EECS 496: Sequential Decision Making

Soumya Ray

sray@case.edu

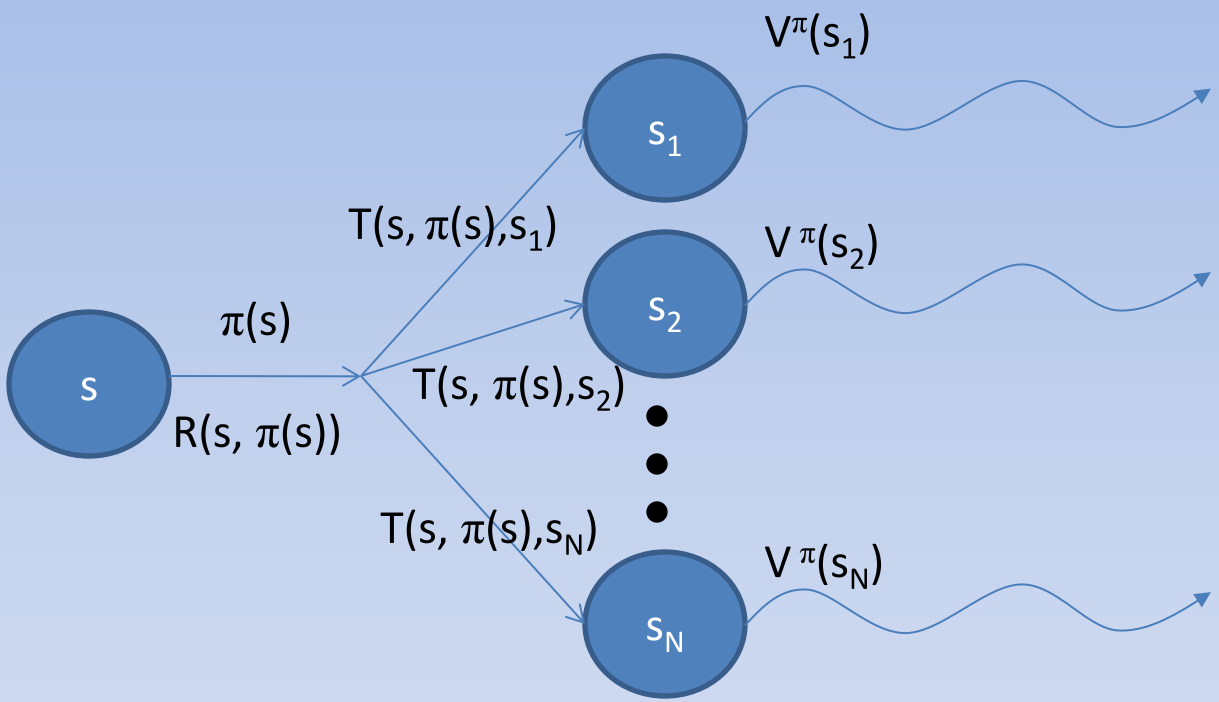
Office: Olin 516

Office hours : T 4-5:30 or by appointment

Recap

- In SDM, what does a utility function do?
- As an agent wanders around it receives periodic ___/ ___ / ___ ___. Its goal is to maximize _____.
- What are some differences between SDM and Classical Planning?
- What is the credit assignment problem?
- What is the exploration/exploitation tradeoff?
- We formalize and SDM with a ___ ___ ___. This has : (i), (ii), (iii), (iv), (v), (vi).
- What does the (3rd component) do?
- What is Markovian about this?
- What is stationary about this?
- What does the (4th component) do?
- What is a policy? Optimal policy?
- What is the utility function we use over a trajectory?
- Why do we use discounting?
- What is our optimality criterion?
- What is the value of a state under a policy?
- V(s)=___ + __* (sum over ___ * ___). This is called the ___ ___.
- What is the intuition behind the above?

Picture



$$V^\pi(s) = R(s, \pi(s)) + \gamma \sum_{s'} T(s, \pi(s), s') V^\pi(s')$$

Bellman equation

Consequences

- Every policy π has a (unique) value function V^π satisfying the Bellman equation
- The optimal policy π^* also has a value function, V^{π^*}
- Remember that the π^* *maximizes* the *expected utility*
  - But this is exactly the value function

Finding the optimal policy

- The optimal policy is the policy with the largest value function:

$$\pi^*(s) = \arg\max_{\pi} V^{\pi}(s) \text{ for all } s$$

$$= \arg\max_{\pi}(R(s,\pi(s)+\gamma\sum_{s'}T(s,\pi(s),s')V^{\pi}(s'))$$

$$= \arg\max_{a}(R(s,a)+\gamma\sum_{s'}T(s,a,s')V(s'))$$

Bellman Optimality Criterion

- As a consequence of the previous slide, the value function of a state under π* can be shown to satisfy:

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

Bellman Optimality Criterion (Bellman 1957)
Necessary *and* sufficient!

Finding the optimal policy

- The Bellman optimality criterion gives us a way to find the optimal policy
- If we can find a value function that satisfies it, that determines the optimal policy
- Unfortunately, this system of equations is nonlinear (because of the max), so we can't solve it directly, but a dynamic programming procedure works

Value Iteration

- Start with an arbitrary value function V_0
- At each iteration i Do

$$V_{i+1}(s) = \max_a R(s,a) + \gamma \sum_{s'} T(s,a,s') V_i(s')$$

- Until $|V_{i+1}(s) - V_i(s)|$ is zero

- Then

$$\pi^*(s) = \arg\max_a R(s,a) + \gamma \sum_{s'} T(s,a,s') V_{final}(s')$$

Convergence of value iteration

- It can be shown that at each step of value iteration, the error of the current value function decreases by a factor of (at least) γ

- For small discount factor, convergence is rapid

Example

Policy Iteration

- Notice that in order to determine the optimal policy, we don't really need the exact values of states
  - All we need is values that result in the correct ordering of actions
- In practice, we get this long before the values themselves converge
- Policy iteration is an algorithm that exploits this idea

Policy Iteration

- Start with an initial policy
- Calculate the value of this policy (*policy evaluation step*)

$$V^{\pi}(s) = R(s,\pi(s)) + \gamma \sum_{s'} T(s,\pi(s),s')V^{\pi}(s')$$

- Calculate a new policy (*policy improvement step*) using:

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

So are we done?

- We’ve looked at two algorithms to solve SDM problems
- But wait, what happened to credit assignment and the exploration-exploitation tradeoff?
  – The value function is how we solved the credit assignment problem
  – But we didn’t solve the exploration-exploitation tradeoff---we avoided it by assuming that the agent knows the characteristics of the world