Motivation for Today’s Class

Planning vs Decision Making

obstacle

big black hole

0.8

0.1

0.1
Today’s Lecture

• Markov Decision Processes
  – Markov Chain
  – Markov Decision Processes
  – Bellman Equation

• Partial Observable Markov Decision Processes
Markov Decision Processes

Mathematical framework for modeling decision processes

Assumption: The world is fully observable

Assumption: cs.utexas.edu/~eladlieb
State Representation

What is the state of the environment?

\[ S = \{ x, y \} \]
\[ x, y \in \{0,1,2,3\} \]

\[ S = \{ x, y, \theta \} \]
\[ x, y \in \{0,1,2,3\} \]
\[ \theta \in \{0,90,180,270\} \]
State Representation

What is the state of the environment?

States $s \in S$, e.g.,
- Joint positions and velocities
- Ball position and velocity
- Opponent position
Markov Property

Property of stochastic process

\[ P(s_{t+1}|s_t) = P(s_{t+1}|s_1, s_2, ..., s_t) \]

State is sufficient statistic

Defines the condition that the decision of an agent only depends on its current state and not on those before
A Markov Chain is a tupel \( \langle S, \mathcal{T} \rangle \)

- \( S \) is a finite set of states \( S = \{s_1, s_2, \ldots, s_n\} \)
- \( \mathcal{T} \) is the state transition probability

\[
\mathcal{T}_{ss'} = P(s_{t+1} = s' | s_t = s)
\]

**Weather in Pittsburgh**

- Rain
  - \( P(s_{t+1} = \text{Sun} | s_t = \text{Rain}) = 0.8 \)
  - \( P(s_{t+1} = \text{Rain} | s_t = \text{Rain}) = 0.2 \)

- Sun
  - \( P(s_{t+1} = \text{Rain} | s_t = \text{Sun}) = 0.7 \)
  - \( P(s_{t+1} = \text{Sun} | s_t = \text{Sun}) = 0.3 \)
Transition Model

For a finite state of states, we can define the transition matrix

\[ T = \begin{bmatrix}
T_{11} & \cdots & T_{1n} \\
\vdots & \ddots & \vdots \\
T_{n1} & \cdots & T_{nn}
\end{bmatrix} \]

Rain  Sun

\[ T = \begin{bmatrix}
0.8 & 0.2 \\
0.7 & 0.3
\end{bmatrix} \]
Markov Decision Processes

A world consisting of states, actions, and rewards

A Markov Decision Process is a tupel \( \langle S, A, T, R, \gamma \rangle \)

- \( S \) is a finite set of states \( S = \{s_1, s_2, ..., s_n\} \)
- \( A \) is a finite set of actions \( A = \{a_1, a_2, ..., a_m\} \)
- \( T \) is the state transition probability
  \[ T_{sas'} = P(s_{t+1} = s'|s_t = s, a_t = a) \]
- \( R \) is a reward function
- \( \gamma \) is a discount factor \( \gamma \in [0,1] \)
Actions

What are actions?

\[ S = \{x, y\} \]
\[ x, y \in \{0,1,2,3\} \]
\[ A = \{\text{forward, left, right, back}\} \]

\[ S = \{x, y, \theta\} \]
\[ x, y \in \{0,1,2,3\} \]
\[ \theta \in \{0,90,180,270\} \]
\[ A = \{\text{move\_forward, rotate\_left, rotate\_right}\} \]
Actions

What are actions?

Model Hitting Movements

• Joint acceleration at each time step

Modeling Strategy

• Choice of hitting movement and
• Where to return the ball
The reward $r(s)$, tells us how good a particular state is
- Only depends on the state you are in: $R(s)$
- Depends on state and action: $R(s,a)$
- Or general case $R(s,a,s')$

The reward may be positive or negative

Here: without loss of generality write only $R(s)$
When moving through our environment we accumulate reward
When moving through our environment we accumulate reward

The total reward from time step $t$:

$$R(s_t) + R(s_{t+1}) + \cdots = \sum_{i=1}^{T} R_{s_{t+i}}$$

What is the expected reward?
A policy is a mapping from states to actions

\[ \pi(a_t | s_t) = P(a_t | s_t) \quad \text{stochastic} \]

\[ \pi(s_t) = a_t \quad \text{deterministic} \]

• A policy tells the system what to do next in each state
• Markov property: it only depends on the current state and not the history
• Policies are stationary \( a_t \sim \pi(\cdot | s_t) \)

Goal is to find an optimal policy \( \pi^* \), i.e., the best solution to our problem
Transition Model

Transition Model now depends on the actions the agent take

Deterministic Action

• $T: S \times A \rightarrow A$. No uncertainty in the execution. State and actions define the next state

Stochastic Action

• $T: S \times A \rightarrow P(s'|s, a)$. Models events not controlled by the agent.
Deterministic policy

Stochastic policy
Policy

Stochastic policy

0.8
0.1
0.1
The reward changes the balance of risk and reward and therefore shapes the policy.
**Value Function**

Quantifies how beneficial a state is for the overall task

The expected reward of a state $V: S \rightarrow \mathbb{R}$ is called the value function

\[
V_\pi(s) = \mathbb{E}\left[ \sum_{t=1}^{H} R(s_t) \middle| s_0 = s, \pi \right]
\]

Finite horizon

\[
V_\pi(s) = \mathbb{E}\left[ \sum_{t=1}^{\infty} \gamma^t R(s_t) \middle| s_0 = s, \pi \right]
\]

Infinite horizon

Discount factor
Discount Factor

\[
V_\pi(s) = \mathbb{E}\left[\sum_{t=1}^{\infty} \gamma^t R(s_t) \mid s_0 = s, \pi\right]
\]

The discount \( \gamma \in [0,1] \) is a
• Way to control the return
• Avoid infinite cycles
• Represents uncertainty in future

Models the agents preference towards immediate rewards
• \( \gamma \) close to 0 leads to
• \( \gamma \) close to 1 leads to
### Value Function

<table>
<thead>
<tr>
<th>Reward</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td></td>
<td>-1</td>
<td>-1</td>
<td>-100</td>
<td></td>
</tr>
<tr>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-100</td>
<td></td>
</tr>
<tr>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[ \gamma = 0 \]
Value Function

Reward

<table>
<thead>
<tr>
<th></th>
<th>-1</th>
<th>-1</th>
<th>-1</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-100</td>
<td></td>
</tr>
<tr>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td></td>
</tr>
<tr>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td></td>
</tr>
<tr>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td></td>
</tr>
</tbody>
</table>

Value

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[
\gamma = 1
\]
Value Function

Value function can be decomposed into two parts

- Immediate reward
- Value of next state

\[
V_\pi(s) = \mathbb{E} \left[ \sum_{t=0}^{\infty} \gamma^t R(s_t) \mid s_0 = s, \pi \right]
\]

\[
= \mathbb{E} [R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + \cdots \mid s_0 = s, \pi]
\]

\[
= \mathbb{E} [R(s_0) + \gamma (\gamma^{t-1} R(s_t)) \mid s_0 = s, \pi]
\]

\[
= \mathbb{E} [R(s_0) \mid s_0 = s, \pi] + \gamma V_\pi(s_{t+1})
\]
The Q-function $Q_{\pi}(s, a)$ is the expected return at state $s$, taking action $a$ and then following policy $\pi$

$$Q_{\pi}(s, a) = \mathbb{E} \left[ \sum_{t=0}^{\infty} \gamma^t R(s_t) \mid s_0 = s, a_0, \pi \right]$$

Tells us how good it is to take a particular action in a state
The Q-function $Q_\pi(s, a)$ is the expected return at state $s$, taking action $a$ and then following policy $\pi$

$$Q_\pi(s, a) = \mathbb{E} \left[ \sum_{t=0}^{\infty} \gamma^t R(s_t) \mid s_0 = s, a_0, \pi \right]$$

$$Q_\pi(s, a) = \mathbb{E} \left[ R(s, a) + \gamma \sum_{t=1}^{\infty} \gamma^{t-1} R(s_t) \mid s, a, \pi \right]$$

$$= \mathbb{E}[R(s, a) \mid s = s, a, \pi] + \gamma Q_\pi(s_{t+1}, a_{t+1})$$
What happens if the policy is stochastic?

\[ P_\pi(s, s') = \sum_{a \in A} \pi(a|s)P(s'|s, a). \]
Value Function

Two step look-ahead:
1. Consider all actions we can take
2. Consider dynamics of system and environment

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

\[ = \sum_{a \in A} \pi(a|s) \left( R(s,a) + \gamma \sum_{s' \in S} P(s,a,s') V_\pi(s') \right) \]
\[
Q_\pi(s, a) = R(s, a) + \gamma \sum_{s' \in S} P(s', a, s)V_\pi(s')
\]

\[
= R(s, a) + \gamma \sum_{a' \in A} \pi(a'|s')Q_\pi(s', a')
\]
Optimal Policy

Goal is to find an optimal policy $\pi^*$, i.e., the best solution to our problem

Value function defines a partial ordering over the policy

$$\pi \geq \pi' \text{ if } V_\pi(s) \geq V_{\pi'}(s) \; \forall s$$

An optimal policy can be defined in terms of the Value and Q-Value function

$$\pi^*(a|s) = \begin{cases} 1 & \text{if } a = \arg\max_{a \in A} Q_{\pi^*}(s, a) \\ 0 & \text{otherwise} \end{cases}$$
Bellman Equation

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

Stochastic process

\[ V_{\pi^*}(s) = \max_a Q_{\pi^*}(s,a) \]

Active, optimal decision process
Bellman Equation

\[ Q_\pi(s, a) = R(s, a) + \gamma \sum_{s' \in S} P(s', a, s)V_\pi(s') \]

Agent does not control the dynamics!
Bellman Equation

\[ V_{\pi^*}(s) = \max_a Q_{\pi^*}(s, a) = \max_a R(s, a) + \gamma \sum_{s' \in S} P(s', a, s)V_{\pi^*}(s') \]
Bellman Equation

\[
Q_{\pi^*}(s, a) = R(s, a) + \gamma \sum_{s' \in S} P(s', a, s) V_{\pi^*}(s')
\]

\[
= R(s, a) + \gamma \sum_{s' \in S} P(s', a, s) \max_{a'} Q_{\pi^*}(s', a')
\]
Bellman Equation

Recursive relationship

Problem:
• No closed form solution due to the max-operator that chooses the optimal action → makes the system nonlinear

Need iterative approach
• Dynamic Programming
• Value Iteration
• Q-learning

\[ V_{\pi^*}(s) = \max_a R(s, a) + \gamma \sum_{s' \in S} P(s', a, s)V_{\pi^*}(s') \]
Partial Observability

**MDP**: Assumption that the state is fully observable

Is this always the case?

System state can not always be determined

→ Partially Observable MDP (PoMDP)

• Action outcomes not fully observable
• Add O: set of observations
• Add Z: observation function

\[ Z(a, o, s) = P(o|s, a) \]
Partial Observable MDP (POMDP)

**MDP**: Assumption that the state is fully observable

Is this always the case?

**POMDP**: Not everything in our world is observable

\[
\begin{align*}
\text{Observation: } & y_t \\
\text{Belief state: } & b_t = P(s_t)
\end{align*}
\]
Solving an MDP

Objective of Reinforcement Learning:
Find a mapping from states to actions such that we maximize the cumulative expected reward

\[ J^\pi = E \left[ \sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \mid \pi \right] \]

Expected return
Solving an MDP

- **Reward** $R$
- **Reinforcement Learning, Optimal Control**
- **Control Policy** $\pi$
- **Dynamical Model** $T$
- **Inverse Reinforcement Learning**
- **Behavioral Cloning**
- **Expert Demonstration**
Summary

MDPs model sequential decision problems

- Model world as state, action, and rewards
- Assume state is fully observable
- State represents all necessary information about the environment and the agent
- Actions: change the state of the world
- Reward: measures the success of an action
- Policy: mapping from states to actions