Reinforcement Learning

11-785, Spring 2020
Defining MDPs, Planning
The story of Flider and Spy

• Flider the spider is at the far corner of the room, and Spy the fly is sleeping happily at the near corner
The story of Flider and Spy

- Flider only walks along edges
- She begins walking along one of the three edges at random
- She takes one minute to cover the distance from one corner to the other along any edge
- When she arrives at the new corner, she randomly chooses one of the three edges and continues walking (she may even turn back)
The story of Flider and Spy

• What is the life expectancy of Spy?
Let $T_i$ be the life expectancy if Flider is at the $i^{th}$ corner.
Flider and Spy

- $1 + T_i$ is the life expectancy if Flider the Spider begins walking towards the $i^{th}$ corner
  - 1 minute to get to the corner plus the time taken to get from that corner to Spy the fly
- 8 Equations, 8 unknowns, trivial to solve
A little terminology

- Markov Process: Does not matter how you got here, only matters where you are
An interesting class of problems

• Is a move good?
  – You will not know until the end of the game
An interesting class of problems

- Is an investment plan good?
  - You will not know for a while
An interesting class of problems

• Do I
  – Change lane left?
  – Change lane right?
  – Accelerate?
  – Decelerate?
Reward-based problems

• And many others

• Common theme: These are control problems where
  – Your actions beget rewards
    • Win the game
    • Make money
    • Get home sooner
  – But not deterministically
    • A world out there that is not predictable

• From experience of *belated* rewards, you must learn to act rationally
General cartoon of the world

• Agent operates in an environment
  – Agent may be you..
  – Environment is the game, the market, the road..
General cartoon of the world

- Agent takes actions which affect the environment
General cartoon of the world

- Agent takes actions which affect the environment
- Which changes in a somewhat unpredictable way
Agent takes actions which affect the environment
Which changes in a somewhat unpredictable way
Which affects the agent’s situation
General cartoon of the world

• The agent also receives rewards..
  – Which may be apparent immediately
  – Or not apparent for a very long time
Challenge

- How must the agent behave to maximize its rewards
Let's formalize the system

• At each time $t$ the agent:
  – Makes an observation $O_t$ of the environment
  – Receives a reward $R_t$
  – Performs an action $A_t$
From the perspective of the Agent

• What the agent perceives..
• The following History:
  \[ H_t = O_0, R_0, A_0, O_1, R_1, A_1, \ldots, O_t, R_t \]

• The total history at any time is the sequence of observations, rewards and actions
• We need to model this sequence such that at any time \( t \), the best \( A_t | H_t \) can be chosen
  – The Strategy that maximizes total reward \( R_0 + R_1 + \cdots + R_T \)
Let's formalize the system

- At each time $t$ the agent:
  - Makes an observation $O_t$ of the environment
  - Receives a reward $R_t$
  - Performs an action $A_t$

- At each time $t$ the environment:
  - Receives an action $A_t$
  - Emits a reward $R_{t+1}$
  - Changes and produces an observation $O_{t+1}$
Can define an environment “state”

• Fully captures the “status” of the system
  – E.g., in an automobile: [position, velocity, acceleration]
  – In traffic: the position, velocity, acceleration of every vehicle on the road
  – In Chess: the state of the board + whose turn it is next
A brief trip to Nostalgia..

• Glider, Flider’s brother, never turns around during his wanderings
  – On arriving at any corner, he chooses one of the two “forward” paths randomly.
    • The future possibilities depend on the edge he arrived from
  – Is he Markovian?
Glider is a Markov dude!

- Any causal system can be viewed as Markov, with appropriately defined state
  - The *information state* $S_t$ may differ from the *apparent* state $s_t$
  - Defining $S_t = s_1, s_2, \ldots, s_t$
  - $P(S_{t+1}|S_0, S_1, \ldots, S_t) = P(S_{t+1}|S_t)$
Markov property

- Assumption: The *information state* of the environment is Markov
  \[ P(S_{t+1}|S_0, S_1, ..., S_t) = P(S_{t+1}|S_t) \]

- The environment’s future only depends on its present
To Maximize Reward

• The agent must *model* this environment process
  – Formulate its own model for the environment, which must ideally match the true values as closely as possible
    • Based only on what it observes

• Agent must formulate winning strategy based on model of environment
The Agent’s Side of the Story

• Agent has an internal representation of the environment state
  – May not match the true one at all

• May be defined in any manner
  – Formally the agent state $S_t = f(H_t)$ is some function of the history
  – The closer the agent’s model is to the true environment state, the better the agent will be able to strategize
Defining Agent State

• What is the outcome?
Defining Agent State

- Different definitions of state result in different predictions
- *True* environment state not really known
  - Would greatly improve prediction if known
The World as we model It

• Definition of Markov property:
  – The state of the system has a Markov property if the future only depends on the present
    \[ P(S_{t+1}|S_0, S_1, ..., S_t) = P(S_{t+1}|S_t) \]

• States can be defined to have this property
A Markov Process

• A Markov process is a random process where the future is only determined by the present
  – Memoryless

• Is fully defined by the set of states $S$, and the state transition probabilities $P(s_i | s_j)$
  – Formally, the tuple $M = \langle S, P \rangle$.
  – $S$ is the (possibly finite) set of states
  – $P$ is the complete set of transition probabilities $P(s | s')$
  – Note $P(s | s')$ stands for $P(S_{t+1} = s | S_t = s')$ at any time $t$
  – Will use the shorthand $P_{s,s'}$
The transition probability

• For processes with a discrete, finite set of states, is generally arranged as transition probability matrix

\[ P = \begin{bmatrix}
  P_{s_1,s_1} & P_{s_2,s_1} & \cdots & P_{s_N,s_1} \\
  P_{s_1,s_2} & P_{s_2,s_2} & \cdots & P_{s_N,s_2} \\
  \vdots & \vdots & \ddots & \vdots \\
  P_{s_1,s_N} & P_{s_2,s_N} & \cdots & P_{s_N,s_N}
\end{bmatrix} \]

• More generally (for continuous-state processes, e.g. the state of an automobile), it is modelled as a parametric distribution

\[ P_{s,s_t} = f(s; \theta_{s_t}) \]
A Markov Reward Process

• A Markov Reward Process (MRP) is a Markov Process where states give you rewards

• At each state $s$, upon arriving at that state, you obtain a reward $r$, drawn from a distribution $P(r|s)$
Markov Reward Process

Reward: Upon arriving at any corner, the spider may catch a fly from the swarm hovering there.

Rewards are corner specific and probabilistic: Different corners have different sized swarms with flies of different sizes. The spider only has a probability of catching a fly, but may not always catch one.

• Flider and the Markov reward process!
Markov Reward Process

• Formally, a Markov Reward Process is the tuple $M = \langle S, \mathcal{P}, \mathcal{R}, \gamma \rangle$
  
  – $S$ is the (possibly finite) set of states
  
  – $\mathcal{P}$ is the complete set of transition probabilities $P_{s,s'}$
  
  – $\mathcal{R}$ is a reward function, consisting of the distributions $P(r|s)$
    
    • Or alternately, the expected value $R_s = E[r|s]$
  
  – $\gamma \in [0,1]$ is a discount factor
Markov Reward Process

• Formally, a Markov Reward Process is the tuple $M = \langle S, P, R, \gamma \rangle$
  – $S$ is the (possibly finite) set of states
  – $P$ is the complete set of transition probabilities $P_{s,s'}$
  – $R$ is a reward function, consisting of the distributions $P(r|s)$
    • Or alternately, the expected value $R_s = E[r|s]$
  – $\gamma \in [0,1]$ is a discount factor

What on earth is this?
Rewards and Expected rewards

- One step expected reward: $R_1$
  - Will this be greater if the spider heads to corner 2 or to corner 3?
Rewards and Expected Rewards

Note: Distinction between expected reward and sample reward.
Sample reward is what we actually get. Will represent by \( r \).
Expected reward is what we may expect to get. Will represent by \( R \).

• One step expected reward: \( R_1 \)
  – Will this greater if the spider heads to corner 2 or to corner 3?

No route to corner 4 except from corner 3.
Where should the spider be?

• Flider has the option of landing on corner 1, 2 or 3 before she begins wandering the room
  – Which is the better corner to land on?
Where should the spider be?

- Flider has the option of landing on corner 1, 2 or 3 before she begins wandering the room
  - Which is the better corner to land on?

Need to know the long-term consequences of landing in the two corners

Where can she expect to get more food in the long term?
Where should the spider be?

Practice

• Assume she is allowed to “practice” once from each corner
  – To plan her future strategy
Where should the spider be?

- Must use her “practice” turn to assign a “value” to each of the corners
  - Guess how much food she would get in the long term from that corner
Flider practices

- Starting from 3, she gets $r_1$, $r_2$, $r_3$...

- Is $r_1 + r_2 + r_3$ ... a realistic representation of what she’d get if she did it again?
Flider practices

\( r_1 \) is somewhat realistic – it is obtained from corner 3

\( r_2 \): she had a choice of 3 corners for her next stop and chose one randomly during practice. Unlikely she’ll go to the same corner in the next run (less representative)

\( r_3 \): she had 9 possible corners to choose from in 2 steps. \( r_3 \) is even less representative of future runs

And so on...

- Starting from 3, she gets \( r_1, r_2, r_3 \)....
- Is \( r_1 + r_2 + r_3 \) ... a realistic representation of what she’d get if she did it again?
**Flider practices**

\[ r_1 \text{ is somewhat realistic – it is obtained from corner 3} \]

\[ r_2: \text{ she had a choice of 3 corners for her next stop and chose one randomly during practice. Unlikely she’ll go to the same corner in the next run (less representative)} \]

\[ r_3: \text{ she had 9 possible corners to choose from in 2 steps. } r_3 \text{ is even less representative of future runs} \]

And so on...

\[ \text{A better guess for how good it is to land at “3”:} \]

\[ r_1 + a_1 r_2 + a_2 r_3 + a_3 r_4 + \ldots \]

Where \( 0 \leq a_i \leq 1 \)

(you “trust” the readings from farther in the future less)

- Is \( r_1 + r_2 + r_3 \ldots \) a realistic representation of what she’d get if she did it again?
Flider practices

\( r_1 \) is somewhat realistic – it is obtained from corner 3

\( r_2 \): she had a choice of 3 corners for her next stop and chose one randomly during practice. Unlikely she’ll go to the same corner in the next run (less representative)

\( r_3 \): she had 9 possible corners to choose from in 2 steps. \( r_3 \) is even less representative of future runs

And so on...

A better guess for how good it is to land at “3”:

\[ r_1 + a_1 r_2 + a_2 r_3 + a_3 r_4 + \cdots \]

Where \( 0 \leq a_i \leq 1 \)

(you “trust” the readings from farther in the future less)

\bullet \text{A “mathematically good” choice: } a_i = \gamma^i \text{ where } 0 \leq \gamma \leq 1 \]

that she’d get if she did it again?
The discounted return

\[ G_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \]

• The return is the total future reward all the way to the end
• But each future step is slightly less “believable” and is hence discounted
  – We trust our own observations of the future less and less
    • The future is a fuzzy place

• The discount factor \( \gamma \) is our belief in the predictability of the future
  – \( \gamma = 0 \): The future is totally unpredictable, only trust what you see immediately ahead of you (myopic)
  – \( \gamma = 1 \): The future is clear; consider all of it (far sighted)

• Part of the Markov Reward Process model
Rewards

- Our spider goes wandering..
  \[ r_1 = 1, r_2 = 2, r_3 = 0.7, r_4 = 1.2, r_5 = 0.5, \ldots \]

- Are these *sample rewards* or *expected rewards*?
• Our spider goes wandering..
  \[ r_1 = 1, r_2 = 2, r_3 = 0.7, r_4 = 1.2, r_5 = 0.5, \ldots \]

• We decide the discounting factor \( \gamma = 1 \)
  – Really trusting the future

• What is the return \( G_t \) at \( t = 1 \)?
• Our spider goes wandering..
  \[ r_1 = 1, r_2 = 2, r_3 = 0.7, r_4 = 1.2, r_5 = 0.5, \ldots \]

• We decide the discounting factor \( \gamma = 1 \)
  – Really trusting the future

• What is the return \( G_t \) at \( t = 1 \)?
• What is the return \( G_t \) at \( t = 7 \)?
• Our spider goes wandering..
  \[ r_1 = 1, r_2 = 2, r_3 = 0.7, r_4 = 1.2, r_5 = 0.5, \ldots \]

• We decide the discounting factor \( \gamma = 1 \)
  – Really trusting the future

• What is the return \( G_t \) at \( t = 1 \)?

• What is the return \( G_t \) at \( t = 7 \)?

• Are these sample returns or expected returns?
Returns

• Discounted sample returns $G_t$ by themselves carry a fuzzy meaning
  – Why should we discount something we already observed?

• However, they make sense as samples of the possible future when you are at any state
  – If you are at any state, what is the expected return $E[G_t]$
Introducing the “Value” function

• The “Value” of a state is the expected total discounted return, starting from that state

  \[ V_s = E[G|S = s] \]

• This is not a function of time
  – i.e. it doesn’t matter *when* you arrive at \( s \), the expected return from that point on is \( V_s \)
The spider again

- The spider gains a reward of value 1 if she consumes the fly
- The spider has infinite patience
- What is the value of starting at each corner?
The spider again

- Regardless of which corner the spider starts at, she will eventually, randomly, nab the fly
- The expected return from any corner is 1!
- The value of being at any corner is 1 for all corners
The *hungry* spider

- The spider is hungry
- She gets a negative reward of -1 for every minute spent finding food
- What is the expected return if she starts at $c_1$
Posing the problem: There is a total reward/penalty associated with each corner
- $-1$ if the corner has no fly
  - Will definitely spend at least one more minute hunting
- $1$ at the corner that has the fly (satisfied!)

Thus $r_{c_x} = -1$ for $c_1 \ldots c_7$

$r_{c_8} = 1$

Note: We could also assign costs/rewards to edges in addition to nodes, if we want more detail, but won’t do so for our lectures
The *hungry* spider

\[ V_{c_1} = -1 + \frac{1}{3} V_{c_2} + \frac{1}{3} V_{c_3} + \frac{1}{3} V_{c_4} \]

\[ V_{c_2} = -1 + \frac{1}{3} V_{c_1} + \frac{1}{3} V_{c_7} + \frac{1}{3} V_{c_8} \]

\[ \vdots \]

\[ V_{c_8} = 1 \]

• A familiar solution

• Assuming \( \gamma = 1 \)
  – A natural fit in this problem
More generally

\[ V_{c_1} = R_{c_1} + \gamma \left( \frac{1}{3} V_{c_2} + \frac{1}{3} V_{c_3} + \frac{1}{3} V_{c_4} \right) \]

\[ V_{c_2} = R_{c_2} + \gamma \left( \frac{1}{3} V_{c_1} + \frac{1}{3} V_{c_7} + \frac{1}{3} V_{c_8} \right) \]

\[ \vdots \]

\[ V_{c_8} = R_{c_8} + \gamma \left( \frac{1}{3} V_{c_2} + \frac{1}{3} V_{c_3} + \frac{1}{3} V_{c_6} \right) \]

• A familiar solution
The Bellman Expectation Equation

• The value function of a state is the *expected discounted return*, when the process begins at that state

\[ G_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \]

\[ V_S = E[G | S = s] \]

• The Bellman Expectation Equation:

\[ V_S = R_S + \gamma \sum_{S'} P_{S',s} V_{S'} \]
Why discounted return?

- In processes with infinite horizon, which can go on for ever, the total undiscounted return will be infinite for every path: \( \sum_{k=0}^{\infty} r_{t+k+1} \) will be infinite for every path.
  - For finite horizon processes, a discount factor \( \gamma = 1 \) is good. It lets us talk in terms of actual total return.
  - For infinite horizon processes, discounting \( \gamma < 1 \) is required for meaningful mathematical analysis: \( \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \)
The Bellman Expectation Equation

\[ V_s = R_s + \gamma \sum_{s'} P_{s,s'} V_{s'} \]

\[
\begin{bmatrix}
V_{s_1} \\
V_{s_2} \\
\vdots \\
V_{s_N}
\end{bmatrix}
= 
\begin{bmatrix}
R_{s_1} \\
R_{s_2} \\
\vdots \\
R_{s_N}
\end{bmatrix}
+ \gamma
\begin{bmatrix}
P_{s_1,s_1} & P_{s_2,s_1} & \cdots & P_{s_N,s_1} \\
P_{s_1,s_2} & P_{s_2,s_2} & \cdots & P_{s_N,s_2} \\
\vdots & \vdots & \ddots & \vdots \\
P_{s_1,s_N} & P_{s_2,s_N} & \cdots & P_{s_N,s_N}
\end{bmatrix}
\begin{bmatrix}
V_{s_1} \\
V_{s_2} \\
\vdots \\
V_{s_N}
\end{bmatrix}
\]

\[ \mathcal{V} = \mathcal{R} + \gamma \mathcal{P} \mathcal{V} \]

- Bellman expectation equation in matrix form
The Bellman Expectation Equation

\[ V = R + \gamma PV \]

• Given the MRP \( M = \langle S, P, R, \gamma \rangle \)
  • I.e. the expected rewards at every state, and the transition probability matrix,
  – the value functions for all states can be easily computed through matrix inversion

\[ V = (I - \gamma P)^{-1}R \]

• Finding the values of states is a key problem in planning and reinforcement learning

• Unfortunately, for very large state spaces, the above matrix inversion is not tractable
  – Also not invertible for small state spaces if \( \gamma = 1 \)
    • Inversion cannot be used to find \( V \) even when it is finite (e.g. our fly problem), if \( \gamma = 1 \)

• Much of what we will deal with is how to tackle this problem
Moving on..

• Up next ... Markov *Decision* Processes
We have assumed so far that the agent behaves randomly
- The agent has no agency
- Let’s make the agent more intelligent..
A more realistic problem

• The spider actively chooses which way to move
  – The agent *takes action*
  – Ideally, it would move in the general direction of the fly

• However, each time the spider moves, the fly jumps up and settles at another corner
  – The agent’s action changes the environment!

How do we model this system?
Redefining the problem

• Each time the spider moves in any direction, the fly randomly jumps
• The fly arrives at a new state but ..
  – The state it arrives in depends on where the fly jumped
  – Which depends on which direction the Spider moved
• The spider’s action *modifies the state transition probabilities!!*
What is $P_{S,S'}^a$?

- Each time the spider moves in any direction, the fly randomly jumps.
- The fly arrives at a new state but..  
  - The state it arrives in depends on where the fly jumped. 
  - Which depends on which direction the Spider moved.
- The spider’s action *modifies the state transition probabilities!!*
What is \( P_{S,S'}^a \)

- Each time the spider moves in any direction, the fly randomly jumps.
- The fly arrives at a new state but ..
  - The state it arrives in depends on where the fly jumped
  - Which depends on which direction the Spider moved
- The spider’s action *modifies the state transition probabilities!!*
Trick Question: Redefining the States

- There are, in fact, only four states, not eight
  - Manhattan distance between fly and spider = 0 ($s_0$)
  - Distance between fly and spider = 1 ($s_1$)
  - Distance between fly and spider = 2 ($s_2$)
  - Distance between fly and spider = 3 ($s_3$)
- Can, in fact, redefine the MRP entirely in terms of these 4 states
- There are two actions $a+$ and $a-$
- Need an idea of the behavior of the fly
The Fly Markov Reward Process

- There are, in fact, only four states, not eight
  - Manhattan distance between fly and spider = 0 (s₀)
  - Distance between fly and spider = 1 (s₁)
  - Distance between fly and spider = 2 (s₂)
  - Distance between fly and spider = 3 (s₃)
- Can, in fact, redefine the MRP entirely in terms of these 4 states
The Markov *Decision* Process: Defining Actions

- Two types of actions:
  - \( a_+ \): Increases distance to fly by 1
  - \( a_- \): Decreases distance to fly by 1
The behavior of the fly:

- If the spider is moving *away from it*, it does nothing
- If the spider is moving *towards* it, it randomly hops to a different adjacent corner
  - 2/3 of the time, it increases the distance to the fly by 1
  - 1/3 of the time, it *decreases* the distance to the fly by 1
The Fly Markov Decision Process
Redefining the problem

• Each time the spider moves in any direction, the fly randomly jumps.

Note: This is a simile for many problems in life, e.g. driving, stock market, advertising, etc.

The agents actions modifies how the environment behaves

  – Which depends on which direction the spider moved

• The spider’s action modifies the state transition probabilities!!
The Markov Decision Process

- A *Markov Decision Process* is a Markov Reward Process, where the agent has the ability to decide its actions!
  - We will represent individual actions as $a$
  - We will represent the action at time $t$ as $A_t$

- The agent’s actions affect the environment’s behavior
  - The transitions made by the environment are functions of the action
  - The rewards returned are functions of the action
The Markov Decision Process

- Formally, a Markov Decision Process is the tuple $M = \langle S, P, \mathcal{A}, R, \gamma \rangle$
  - $S$ is a (possibly finite) set of states: $S = \{s\}$
  - $\mathcal{A}$ is a (possibly finite) set of actions: $\mathcal{A} = \{a\}$
  - $P$ is the set of action conditioned transition probabilities $P_{s,s'}^a = P(S_{t+1} = s|S_t = s', A_t = a)$
  - $R$ is an action conditioned reward function $R_s^a = E[r|S = s, A = a]$
  - $\gamma \in [0,1]$ is a discount factor
Introducing: Policy

• The policy is the probability distribution over actions that the agent may take at any state

\[ \pi(a|s) = P(A_t = a|S_t = s) \]

– What are the preferred actions of the spider at any state

• The policy may be deterministic, i.e.

\[ \pi(a|s) = 1 \text{ for } a = a_s; \quad 0 \text{ for } a \neq a_s \]

– where \( a_s \) is the preferred action in state \( s \)
An example of a policy

- Assuming the fly does not move
  - This example is not a particularly good policy for the spider
An example of a policy

- What are the (action dependent) transition probabilities of the states here?
An example of a policy

The transition probabilities depend on actions, but not on policy

• What are the (action dependent) transition probabilities of the states here?
An example of a policy

• Assuming the fly does not move
  – This is a different *optimal* policy
  – What are the transition probabilities here?
The value function of an MDP

• The *expected return* from any state depends on the policy you follow
The Fly MDP: Policy 1

\[ V_{s_1} = R_{s_1} + \gamma V_{s_1} \]

\[ V_{s_2} = R_{s_2} + \gamma \left( \frac{1}{3} V_{s_0} + \frac{2}{3} V_{s_2} \right) \]

\[ V_{s_3} = R_{s_3} + \gamma \left( \frac{2}{3} V_{s_1} + \frac{1}{3} V_{s_3} \right) \]
The Fly MDP: Policy 2 (optimal)

\[ V_{s_1} = R_{s_1} + \gamma V_{s_2} \]

\[ V_{s_2} = R_{s_2} + \gamma \left( \frac{1}{3} V_{s_0} + \frac{2}{3} V_{s_2} \right) \]

\[ V_{s_3} = R_{s_3} + \gamma \left( \frac{2}{3} V_{s_1} + \frac{1}{3} V_{s_3} \right) \]
The Fly MDP: Stochastic Policy
The Fly MDP: Stochastic Policy

\[ V_{s_1} = \frac{2}{3} (R_{s_1} + \gamma V_{s_2}) + \frac{1}{3} (R_{s_1} + \gamma V_{s_1}) \]

\[ V_{s_2} = \frac{1}{3} (R_{s_2} + \gamma V_{s_3}) + \frac{2}{3} \left( R_{s_2} + \gamma \left( \frac{1}{3} V_{s_0} + \frac{2}{3} V_{s_2} \right) \right) \]

\[ V_{s_3} = R_{s_3} + \gamma \left( \frac{2}{3} V_{s_1} + \frac{1}{3} V_{s_3} \right) \]
The **state value** function of an MDP

- The *expected return* from any state depends on the policy you follow.
- We will index the value of any state by the policy to indicate this.

\[
\nu_\pi(s) = \sum_{a \in A} \pi(a|s) \left( R_s^a + \gamma \sum_{s'} P_{s,s'}^a \nu_\pi(s') \right)
\]

**Bellman Expectation Equation for State Value Functions of an MDP**

*Note: Although reward was not dependent on action for the fly example, more generally it will be.*
The action value function of an MDP

- There are different value equations associated with different actions.
- So we can actually associate value to **state action pairs**.
- **Note:** The LHS in the equation is the action-specific value at the source state, but the RHS is the overall value of the target states.
The action value function of an MDP

- The expected return from any state under a given policy, when you follow a specific action

$$q_\pi(s, a) = R_s^a + \gamma \sum_{s'} P_{s,s',\nu_\pi(s')}$$

Bellman Expectation Equation for Action Value Functions of an MDP
All together now

• The Bellman expectation equation for state value function

\[ v_\pi(s) = \sum_{a \in A} \pi(a|s) \left( R_s^a + \gamma \sum_{s'} P_{s,s'}^a v_\pi(s') \right) \]

• For action value function

\[ q_\pi(s, a) = R_s^a + \gamma \sum_{s'} P_{s,s'}^a v_\pi(s') \]

• Giving you (obviously)

\[ v_\pi(s) = \sum_{a \in A} \pi(a|s) q_\pi(s, a) \]

• And

\[ q_\pi(s, a) = R_s^a + \gamma \sum_{s'} P_{s,s'}^a \sum_{a' \in A} \pi(a'|s') q_\pi(s', a') \]
The Bellman Expectation Equations

- The Bellman expectation equation for state value function

\[ \nu_\pi(s) = \sum_{a \in \mathcal{A}} \pi(a|s) \left( R_s^a + \gamma \sum_{s'} P_{s,s'}^a \nu_\pi(s') \right) \]

- The Bellman expectation equation for action value function

\[ q_\pi(s, a) = R_s^a + \gamma \sum_{s'} P_{s,s'}^a \sum_{a' \in \mathcal{A}} \pi(a'|s') q_\pi(s', a') \]
“Computing” the MDP

• Finding the state and/or action value functions for the MDP:
  – Given complete MDP (all transition probabilities $P_{s,s'}^a$, expected rewards $R_s^a$, and discount $\gamma$)
  – and a policy $\pi$
  – find all value terms $v_\pi(s)$ and/or $q_\pi(s,a)$

• The Bellman expectation equations are simultaneous equations that can be solved for the value functions
  – Although this will be computationally intractable for very large state spaces
Computing the MDP

\[ \mathcal{V}_\pi = \mathcal{R}_\pi + \gamma \mathcal{P}_\pi \mathcal{V}_\pi \]

• Given the expected rewards at every state, the transition probability matrix, the discount factor and the policy:

\[ \mathcal{V}_\pi = (I - \gamma \mathcal{P}_\pi)^{-1} \mathcal{R}_\pi \]

• Matrix inversion \( O(N^3) \); intractable for large state spaces
Optimal Policies

• Different policies can result in different value functions

• What is the optimal policy?

• The optimal policy is the policy that will maximize the expected total discounted reward at every state:

\[
E[G_t | S_t = s]
\]

\[
= E \left[ \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | S_t = s \right]
\]
Optimal Policies

• Different policies can result in different value functions

• What is the optimal policy?

• The optimal policy is the policy that will maximize the expected total discounted reward at every state:

\[ E[G_t | S_t = s] \]

\[ = E \left[ \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | S_t = s \right] \]

– Recall: why do we consider the discounted return, rather than the actual return \( \sum_{k=0}^{\infty} r_{t+k+1} \)?
Policy Ordering Definition

• A policy $\pi$ is “better” than a policy $\pi'$ if the value function under $\pi$ is greater than or equal to the value function under $\pi'$ at all states

$$\pi \geq \pi' \Rightarrow \nu_\pi(s) \geq \nu_{\pi'}(s) \ \forall s$$

• Under the better policy, you will expect better overall outcome no matter what the current state
The optimal policy theorem

• **Theorem**: For any MDP there exists an optimal policy $\pi_*$ that is better than or equal to every other policy:
  \[ \pi_* \geq \pi \quad \forall \pi \]

• **Corollary**: If there are *multiple* optimal policies $\pi_{opt1}, \pi_{opt2}, \ldots$ all of them achieve the same value function
  \[ v_{\pi_{opti}}(s) = v_*(s) \quad \forall s \]

• All optimal policies achieve the same action value function
  \[ q_{\pi_{opti}}(s, a) = q_*(s, a) \quad \forall s, a \]
How to find the optimal policy

• For the optimal policy:

\[ \pi_*(a|s) = \begin{cases} 
1 & \text{for} \quad \arg \max_{a'} q_*(s, a') \\
0 & \text{otherwise}
\end{cases} \]

• Easy to prove
  – For any other policy \( \pi \), \( q_\pi(s, a) \leq q_*(s, a) \)

• Knowing the optimal action value function \( q_*(s, a) \) \( \forall s, a \) is sufficient to find the optimal policy
The optimal value function

\[ \pi_*(a|s) = \begin{cases} 
1 & \text{for } \arg\max_{a'} q_*(s, a') \\
0 & \text{otherwise}
\end{cases} \]

• Which gives us

\[ \nu_*(s) = \max_a q_*(s, a) \]
Pictorially

Figures from Sutton

\[ v_*(s) = \max_a q_*(s, a) \]

- Blank circles are states, filled dots are state-action pairs
The optimal value function

\[ \pi_*(a|s) = \begin{cases} 
1 & \text{for } \arg\max_{a'} q_*(s, a') \\
0 & \text{otherwise} 
\end{cases} \]

- Which gives us
  \[ \nu_*(s) = \max_a q_*(s, a) \]
- But, for the optimal policy we also have
  \[ q_*(s, a) = R_s^a + \gamma \sum_{s'} P_{s,s'}^a \nu_*(s') \]
Backup Diagram

Figures from Sutton

\[ v_*(s) = \max_a q_*(s, a) \]

\[ q_*(s, a) = R_s^a + \gamma \sum_{s'} p_{s,s'}^a v_*(s') \]
$v_*(s) = \max_a R_s^a + \gamma \sum_{s'} P_{s,s'}^a v_*(s')$
Backup Diagram

Figures from Sutton

\[ q_*(s, a) = R_s^a + \gamma \sum_{s'} P_{s,s'}^a v_*(s') \]
$q_*(s, a) = R_s^a + \gamma \sum_{s'} P_{s,s'}^a v_*(s')$

$v_*(s') = \max_{a'} q_*(s', a')$
Backup Diagram

Figures from Sutton

\[ q^*(s, a) = R_s^a + \gamma \sum_{s', a'} P_{s,s'}^a \max_{a'} q^*(s', a') \]
Bellman Optimality Equations

• Optimal value function equation

\[ v_*(s) = \max_a R_s^a + \gamma \sum_{s'} P_{s,s'} v_*(s') \]

• Optimal action value equation

\[ q_*(s, a) = R_s^a + \gamma \sum_{s', a'} P_{s,s'} \max_{a'} q_*(s', a') \]
Optimality Relationships

• Given the MDP: \( \langle S, P, A, R, \gamma \rangle \)
• Given the optimal action value functions, the optimal value function can be found

\[
v_*(s) = \max_a q_*(s, a)
\]

• Given the optimal value function, the optimal action value function can be found

\[
q_*(s, a) = R_s + \gamma \sum_{s'} P_{s,s'}^a v_*(s')
\]

• Given the optimal action value function, the optimal policy can be found

\[
\pi_*(a|s) = \begin{cases} 
1 & \text{for } \arg\max_{a'} q_*(s, a') \\
0 & \text{otherwise}
\end{cases}
\]
“Solving” the MDP

• Solving the MDP equates to finding the optimal policy $\pi_*(a|s)$

• Which is equivalent to finding the optimal value function $v_*(s)$

• Or finding the optimal action value function $q_*(s,a)$

• Various solutions will estimate one or the other
  – Value based solutions solve for $v_*(s)$ and $q_*(s,a)$ and derive the optimal policy from them
  – Policy based solutions directly estimate $\pi_*(a|s)$
Solving the Bellman Optimality Equation

• No closed form solutions

• Solutions are iterative

• Given the MDP (Planning):
  – Value iterations
  – Policy iterations

• Not given the MDP (Reinforcement Learning):
  – Q-learning
  – SARSA
QUESTIONS before we dive?