Sarsa: On-Policy TD Control

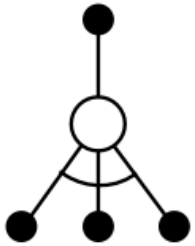
Turn this into a control method by always updating the policy to be greedy with respect to the current estimate:

```
Initialize Q(s, a) arbitrarily
Repeat (for each episode):
    Initialize s
    Choose a from s using policy derived from Q (e.g., ε-greedy)
    Repeat (for each step of episode):
        Take action a, observe r, s'
        Choose a' from s' using policy derived from Q (e.g., ε-greedy)
        Q(s, a) ← Q(s, a) + α[r + γQ(s', a') − Q(s, a)]
        s ← s'; a ← a';
    until s is terminal
```

Q-Learning: Off-Policy TD Control

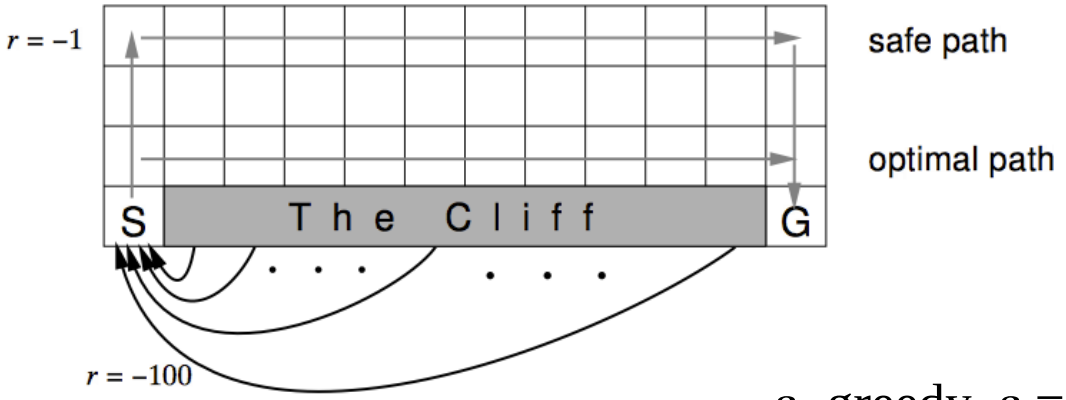
One-step Q-learning:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left[r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t) \right]$$

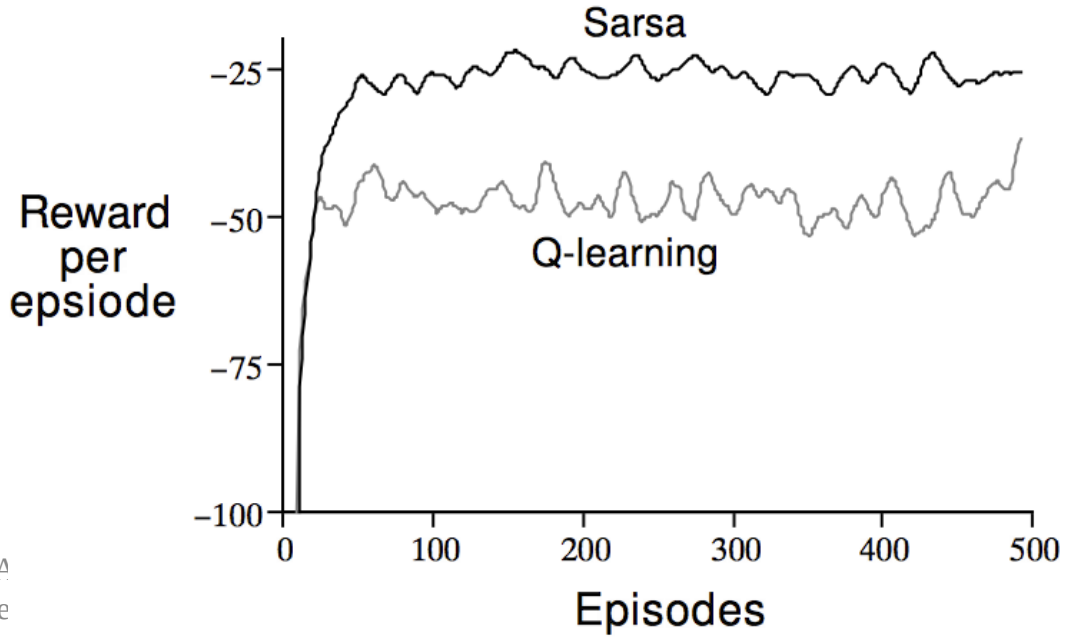


```
Initialize Q(s, a) arbitrarily
Repeat (for each episode):
    Initialize s
    Repeat (for each step of episode):
        Choose a from s using policy derived from Q (e.g., ε-greedy)
        Take action a, observe r, s'
        Q(s, a) ← Q(s, a) + α[r + γ max_a' Q(s', a') − Q(s, a)]
        s ← s';
    until s is terminal
```

Cliffwalking



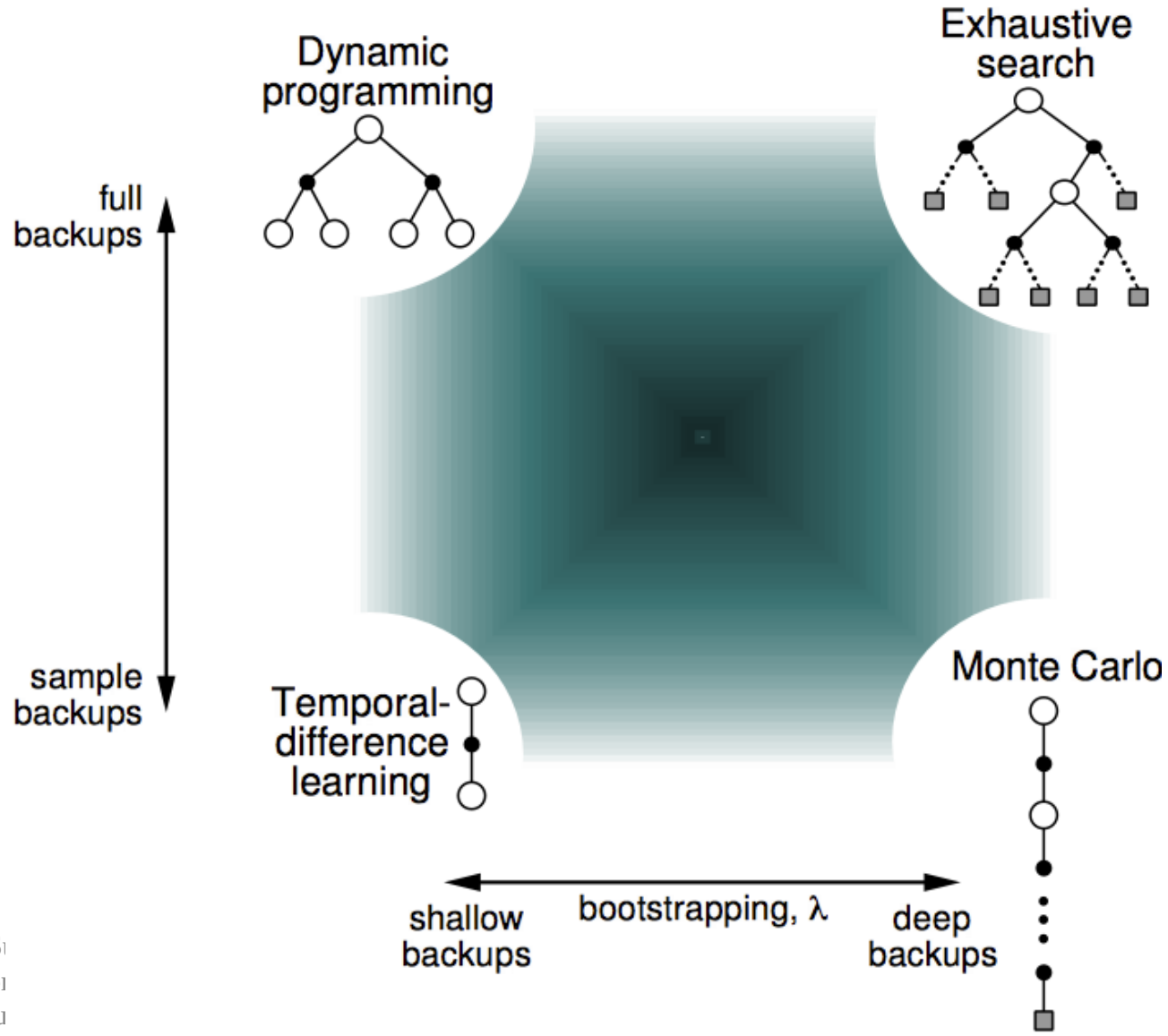
ε–greedy, ε = 0.1



The Book

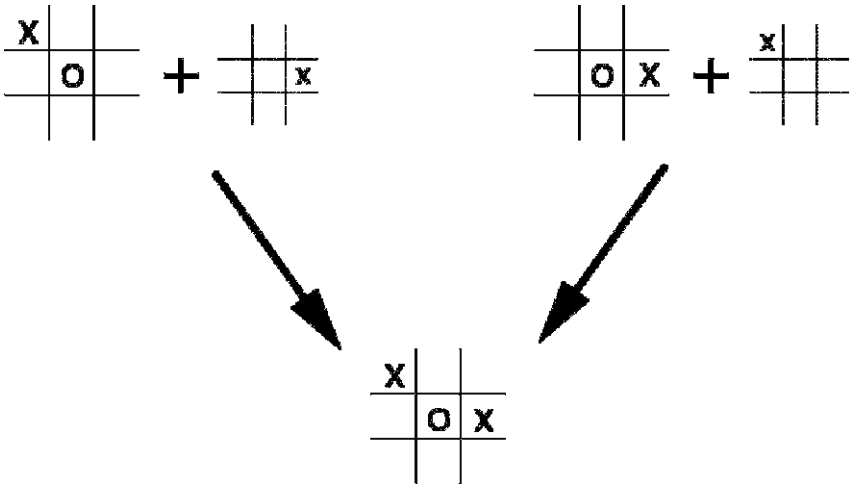
- Part I: The Problem
  - Introduction
  - Evaluative Feedback
  - The Reinforcement Learning Problem
- Part II: Elementary Solution Methods
  - Dynamic Programming
  - Monte Carlo Methods
  - Temporal Difference Learning
- Part III: A Unified View
  - Eligibility Traces
  - Generalization and Function Approximation
  - Planning and Learning
  - Dimensions of Reinforcement Learning
  - Case Studies

Unified View



Afterstates

- Usually, a state-value function evaluates states in which the agent can take an action.
- But sometimes it is useful to evaluate states after agent has acted, as in tic-tac-toe.
- Why is this useful?



Summary

- TD prediction
- Introduced *one-step tabular model-free TD methods*
- Extend prediction to control by employing some form of GPI
  - On-policy control: Sarsa
  - Off-policy control: Q-learning
- These methods bootstrap and sample, combining aspects of DP and MC methods

Questions – for Discussion

- What is common to all three classes of methods? – DP, MC, TD
- What are the principle strengths and weaknesses of each?
  - How could you use each to “solve” black jack?
  - What about Mario?
- In what sense is our RL view complete?
- In what senses is it incomplete?
  - What are the principal things missing?
- The broad applicability of these ideas…
- What does the term bootstrapping refer to?
- What is the relationship between DP and learning?