Reinforcement Learning

Lecture 6

Model-free Control

Monte Carlo

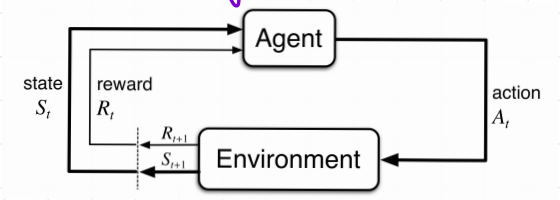
TD: Sarsa and Q-Learning

Robert Peharz

Institute of Theoretical Computer Science
Graz University of Technology
Winter Term 2023/24

Recap

policy $\pi(a|s)$



MDP $(\mathcal{S}, \mathcal{A}, p)$

discounted return $G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} = R_{t+1} + \gamma G_{t+1}$ $0 \le \gamma \le 1$

value function $v_\pi(s) := \mathbb{E}_\pi[G_t \mid S_t = s]$

q-function $q_\pi(s,a) := \mathbb{E}_\pi[G_t \mid S_t = s, A_t = a]$

greedy policy

$$\pi'(a|s) = \begin{cases} 1 & \text{if } a = \arg\max_{a'} q_\pi(s,a') \\ 0 & \text{otherwise} \end{cases}$$

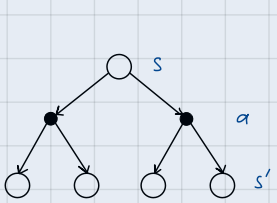
Policy Improvement Theorem

$v_\pi(s) \le v_{\pi'}(s) \quad \forall s \in \mathcal{S}$

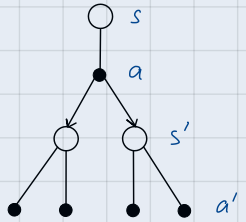
Theorem: Global Optimal Policy

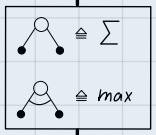
For any initial policy π_0, policy iteration converges to an optimal policy with value function v^*

Bellman Expectation Equation in v

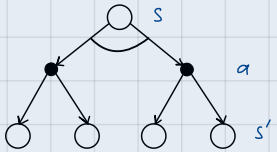


Bellman Expectation Equation in q

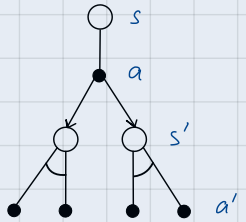


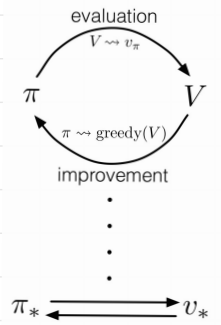


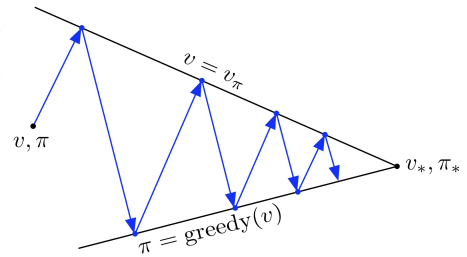
Bellman Optimality Equation in v



Bellman Optimality Equation in q





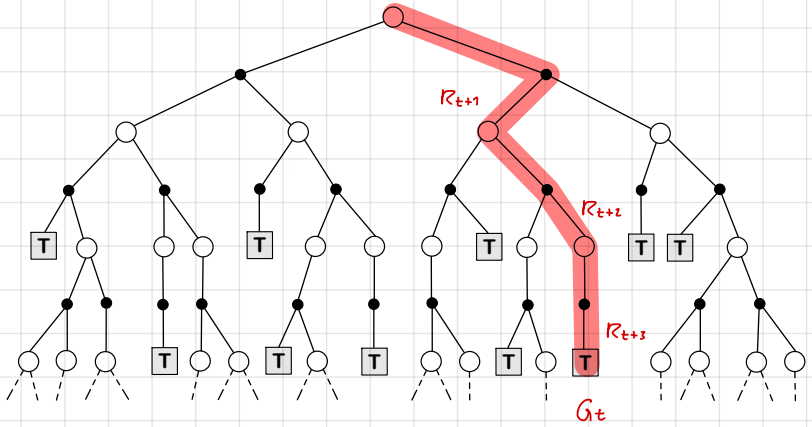


(Generalized) Policy Iteration

Recap: Model-free Prediction

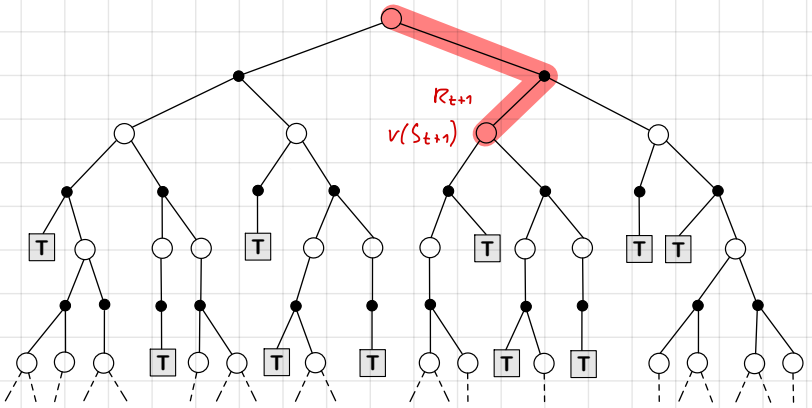
Monte Carlo

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



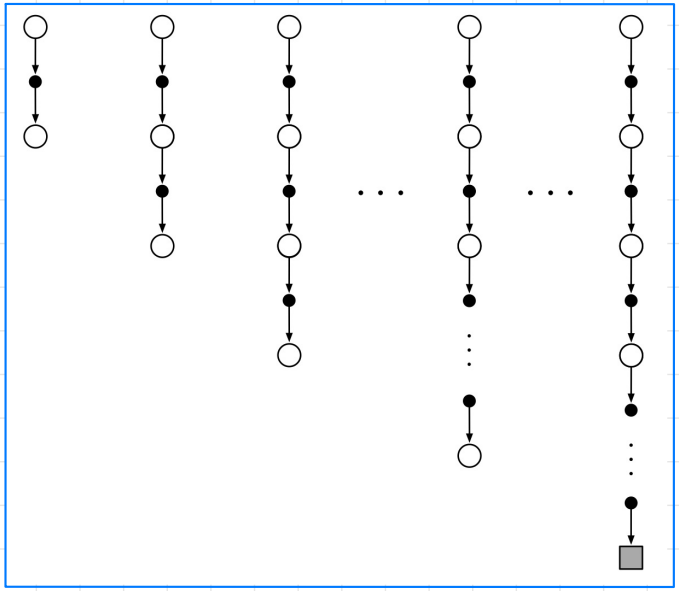
Temporal Differences (TD)

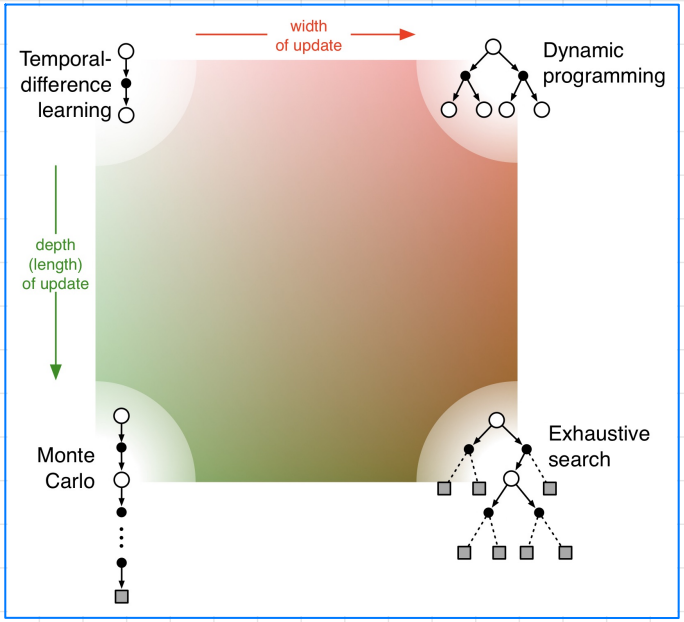
$$V(S) \leftarrow V(S) + \alpha\left(R_{t+1} + \gamma V(S') - V(S)\right)$$



N-step TD

TD ——— MC



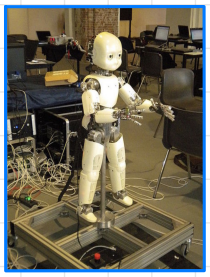


Model-free Control

Why Model-Free Control?

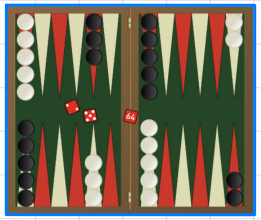
Images: Wikipedia

- MDP might be unknown

  robotics 

  autonomous navigation 

- We are too lazy to describe the MDP explictly

  games 

- MDP too large

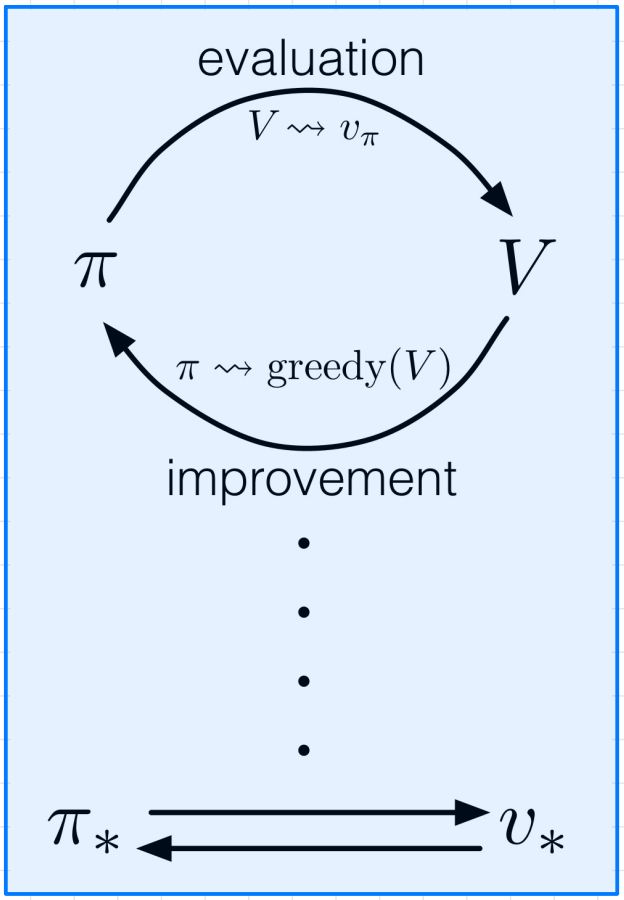
  atari games 

  160×192 pixels

  128 colors

  128^{30720} states

Generalised Policy Iteration (GPI)



- Alternating between (approximately) computing v_π and (approximately) computing greedy(v_π)
- Worked for model-based RL
- We know already two model-free techniques to compute v_π, MC and TD
- Essentially, we will plug them into GPI
- There are two subtleties, however

Image: Sutton & Barto

Subtlety 1: We need q

- Policy improvement:

$$\hat{\pi}'(a|s) = \begin{cases} 1 & \text{if } a = \arg\max_{a'} q_{\hat{\pi}}(s,a') \\ 0 & \text{o.w.} \end{cases}$$

For planning, it was enough to estimate $V_{\hat{\pi}}(s)$, since

$$q_{\hat{\pi}}(s,a) = \underbrace{r(s,a) + \gamma \sum_{s'} p(s'|s,a)}_{\text{MDP, now unknown}} V_{\hat{\pi}}(s')$$

- Thus, we need to directly estimate $q_{\hat{\pi}}$
- No problem, MC and TD carry over to (s,a) pairs

Basically, the same updates

- Monte Carlo

$$v(s) \leftarrow v(s) + \frac{1}{N(s)}(G_t - v(s))$$

$$q(s,a) \leftarrow q(s,a) + \frac{1}{N(s)}(G_t - q(s,a))$$

- Temporal Differences (TD)

$$v(s) \leftarrow v(s) + \alpha(R_{t+1} + \gamma v(s') - v(s))$$

$$q(s,a) \leftarrow q(s,a) + \alpha(R_{t+1} + \gamma q(s',a') - q(s,a))$$

- But, more expensive

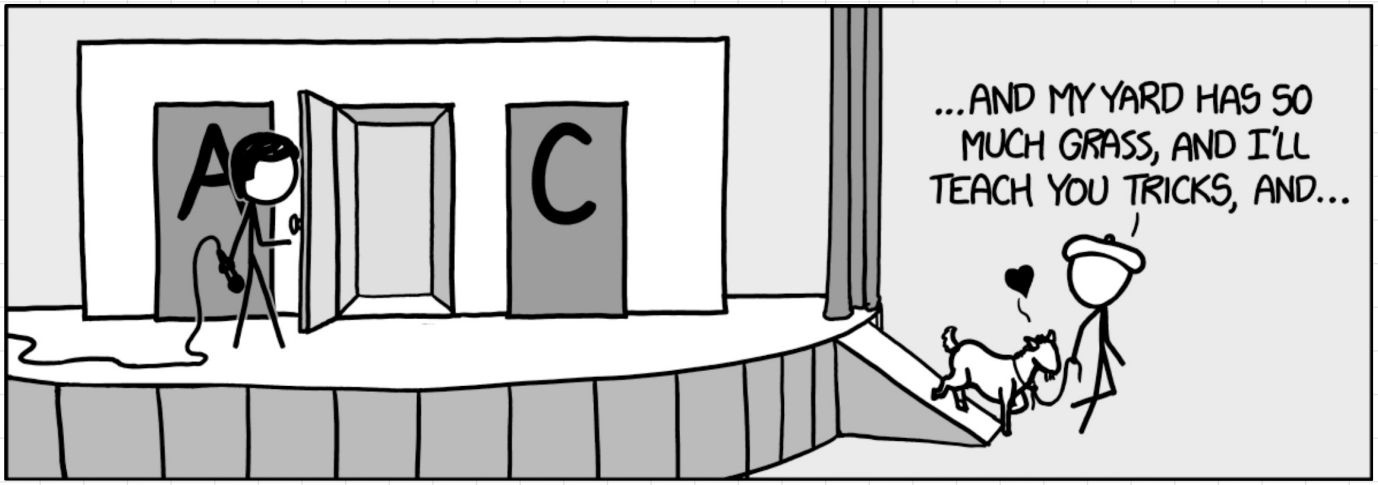
v_π ... $|S|$ values

q_π ... $|S| \times |A|$ values

$$\begin{pmatrix} v_\pi(s_1) \\ v_\pi(s_2) \\ \vdots \\ v_\pi(s_N) \end{pmatrix} \qquad \begin{pmatrix} q_\pi(s_1,a_1) & \dots & q_\pi(s_1,a_M) \\ \vdots & \ddots & \vdots \\ q_\pi(s_N,a_1) & \dots & q_\pi(s_N,a_M) \end{pmatrix}$$

Subtlety 2: Exploration

Image: xkcd.com



- You have the choice between doors A, B, and C.
  - you pick door A and get 5 reward
  - you pick door B and get 0 reward
  - you pick door C and get 1 reward
  - door A ⟶ 3 reward
  - door A ⟶ 4 reward
  - door A ⟶ 7 reward

- Is door A the best? ⟶ **Exploration vs. Exploitation**

Fundamental trade-off in RL

Monte Carlo Control with Exploring Starts

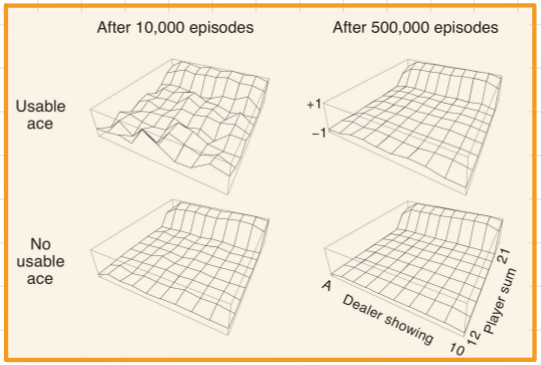
- initialize $q(s, a)$ arbitrarily $\forall s \in \mathcal{S}, a \in \mathcal{A}$
- $N(s, a) \leftarrow 0 \quad \forall s \in \mathcal{S}, a \in \mathcal{A}$ // visit counter
- repeat // assumes episodic tasks
  - select S_0, A_0 randomly, with $p(s, a) > 0, \forall s \in \mathcal{S}, a \in \mathcal{A}$ // exploring starts
  - generate episode $S_0, A_0, R_1, S_1, A_1, R_2, \ldots, S_{T-1}, A_{T-1}, R_T, S_T$ where $A_t = \underset{a'}{\operatorname{argmax}}\, q(S_t, a')$ // $\Uparrow$ = greedy(q)
  - $G \leftarrow 0$
  - for $t = T-1, \ldots, 0$:
    - $G \leftarrow R_{t+1} + \gamma G$
    - $N(S_t, A_t) \leftarrow N(S_t, A_t) + 1$
    - $q(S_t, A_t) \leftarrow q(S_t, A_t) + \frac{1}{N(S_t, A_t)} \left(G - q(S_t, A_t)\right)$ // incremental average

Monte Carlo Control with Exploring Starts cont'd

- $N(s,a) \to \infty$ (since $p(s,a) > 0$) $\quad \forall s \in \mathcal{S}, a \in \mathcal{A}$
- Thus, eventually we will have fully explored the environment
- π cannot converge to suboptimal policy, since q would converge to q_π, leading to a change of π.
- Thus, if algorithm converges, it converges to π^*, q^* ✓
- **Does it always converge?** 
- Empirically, yes.
- Theoretically, open problem.
- Convergence with probability 1 has been proven for synchronous version, i.e. when we simulate an episode for each s,a pair, before updating q and $\gamma < 1$.

[Tsitsiklis, JMLR, 2002]

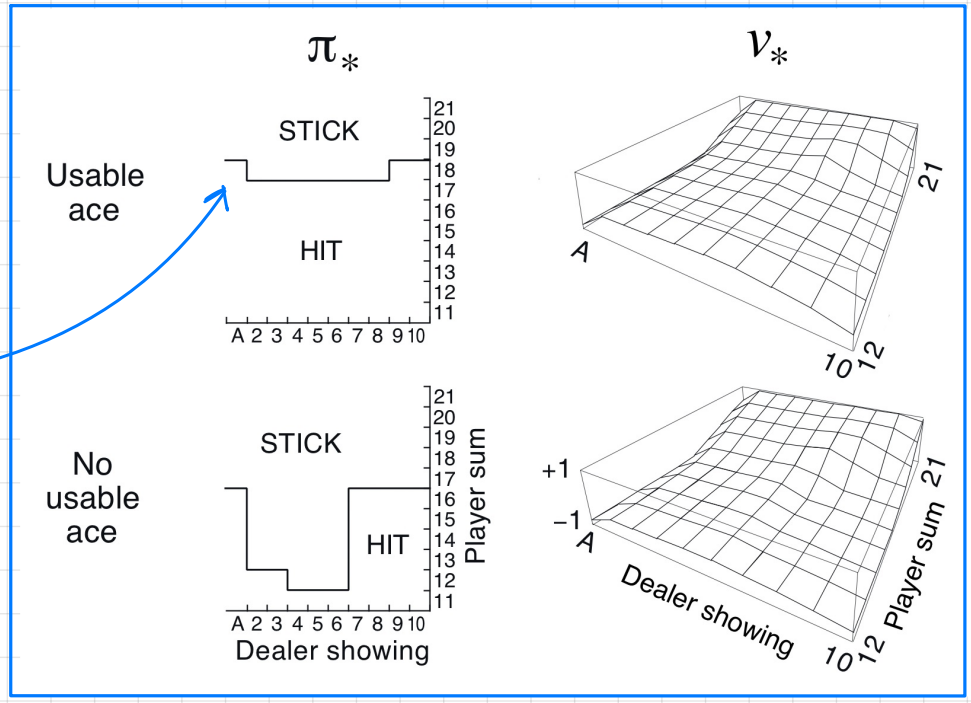
Black Jack cont'd



Recall: Last lecture we evaluated Black Jack with (sub-optimal) "Hit-until-20" strategy

Using MC control, we can find an optimal policy:

"Basic strategy" by Thorp (1966), except for this notch



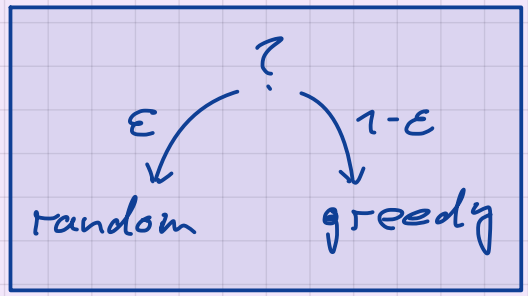
Epsilon Greedy Policies

Image: pinterest.uk

- Exploring starts are unrealistic / impractical...
- Instead, we can incorporate exploration in π

ε-soft Policy: $\pi(a|s) \geq \varepsilon \quad \forall s,a$

ε-greedy Policy: With probability ε, take random action (uniformly), otherwise take greedy action



$$\pi(a|s) = \begin{cases} 1-\varepsilon + \frac{\varepsilon}{|\mathcal{A}(s)|} & \text{if } a = \arg\max_{a'} q(s,a') \\ \frac{\varepsilon}{|\mathcal{A}(s)|} & \text{o.w.} \end{cases}$$

- ε-greedy: the "hardest" ε-soft policy
- for $\varepsilon = 0$: greedy policy

GLIE

Greedy in the Limit with Infinite Exploration

- All state-action pairs are visited infinitely often
- Policy converges to greedy policy

In the GLIE setting, if MC control converges,* it converges to π^* and q^*

* it probably always does

- assume that any state s' can be reached from any state s
- then ε-greedy policy with $\varepsilon = \frac{1}{k}$ fulfills the GLIE property, where k is the episode counter
- thus, ε-greedy policy will converge to optimal policy, with arbitrary starting states.

On Policy vs. Off Policy

Exploration

Exploitation

On-policy methods

"Learning on the job"

need to find a compromise

e.g exploring starts, ϵ-greedy, GLIE

Off-policy methods

"Looking over the shoulder"

separate the problem

Behavior policy b

generates behavior, data

Target policy π

learns from b's data

Importance Sampling

Monte Carlo: $\mathbb{E}[f(\underline{X})] \approx \hat{E} = \frac{1}{N}\sum_{i=1}^{N} f(\underline{X}_i)$, where $\underline{X}_i \overset{\text{i.i.d.}}{\sim} p_{\underline{X}}$

Sometimes hard to sample $p_{\underline{X}}$, but easy to sample a "proxy" q:

$$\mathbb{E}_p[f(\underline{X})] = \int p_{\underline{x}}(\underline{x}) f(\underline{x}) \, d\underline{x}$$

$$= \int q(\underline{x}) \frac{p_{\underline{x}}(\underline{x})}{q(\underline{x})} f(\underline{x}) \, d\underline{x} = \mathbb{E}_q\left[\frac{p_{\underline{x}}(\underline{X})}{q(\underline{X})} \underbrace{f(\underline{X})}_{\tilde{f}(\underline{X})}\right]$$

Importance Sampling Estimator

$$\mathbb{E}[f(\underline{X})] \approx \hat{E}_{IS} = \frac{1}{N}\sum_{i=1}^{N} \frac{p_{\underline{x}}(\underline{X}_i)}{q(\underline{X}_i)} f(\underline{X}_i), \quad \text{where } \underline{X}_i \overset{\text{i.i.d.}}{\sim} q$$

importance weights

Off-policy Monte Carlo

- Generate an episode with behavior policy b:

$$S_0, A_0, R_1, S_1, A_1, R_2, \ldots, S_{T-1}, A_{T-1}, R_T, S_T$$

- Probability of episode under target policy π:

$$p_\pi(A_0, R_1, \ldots \mid S_0) = \pi(A_0 \mid S_0)\, \underbrace{p(S_1, R_1 \mid S_0, A_0)}_{\text{MDP}}\, \pi(A_1 \mid S_1) \ldots \underbrace{p(S_T, R_T \mid S_{T-1}, A_{T-1})}_{\text{MDP}}$$

- Probability of episode under behavior policy b:

$$p_b(A_0, R_1, \ldots \mid S_0) = b(A_0 \mid S_0)\, p(S_1, R_1 \mid S_0, A_0)\, b(A_1 \mid S_1) \ldots p(S_T, R_T \mid S_{T-1}, A_{T-1})$$

- Importance weight for episode:

unknown dynamics cancel!

$$\frac{p_\pi(A_0, R_1, \ldots \mid S_0)}{p_b(A_0, R_1, \ldots \mid S_0)} = \frac{\pi(A_0 \mid S_0)\, \cancel{p(S_1, R_1 \mid S_0, A_0)}\, \pi(A_1 \mid S_1) \ldots \cancel{p(S_T, R_T \mid S_{T-1}, A_{T-1})}}{b(A_0 \mid S_0)\, \cancel{p(S_1, R_1 \mid S_0, A_0)}\, b(A_1 \mid S_1) \ldots \cancel{p(S_T, R_T \mid S_{T-1}, A_{T-1})}}$$

$$= \prod_{t=0}^{T} \frac{\pi(A_t \mid S_t)}{b(A_t \mid S_t)}$$

Off-policy Monte Carlo cont'd

- initialize $q(s, a)$ arbitrarily $\forall s \in \mathcal{S}, a \in \mathcal{A}$
- $N(s, a) \leftarrow 0 \quad \forall s \in \mathcal{S}, a \in \mathcal{A}$
- repeat
  - generate episode using b

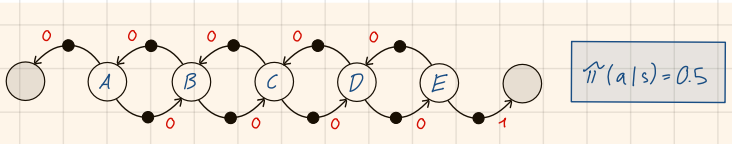
    $S_0, A_0, R_1, S_1, A_1, R_2, \ldots, S_{T-1}, A_{T-1}, R_T$
  - $G \leftarrow 0$
  - $W \leftarrow 1$
  - for $t = T-1, \ldots, 0$:
    - $G \leftarrow R_{t+1} + \gamma G$
    - $W \leftarrow W \frac{\pi(A_t \mid S_t)}{b(A_t \mid S_t)}$ // recursive

      $\prod_{t=t'}^{T} \frac{\pi(A_t \mid S_t)}{b(A_t \mid S_t)}$
    - $N(S_t, A_t) \leftarrow N(S_t, A_t) + 1$
    - $q(S_t, A_t) \leftarrow q(S_t, A_t) + \frac{1}{N(S_t, A_t)} \left(W G - q(S_t, A_t) \right)$

Correct, but silly algorithm... Why? (How is π defined?)

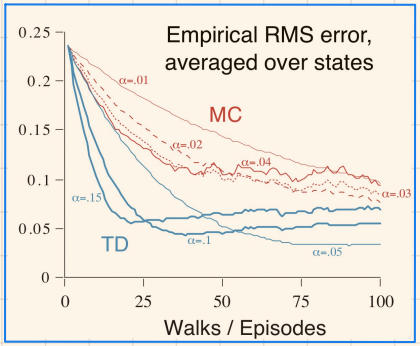
Model-free Control using TD

Sarsa: On-Policy TD Control



TD update for evaluation:

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

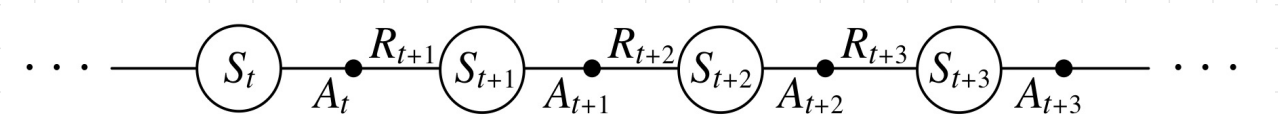


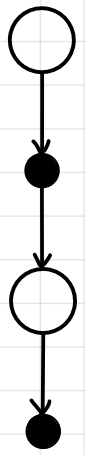
For control, we need TD update for q:

$$q(S_t,A_t) \leftarrow q(S_t,A_t) + \alpha\left(R_{t+1} + \gamma\, q(S_{t+1},A_{t+1}) - q(S_t,A_t)\right)$$

This rule is applied to episodes following current π (on-policy), an to each consecutive quintuple $(S_t, A_t, R_{t+1}, S_{t+1}, A_{t+1})$

SARSA





Images: Sutton & Barto

Sarsa: On-Policy TD Control

Let $\varepsilon > 0$, $\alpha \in (0, 1]$ // hyperparameters for ε-greedy, stepsize

Initialize $q(s, a)$ arbitrarily, except $q(s, a) = 0$ for terminal s

- repeat: // for continuing tasks => one repetition
  - init S
  - generate A from ε-greedy$(q(S, \cdot))$
  - repeat until S is terminal: // forever for continuing task
    - take action A
    - observe S', R
    - generate A' from ε-greedy$(q(S', \cdot))$
    - $q(S, A) \leftarrow q(S, A) + \alpha \left(R + \gamma q(S', A') - q(S, A) \right)$
    - $S, A \leftarrow S', A'$

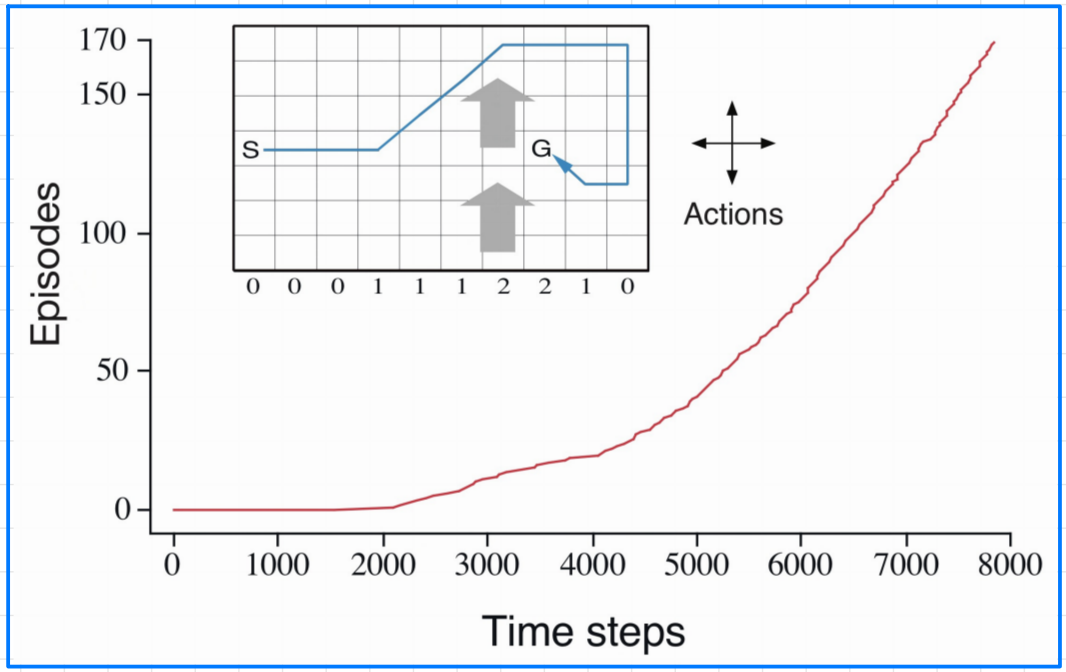
Sarsa converges to q^* and $\hat{\pi}^*$ if

- GLIE holds
- $\sum_{k=0}^{\infty} \alpha_t = \infty$ and $\sum_{k=0}^{\infty} \alpha_t^2 < \infty$ (e.g. $\alpha = \frac{1}{t}$)

[Singh et al., Machine Learning, 2000]

Windy Grid World with Sarsa

- dedicated start state S and goal G
- wind in the center, displacing agent by 1 or 2 cells
- Sarsa with $\alpha = 0.5$ and $\varepsilon = 0.1$
- over time, agent learns to reach goal quicker



read: more completed episodes in fewer time steps

- Monte Carlo would learn very slowly, since it waits until end of episode before update — this takes very long in this grid world!

Image: Sutton & Barto

Importance-sampled Sarsa

$$\hat{E}_{IS} = \frac{1}{N} \sum_{i=1}^{N} \frac{p_{\underline{X}}(\underline{x}_i)}{q(\underline{x}_i)} f(\underline{x}_i), \quad \text{where } \underline{X}_i \overset{\text{i.i.d.}}{\sim} q$$

Sarsa:

$\pi = \varepsilon\text{-greedy}(q)$

$A_{t+1} \sim \pi(a|s)$

$q(s,a) \leftarrow q(s,a) + \alpha\left(R_{t+1} + \gamma\, q(S_{t+1}, A_{t+1}) - q(s,a)\right)$

- On-policy, π is learned while following it
- Using importance sampling, we can make Sarsa off-policy

Importance-weighted Sarsa:

$b = \varepsilon\text{-greedy}(q)$ // behavior

$\pi = \text{greedy}(q)$ // target

$A_{t+1} \sim b(a|s)$

$$q(s,a) \leftarrow q(s,a) + \alpha\left(R_{t+1} + \gamma \frac{\pi(A_{t+1}|S_{t+1})}{b(A_{t+1}|S_{t+1})} q(S_{t+1}, A_{t+1}) - q(s,a)\right)$$

- no GLIE required, ε can be constant
- has rather large variance $\longrightarrow$ Q-learning generally preferred

Q-learning: Off-Policy TD Control

[Watkins 1989]

Sarsa: $q(s,a) \leftarrow q(s,a) + \alpha\big(R_{t+1} + \gamma\, q(S_{t+1}, A_{t+1}) - q(s,a)\big)$

$\mathbb{E}[\delta_t] = 0 \triangleq$ Bellman expectation equation

Q-learning $q(s,a) \leftarrow q(s,a) + \alpha\big(R_{t+1} + \gamma \max_{a'} q(S_{t+1}, a') - q(s,a)\big)$

$\mathbb{E}[\delta_t] = 0 \triangleq$ Bellman optimality equation

Let $\varepsilon > 0$, $\alpha \in (0,1]$ // hyperparameters for ε-greedy, stepsize
Initialize $q(s,a)$ arbitrarily, except $q(s,a) = 0$ for terminal s
- repeat:
  - init S
  - repeat until S is terminal:
    - take action A from behavioral policy, e.g. ε-greedy$(q(S,\cdot))$
    - observe S', R
    - $q(S,A) \leftarrow q(S,A) + \alpha\big(R + \gamma \max_{a'} q(S',a') - q(S,A)\big)$
    - $S \leftarrow S'$

Off-policy: learn (hard) greedy policy while following ε-greedy policy.
ε can be constant (no GLIE). Convergence to q^* for $\sum_k \alpha_k = \infty$, $\sum_k \alpha_k^2 < \infty$.

The Cliff

- grid world with starting state and goal
- the grey area (the cliff) yields large penalty ($R=-100$) and transition to the start
- when running Sarsa and Q-learning with constant $\varepsilon = 0.1$, one will learn the optimal path and the other the safer path.
- which algorithm will learn which path?

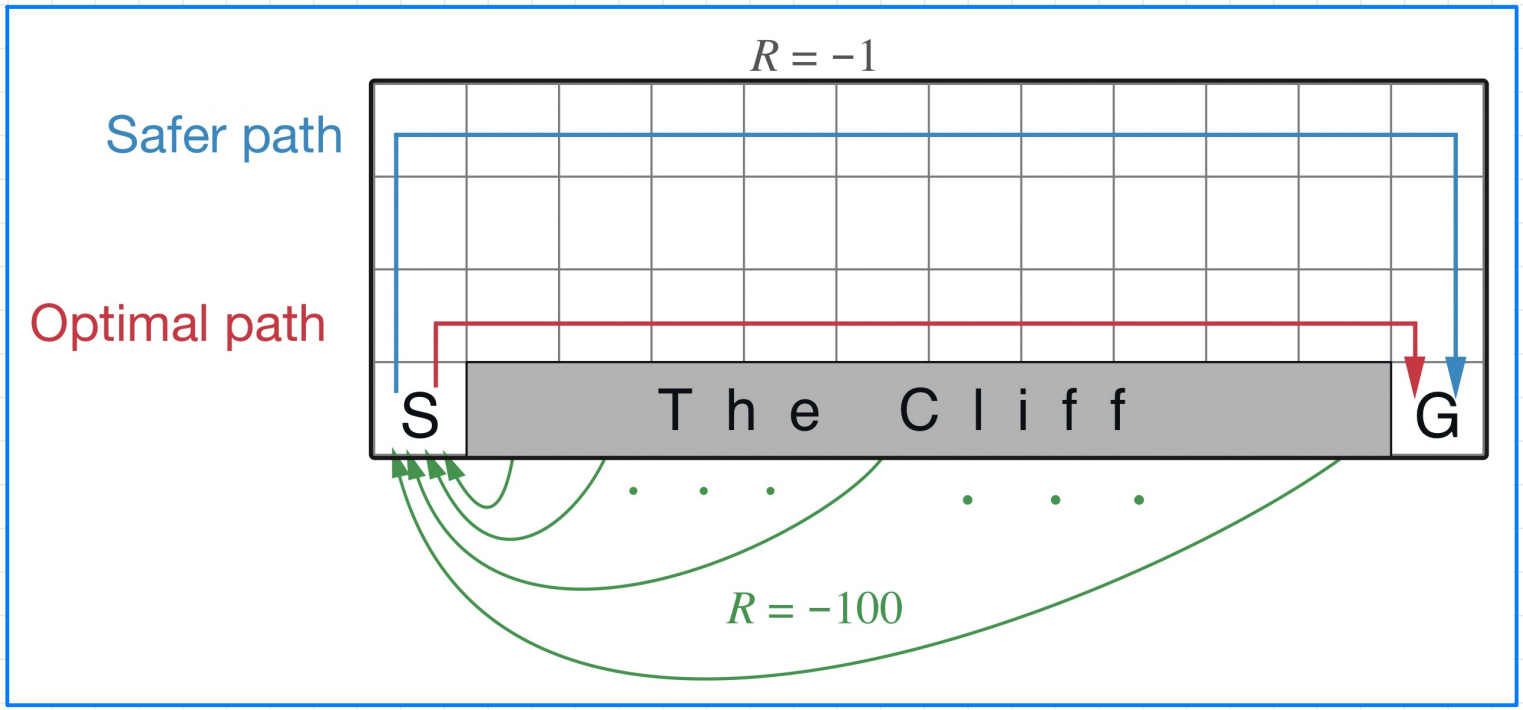


Image: Sutton & Barto

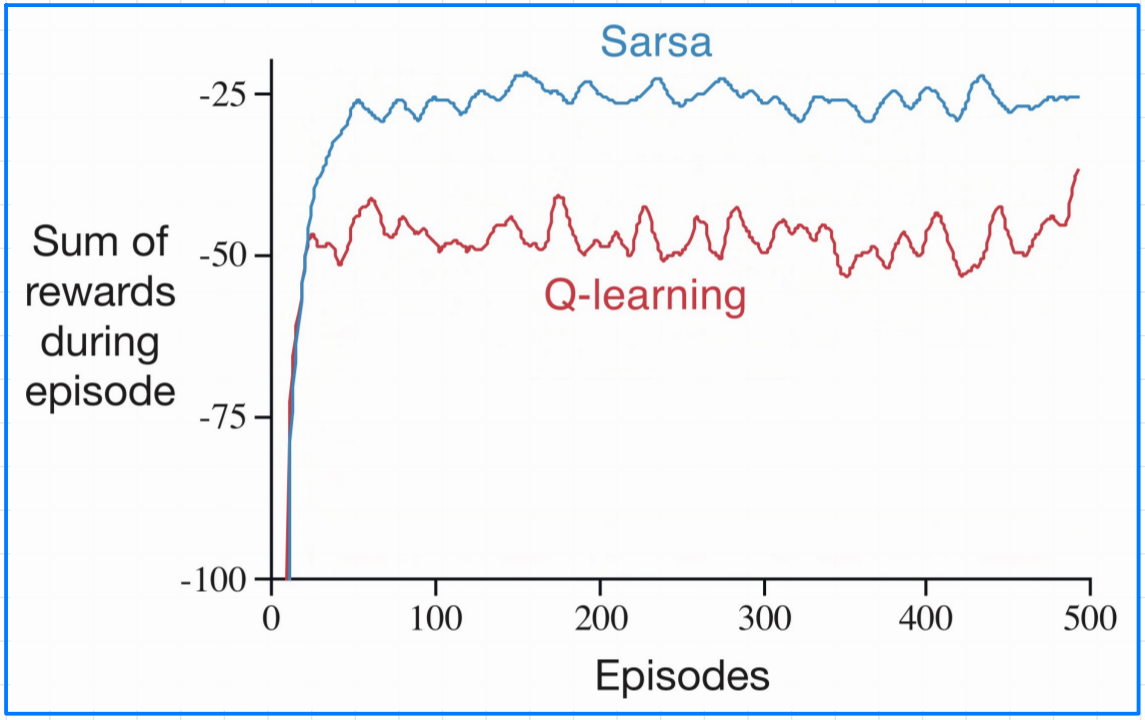
The Cliff cont'd

Q-learning: optimal path

Sarsa: safer path

Sarsa is on-policy: return of behavior policy is maximized

Q-learning is off-policy: return of behavior policy not a goal per se



If $\varepsilon = 1/k$ (GLIE)

- Sarsa converges to π^*
- Q-learning achieves optimal online returns

Image: Sutton & Barto

Expected Sarsa

greedy policy: $\hat{\pi}'(a|s) = \begin{cases} 1 & \text{if } a = \arg\max_{a'} q(s,a') \\ 0 & \text{otherwise} \end{cases}$

Q-Learning: $q(s,a) \leftarrow q(s,a) + \alpha\left(R_{t+1} + \gamma \max_{a'} q(S_{t+1}, a') - q(s,a)\right)$

- off-policy
- behavior ε-greedy
- target greedy

$$\mathbb{E}_{\hat{\pi}'}\left[q(S_{t+1}, A_{t+1})\right] = \sum_{a'} \hat{\pi}'(a'|S_{t+1})\, q(S_{t+1}, a')$$

Sarsa: $q(s,a) \leftarrow q(s,a) + \alpha\left(R_{t+1} + \gamma\, q(S_{t+1}, A_{t+1}) - q(s,a)\right)$

- on-policy
- ε-greedy (GLIE)

$$\hat{E} \approx \mathbb{E}_{\hat{\pi}}\left[q(S_{t+1}, A_{t+1})\right], \quad A_{t+1} \sim \hat{\pi}(a|s)$$

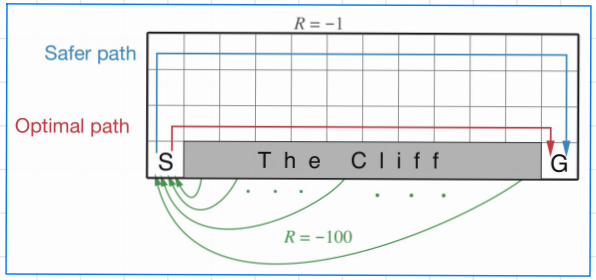
(single sample Monte Carlo)

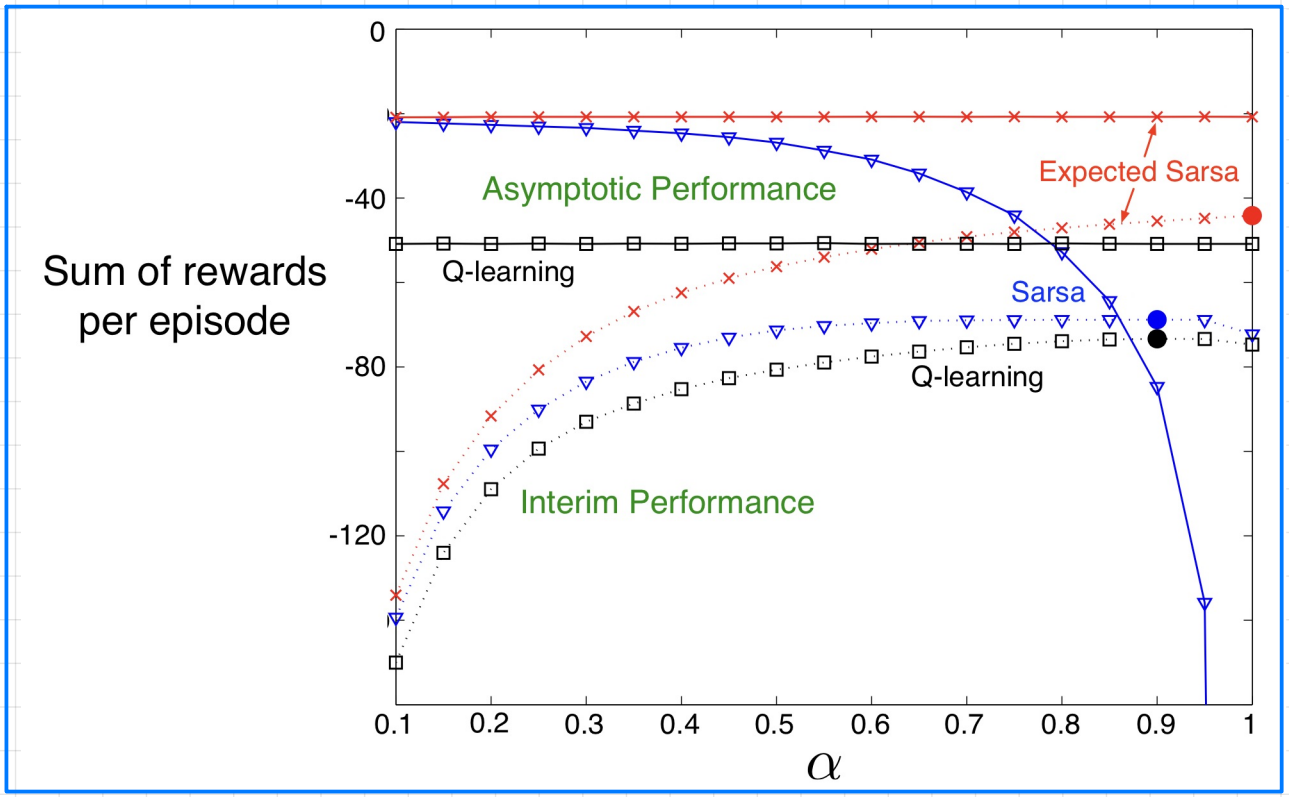
Expected Sarsa

$$q(s,a) \leftarrow q(s,a) + \alpha\left(R_{t+1} + \gamma \sum_{a'} \hat{\pi}(a'|S_{t+1})\, q(S_{t+1}, a') - q(s,a)\right)$$

- on-policy
- ε-greedy (GLIE)
- lower variance than Sarsa
- turns into Q-learning when using $\hat{\pi}'$ instead

The Cliff cont'd

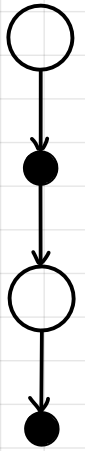
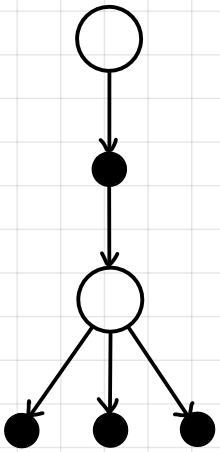




- all algorithms used $\varepsilon = 0.1$
- interim performance: 100 episodes
  asymptotic performance: 100,000 episodes
- expected Sarsa learns faster than Sarsa
  also less sensitive to choice of α

Image: Sutton & Barto

Summary TD Control

	on-policy	off-policy
sample A	Sarsa 	Importance-weighted Sarsa  $\frac{\tilde{\pi}(A_{t+1} \mid S_{t+1})}{b(A_{t+1} \mid S_{t+1})}$
$\mathbb{E}_A$	Expected Sarsa 	Q-learning (Sarsa Max)