⑨

MDP: Part 2

★ **Convergence**

Case 1: If the tree has maximum depth M, then V_M holds the actual untruncated values.

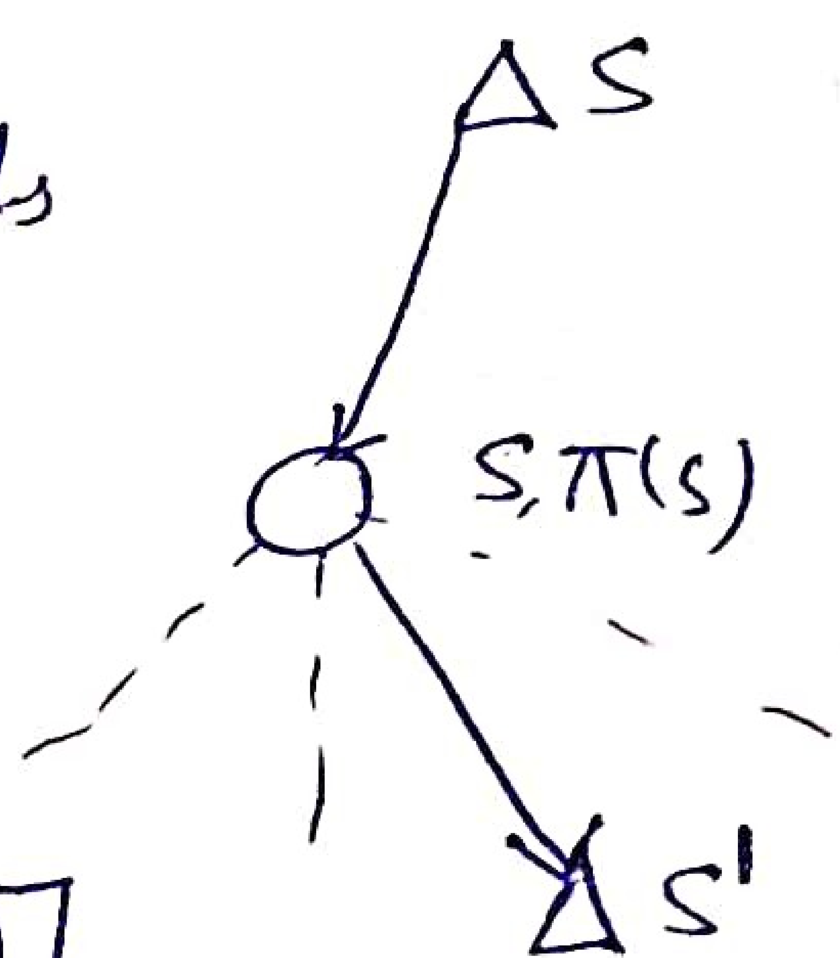
Case 2: If the discount is less than 1

★ **Policy Evalution**

⇒ Given the policy, you want to know, how good is that policy.

★ **Utilities for a fixed Policy**

$V^{\pi}(s)$ = Expected total discounted rewards starting in s and following π



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

⇒ How do we calculate the V's for a fixed polics π?

→ Idea 1: Turn recursive Bellman equations into updates.

→ Idea 2: Without the maxes, the bellman equations are just a linear system.

* Computing Actions from Values

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

↳ Policy Extraction

* Computing Actions from Q-Values

$$\pi^*(s) = \underset{a}{\operatorname{argmax}}\, Q^*(s,a)$$

"actions are easier to select from" q-values than values!

* Problems with Value Iteration

Problem 1: It's slow – $O(S^2A)$ per iteration

Problem 2: The "max" at each state rarely changes

Problem 3: The policy often converges long before the value.

* Policy Iteration

Step 1: (Policy Evaluation)

Calculate utilities for some fixed policy (not optimal utilities) until convergence

Step 2: (Policy Improvement)

Update policy using one-step look-ahead with resulting converged (but not optimal) utilities as future values

⇒ Repeat steps until policy converges.

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

② $\pi_{i+1}(s) = \arg\max_{a} \sum_{s'} T(s,a,s')\left[R(s,a,s') + \gamma V^{\pi_i}(s')\right]$

⇒ Both Value Iteration and Policy Iteration are dynamic programs for solving MDPs

★ Reinforcement Learning

⇒ Important ideas in reinforcement Learning:

* **Exploration**: You have to try unknown actions to get information
* **Exploitation**: Eventually, you have to use what you know
* **Regret**: Even if you learn intelligently, you make mistakes.
* **Sampling**: Because of chance, you have to try things repeatedly.
* **Difficulty**: Learning can be harder that solving a known MDP.