RLearning:
Short guides to reinforcement learning

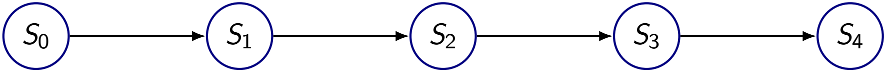
Unit 2-2: Markov Decision Processes

Davud Rostam-Afschar (Uni Mannheim)

How to take actions based on predictions?

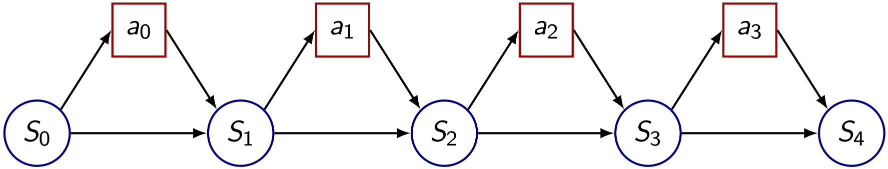
Markov Decision Process

- Markov process augmented with...
  - Actions e.g., a_t
  - Rewards e.g., r_t

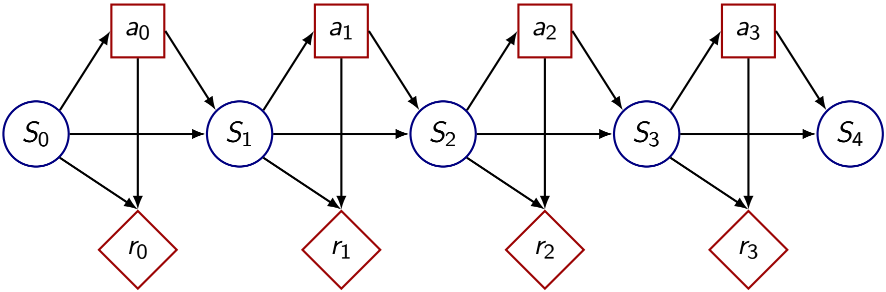


Markov Decision Process

- Markov process augmented with...
  - Actions e.g., a_t
  - Rewards e.g., r_t



- Markov process augmented with...
  - Actions e.g., a_t
  - Rewards e.g., r_t



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)

Markov Decision Process

- Definition
  - Set of states: S
  - Set of actions: A
  - Transition model: $\mathbb{P}\left(s_t \mid s_{t-1}, a_{t-1}\right)$
  - Reward model: $R\left(s_t, a_t\right)$
- Goal: find optimal policy

Readings: Intro to Markov decision processes

?, chapter 3

?, chapter 2

?, sections 17.1-17.2, 17.4

?, chapters 2, 4, 5

(Discounted) Rewards and Values

What are the Rewards?

- **Rewards**: $r_t \in \mathbb{R}$
- **Reward function**: $R(s_t, a_t) = r_t$ mapping from state-action pairs to rewards
- 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 rewards** $\sum_t R(s_t, a_t)$

Discounted Rewards

- If process infinite, isn't $\sum_t R(s_t, a_t)$ infinite?
- Solution: 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 (per-period) time-preference rate of $\frac{1}{\gamma} - 1$
  - Intuition: prefer utility sooner than later

- Definition
  - Set of states: S
  - Set of actions: A
  - Transition model: $\mathbb{P}\left(s_t \mid s_{t-1}, a_{t-1}\right)$
  - Reward model: $R\left(s_t, a_t\right)$
  - 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

Inventory Management

- Markov Decision Process
  - States: inventory levels
  - Actions: $\{$doNothing, orderGoods$\}$
  - 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

Policies to Max Expected Utility

What is a 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 evaluation:
  - Compute expected utility

$$V^{\pi}(s_0) = \sum_{t=0}^{h} \gamma^t \sum_{s_0} \mathbb{P}(s_t \mid \ldots, s_0, \pi) \, R(s_t, \pi(s_t))$$

- 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

References I

PUTERMAN, M. L. (2014): *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.

RUSSELL, S. J., AND P. NORVIG (2016): *Artificial intelligence: a modern approach*. Pearson.

SUTTON, R. S., AND A. G. BARTO (2018): "Reinforcement learning: An introduction," *A Bradford Book*, Available at `http://incompleteideas.net/book/the-book-2nd.html`.

SZEPESVÁRI, C. (2022): *Algorithms for reinforcement learning*. Springer nature, Available at `https://sites.ualberta.ca/~szepesva/RLBook.html`.

Takeaways

How Can Agents Choose Actions to Maximize Expected Rewards?

- Markov Decision Processes (MDPs) extend Markov processes with actions and rewards
- Goal: Find a policy that maps states to actions to maximize expected cumulative rewards
- Policies can be optimized via value iteration, policy iteration, or other algorithms
- Discounting helps handle infinite horizons and captures preference for earlier rewards