Reinforcement Learning: Model-Free RL

AI/ML Teaching

Goals & Keyword

- Model-free prediction
  - Monte-Carlo (MC)
  - Temporal Difference (TD)

- Model-free control
  - ϵ-greedy exploration
  - SARSA

  - On-policy & off-policy learning
  - Q-learning

Model-free prediction

Monte-Carlo RL

- MC methods learn directly from episodes of experience
- MC is model-free: **no knowledge of MDP transitions / rewards**
- MC learns from complete episodes: all episodes must terminate
- MC uses the simplest possible idea: value = mean return

- Goal: learn v_π from episodes of experience under policy π

$$S_1, A_1, R_2, \dots, S_k \sim \pi$$

- Return is the total discounted reward

$$G_t = R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{T-1} R_T$$

- Value function is the expected return

$$v_\pi(s) = \mathbb{E}[G_t | S_t = s]$$

Empirical mean update

- Mean can be varied (non-stationary)

$$\begin{aligned}\mu_k &= \frac{1}{k}\sum_{j=1}^{k} x_j = \frac{1}{k}\left(x_k + \sum_{j=1}^{k-1} x_j\right) \\ &= \frac{1}{k}\left(x_k + (k-1)\mu_{k-1}\right) \\ &= \mu_{k-1} + \frac{1}{k}\left(x_k - \mu_{k-1}\right)\end{aligned}$$

- Update $V(s)$ incrementally after episode

$$N(S_t) \leftarrow N(S_t) + 1$$
$$V(S_t) \leftarrow V(S_t) + \frac{1}{N(S_t)}\left(G_t - V(S_t)\right)$$

$$V(S_t) \leftarrow V(S_t) + \alpha\left(G_t - V(S_t)\right)$$

Temporal-Difference Learning

- TD methods learn directly from episodes of experience
- TD is model-free: **no knowledge of MDP transitions / rewards**
- TD learns from incomplete episodes
- TD updates a guess towards a guess

- Simplest temporal-difference learning algorithm: TD(0)
  - Update value $V(S_t)$ toward estimated return $R_{t+1} + \gamma V(S_t + 1)$

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

*Monte-Carlo RL

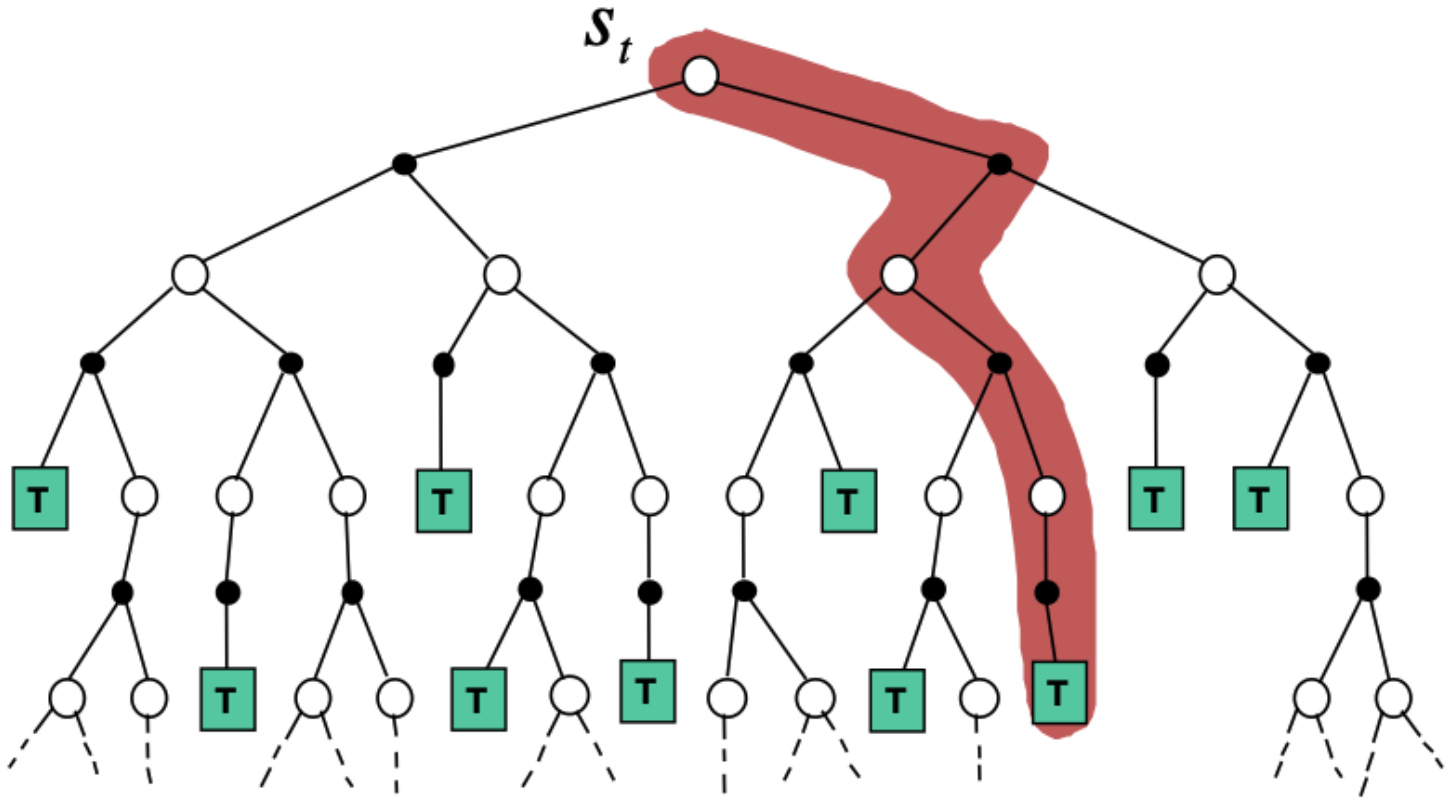
$$V(S_t) \leftarrow V(S_t) + \alpha\big(G_t - V(S_t)\big)$$

MC vs. TD

MC	TD
must wait until end of episode	can learn before knowing the final outcome
can only learn from complete sequences	can learn from incomplete sequences
only works for episodic (terminating) environments	works in continuing (non-terminating) environments
unbiased estimate of $v_\pi(S_t)$	biased estimate of $v_\pi(S_t)$
Higher variance (depending on random actions, transitions, rewards)	Lower variance than the return: depends on one random action, transition, reward
Good convergence properties (even with function approximation)	Usually more efficient than MC TD(0) converges to $v_\pi(s)$ (but not always with function approximation)

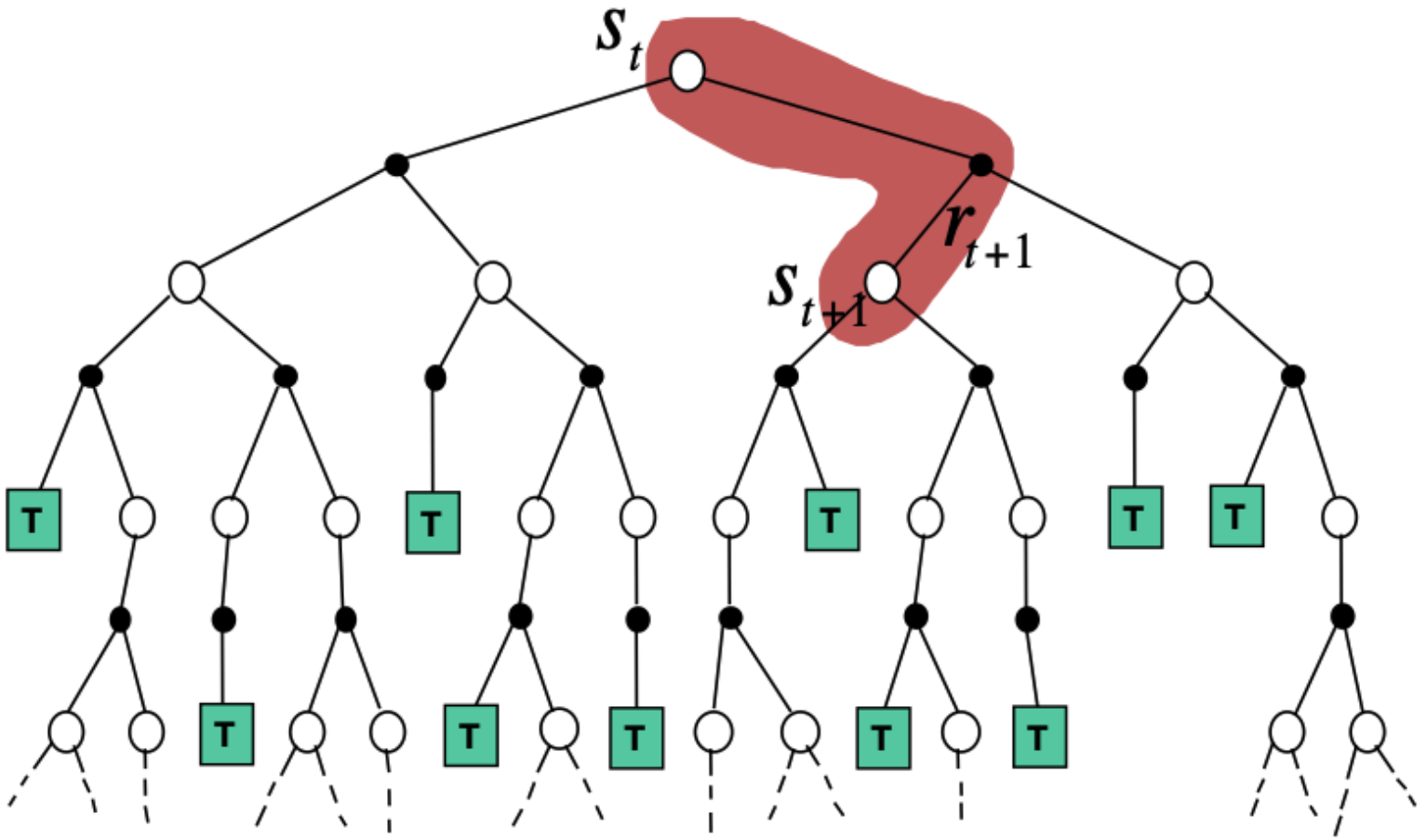
Monte-Carlo Backup

$$V(S_t) \leftarrow V(S_t) + \alpha\big(G_t - V(S_t)\big)$$



Temporal-Difference Backup

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



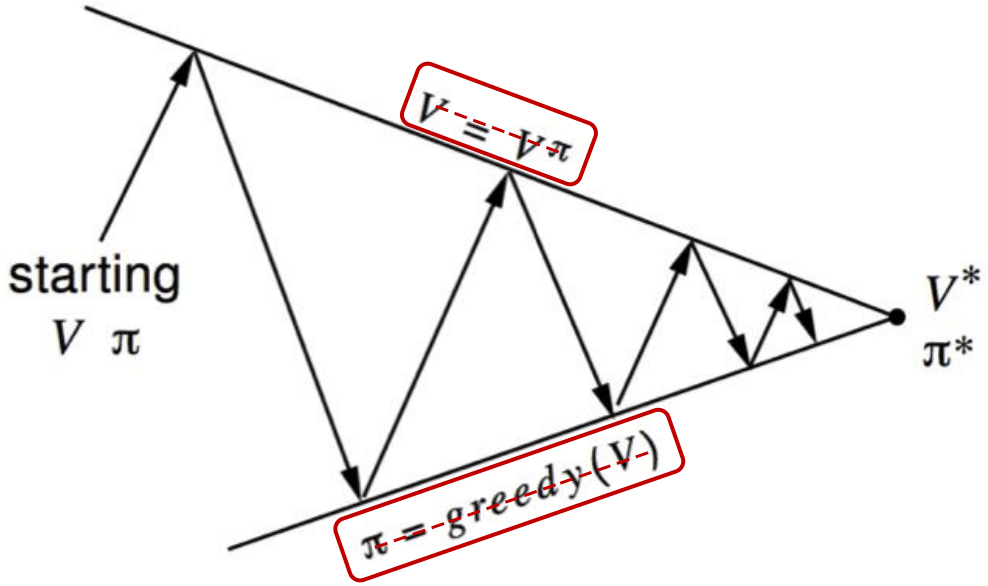
Model-free control

Generalized Policy Iteration

GPI with MC Evaluation

- Policy evaluation, $V = V^{\pi}$
- Policy improvement

- MC method
  - Policy evaluation by $Q = q_{\pi}$



Greedy policy improvement over $V(s)$ requires model of MDP, doesn't know R and s'

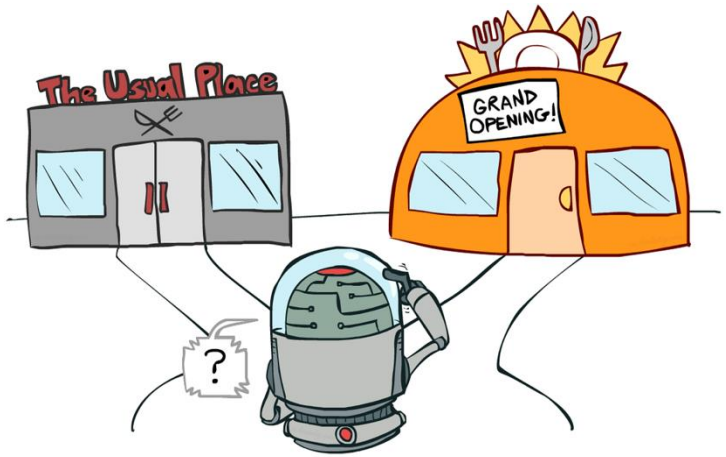
$$\pi'(s) = \arg\max_{a \in \mathcal{A}} R_s^a + \mathcal{P}_{ss'}^a V(s')$$

Instead, greedy policy improvement over $Q(s,a)$ is model-free

$$\pi'(s) = \arg\max_{a \in \mathcal{A}} Q(s,a)$$

GPI with MC Evaluation

- Policy evaluation, ~~$V = V^{\pi}$~~
- Policy improvement, ~~greedy improvement~~



- MC method
  - ϵ-greedy exploration

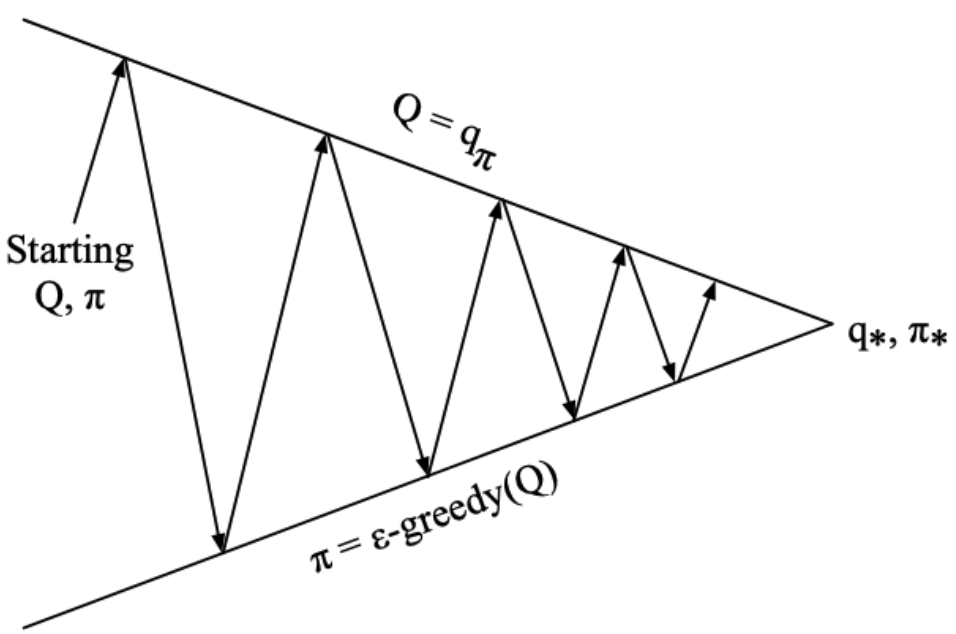
Greedy policy improvement → stuck at local optimum policy

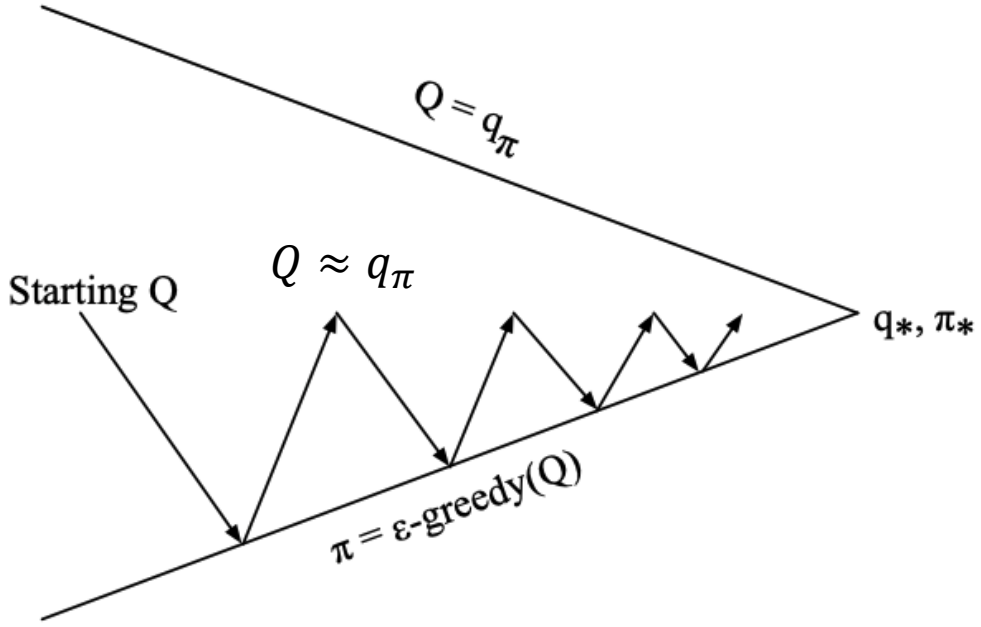
In model-free setting, we can't see all the possible states

https://medium.com/operations-research-bit/optimizing-decision-making-in-ai-fine-tuning-the-epsilon-greedy-algorithm-for-enhanced-performance-ea61e86d6f1d

GPI with MC Evaluation

- Policy evaluation: Monte-Carlo policy evaluation ($Q = q_\pi$)
- Policy improvement: ϵ-greedy policy improvement





Intermediate Summary & TD

- Model-based RL: value/policy iteration
  - Iterative policy evaluation & greedy policy update

- Model-free RL
  - MC: action value evaluation & ϵ-greedy policy improvement

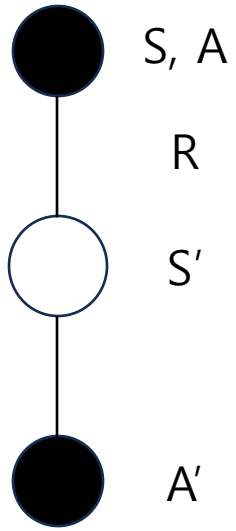
---

  - Temporal-difference (TD) learning
    - Lower variance, online, incomplete sequences
    - Apply TD to model-free RL with
      - $Q(S, A)$
      - Use ϵ-greedy policy improvement
      - Update every time-step

SARSA

$$Q(S,A) \leftarrow Q(S,A) + \alpha\big(R + \gamma Q(S',A') - Q(S,A)\big)$$

- Every time-step
  - Policy evaluation: SARSA, $Q \approx q_\pi$
  - Policy improvement: ϵ-greedy policy improvement



SARSA pseudo code

- Initialize $Q(s,a), \forall s \in \mathcal{S}, a \in \mathcal{A}(s)$ and $Q(terminal\ state,\cdot) = 0$
- 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)
    - $Q(S,A) \leftarrow Q(S,A) + \alpha[R + \gamma Q(S',A') - Q(S,A)]$
    - $S \leftarrow S'$; $A \leftarrow A'$;
  - Until S is terminal

On and Off-Policy Learning

- On-policy learning
  - Learn on the job
  - Learn about policy π from experience sampled from π

- Off-policy learning
  - Look over someone's shoulder
  - Learn about policy π from experience sampled from μ

  → Learn from observing humans or other agents

  → Re-use experience generated from old policies $\pi_1, \pi_2, \ldots, \pi_{t-1}$

  → Learn about optimal policy while following exploratory policy

Off-Policy TD

- Evaluate target policy $\pi(a|s)$ to compute $v_\pi(s)$ or $q_\pi(s,a)$
- Behavior policy $\mu(a|s)$

$$\{S_1, A_1, R_2, \dots, S_T\} \sim \mu$$

- Importance sampling
  - $\mathbb{E}_{X\sim P}[f(X)] = \sum P(X) f(X) = \sum Q(X) \frac{P(X)}{Q(X)} f(X) = \mathbb{E}_{X\sim Q}\left[\frac{P(X)}{Q(X)} f(X)\right]$
- Importance sampling for Off-Policy TD
  - $V(S_t) \leftarrow V(S_t) + \alpha \frac{\pi(A_t|S_t)}{\mu(A_t|S_t)} \left(R_{t+1} + \gamma V(S_{t+1}) - V(S_t)\right)$

Q-Learning

- Off-policy learning of action-values $Q(s, a)$
- No importance sampling is required
- Next action is chosen using behavior policy $A_{t+1} \sim \mu(\cdot \mid S_t)$
- But we consider alternative successor action $A' \sim \pi(\cdot \mid S_t)$
- Update $Q(S_t, A_t)$ towards value of alternative action

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha\big(R_{t+1} + \gamma Q(S_{t+1}, A') - Q(S_t, A_t)\big)$$

Off-Policy Control with Q-Learning

- Target policy π: greedy w.r.t. $Q(s,a)$

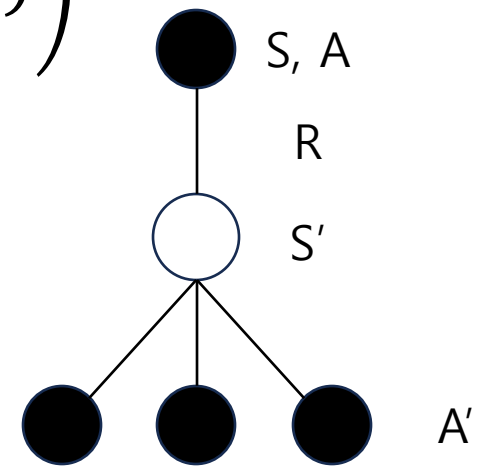
$$\pi(S_{t+1}) = \arg\max_{a'} Q(S_{t+1}, a')$$

- Behavior policy μ: ϵ-greedy w.r.t. $Q(s,a)$

$$\begin{aligned} & R_{t+1} + \gamma Q(S_{t+1}, A') \\ &= R_{t+1} + \gamma Q\left(S_{t+1}, \arg\max_{a'} Q(S_{t+1}, a')\right) \\ &= R_{t+1} + \max_{a'} \gamma Q(S_{t+1}, a') \end{aligned}$$

Q-learning

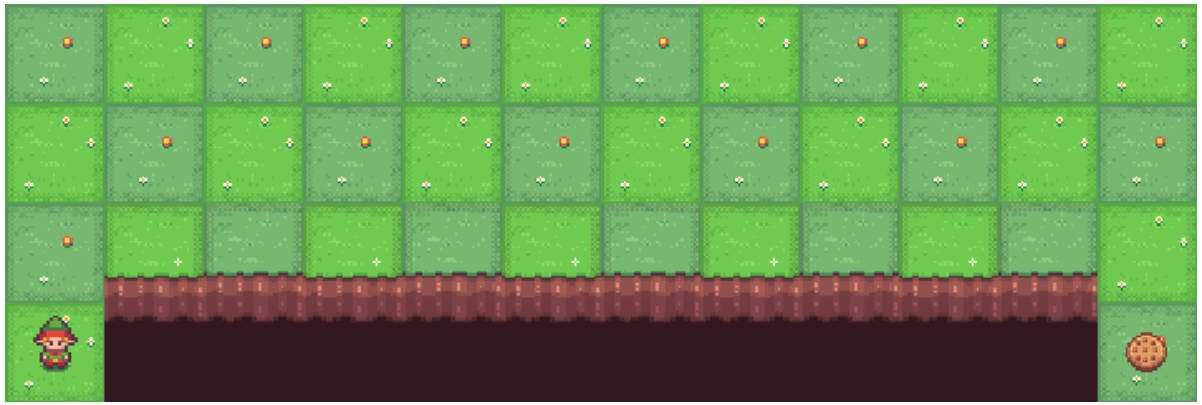
$$Q(S,A) \leftarrow Q(S,A) + \alpha\left(R + \gamma \max_a Q(S',a') - Q(S,A)\right)$$

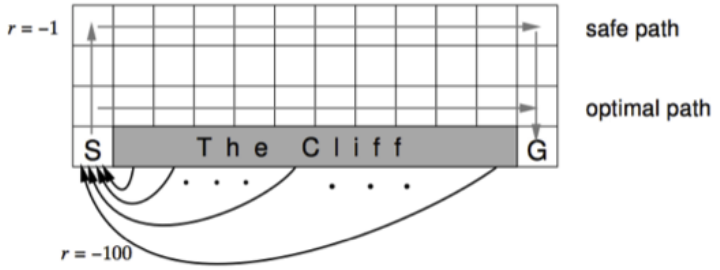


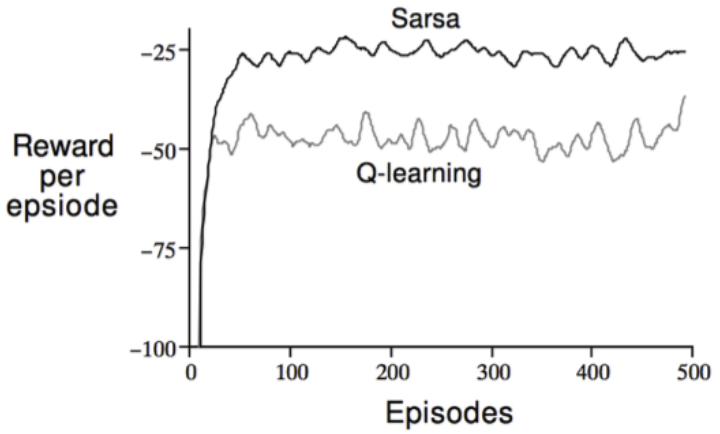
- Initialize $Q(s,a), \forall s \in \mathcal{S}, a \in \mathcal{A}(s)$ and $Q(terminal\ state,\cdot) = 0$
- Repeat (for each episode):
  - Initialize S
  - Choose A from S using policy derived from Q (e.g., ϵ-greedy)
  - Repeat (for each step of episode):
    - Take action A, observe R, S'
    - $Q(S,A) \leftarrow Q(S,A) + \alpha\left[R + \gamma \max_a Q(S',a) - Q(S,A)\right]$
    - $S \leftarrow S'$; $A \leftarrow A'$;
  - Until S is terminal

SARSA & Q-learning

- SARSA devalue the state when falls down the cliff
- Q-learning still finds the optimal path even after falling down
- While training, SARSA could be better
- With stochastic environment, Q-learning could be worse







Reference

- David Silver, COMPM050/COMPGI13 Lecture Notes
- Richard S. Sutton and Andrew G. Barto, “Reinforcement Learning: An Introduction,” 2nd Ed.