Lecture 1b: Markov Decision Processes
CS885 Reinforcement Learning

2022-09-09

Complementary readings: [SutBar] Chap. 3, [Sze] Chap. 2, [RusNor] Sec. 15.1, 17.1-17.2, 17.4, [Put] Chap. 2, 4, 5

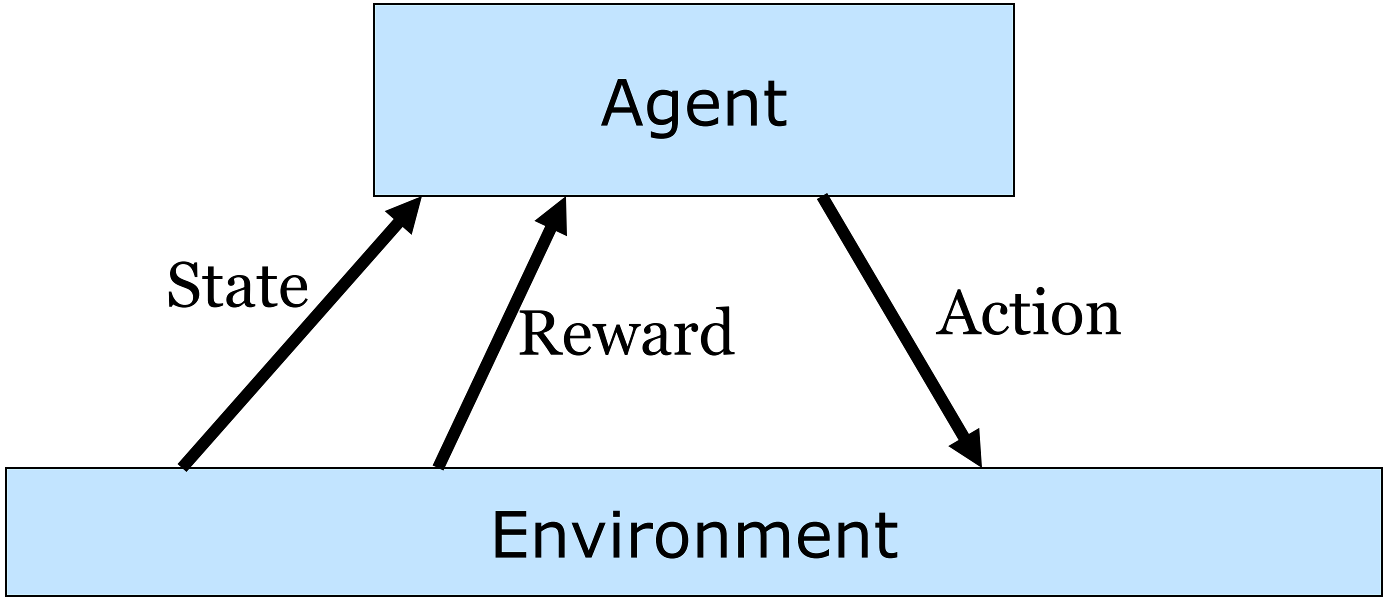
Pascal Poupart
David R. Cheriton School of Computer Science

UNIVERSITY OF WATERLOO

Outline

- Markov Decision Processes
- Value Iteration

Recall: RL Problem



Goal: Learn to choose actions that maximize rewards

Unrolling the Problem

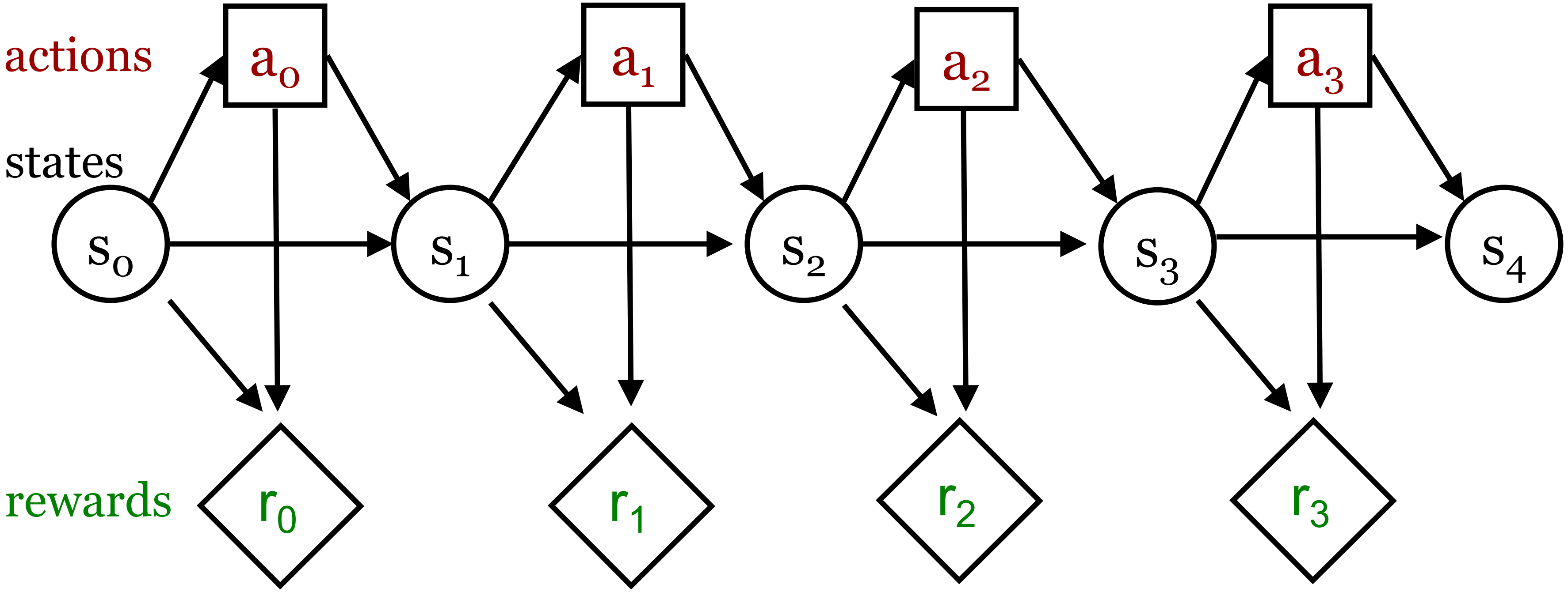
- Unrolling the control loop leads to a sequence of states, actions and rewards:

$$s_0, a_0, r_0, s_1, a_1, r_1, s_2, a_2, r_2, \ldots$$

- This sequence forms a stochastic process (due to some uncertainty in the dynamics of the process)

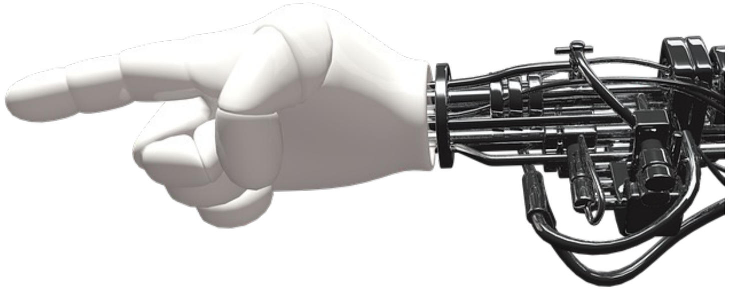
Markov Decision Processes

- Probabilistic graphical model



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

UNIVERSITY OF WATERLOO

Markov Decision Processes

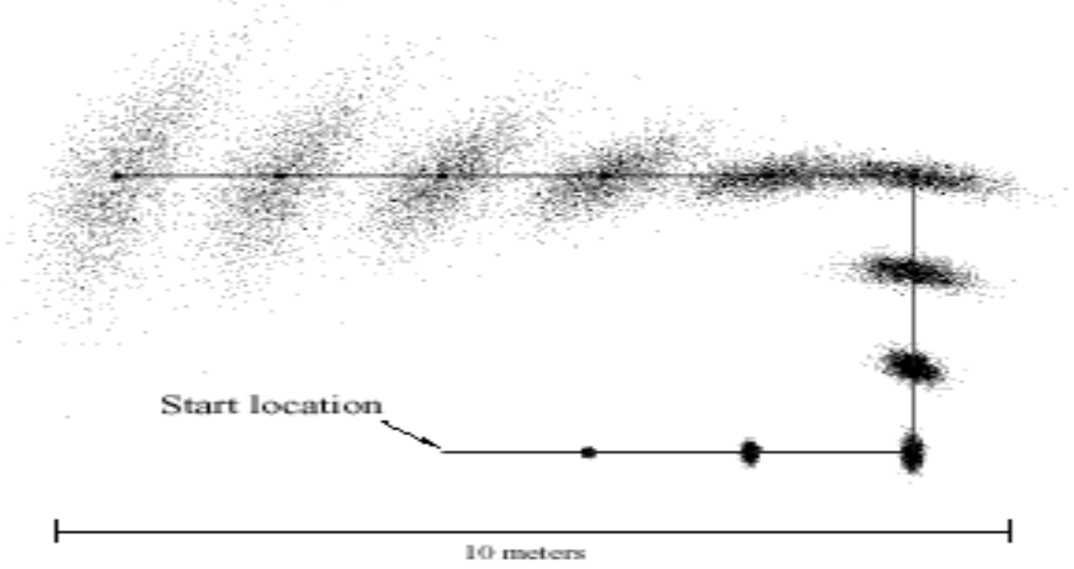
- Formal Definition
  - States: $s \in S$
  - Actions: $a \in A$
  - Rewards: $r \in \mathbb{R}$
  - Transition model: $\Pr(s_t|s_{t-1}, a_{t-1})$
  - Reward model: $\Pr(r_t|s_t, a_t)$, $R(s_t, a_t) = \sum_{r_t} r_t \Pr(r_t|s_t, a_t)$
  - Discount factor: $0 \leq \gamma \leq 1$
    - discounted: $\gamma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

- Goal: find optimal policy

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** $\forall t$

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



Reward Model

- Rewards: $r_t \in \Re$
- Reward function: $R(s_t, a_t) = \sum_{r_t} r_t \Pr(r_t | s_t, a_t)$

- Common assumption: stationary reward function
  - $R(s_t, a_t)$ is the same $\forall t$
- Exception: terminal reward function often different
  - E.g., in a game: 0 reward at each turn and +1/-1 at the end for winning/losing

- Goal: **maximize sum of expected rewards** $\boldsymbol{\sum_t R(s_t, a_t)}$

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 of $1/\gamma - 1$
  - 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

Policy Optimization

- Policy evaluation:
  - Compute expected utility

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

- Optimal policy:
  - Policy with highest expected utility

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

Policy Optimization

- Several classes of algorithms:
  - Value iteration
  - Policy iteration
  - Linear Programming
  - Search techniques

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

Value Iteration Algorithm

valueIteration(MDP)

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

For $n = 1$ to h do

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

Return V^*

Optimal policy π^*

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

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

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

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)

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

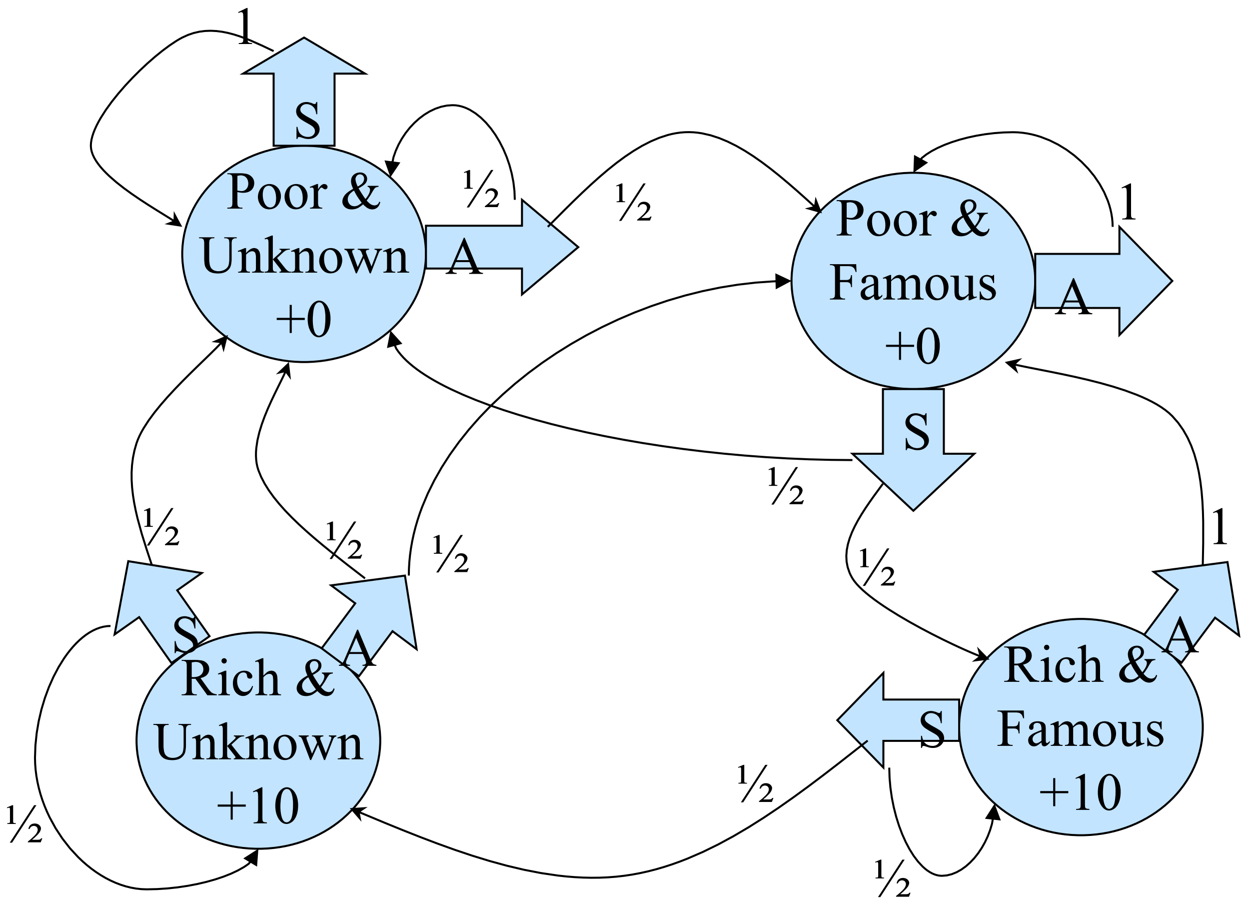
For $t = 1$ to h do

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

Return V^*

UNIVERSITY OF WATERLOO

A Markov Decision Process

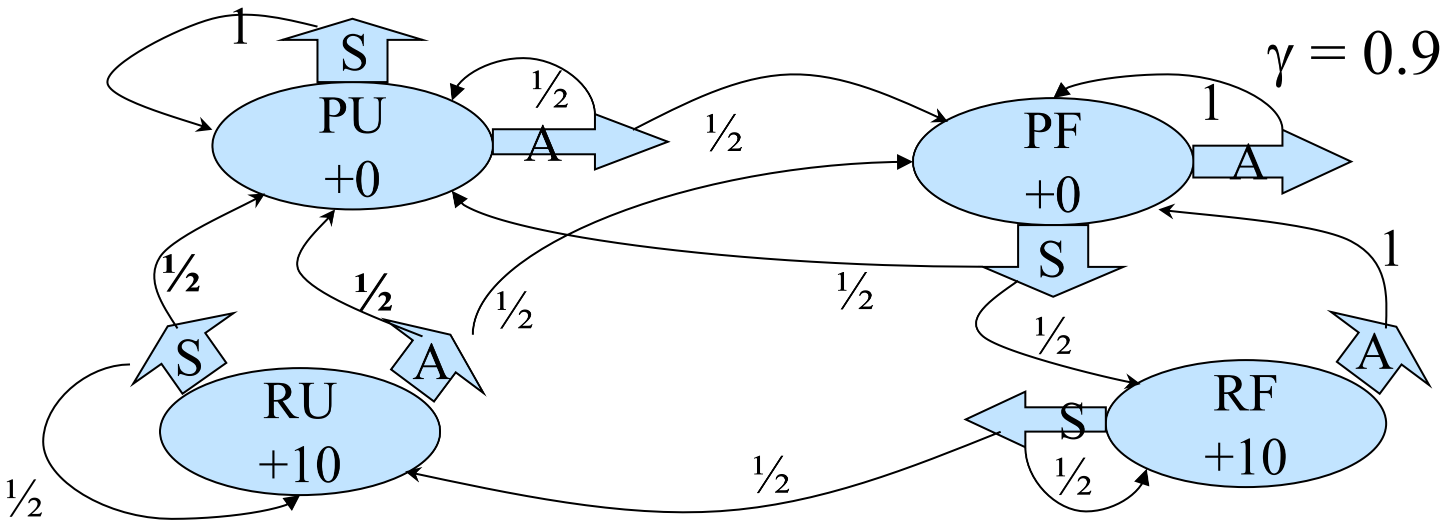


$\gamma = 0.9$

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

Horizon Effect

- Finite h:
  - Non-stationary optimal policy
  - Best action different at each time step
  - Intuition: best action varies with the amount of time left

- Infinite h:
  - Stationary optimal policy
  - Same best action at each time step
  - Intuition: same (infinite) amount of time left at each time step, hence same best action
  - 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