CS260R Discussion
Jan 23, 2026

Info

No participation will be recorded for any discussion.

You can go whatever discussion session.

Discussion 1A Friday 1200 - 1350: Matthew Leng (matthewleng@cs.ucla.edu)

Discussion 1B Friday 1400 - 1550: Haoyuan Cai (haoyuan@cs.ucla.edu)

Agenda Today

1. Value iteration
2. Comparing value iteration & policy iteration
3. Q&A

Value Iteration

```
# Training
```

Initialize the values

For training iteration n = 0, 1, . . ., N (while the values are changing):

  For all states s:

    Compute new value V(s):

    $V_{n+1}(s) \leftarrow \max_{a \in \mathcal{A}} \mathbb{E}_{s' \sim P(s,a)}[r(s,a) + \gamma V_n(s')]$

```
# Control
```

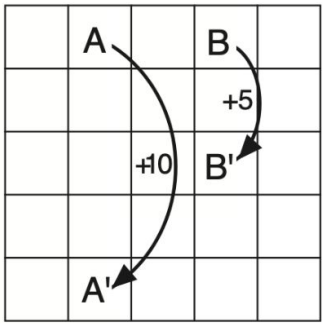
For step t = 1, ..., H:

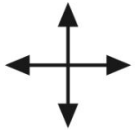
  $a_t \leftarrow \arg\max_{a \in \mathcal{A}} \mathbb{E}_{s' \sim \mathbb{P}(\cdot|s,a)}[r(s,a) + \gamma V_N(s')]$

  s_{t+1} **= env.step(**a_t**)**

Value Iteration

Example 3.5: Gridworld Figure 3.2 (left) shows a rectangular gridworld representation of a simple finite MDP. The cells of the grid correspond to the states of the environment. At each cell, four actions are possible: `north`, `south`, `east`, and `west`, which deterministically cause the agent to move one cell in the respective direction on the grid. Actions that would take the agent off the grid leave its location unchanged, but also result in a reward of -1. Other actions result in a reward of 0, except those that move the agent out of the special states A and B. From state A, all four actions yield a reward of $+10$ and take the agent to A'. From state B, all actions yield a reward of $+5$ and take the agent to B'.





3.3	8.8	4.4	5.3	1.5
1.5	3.0	2.3	1.9	0.5
0.1	0.7	0.7	0.4	-0.4
-1.0	-0.4	-0.4	-0.6	-1.2
-1.9	-1.3	-1.2	-1.4	-2.0

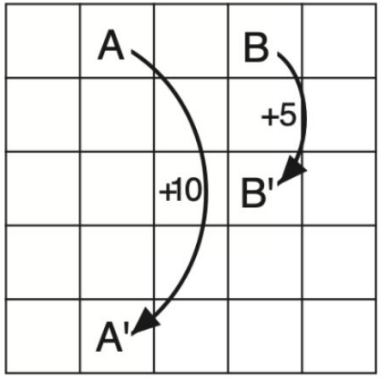
Figure 3.2: Gridworld example: exceptional reward dynamics (left) and state-value function for the equiprobable random policy (right).

Value Iteration

Initialize all values to 0

MDP Summary:

- Four actions: Up, Down, Left, Right
- Initial Policy is uniformly random
- R(A, A') = +10
- R(B, B') = +5
- Action takes off-grid is invalid, R(S, S) = -1



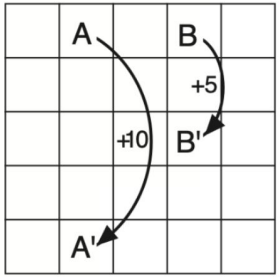
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

Value Iteration: 1st iteration

$$V_{n+1}(s) \leftarrow \max_{a \in \mathcal{A}} \mathbb{E}_{s' \sim P(s,a)}[r(s,a) + \gamma V_n(s')]$$

MDP Summary:

- Four actions: Up, Down, Left, Right
- Assume uniform random action
- R(A, A') = +10
- R(B, B') = +5
- Action takes off-grid is invalid, R(S, S) = -1
- Gamma = 1



0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

Table at iteration=0

0	+10	0	+5	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

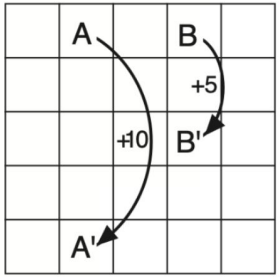
Table at iteration=1

Value Iteration: 2nd iteration

$$V_{n+1}(s) \leftarrow \max_{a \in \mathcal{A}} \mathbb{E}_{s' \sim P(s,a)}[r(s,a) + \gamma V_n(s')]$$

MDP Summary:

- Four actions: Up, Down, Left, Right
- Assume uniform random action
- R(A, A') = +10
- R(B, B') = +5
- Action takes off-grid is invalid, R(S, S) = -1
- Gamma = 1



0	+10	0	+5	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

Table at iteration=1

+10				

Table at iteration=2

For Upper Left Cell:

Up: -1

Down: 0

Left: -1

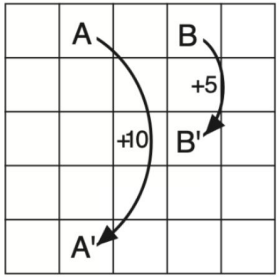
Right: +10

Value Iteration: 2nd iteration

$$V_{n+1}(s) \leftarrow \max_{a \in \mathcal{A}} \mathbb{E}_{s' \sim P(s,a)}[r(s,a) + \gamma V_n(s')]$$

MDP Summary:

- Four actions: Up, Down, Left, Right
- Assume uniform random action
- R(A, A') = +10
- R(B, B') = +5
- Action takes off-grid is invalid, R(S, S) = -1
- Gamma = 1



0	+10	0	+5	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

Table at iteration=1

+10	+10	+10	+5	+5
0	+10	0	+5	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

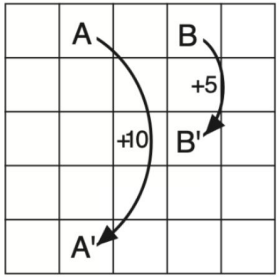
Table at iteration=2

Value Iteration: 3rd iteration

$$V_{n+1}(s) \leftarrow \max_{a \in \mathcal{A}} \mathbb{E}_{s' \sim P(s,a)}[r(s,a) + \gamma V_n(s')]$$

MDP Summary:

- Four actions: Up, Down, Left, Right
- Assume uniform random action
- R(A, A') = +10
- R(B, B') = +5
- Action takes off-grid is invalid, R(S, S) = -1
- Gamma = 1



+10	+10	+10	+5	+5
0	+10	0	+5	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

Table at iteration=2

+10				

Table at iteration=3

For the highlighted cell:

Up: +10

Down: 0

Left: -1

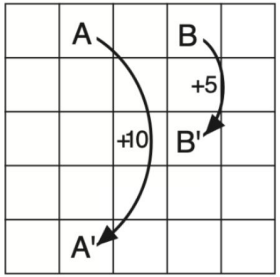
Right: +10

Value Iteration: 3rd iteration

$$V_{n+1}(s) \leftarrow \max_{a \in \mathcal{A}} \mathbb{E}_{s' \sim P(s,a)}[r(s,a) + \gamma V_n(s')]$$

MDP Summary:

- Four actions: Up, Down, Left, Right
- Assume uniform random action
- R(A, A') = +10
- R(B, B') = +5
- Action takes off-grid is invalid, R(S, S) = -1
- Gamma = 1



+10	+10	+10	+5	+5
0	+10	0	+5	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

Table at iteration=2

+10	+10	+10	+5	+5
+10	+10	+10	+5	+5
0	+10	0	+5	0
0	0	0	0	0
0	0	0	0	0

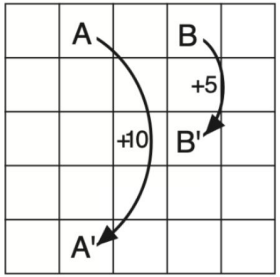
Table at iteration=3

Value Iteration: 4th iteration

$$V_{n+1}(s) \leftarrow \max_{a \in \mathcal{A}} \mathbb{E}_{s' \sim P(s,a)}[r(s,a) + \gamma V_n(s')]$$

MDP Summary:

- Four actions: Up, Down, Left, Right
- Assume uniform random action
- R(A, A') = +10
- R(B, B') = +5
- Action takes off-grid is invalid, R(S, S) = -1
- Gamma = 1



+10	+10	+10	+5	+5
+10	+10	+10	+5	+5
0	+10	0	+5	0
0	0	0	0	0
0	0	0	0	0

Table at iteration=3

+10	+10	+10	???	**+5**
+10	+10	+10	???	+5
+10	+10	+10	+5	+5
0	+10	0	+5	0
0	0	0	0	0

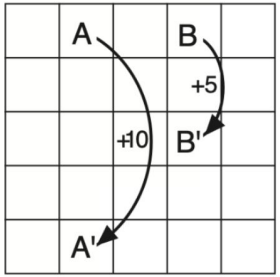
Table at iteration=4

Value Iteration: 4th iteration

$$V_{n+1}(s) \leftarrow \max_{a \in \mathcal{A}} \mathbb{E}_{s' \sim P(s,a)}[r(s,a) + \gamma V_n(s')]$$

MDP Summary:

- Four actions: Up, Down, Left, Right
- Assume uniform random action
- R(A, A') = +10
- R(B, B') = +5
- Action takes off-grid is invalid, R(S, S) = -1
- Gamma = 1



+10	+10	+10	+5	+5
+10	+10	+10	+5	+5
0	+10	0	+5	0
0	0	0	0	0
0	0	0	0	0

Table at iteration=3

+10	+10	+10	+10	+5
+10	+10	+10	+10	+5
+10	+10	+10	+5	+5
0	+10	0	+5	0
0	0	0	0	0

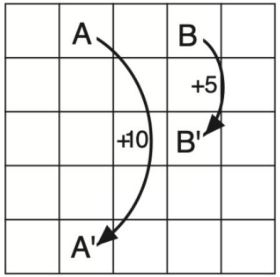
Table at iteration=4

Value Iteration: 5th iteration

$$V_{n+1}(s) \leftarrow \max_{a \in \mathcal{A}} \mathbb{E}_{s' \sim P(s,a)}[r(s,a) + \gamma V_n(s')]$$

MDP Summary:

- Four actions: Up, Down, Left, Right
- Assume uniform random action
- R(A, A') = +10
- R(B, B') = +5
- Action takes off-grid is invalid, R(S, S) = -1
- Gamma = 1



+10	+10	+10	+10	+5
+10	+10	+10	+10	+5
+10	+10	+10	+5	+5
0	+10	0	+5	0
0	0	0	0	0

Table at iteration=4

+10	+10	+10	???	+10
+10	+10	+10	???	+10
+10	+10	+10	???	+5
+10	+10	+10	+5	+5
0	+10	0	+5	0

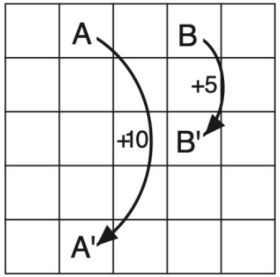
Table at iteration=5

Value Iteration: 5th iteration

$$V_{n+1}(s) \leftarrow \max_{a \in \mathcal{A}} \mathbb{E}_{s' \sim P(s,a)}[r(s,a) + \gamma V_n(s')]$$

MDP Summary:

- Four actions: Up, Down, Left, Right
- Assume uniform random action
- R(A, A') = +10
- R(B, B') = +5
- Action takes off-grid is invalid, R(S, S) = -1
- Gamma = 1



+10	+10	+10	+10	+5
+10	+10	+10	+10	+5
+10	+10	+10	+5	+5
0	+10	0	+5	0
0	0	0	0	0

Table at iteration=4

+10	+10	+10	+10	+10
+10	+10	+10	+10	+10
+10	+10	+10	+10	+5
+10	+10	+10	+5	+5
0	+10	0	+5	0

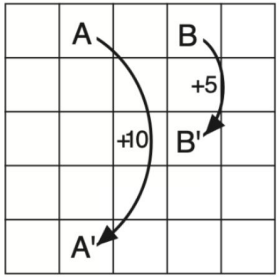
Table at iteration=5

Value Iteration: 6th iteration

$$V_{n+1}(s) \leftarrow \max_{a \in \mathcal{A}} \mathbb{E}_{s' \sim P(s,a)}[r(s,a) + \gamma V_n(s')]$$

MDP Summary:

- Four actions: Up, Down, Left, Right
- Assume uniform random action
- R(A, A') = +10
- R(B, B') = +5
- Action takes off-grid is invalid, R(S, S) = -1
- Gamma = 1



+10	+10	+10	+10	+10
+10	+10	+10	+10	+10
+10	+10	+10	+10	+5
+10	+10	+10	+5	+5
0	+10	0	+5	0

Table at iteration=5

+10	???	+10	???	+10
+10	+10	+10	+10	+10
+10	+10	+10	+10	+10
+10	+10	+10	+10	+5
+10	+10	+10	+5	+5

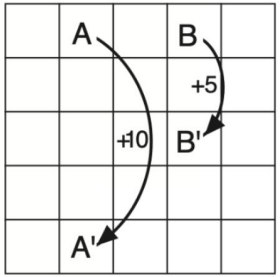
Table at iteration=6

Value Iteration: 6th iteration

$$V_{n+1}(s) \leftarrow \max_{a \in \mathcal{A}} \mathbb{E}_{s' \sim P(s,a)}[r(s,a) + \gamma V_n(s')]$$

MDP Summary:

- Four actions: Up, Down, Left, Right
- Assume uniform random action
- R(A, A') = +10
- R(B, B') = +5
- Action takes off-grid is invalid, R(S, S) = -1
- Gamma = 1



+10	+10	+10	+10	+5
+10	+10	+10	+10	+5
+10	+10	+10	+10	+5
+10	+10	+10	+5	+5
0	+10	0	+5	0

Table at iteration=5

+10	+20	+10	???	+10
+10	+10	+10	+10	+10
+10	+10	+10	+10	+10
+10	+10	+10	+10	+5
+10	+10	+10	+5	+5

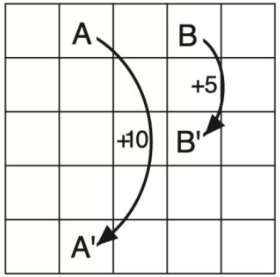
Table at iteration=6

Value Iteration: 6th iteration

$$V_{n+1}(s) \leftarrow \max_{a \in \mathcal{A}} \mathbb{E}_{s' \sim P(s,a)}[r(s,a) + \gamma V_n(s')]$$

MDP Summary:

- Four actions: Up, Down, Left, Right
- Assume uniform random action
- R(A, A') = +10
- R(B, B') = +5
- Action takes off-grid is invalid, R(S, S) = -1
- Gamma = 1



+10	+10	+10	+10	+5
+10	+10	+10	+10	+5
+10	+10	+10	+10	+5
+10	+10	+10	+5	+5
0	+10	0	+5	0

Table at iteration=5

+10	+20	+10	+15	+10
+10	+10	+10	+10	+10
+10	+10	+10	+10	+10
+10	+10	+10	+10	+5
+10	+10	+10	+5	+5

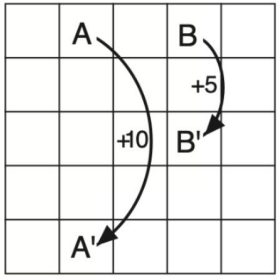
Table at iteration=6

Value Iteration: 7th iteration

$$V_{n+1}(s) \leftarrow \max_{a \in \mathcal{A}} \mathbb{E}_{s' \sim P(s,a)}[r(s,a) + \gamma V_n(s')]$$

MDP Summary:

- Four actions: Up, Down, Left, Right
- Assume uniform random action
- R(A, A') = +10
- R(B, B') = +5
- Action takes off-grid is invalid, R(S, S) = -1
- Gamma = 1



+10	+20	+10	+15	+10
+10	+10	+10	+10	+10
+10	+10	+10	+10	+10
+10	+10	+10	+10	+5
+10	+10	+10	+5	+5

Table at iteration=6

+20	+20	+20	+15	+15
+10	+20	+10	+15	+10
+10	+10	+10	+10	+10
+10	+10	+10	+10	+10
+10	+10	+10	+10	+5

Table at iteration=7

Value explosion coming soon!

That's why $\gamma < 1.0$

Value Iteration

Assuming an argmax policy!

The values should be an values estimator w.r.t. the optimal policy, but at the intermediate iterations the values is biased!

(in policy iteration, after each policy eval loop the values is the unbiased value estimator w.r.t. current policy)

```
# Training
```

Initialize the values

For training iteration n = 0, 1, . . ., N (while the values are changing):

  For all states s:

    Compute new value V(s):

    $V_{n+1}(s) \leftarrow \max_{a \in \mathcal{A}} \mathbb{E}_{s' \sim P(s,a)}[r(s,a) + \gamma V_n(s')]$

```
# Control
```

For step t = 1, ..., H:

  $a_t \leftarrow \arg\max_{a \in \mathcal{A}} \mathbb{E}_{s' \sim \mathbb{P}(\cdot|s,a)}[r(s,a) + \gamma V_N(s')]$

  s_{t+1} **= env.step(**a_t**)**

Policy Iteration

```
# Training
Initialize policy π_0
For training iteration n=1,..,N
(while the policy is changing):
  # Policy Evaluation:
  For policy evaluation step k=1,.., K
  (while the values are changing):
    For all states s:
      Compute k step's new value V(s):
        V^{π_n}_{k+1}(s) ← E_{a∼π_n(·|s),s'∼P(·|s,a)}[r(s,a) + γV^{π_n}_k(s')]
    Update the value table
  # Policy Improvement
  For all states s:
    Update policy π_{n+1}(s) ← arg max_{a∈A} E_{s'∼P(·|s,a)}[r(s,a) + γV^{π_n}_K(s')]

# Control
For step t = 1, ..., H:
  a_t ← π_N(s_t)
  s_{t+1} = env.step(a_t)
```

Value Iteration

```
# Training
Initialize the values
For training iteration n = 0, 1, . . ., N
(while the values are changing):
  For all states s:
    Compute new value V(s):
      V_{n+1}(s) ← max_{a∈A} E_{s'∼P(s,a)}[r(s,a) + γV_n(s')]

# Control
For step t = 1, ..., H:
  a_t ← arg max_{a∈A} E_{s'∼P(·|s,a)}[r(s,a) + γV_N(s')]
  s_{t+1} = env.step(a_t)
```

Diff:

- The policy in Bellman Equation
- After each policy eval, the V is an unbiased estimator w.r.t. current policy. (in value iteration it's biased until converged)

Info

No participation will be recorded for any discussion.

You can go whatever discussion session.

Discussion 1A Friday 1200 - 1350: Matthew Leng (matthewleng@cs.ucla.edu)

Discussion 1B Friday 1400 - 1550: Haoyuan Cai (haoyuan@cs.ucla.edu)

Agenda Today

1. Value iteration
2. Comparing value iteration & policy iteration
3. Q&A