Reinforcement Learning

Part 1

12 October 2020
Exam 1

Two hours (+ 30 min to deal with printing/scanning)
Due 11:59 p.m. tonight

Assignment 1

Should be done or finishing very soon

Assignment 2

Should have started it
Recall

Markov decision process (MDP):

A set of states $s \in S$

A set of actions (per state) $A$

A model $T(s, a, s')$

A reward function $R(s, a, s')$
Double bandits
Double-bandit MDP

Actions: Blue, Red

States: Win, Lose

No discount
10 time steps
Both states have the same value
Double-bandits MDP

No discount
10 time steps
Both states have the same value

Play Blue: Value is $10

Play Red: Value is $15
Double-bandits MDP

Play **Blue**: Value is $10

Play **Red**: Value is $15

This needs to be finite or they’d both have infinite value.
Offline planning

Solving MDPs is offline planning.

You determine all quantities through computation.
You need to know the details of the MDP.
You don’t need to actually play the game!
Let's play!
Let’s play!

$2
Let’s play!
Let’s play!

$2  $2  $0
Let’s play!

$2 $2 $0 $2
Let’s play!

$2  $2  $0  $2  $2
Let’s play!

$2  $2  $0  $2  $2

$2
Let’s play!

$2 \quad $2 \quad $0 \quad $2 \quad $2

$2 \quad $2
Let’s play!
Let’s play!

$2 $2 $0 $2 $2
$2 $2 $0 $0
Let’s play!

$2 $2 $0 $2 $2

$2 $2 $0 $0 $0 $0
Online planning

Rules changed! The chance of winning on the red machine is different.
Let’s play!
Let’s play!
Let’s play!

$0  $0
Let’s play!
Let’s play!
Let’s play!

$1

$0 $0 $0 $2
Let’s play!
Let’s play!

$1, $1, $1

$0, $0, $0, $2
What just happened?

That wasn’t planning; it was learning!

Specifically, it was reinforcement learning (RL).

There was an MDP, but you couldn’t solve it with just computation; you needed to actually act to figure it out.

Important ideas in RL that came up:

*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 much harder than solving a known MDP.
Reinforcement learning
Reinforcement learning

Still assume a Markov decision process (MDP):

- A set of states $s \in S$
- A set of actions (per state) $A$
- A model $T(s, a, s')$
- A reward function $R(s, a, s')$

Still looking for a policy $\pi(s)$
Reinforcement learning

Still assume a Markov decision process (MDP):

- A set of states $s \in S$
- A set of actions (per state) $A$
- A model $T(s, a, s')$
- A reward function $R(s, a, s')$

Still looking for a policy $\pi(s)$

New twist: *don’t know $T$ or $R$.*

I.e., we don’t know which states are good or what the actions do.
We need to actually try actions and states out to learn.
Basic idea:

Receive feedback in the form of *rewards*

Agent's utility is defined by the reward function

Must learn how to act to *maximize expected rewards*

All learning is based on observed samples of outcomes!
Example: Learning to walk

- Initial
- A learning trial
- After learning (1000 trials)
Example: Learning to walk

Initial
Example: Learning to walk
Example: Learning to walk

Training
Example: Learning to walk

Training
Example: Learning to walk

Finished
Example: Learning to walk

Finished
The crawler!
average speed: 2.31191464863606509
DeepMind Atari

© Two Minute Lectures
DeepMind Atari
Reinforcement learning

Still assume a Markov decision process (MDP):

A set of states $s \in S$
A set of actions (per state) $A$
A model $T(s, a, s')$
A reward function $R(s, a, s')$

Still looking for a policy $\pi(s)$
Reinforcement learning

Still assume a Markov decision process (MDP):

- A set of states $s \in S$
- A set of actions (per state) $A$
- A model $T(s, a, s')$
- A reward function $R(s, a, s')$

Still looking for a policy $\pi(s)$

New twist: don’t know $T$ or $R$.

I.e., we don’t know which states are good or what the actions do.

We need to actually try actions and states out to learn.
Offline (MDPs) vs online (RL)
Offline solution
Offline (MDPs) vs online (RL)

Offline solution

Online learning
Reinforcement learning

Model-based learning

Model-free learning
Model-based learning
Model-based idea:

Learn an approximate model based on experiences.
Solve for values as if the learned model were correct.
Model-based idea:

Learn an approximate model based on experiences.
Solve for values as if the learned model were correct.

**Step 1: Learn empirical MDP model**

Count the number of times we end up in $s'$ when we take action $a$ in state $s$, i.e., after entering q-state $(s, a)$.

Normalize the counts to give $\hat{T}(s, a, s')$, an estimate of the transition function $T$. (Make the count for each $s'$ into a probability by dividing it by the total number of instances.)

Discover each $\hat{R}(s, a, s')$ when we experience $(s, a, s')$. 
Model-based idea:

Learn an approximate model based on experiences. Solve for values as if the learned model were correct.

**Step 1: Learn empirical MDP model**

Count the number of times we end up in $s'$ when we take action $a$ in state $s$, i.e., after entering q-state $(s, a)$.

Normalize the counts to give $\hat{T}(s, a, s')$, an estimate of the transition function $T$. (Make the count for each $s'$ into a probability by dividing it by the total number of instances.)

Discover each $\hat{R}(s, a, s')$ when we experience $(s, a, s')$.

**Step 2: Solve the learned MDP**

For example, use value iteration, as before.
Example: Model-based learning

Input policy $\pi$

Assume: $\gamma = 1$
Example: Model-based learning

**Input policy** $\pi$

Assume: $\gamma = 1$

**Observed episodes (training)**

**Episode 1**
- B, east, C, $-1$
- C, east, D, $-1$
- D, exit, $x$, $+10$

**Episode 2**
- B, east, C, $-1$
- C, east, D, $-1$
- D, exit, $x$, $+10$

**Episode 3**
- E, north, C, $-1$
- C, east, D, $-1$
- D, exit, $x$, $+10$

**Episode 4**
- E, north, C, $-1$
- C, east, A, $-1$
- A, exit, $x$, $-10$
Example: Model-based learning

Input policy $\pi$

Observe episodes (training)

Assume: $\gamma = 1$

<table>
<thead>
<tr>
<th>Episode 1</th>
<th>Episode 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>B, east, C, $-1$</td>
<td>B, east, C, $-1$</td>
</tr>
<tr>
<td>C, east, D, $-1$</td>
<td>C, east, D, $-1$</td>
</tr>
<tr>
<td>D, exit, $x$, $+10$</td>
<td>D, exit, $x$, $+10$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Episode 3</th>
<th>Episode 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>E, north, C, $-1$</td>
<td>E, north, C, $-1$</td>
</tr>
<tr>
<td>C, east, D, $-1$</td>
<td>C, east, A, $-1$</td>
</tr>
<tr>
<td>D, exit, $x$, $+10$</td>
<td>A, exit, $x$, $-10$</td>
</tr>
</tbody>
</table>
Example: Model-based learning

**Input policy** \( \pi \)

**Observed episodes (training)**

<table>
<thead>
<tr>
<th>Episode 1</th>
<th>Episode 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>B, east, C, -1</td>
<td>B, east, C, -1</td>
</tr>
<tr>
<td>C, east, D, -1</td>
<td>C, east, D, -1</td>
</tr>
<tr>
<td>D, exit, x, +10</td>
<td>D, exit, x, +10</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Episode 3</th>
<th>Episode 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>E, north, C, -1</td>
<td>E, north, C, -1</td>
</tr>
<tr>
<td>C, east, D, -1</td>
<td>C, east, A, -1</td>
</tr>
<tr>
<td>D, exit, x, +10</td>
<td>A, exit, x, -10</td>
</tr>
</tbody>
</table>

Assume: \( \gamma = 1 \)
<table>
<thead>
<tr>
<th>$s$</th>
<th>$a$</th>
<th>$s'$</th>
<th>count</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>exit</td>
<td>x</td>
<td>1</td>
</tr>
<tr>
<td>B</td>
<td>east</td>
<td>C</td>
<td>2</td>
</tr>
<tr>
<td>C</td>
<td>east</td>
<td>A</td>
<td>1</td>
</tr>
<tr>
<td>C</td>
<td>east</td>
<td>D</td>
<td>3</td>
</tr>
<tr>
<td>D</td>
<td>exit</td>
<td>x</td>
<td>3</td>
</tr>
<tr>
<td>E</td>
<td>north</td>
<td>C</td>
<td>2</td>
</tr>
<tr>
<td>$s$</td>
<td>$a$</td>
<td>$s'$</td>
<td>count</td>
</tr>
<tr>
<td>-----</td>
<td>------</td>
<td>------</td>
<td>-------</td>
</tr>
<tr>
<td>A</td>
<td>exit</td>
<td>x</td>
<td>1</td>
</tr>
<tr>
<td>B</td>
<td>east</td>
<td>C</td>
<td>2</td>
</tr>
<tr>
<td>C</td>
<td>east</td>
<td>A</td>
<td>1</td>
</tr>
<tr>
<td>C</td>
<td>east</td>
<td>D</td>
<td>3</td>
</tr>
<tr>
<td>D</td>
<td>exit</td>
<td>x</td>
<td>3</td>
</tr>
<tr>
<td>E</td>
<td>north</td>
<td>C</td>
<td>2</td>
</tr>
</tbody>
</table>

### Transition function

\[
\hat{T}(A, \text{exit}, x) = \frac{\#(A, \text{exit}, x)}{\#(A, \text{exit})} = \frac{1}{1} = 1
\]

\[
\hat{T}(B, \text{east}, C) = \frac{\#(B, \text{east}, C)}{\#(B, \text{east})} = \frac{2}{2} = 1
\]

\[
\hat{T}(C, \text{east}, A) = \frac{\#(C, \text{east}, A)}{\#(C, \text{east})} = \frac{1}{4} = 0.25
\]

\[
\hat{T}(C, \text{east}, D) = \frac{\#(C, \text{east}, D)}{\#(C, \text{east})} = \frac{3}{4} = 0.75
\]

\[
\hat{T}(D, \text{exit}, x) = \frac{\#(D, \text{exit}, x)}{\#(D, \text{exit})} = \frac{3}{3} = 1
\]

\[
\hat{T}(E, \text{north}, C) = \frac{\#(E, \text{north}, C)}{\#(E, \text{north})} = \frac{2}{2} = 1
\]
Example: Model-based learning

Input policy $\pi$

Assume: $\gamma = 1$

Observed episodes (training)

Episode 1
- B, east, C, −1
- C, east, D, −1
- D, exit, x, +10

Episode 2
- B, east, C, −1
- C, east, D, −1
- D, exit, x, +10

Episode 3
- E, north, C, −1
- C, east, D, −1
- D, exit, x, +10

Episode 4
- E, north, C, −1
- C, east, A, −1
- A, exit, x, −10

Learned model

$\hat{T}(s, a, s')$
- $T(B, \text{east, C}) = 1.00$
- $T(C, \text{east, D}) = 0.75$
- $T(C, \text{east, A}) = 0.25$
- ...

$\hat{R}(s, a, s')$
- $R(B, \text{east, C}) = -1$
- $R(C, \text{east, D}) = -1$
- $R(D, \text{exit, x}) = +10$
- ...

Reinforcement learning

Model-based learning

Model-free learning
Example: Expected age

Goal: Compute expected age of CMPU 365 students
Example: Expected age

Goal: Compute expected age of CMPU 365 students

Known $P(A)$
Example: Expected age

**Goal:** Compute expected age of CMPU 365 students

<table>
<thead>
<tr>
<th>Known $P(A)$</th>
</tr>
</thead>
</table>

\[
E[A] = \sum_a P(a) \cdot a
\]
Example: Expected age

**Goal:** Compute expected age of CMPU 365 students

**Known \( P(A) \)**

\[
E[A] = \sum_a P(a) \cdot a = 0.35 \times 20 + \ldots
\]
Example: Expected age

**Goal:** Compute expected age of CMPU 365 students

**Known P(A)**

\[ E[A] = \sum_{a} P(a) \cdot a = 0.35 \times 20 + \ldots \]

Without \( P(A) \), instead collect samples \([a_1, a_2, \ldots, a_N]\)
Example: Expected age

**Goal:** Compute expected age of CMPU 365 students

<table>
<thead>
<tr>
<th>Known $P(A)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E[A] = \sum_a P(a) \cdot a = 0.35 \times 20 + \ldots$</td>
</tr>
</tbody>
</table>

Without $P(A)$, instead collect samples $[a_1, a_2, \ldots, a_N]$

<table>
<thead>
<tr>
<th>Unknown $P(A)$: Model-based</th>
</tr>
</thead>
</table>
Example: Expected age

Goal: Compute expected age of CMPU 365 students

**Known P(A):**

\[ E[A] = \sum_a P(a) \cdot a = 0.35 \times 20 + \ldots \]

Without \( P(A) \), instead collect samples \([a_1, a_2, \ldots, a_N]\)

**Unknown P(A): Model-based**

\[ \hat{P}(a) = \frac{\text{num}(a)}{N} \]
Example: Expected age

**Goal:** Compute expected age of CMPU 365 students

**Known P(A)**

\[ E[A] = \sum_a P(a) \cdot a = 0.35 \times 20 + \ldots \]

Without P(A), instead collect samples \([a_1, a_2, \ldots, a_N]\)

**Unknown P(A): Model-based**

\[ \hat{P}(a) = \frac{\text{num}(a)}{N} \]

\[ E[A] \approx \sum_a \hat{P}(a) \cdot a \]
**Goal:** Compute expected age of CMPU 365 students

**Known P(A):**

\[
E[A] = \sum_a P(a) \cdot a = 0.35 \times 20 + \ldots
\]

Without \(P(A)\), instead collect samples \([a_1, a_2, \ldots, a_N]\)

**Unknown P(A): Model-based**

\[
\hat{P}(a) = \frac{\text{num}(a)}{N}
\]

\[
E[A] \approx \sum_a \hat{P}(a) \cdot a
\]

Why does this work? Because eventually you learn the right model.
Example: Expected age

Goal: Compute expected age of CMPU 365 students

<table>
<thead>
<tr>
<th>Known $P(A)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E[A] = \sum_a P(a) \cdot a = 0.35 \times 20 + \ldots$</td>
</tr>
</tbody>
</table>

Without $P(A)$, instead collect samples $[a_1, a_2, \ldots, a_N]$.

<table>
<thead>
<tr>
<th>Unknown $P(A)$: Model-based</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\hat{P}(a) = \frac{\text{num}(a)}{N}$</td>
</tr>
<tr>
<td>$E[A] \approx \sum_a \hat{P}(a) \cdot a$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Unknown $P(A)$: Model-free</th>
</tr>
</thead>
</table>

Why does this work? Because eventually you learn the right model.
Example: Expected age

**Goal:** Compute expected age of CMPU 365 students

<table>
<thead>
<tr>
<th>Known $P(A)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E[A] = \sum_a P(a) \cdot a = 0.35 \times 20 + \ldots$</td>
</tr>
</tbody>
</table>

Without $P(A)$, instead collect samples $[a_1, a_2, \ldots, a_N]$

<table>
<thead>
<tr>
<th>Unknown $P(A)$: Model-based</th>
<th>Unknown $P(A)$: Model-free</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\hat{P}(a) = \frac{\text{num}(a)}{N}$</td>
<td>$E[A] \approx \frac{1}{N} \sum_i a_i$</td>
</tr>
<tr>
<td>$E[A] \approx \sum_a \hat{P}(a) \cdot a$</td>
<td></td>
</tr>
</tbody>
</table>
**Example: Expected age**

**Goal:** Compute expected age of CMPU 365 students

<table>
<thead>
<tr>
<th>Known $P(A)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E[A] = \sum_a P(a) \cdot a = 0.35 \times 20 + \ldots$</td>
</tr>
</tbody>
</table>

Without $P(A)$, instead collect samples $[a_1, a_2, \ldots, a_N]$.

<table>
<thead>
<tr>
<th>Unknown $P(A)$: Model-based</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\hat{P}(a) = \frac{\text{num}(a)}{N}$</td>
</tr>
<tr>
<td>$E[A] \approx \sum_a \hat{P}(a) \cdot a$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Unknown $P(A)$: Model-free</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E[A] \approx \frac{1}{N} \sum_i a_i$</td>
</tr>
</tbody>
</table>

Why does this work? Because eventually you learn the right model.

Why does this work? Because eventually you learn the right model.

Why does this work? Because samples appear with the right frequencies.
Model-free learning
Reinforcement learning

Model-based learning

Model-free learning

Passive

Active
Passive reinforcement learning
Passive reinforcement learning

**Simplified task:** policy evaluation

*Input:* a fixed policy \( \pi(s) \)

You don’t know the transitions \( T(s, a, s') \)

You don’t know the rewards \( R(s, a, s') \)

*Goal:* learn the state values

In this case:

Learner is “along for the ride” – no choice about what actions to take.

Just execute the policy and learn from experience!

This is *not* offline planning! You actually take actions in the world.
**Goal:** Compute values for each state under policy $\pi$.

**Idea:** Average together observed sample values.
**Goal**: Compute values for each state under policy $\pi$.

**Idea**: Average together observed sample values.

- Act according to $\pi$.
- Every time you visit a state, write down what the sum of discounted rewards turned out to be.
- Average those samples.

This is called *direct evaluation*.
Example: Direct evaluation

*Input policy* $\pi$

```
      A
     B ▶ C ▶ D
      ▲ E
```

Assume: $\gamma = 1$
Example: Direct evaluation

Input policy $\pi$  

Observed episodes (training)

Assume: $\gamma = 1$
Example: Direct evaluation

Input policy $\pi$

Observed episodes (training)

Episode 1

B, east, C, $-1$
C, east, D, $-1$
D, exit, $x$, $+10$

Assume: $\gamma = 1$
Example: Direct evaluation

**Input policy $\pi$**

- A
- B
- C
- D
- E

**Observed episodes (training)**

**Episode 1**
- B, east, C, −1
- C, east, D, −1
- D, exit, $x$, +10

**Episode 2**
- B, east, C, −1
- C, east, D, −1
- D, exit, $x$, +10

Assume: $\gamma = 1$
Example: Direct evaluation

**Input policy** \( \pi \)

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Assume: \( \gamma = 1 \)

**Observed episodes (training)**

**Episode 1**
- B, east, C, -1
- C, east, D, -1
- D, exit, x, +10

**Episode 2**
- B, east, C, -1
- C, east, D, -1
- D, exit, x, +10

**Episode 3**
- E, north, C, -1
- C, east, D, -1
- D, exit, x, +10
Example: Direct evaluation

Input policy $\pi$

Assume: $\gamma = 1$

Observed episodes (training)

Episode 1
- B, east, C, −1
- C, east, D, −1
- D, exit, $x$, +10

Episode 2
- B, east, C, −1
- C, east, D, −1
- D, exit, $x$, +10

Episode 3
- E, north, C, −1
- C, east, D, −1
- D, exit, $x$, +10

Episode 4
- E, north, C, −1
- C, east, A, −1
- A, exit, $x$, −10
Example: Direct evaluation

Input policy $\pi$

Assume: $\gamma = 1$

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Observed episodes (training)

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Episode 1</th>
<th>Episode 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>B, east, C, −1</td>
<td>B, east, C, −1</td>
</tr>
<tr>
<td>C, east, D, −1</td>
<td>C, east, D, −1</td>
</tr>
<tr>
<td>D, exit, x, +10</td>
<td>D, exit, x, +10</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Episode 3</th>
<th>Episode 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>E, north, C, −1</td>
<td>E, north, C, −1</td>
</tr>
<tr>
<td>C, east, D, −1</td>
<td>C, east, A, −1</td>
</tr>
<tr>
<td>D, exit, x, +10</td>
<td>A, exit, x, −10</td>
</tr>
</tbody>
</table>

Output values

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example: Direct evaluation

Assume: $\gamma = 1$

Input policy $\pi$

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A</td>
<td></td>
<td>D</td>
</tr>
<tr>
<td>B</td>
<td>C</td>
<td></td>
<td>E</td>
</tr>
</tbody>
</table>

Output values

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>+8</td>
<td>C</td>
<td>D</td>
</tr>
</tbody>
</table>

Observed episodes (training)

**Episode 1**
- B, east, C, $-1$
- C, east, D, $-1$
- D, exit, x, +10

**Episode 2**
- B, east, C, $-1$
- C, east, D, $-1$
- D, exit, x, +10

**Episode 3**
- E, north, C, $-1$
- C, east, D, $-1$
- D, exit, x, +10

**Episode 4**
- E, north, C, $-1$
- C, east, A, $-1$
- A, exit, x, $-10$
Example: Direct evaluation

Input policy \( \pi \)

```
\begin{bmatrix}
  & A & \\
  B & C & D \\
  & E \\
\end{bmatrix}
```

Assume: \( \gamma = 1 \)

Observed episodes (training)

<table>
<thead>
<tr>
<th>Episode 1</th>
<th>Episode 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>B, east, C, -1</td>
<td>B, east, C, -1</td>
</tr>
<tr>
<td>C, east, D, -1</td>
<td>C, east, D, -1</td>
</tr>
<tr>
<td>D, exit, x, +10</td>
<td>D, exit, x, +10</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Episode 3</th>
<th>Episode 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>E, north, C, -1</td>
<td>E, north, C, -1</td>
</tr>
<tr>
<td>C, east, D, -1</td>
<td>C, east, A, -1</td>
</tr>
<tr>
<td>D, exit, x, +10</td>
<td>A, exit, x, -10</td>
</tr>
</tbody>
</table>

Output values

```
\begin{bmatrix}
  & A & \\
  B & +8 & C \\
  & D & \\
  & E & -2 \\
\end{bmatrix}
```
**Example: Direct evaluation**

**Input policy** $\pi$

Assume: $\gamma = 1$

**Observed episodes (training)**

**Episode 1**

- B, east, C, $-1$
- C, east, D, $-1$
- D, exit, x, +10

**Episode 2**

- B, east, C, $-1$
- C, east, D, $-1$
- D, exit, x, +10

**Episode 3**

- E, north, C, $-1$
- C, east, D, $-1$
- D, exit, x, +10

**Episode 4**

- E, north, C, $-1$
- C, east, A, $-1$
- A, exit, x, $-10$

**Output values**
Example: Direct evaluation

**Input policy** $\pi$

**Observed episodes (training)**

- **Episode 1**
  - B, east, C, $-1$
  - C, east, D, $-1$
  - D, exit, x, $+10$

- **Episode 2**
  - B, east, C, $-1$
  - C, east, D, $-1$
  - D, exit, x, $+10$

- **Episode 3**
  - E, north, C, $-1$
  - C, east, D, $-1$
  - D, exit, x, $+10$

- **Episode 4**
  - E, north, C, $-1$
  - C, east, A, $-1$
  - A, exit, x, $-10$

Assume: $\gamma = 1$
Example: Direct evaluation

Input policy $\pi$

<table>
<thead>
<tr>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>E</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Assume: $\gamma = 1$

Observed episodes (training)

**Episode 1**
- B, east, C, −1
- C, east, D, −1
- D, exit, x, +10

**Episode 2**
- B, east, C, −1
- C, east, D, −1
- D, exit, x, +10

**Episode 3**
- E, north, C, −1
- C, east, D, −1
- D, exit, x, +10

**Episode 4**
- E, north, C, −1
- C, east, A, −1
- A, exit, x, −10

Output values

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>−10</td>
<td>+8</td>
<td>+4</td>
<td>+10</td>
</tr>
<tr>
<td>+8</td>
<td>+10</td>
<td>−2</td>
<td>−10</td>
</tr>
</tbody>
</table>
Problems with direct evaluation

What’s good about direct evaluation?

It’s easy to understand.

It doesn’t require any knowledge of $T$, $R$.

It eventually computes the correct average values, using just sample transitions.
Problems with direct evaluation

What’s good about direct evaluation?

- It's easy to understand.
- It doesn't require any knowledge of $T$, $R$.
- It eventually computes the correct average values, using just sample transitions.

What bad about it?

- It wastes information about state connections.
- Each state must be learned separately.
- So, it takes a long time to learn.
Problems with direct evaluation

What’s good about direct evaluation?

It’s easy to understand.
It doesn’t require any knowledge of $T, R$.
It eventually computes the correct average values, using just sample transitions.

What bad about it?

It wastes information about state connections.
Each state must be learned separately.
So, it takes a long time to learn.

If B and E both go to C under this policy, how can their values be different?
Acknowledgments

The lecture incorporates material from:

Pieter Abbeel, Dan Klein, et al., University of California, Berkeley; ai.berkeley.edu
Ketrina Yim (illustrations)