	33.21	S

Value Iteration Step 1 (for state RF)

- For all the states — the value is the reward
- For all the states — Policy is any of the action

$$V_0(RF) = max\{R(RF, A), R(RF, S)\}$$
$$= max\{10,10\}$$
$$= 10$$

$$\pi_0(RF) = \arg\max\ \{R(RF, A), R(RF, S)\}$$
$$= \{A, S\}$$

Value Iteration Step 2 (for state RF)

- For step 2 we should take a value iteration step with $n = 1$
- The value updates are as follows:

$$V_1(RF) = \max_a R(RF, a) + \gamma \sum_{s'} P(s' | RF, a) V_0(s')$$

$$= \max\{10 + 0.9 \times 1 \times 0, 10 + 0.9(0.5 \times 10 + 0.5 \times 10\}$$

$$= \max\{10, 19\}$$

$$= 19$$

- Policy is the action that maximizes value:

$\pi_1(RF) = S$

UNIVERSITY OF WATERLOO

Value Iteration Step 3 (for state RF)

- For step 3 we should take a value iteration step with $n = 2$
- The value updates are as follows:

$$V_2(RF) = \max_a R(RF, a) + \gamma \sum_{s'} P(s' | RF, a) V_1(s')$$

$$= \max\{10 + 0.9 \times 1 \times 4.5, 10 + 0.9(0.5 \times 19 + 0.5 \times 14.5\}$$

$$= \max\{14.05, 25.08\}$$

$$= 25.08$$

- Policy is the action that maximizes value:

$$\pi_2(RF) = S$$

UNIVERSITY OF WATERLOO

Horizon Effect

- Finite h:
  - Non-stationary optimal policy
  - No guarantee to converge
  - Best action different at each time step
  - Intuition: Best action varies with changing value estimate
- Infinite h:
  - Stationary optimal policy
  - Value iteration converges
  - Same best action at each time step
  - Intuition: Best action same with non-changing value estimate
  - Problem: Value iteration does infinite # of iterations…

Infinite Horizon

- Assuming a discount factor γ, after n time steps, rewards are scaled down by γ^n
- For large enough n, rewards become insignificant since $\gamma^n \to 0$

- Solution:
  - pick large enough n
  - run value iteration for n steps
  - Execute policy found at the n^{th} iteration

- Solution 2:
  - Continue iterating until $| Vn - V_{n-1} |_\infty \leq \epsilon$
  - ϵ is called threshold or tolerance

Policy Optimization

- Value Iteration
  - Optimize value function
  - Extract induced policy

- Can we directly optimize the policy?
  - Yes, by **policy iteration**

Policy Iteration

- Alternate between two steps

  - Policy Evaluation

$$V^{\pi}(s) = R\left(s, \pi(s)\right) + \gamma \sum_{s'} \Pr\left(s' \middle| s, \pi(s)\right) V^{\pi}(s') \ \forall s$$

  - Policy Improvement

$$\pi(s) \leftarrow \underset{a}{\operatorname{argmax}} R(s, a) + \gamma \sum_{s'} \Pr(s' | s, a) V^{\pi}(s') \ \forall s$$

UNIVERSITY OF WATERLOO

Algorithm

policyIteration(MDP)

Initialize π_0 to any policy

$n \leftarrow 0$

Repeat

Eval: $V_n = R^{\pi_n} + \gamma T^{\pi_n} V_n$

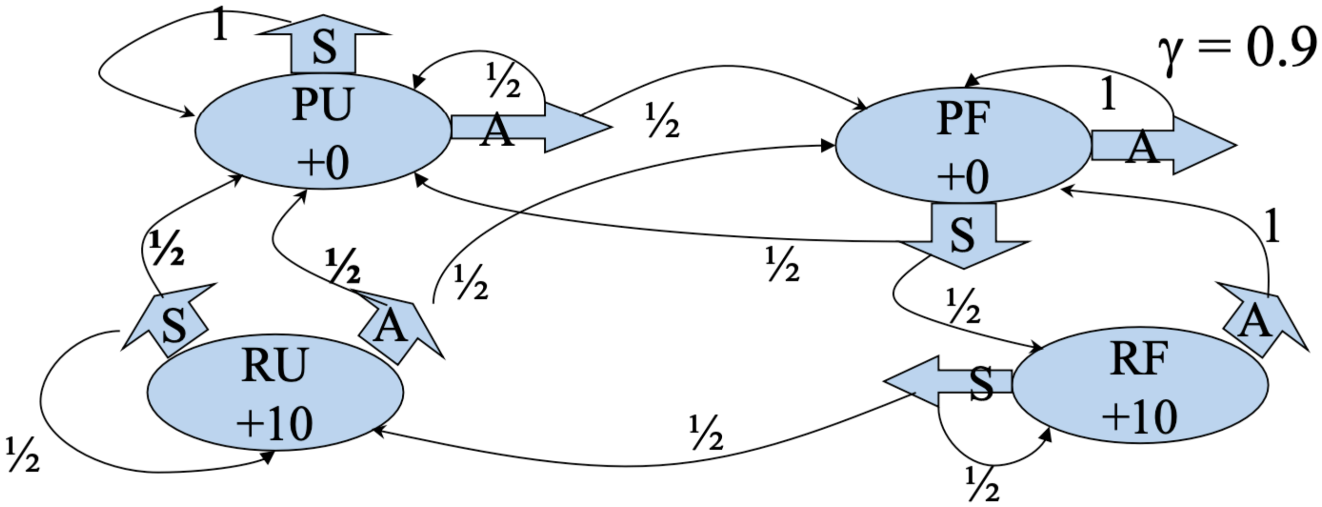
Improve: $\pi_{n+1} \leftarrow argmax_a \; R^a + \gamma T^a V_n$

$n \leftarrow n + 1$

Until $\pi_{n+1} = \pi_n$

Return π_n

Example (Policy Iteration)



n	$V(PU)$	$\pi(PU)$	$V(PF)$	$\pi(PF)$	$V(RU)$	$\pi(RU)$	$V(RF)$	$\pi(RF)$
0	0	A	0	A	10	A	10	A
1	31.6	A	38.6	S	44.0	S	54.2	S
2	31.6	A	38.6	S	44.0	S	54.2	S

Complexity

- Value Iteration:
  - Each iteration: $O(|S|^2|A|)$
  - Many iterations: linear convergence

- Policy Iteration:
  - Each iteration: $O(|S|^3 + |S|^2|A|)$
  - Few iterations: linear-quadratic convergence

Summary

- Markov Decision Processes
  - Models sequential decision making
  - (Possibly) Infinite or Indefinite horizon for decision making
  - Objective: Find the optimal policy
- Policy Optimization
  - Value Iteration
  - Policy Iteration
  - Examples
- Think about:
  - What happens if the transition model and/or reward model is not given?

UNIVERSITY OF WATERLOO