6/21/2021

**Lecture 2: Markov Decision Processes**

**1/ Introduction to MDPs:**

* MDP formally describe an environment for RL
* Where the environment is **fully observable**
* The current state completely characterizes the process
* Almost all RL problems can be formalize as MDP:

+ Optimal control primarily deals with continuous MDPs

+ Partially observable problems can be converted into MDPs

**+ Bandits are MDPs with one state**

**2/ Markov Property:**

* The future is independent of the past given the present.
* The state captures all relevant info from the history
* Once the state is known, the history may be thrown away meaning that the state is a sufficient statistic of the future.

**3/ State Transition Matrix:**

* For a Markov state s and successor state s’, the state transition probability is defined by:

Text

Description automatically generated

* State transition matrix P defines transition probabilities from all states s to all successor states s’

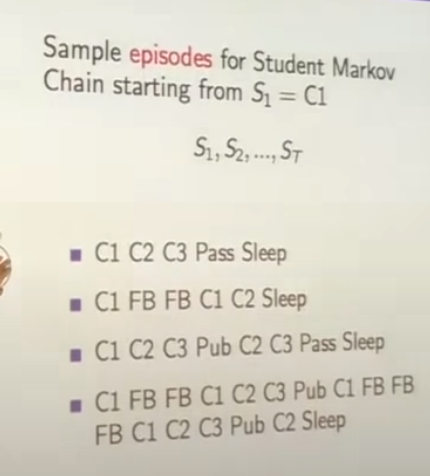
**4/ Markov Process:**

* A Markov process is a memoryless random process, is a sequence of random states S1, S2,…. With the Markov property.

Graphical user interface, text

Description automatically generated

Diagram

Description automatically generated

Diagram

Description automatically generated

* Transition matrix = what’s the probability change from 1 state to another.

**5/ Markov Reward Process:**

* A Markov reward process = a Markov chain with values

Text

Description automatically generated

* R = if we are in a state, how much reward do we get from that state. This is immediate reward = how much reward we got in that state at that moment. This is also called “reward in 1 steps”

a/ Return - reward:

* The return G\_t is the total discounted reward from time-step t. G = Goal

A picture containing scatter chart

Description automatically generated

* The discount gamma belong [0, 1] is the present value of future rewards.
* The value of receiving reward R after k+1 time-steps if gamma^k\*R
* This values immediate reward above delayed reward

+ gamma close to 0 leads to “myopic” evaluation – care about the latest reward

+ gamma close to 1 leads to “far sighted” evaluation – care about previous reward

**- Why there are no expectation?**

- Because G is considered at this point as random goal.

* **Why we do discount?**

+ Mathematically convenient to discount rewards

+ Avoid infinite returns in cyclic Markov Processes

+ Represent the fact that we do not have a perfect model.

+ Uncertainty about the future may not be fully represented

+ If the reward is financial, immediate rewards may earn more interest than delayed rewards (interested rate)

+ Animal/human behaviors shows preference for immediate reward

+ It is possible sometimes to use miscounted Markov reward processes (gamma = 1) if all sequences terminate.

b/ Value Function:

* The value function v(s) gives the long-term value of state s = what’s the total reward you get

Text

Description automatically generated

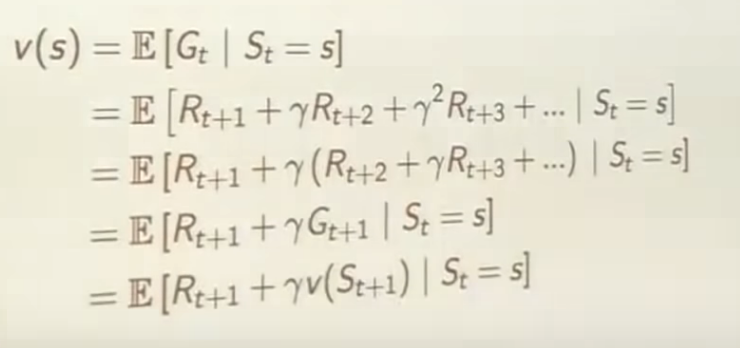
c/ Bellman Equation for Markov Rewards Process:

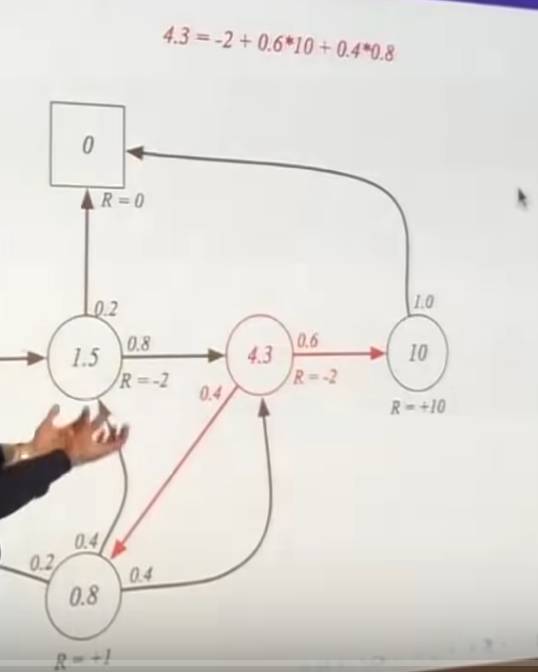
The value function can be decomposed into 2 parts:

- Immediate Rewards R(t+1)

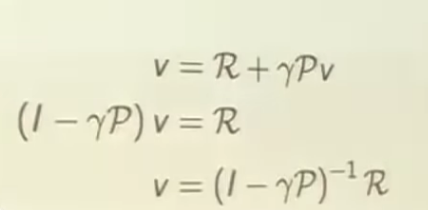
- Discounted value of successor state

- “The value function in state S is equal to the immediate rewards plus the value function of the next state”





d/ Solving the Bellman Equation:

- The Bellman equation is a linear equation

- It can be solved directly:

- Computational complexity is O(n^3) for n states

- Direct solution only possible for small MRPs

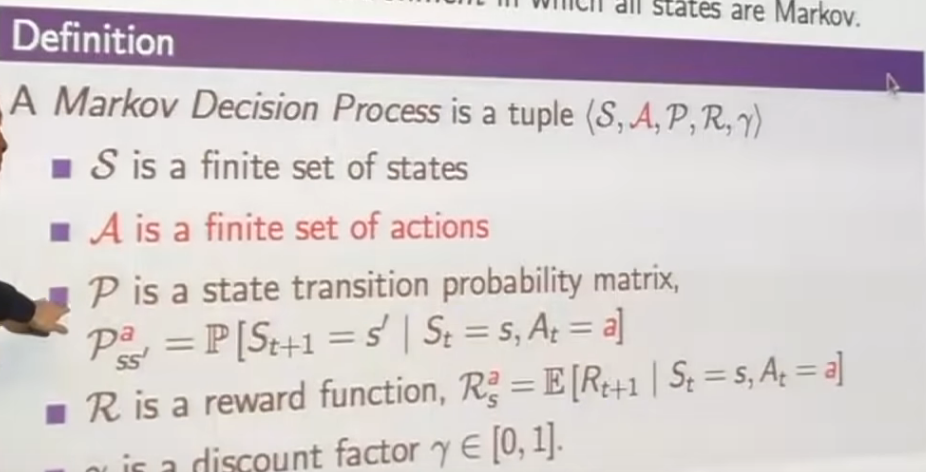
- There are many iterative methods for large MRPs:  
 + Dynamics programming

+ Monte-Carlo Evaluation

+ Temporal-Difference learning

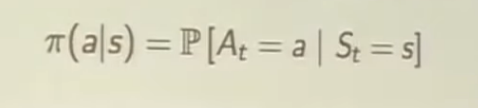
**6/ Markov Decision Processes:**

- A MDP is a Markov reward process with decisions. It is an environment in which all states are Markov.



A/ Policies:

- A policy pi is a distribution over actions given states.



- A policy fully defines the behaviors of an agent

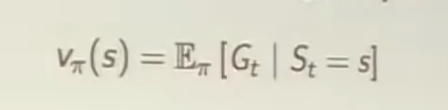
- MDP policies depend on the current state (not the history)

- Given an MDP M = <S,A,P,R> and a policy pi. The state sequence S1, S2,…. Is a Markov process <S,P>

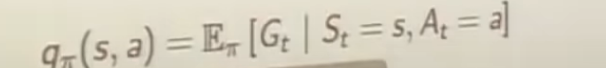
- The state and reward sequence S1, R2,S2,….is a Markov rewards process.

B/ Value Function:

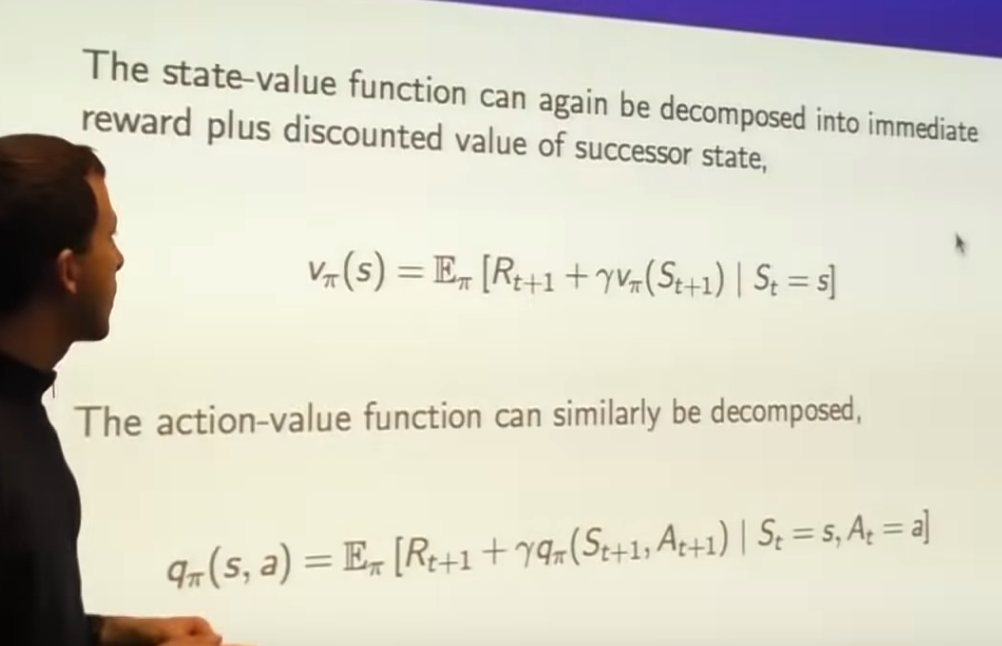
- The state-value function V(s) of an MDP is the expected return starting from value s, and then following policy pi



- The action- value function q(s,a) is the expected return starting from state s, taking action a, and then following policy pi

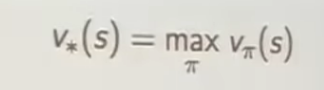


C/ Bellman Expectation Equation:

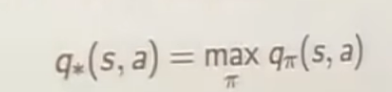


**7/ Optimal Value Function:**

- The optimal state-valiue function v(s) is the maximum value function over all policies:



- The optimal action-value function q(s,a) is the maximum action-value function over all policies:



- The optimal value function specifies the best possible performance in the MDP

- The MDP is solved when we know the optimal function.

**8/ Optimal Policy:**

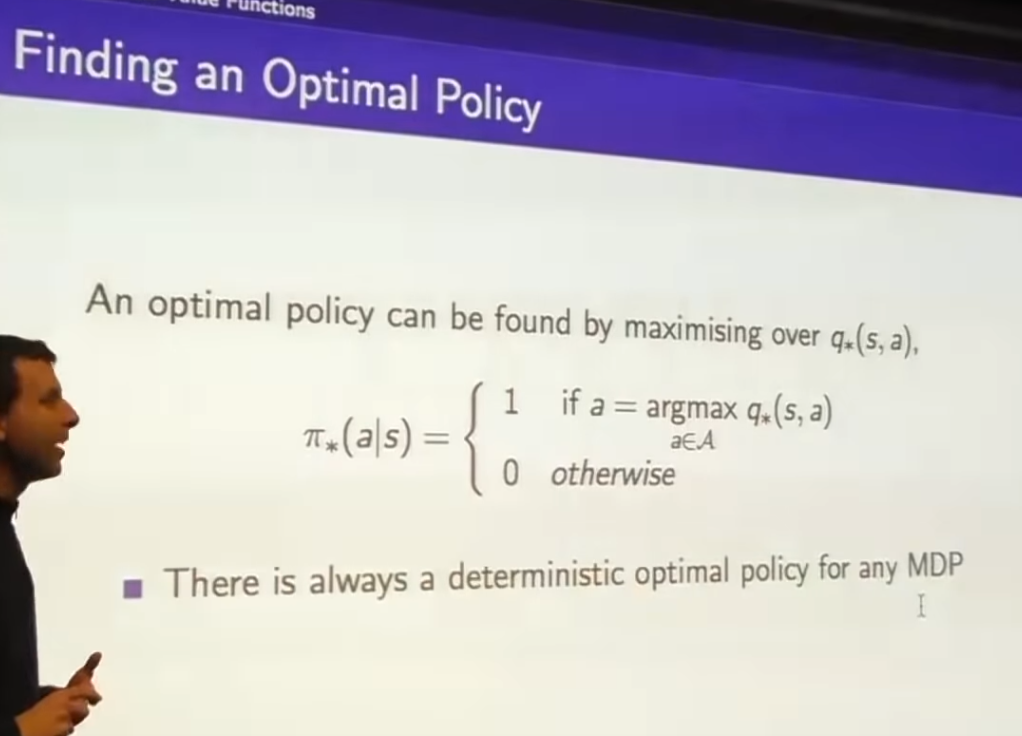
- For any MDP:

+ There exist an optimal policy pi\* that is better than or equal to all other policies pi\* > pi, for all pi

+ All optimal policies achieve the optimal value function, V\_pi\*(s) = V\*(s)

+ All optimal policies achieve the optimal action-value functions, Q\_pi\*(s,a) = A\*(s,a)

**Note**: It is possible to have more than 1 optimal policies. For instance, 2 paths get to same goals.



9/ Solving the Bellman Optimality Equation:

- BOE is non-linear

- No closed form solution (in general)

- Many iterative solution methods:

+ Value Iteration

+ Policy Iteration

+ Q Learning

+ Sarsa