Passive Reinforcement Learning

Bert Huang
Introduction to Artificial Intelligence

Notation Review

- Recall the Bellman Equation:

$$U(s) = R(s) + \gamma \max_{a \in A(s)} \sum_{s'} P(s'|s,a)U(s')$$

alternate version

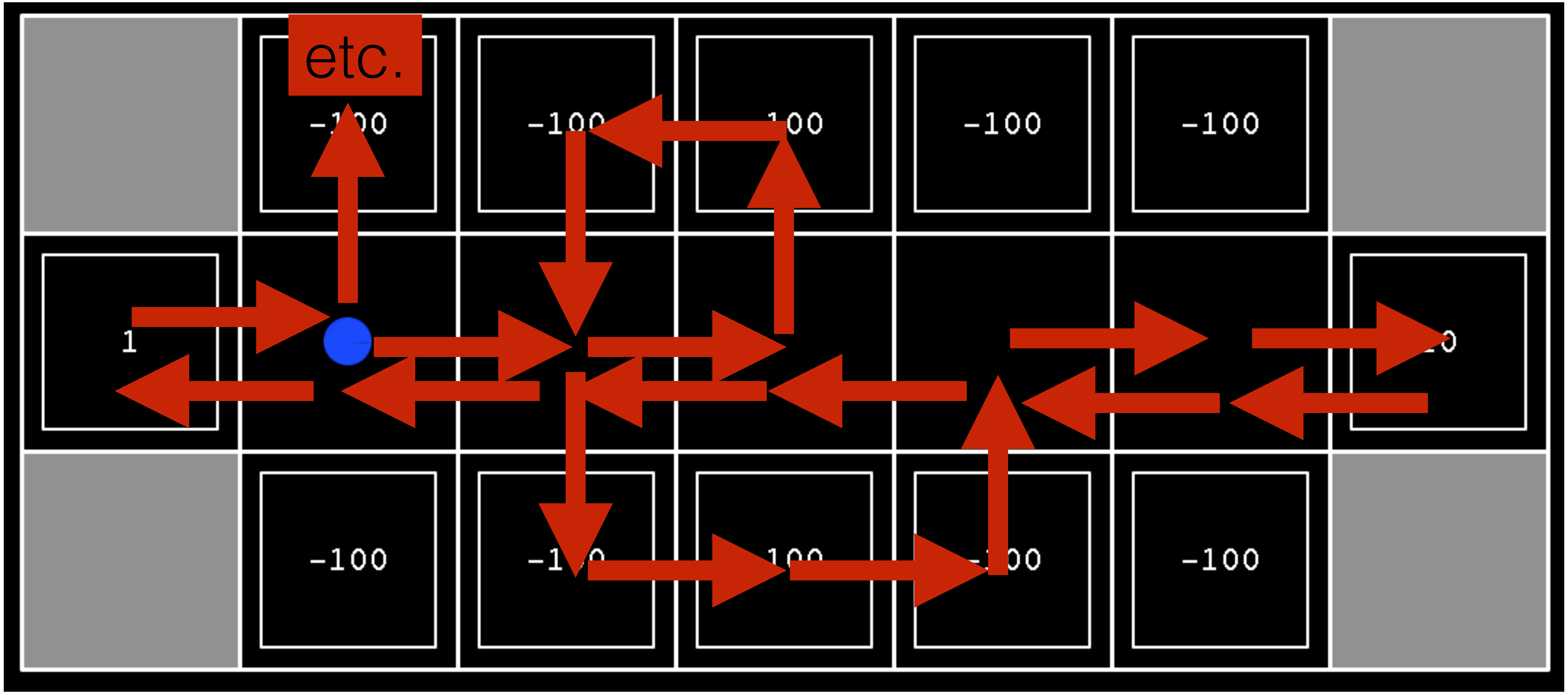
$$U(s) = \max_{a \in A(s)} R(s,a) + \gamma \sum_{s'} P(s'|s,a)U(s')$$

$$\pi^*(s) = \arg\max_{a \in A(s)} \sum_{s'} P(s'|s,a)U(s')$$

Value Iteration Drawbacks

- Computes utility for every state
- Needs exact transition model
- Needs to fully observe state
- Needs to know exact reward for each state

Slippery Bridge



	Value Iteration	Passive Learning	Active Learning
States and rewards	Observes all states and rewards in environment	Observes only states (and rewards) visited by agent	Observes only states (and rewards) visited by agent
Transitions	Observes all action-transition probabilities	Observes only transitions that occur from chosen actions	Observes only transitions that occur from chosen actions
Decisions	N/A	Learning algorithm does not choose actions	Learning algorithm chooses actions

Passive Learning

- Recordings of agent running fixed policy
- Observe states, rewards, actions
  - Direct utility estimation
  - Adaptive dynamic programming (ADP)
  - Temporal-difference (TD) learning

Direct Utility Estimation

$$U(s) = R(s) + \gamma \max_{a \in A(s)} \sum_{s'} P(s'|s, a) U(s')$$

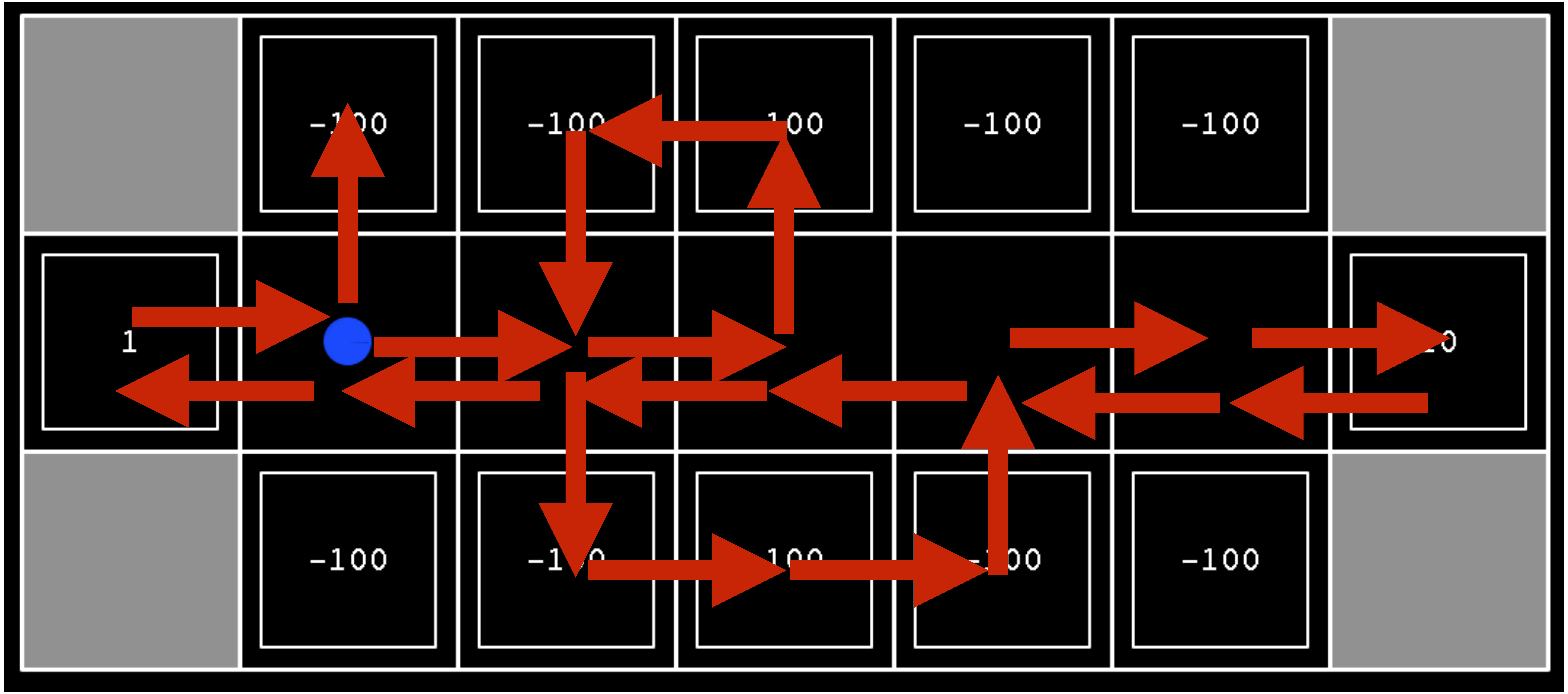
$$U^{\pi}(s) = R(s) + \gamma \sum_{s'} P(s'|s, \pi(s)) U^{\pi}(s')$$

future reward of state assuming we use this policy

Direct utility estimation: use observed rewards and future rewards to estimate U (i.e., take average of samples from data)

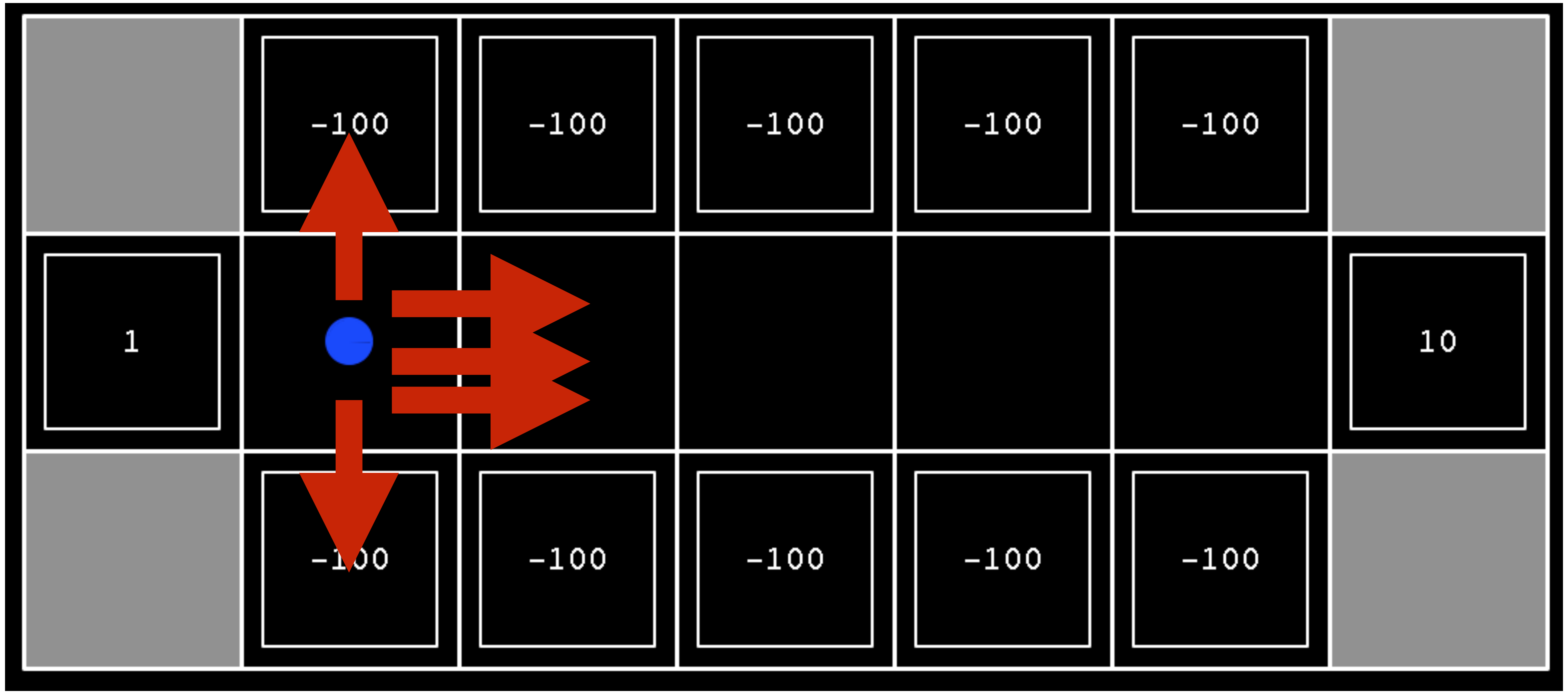
Adaptive Dynamic Programming

- Run value iteration using estimated rewards and transition probabilities



Adaptive Dynamic Programming

- Run value iteration using estimated rewards and transition probabilities



Action	Result
RIGHT	UP
RIGHT	RIGHT
RIGHT	RIGHT
RIGHT	DOWN
RIGHT	RIGHT

Adaptive Dynamic Programming

- Run value iteration using estimated rewards and transition probabilities

Estimate of $\quad$ Estimate of

$$U_{i+1}(s) \leftarrow R(s) + \gamma \max_{a \in A(s)} \sum_{s'} P(s'|s,a) U_i(s')$$

Temporal-Difference Learning

$$U^\pi(s) = R(s) + \gamma \sum_{s'} P(s'|s, \pi(s)) U^\pi(s')$$

$$U^\pi(s) = R(s) + \gamma \mathrm{E}_{s'}[U^\pi(s')]$$

$$U^\pi(s) = \mathrm{E}_{s'}[R(s) + \gamma U^\pi(s')]$$

learning rate parameter

current estimate of utility

$$U^\pi(s) \leftarrow U^\pi(s) + \alpha(R(s) + \gamma U^\pi(s') - U^\pi(s))$$

observed utility

Temporal-Difference Learning

$$U^\pi(s) \leftarrow U^\pi(s) + \alpha(\boxed{R(s) + \gamma U^\pi(s')} - \boxed{U^\pi(s)})$$

Run each time we transition from state **s** to **s'**

Converges slower than ADP, but much simpler update.

Leads to famous q-learning algorithm (next video)

Passive Learning

- Recordings of agent running fixed policy
- Observe states, rewards, actions
  - Direct utility estimation
  - Adaptive dynamic programming (ADP)
  - Temporal-difference (TD) learning