Markov Decision Process

Generalization of Search problem, where you are not entirely sure what your actions are going to do, until you try them.

(Non-Deterministic Search)

* <u>Noisy Movement</u>: Action do not always go as planned.

* The agent receives rewards each time step.

* <u>Goal</u>: Maximize sum of rewards

* <u>Markov Decision Process</u>

⇒ An MDP is defined by:

- A set of state $s \in S$
- A set of actions $a \in A$
- A transition function $T(s,a,s')$
  - → Probability that a from s leads to s' i.e. $P(s'|s,a)$
  - → Also called the model or the dynamics
- A reward function $R(s,a,s')$
- A start state
- Maybe a terminal state.

* What is Markov about MDPs?

⇒ For markov decision processs, "Markov" means action outcomes depends only on the current state:

$$\boxed{P(S_{t+1} \mid S_t, A_t, S_{t-1}, A_{t-1} \cdots S_0) = P(S_{t+1} \mid S_t, A_t)}$$

* Policies

⇒ In deterministic single-agent search problems we wanted an obtimal plan.

{Sequence of action from start to Goal}

⇒ For MDPs, we want an optimal policy $\pi^* : S \to A$

{An optimal policy is the one that maximizes expected utility if followed}

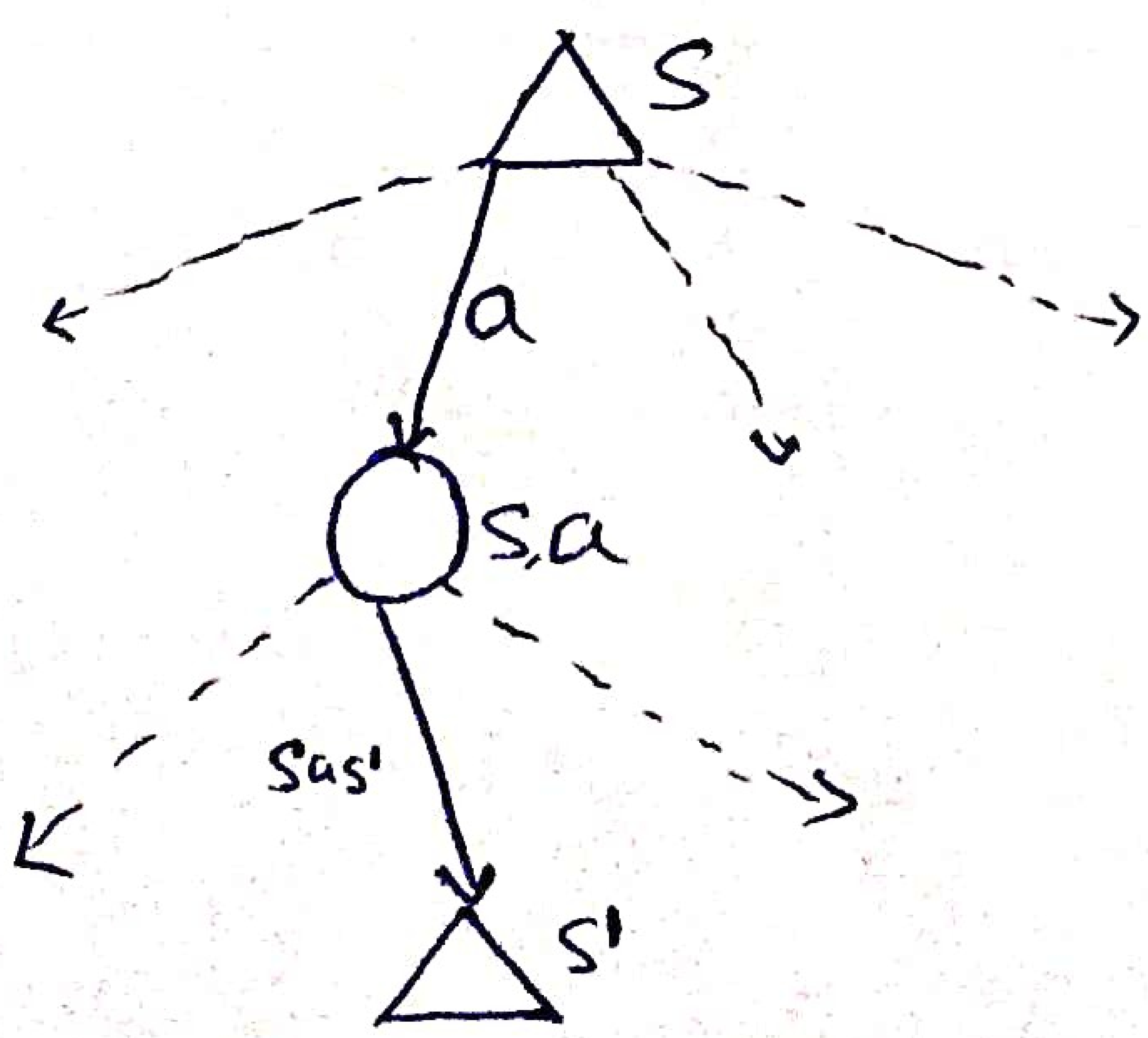
{A policy π gives action for each state}

⇒ A explicit policy defines a reflex agent.

* MDP Search Tree

⇒ Each MDP state projects an expectimax-like search tree.

(s,a) is the q-state



(s a s') called a transition

$T(s,a,s') = P(s' \mid s,a)$

$R(s,a,s')$

★ Discounting

⇒ It's reasonable to maximize the sum of rewards

⇒ It is also reasonable to prefer rewards now to rewards later.

⇒ One Solution: Values of rewards decay exponentially

$$1 \longrightarrow \gamma \longrightarrow \gamma^2$$

★ Stationary Preferences

- Theorem: If we assume stationary preferences:

$$[a_1, a_2, \dots] \succ [b_1, b_2 \dots]$$

$$[r, a_1, a_2 \dots] \succ [r\, b_1, b_2 \dots]$$

⇒ There are only two ways of defining utilities:

- Additive utility: $U([r_0, r_1, r_2 \dots]) = r_0 + r_1 + r_2 + \dots$
- Discounted utility: $U([r_0\, r_1, r_2 \dots]) = r_0 + \gamma r_1 + \gamma^2 r_2 + \dots$

★ Infinite Utilities

Problem: What if the game last forever? Do we get infinite rewards?

Solution

① Finite horizon (Similar to depth-limited Search)

- Terminate episode after a fixed T step.
- Gives non stationary policies (π depends of time left)

54:30

② Discounting: Use $0 < \gamma < 1$

→ $U([r_0, r_1, \ldots r_\infty]) = \sum_{t=0}^{\infty} \gamma^t r_t \leq \frac{R_{max}}{1-\gamma}$

→ Smaller γ means smaller "horizon"
(shorter term focus)

③ Absorbing State

→ Guarantee that for every policy, a terminal state will eventually be reached.

Markov Decision Process

* Solving MDPs

Given:

- States
- Actions possible at each state
- Transition Probabilities
- Rewards

$\xrightarrow{\text{Solving MDP}}$ (Policy $\pi^*: S \to A$)

Output

* Optimal Quantities

⇒ The value (utility) of a state s:

$V^*(s)$ = Expected utility starting in s and acting optimally.

⇒ The value (utility) of a q-state (s,a);

$Q^*(s,a)$ = Expected utility starting out having taken action a from state s & thereafter acting optimally.

⇒ The Optimal policy:

$\pi^*(s)$ = optimal action from state s.

s is a state

(s,a) is a q-state

(s,a,s') is a transition

* Values of State

Fundamental operation: Compute the (expectimax) value of a state.

↳ Average sum of (discounted) rewards.

$$V^*(s) = \max_a Q^*(s,a)$$

$$Q^*(s,a) = \sum_{s'} T(s,a,s')\left[R(s,a,s') + \gamma V^*(s')\right]$$

$$V^*(s) = \max_a \sum_{s'} T(s,a,s')\left[R(s,a,s') + \gamma V^*(s')\right]$$

* Time-Limited Values

⇒ Define $V_k(s)$ to be the optimal value of s if the game ends in K more time steps.

* Value Iteration

- Start with $V_0(s) = 0$: no time steps left means an expected reward sum of zero.
- Given vector of $V_k(s)$ values, do one ply of expectimax for each state

$$V_{k+1}(s) \leftarrow \max_a \sum_{s'} T(s,a,s')\left[R(s,a,s') + \gamma V_k(s')\right]$$

- Repeat until convergence.
- Complexity of each iteration: $O(S^2A)$


-10
F
1
0.5 +1
S
+1
0.5
S
Cool
Warm
Overheated
1 +1
0.5
+2 0.5
F +2