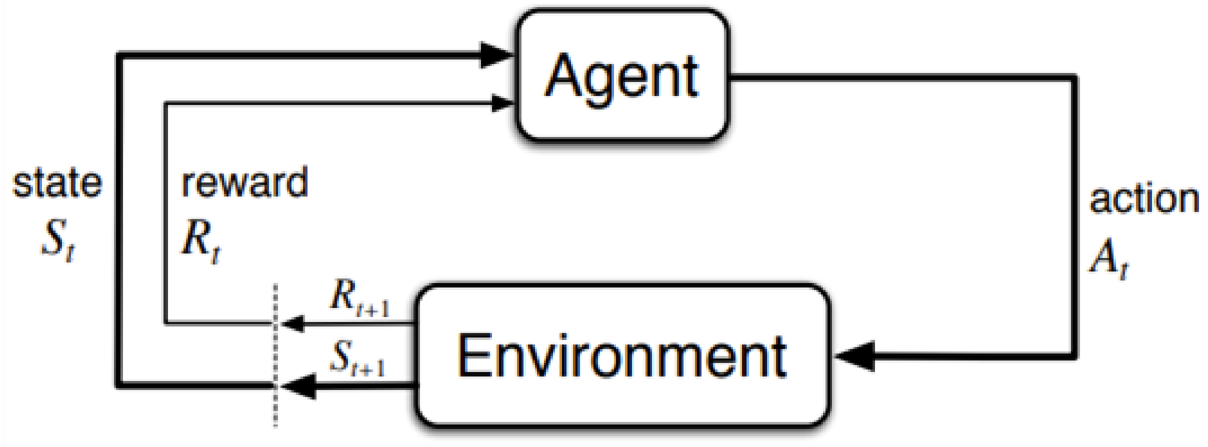
Fundamentals of Reinforcement Learning

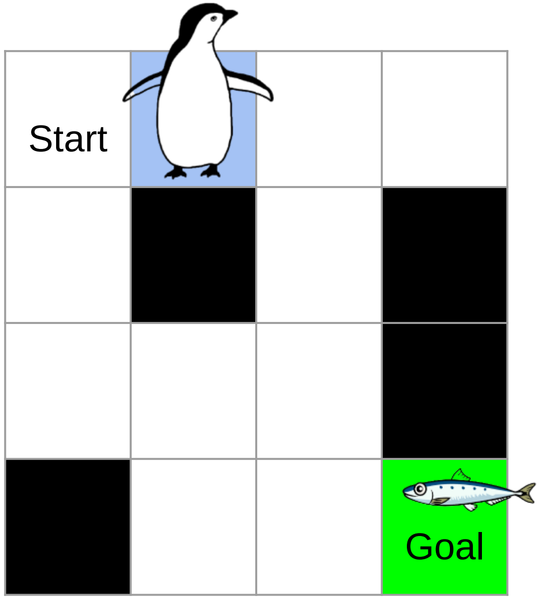
Presenter: Kyler Sood
Mentor: Dr. Loren Anderson

Reinforcement Learning Problem



- Interaction produces a *trajectory*: $S_0, A_0, R_1, S_1, A_1, R_2, \ldots$
- Maximize expected *return*: $G_t = R_{t+1} + R_{t+2} + \cdots + R_T$

Gridworld: Frozen Lake Environment



States: individual grid squares

Terminal States: hole squares, goal squares

Actions: up, down, left, right

Rewards:

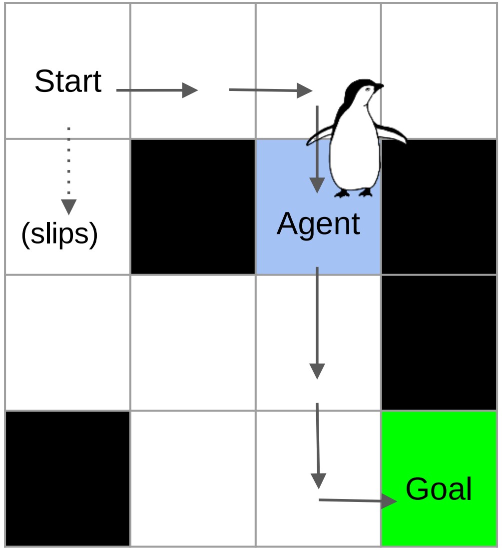
- -1 per action
- -100 for falling into a hole
- +100 for reaching goal state

Dynamics:

- 75% of the time, intended action
- Slips 25% of the time, random action

Goal: Reach goal state as quickly as possible

Value Functions and the Bellman Equation



Agent's *policy*: $\pi(a|s)$

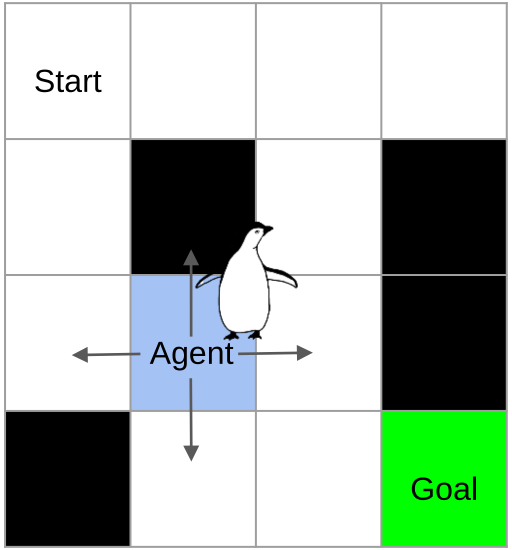
Value function of policy:

$$v_\pi(s) = \mathbb{E}_\pi[R_{t+1} + R_{t+2} + \cdots + R_T | S_t = s]$$

Bellman Equation

$$v_\pi(s) = \sum_a \pi(a|s) \sum_{s',r} p(s', r|s, a) \left[r + \gamma v_\pi(s')\right]$$

Dynamic Programming



- Goal: Learn an *optimal* policy.

$$v_*(s) \geq v_\pi(s) \text{ for all } s \in \mathcal{S}, \text{ policies } \pi$$

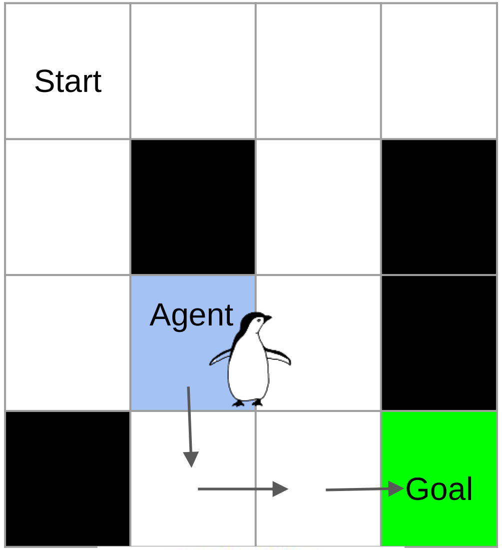
- *Policy evaluation*:

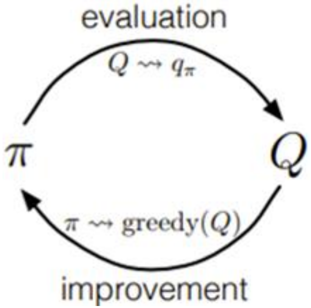
$$V(s) \leftarrow \sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a)[r + \gamma V(s')]$$

- *Policy improvement* (Bellman Eq.):

$$\pi(s) \leftarrow \arg\max_a \sum_{s',r} p(s',r|s,a)[r + \gamma V(s')]$$

Monte Carlo Methods





$$S_0, A_0, R_1, S_1, A_1, R_2, \ldots$$

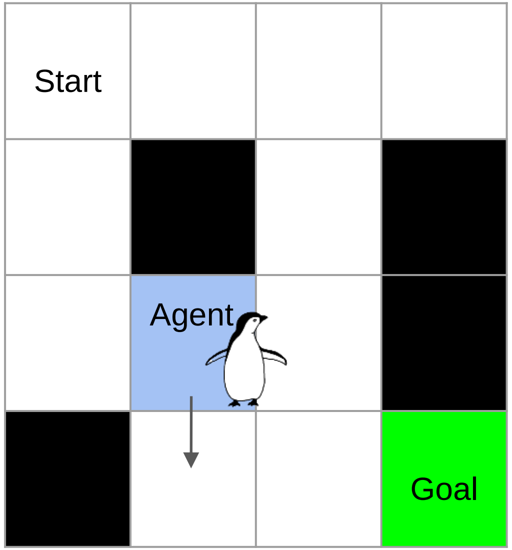
- Learn optimal policy through experience *only*
- MC Prediction: (*average* over many episodes)

$$G \leftarrow \gamma G + R_{t+1}$$

- MC Control (optimize greedy policy):

$$\pi(S_t) \leftarrow \underset{a}{\operatorname{argmax}}\ Q(S_t, a)$$

Temporal-Difference Learning



Iterative update for an MC method

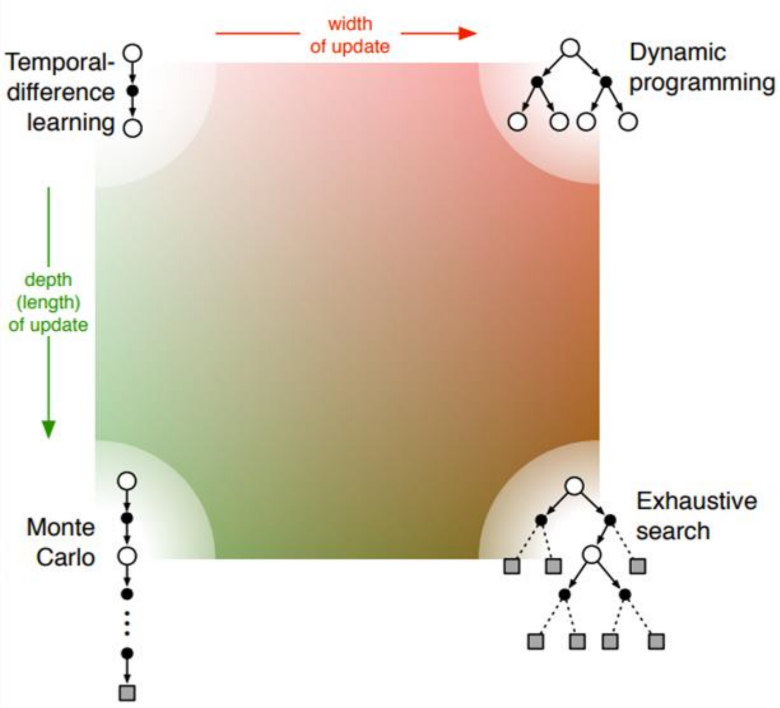
$$V(S_t) \leftarrow V(S_t) + \alpha[G_t - V(S_t)]$$

Iterative update for one-step TD method (policy evaluation)

$$V(S_t) \leftarrow V(S_t) + \alpha[R_{t+1} + \gamma V(S_{t+1}) - V(S_t)]$$

- Combining DP and MC techniques.
- Like MC, learning from experience

Comparing Tabular Solution Methods



- DP is *model-based*, learn from model
- MC and TD methods are *model-free*, learn from experience only.

Iterative MC Update

$$V(S_t) \leftarrow V(S_t) + \alpha[G_t - V(S_t)]$$

Iterative TD Update

$$V(S_t) \leftarrow V(S_t) + \alpha[R_{t+1} + \gamma V(S_{t+1}) - V(S_t)]$$

- TD typically converges faster than MC

References & Acknowledgments

- Reference: Reinforcement Learning (2018), 2nd ed. R. S. Sutton and A. G. Barto

- FrozenLake-v0 environment is from OpenAI Gym (SF startup)

- Really appreciate Loren for being a great mentor!

- For inquiries, my email is sood0027@umn.edu