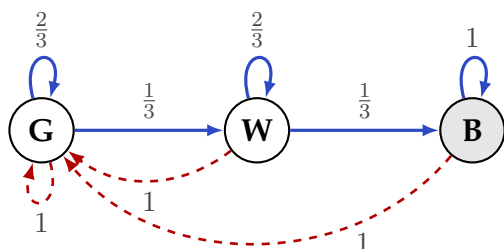


Lecture 10: On-Policy Control: SARSA

How to use these notes. Board content in board order: figures, equations, tables, one-line statements. Today's task is *control without a model*: the learner chooses the actions on a three-state machine, sees the rewards and the states and never P or r , and must end with a rule as good as the best one. The note stands on its own: every symbol is defined where it is first used. Reading: [Sutton and Barto \(2018, Sections 5.4, 6.4, and 6.6\)](#); optional, [Szepesvári \(2010, Section 4.4.1\)](#). Throughout the course, the notes draw on the public materials of Stanford CS 234 ([Brunskill, 2026](#)), UIUC CS 443 ([Jiang, 2026](#)), and Cornell CS 4789 ([Sun, 2025](#)); some examples are adapted from them.

The Task: Control Without a Model

A workshop runs one machine. Each week it is good (G), worn (W), or broken (B), and the manager runs it or has it repaired. Nobody hands the manager a rule: *control* asks for the best rule, found from the weeks themselves. The machine's numbers below are the truth, for checking; the learner never sees them.



The machine, both actions. Run (solid): a good machine earns 8 and stays good w.p. $\frac{2}{3}$, else wears; a worn one earns 4 and stays worn w.p. $\frac{2}{3}$, else breaks; a broken one earns 0 and stays broken. Repair (dashed): costs 10, leaves it good. Gray: hidden from the learner.

The truth, for checking	G	W	B
$Q^*(s, \text{run})$	44.30	32.41	26.89
$Q^*(s, \text{repair})$	29.87	29.87	29.87
$\pi^*(s)$: the larger entry	run	run	repair
$V^*(s) = \max_a Q^*(s, a)$	44.30	32.41	29.87

Known to the learner	Hidden
$S = \{G, W, B\}$, $\mathcal{A} = \{\text{run}, \text{repair}\}$	P : 18 numbers
$\gamma = 0.9$; its own actions	r : 6 numbers

"Repair when broken" is optimal; a wrong rule at B costs $29.87 - 26.89$ or worse. The learner may exercise all 24 hidden numbers: it chooses the action.

Control: what is wanted (Sutton and Barto, 2018, Sections 3.5–3.6). The machine is a Markov decision process $(\mathcal{S}, \mathcal{A}, P, r, \gamma)$: action a in state s pays a reward with mean $r(s, a)$ (on the machine, exactly $r(s, a)$) and moves to s' with probability $P(s' | s, a)$; $\gamma \in [0, 1]$ discounts. A policy $\pi(a | s)$ draws the action; a deterministic one is written $\pi(s)$.

$$Q^\pi(s, a) := \mathbb{E}[R_{t+1} + \gamma R_{t+2} + \dots | S_t = s, A_t = a], \quad V^\pi(s) = \sum_a \pi(a | s) Q^\pi(s, a)$$

under π : do a now, follow π after; V^π averages Q^π over the first action

$$Q^\pi(s, a) = r(s, a) + \gamma \sum_{s'} P(s' | s, a) \sum_{a'} \pi(a' | s') Q^\pi(s', a') =: (\mathcal{T}^\pi Q^\pi)(s, a)$$

Bellman's equation on pairs: this week, then next week's pair under π

$$Q^*(s, a) = r(s, a) + \gamma \sum_{s'} P(s' | s, a) \max_{a'} Q^*(s', a') =: (\mathcal{T} Q^*)(s, a)$$

the optimality equation: this week, then the best pair next week

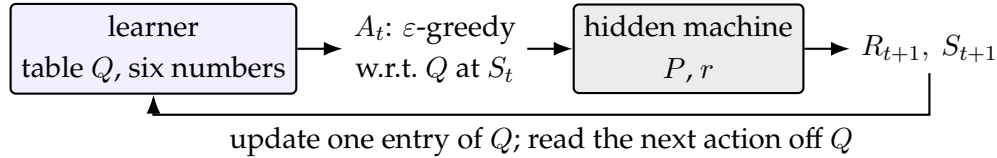
$$\pi^*(s) = \operatorname{argmax}_a Q^*(s, a), \quad V^*(s) = \max_a Q^*(s, a)$$

the best rule reads its action off Q^* : a lookup

$$\|\mathcal{T}^\pi Q - \mathcal{T}^\pi Q'\|_\infty \leq \gamma \|Q - Q'\|_\infty \text{ and the same for } \mathcal{T}; \quad \|Q\|_\infty := \max_{s,a} |Q(s, a)|$$

stated: an entry of the difference is γ times an average, or a max, of entries of $Q - Q'$

So Q^π and Q^* are the only solutions of their equations (Banach, 1922), and π is optimal if and only if it is greedy with respect to its own Q^π (stated). Rewards on the machine lie in $[-10, 8]$, so $|Q| \leq 10/(1 - \gamma) = 100$. With P and r in hand, control is the optimality equation, six unknowns. Today P and r are hidden.



Control without a model. The learner acts, the machine answers, the learner updates one of its six numbers and reads its next action off them. Nothing is stored but Q : the policy is whatever Q says.

The data: one stream, the learner's own actions.

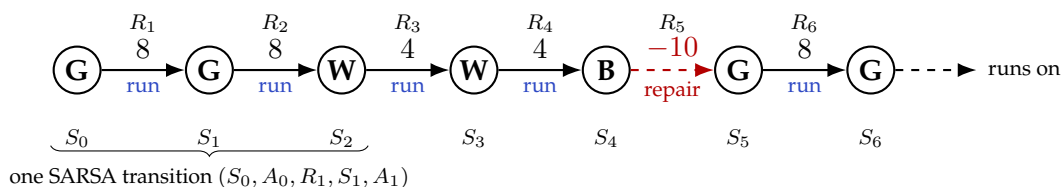
$S_0, A_0, R_1, S_1, A_1, R_2, S_2, A_2, R_3, \dots$ *now A_t is drawn by the learner from its current table*

$S_{t+1} \sim P(\cdot | S_t, A_t), \quad R_{t+1} = r(S_t, A_t)$ *on the machine*

one step of the hidden machine; in general R_{t+1} has mean $r(S_t, A_t)$

- One transition: $(S_t, A_t, R_{t+1}, S_{t+1})$, and today's update reads one more symbol, A_{t+1} .

- No end: the machine is repaired, never retired. The learner never sees a complete return.
- The learner acts *and* learns from the same stream. Wanted: a table whose greedy policy is π^* .



Six weeks of the stream, used by hand below. Every action is the learner's: the table starts at zero, so every choice is a coin flip, and the repair at B is one of them. The actions are part of the record.

Control from one stream, with no P , no r , and the learner choosing every action: which table makes the greedy step a lookup, why must the learner keep trying the other action, and what does trying it cost?

Today's targets.

1. From V to Q : with a Q table the greedy step is $\arg\max_a Q(s, a)$, a lookup; the sampled backup $R_{t+1} + \gamma Q(S_{t+1}, A_{t+1})$ replaces the lookahead's P and r . The loop without a model: evaluate Q from the stream, improve ε -greedily.
2. Theorem, three lines from the policy improvement theorem: ε -greedy w.r.t. Q^π improves any ε -soft π . On the machine at $\varepsilon = 0.1$: (29.12, 14.43, 5.59) $\rightarrow$ (33.30, 20.07, 19.86) $\rightarrow$ (39.82, 28.87, 25.70), then unchanged, 4.49 below V^* at G.
3. SARSA: the template on Q , one transition at a time. By hand, six weeks from $Q = 0$ with $\alpha = \frac{1}{2}$: $Q(\text{G}, \text{run}) = 9.7$, $Q(\text{W}, \text{run}) = 3$, $Q(\text{B}, \text{repair}) = -2.3$, and the greedy policy says never repair.
4. Expected SARSA: the coin averaged out. By computer: $\varepsilon = 0$ stuck; $\varepsilon = 0.1$ reads π^* off the table by week 1,000, settles 4 from Q^* ; shrinking ε goes lower ($1/\sqrt{t}$) or dies too soon ($1/t$). The convergence theorem is a reading box.

From V to Q : the Greedy Step Without a Model

The planning loop: evaluate the current policy, then improve it greedily, repeat. It needs the model in exactly one place.

Policy iteration, and where the model is used (Sutton and Barto, 2018, Sections 4.3 and 6.1). Start from any π .

evaluate: $V^\pi(s) = r(s, \pi(s)) + \gamma \sum_{s'} P(s' | s, \pi(s)) V^\pi(s')$ *a linear solve with P and r*

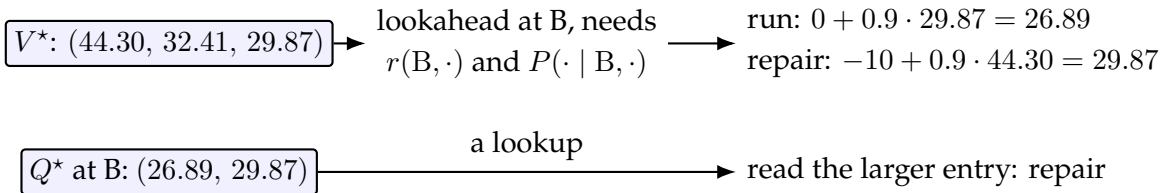
or, from the stream of π : $V(S_t) \leftarrow V(S_t) + \alpha(R_{t+1} + \gamma V(S_{t+1}) - V(S_t))$
TD(0), stated: it converges to V^π ; no P , no r

improve: $\pi'(s) = \operatorname{argmax}_a r(s, a) + \gamma \underbrace{\sum_{s'} P(s' | s, a) V^\pi(s')}_{= Q^\pi(s, a)}$

the lookahead: r and P at every (s, a) , including the actions π never takes

stop when $\pi' = \pi$: then π is greedy w.r.t. Q^π , so optimal *the test stated in the first box*

With Q^π in hand, $\pi'(s) = \operatorname{argmax}_a Q^\pi(s, a)$ is a lookup: no P , no r . **Learn Q^π , not V^π .**



The same decision twice. From V^* the lookahead at B needs two rewards and two rows of P , all hidden. From Q^* it needs nothing: the comparison is already in the table.

Q^π from the stream: the sampled backup (Sutton and Barto, 2018, Section 6.4). For any table Q , any pair (s, a) , and $A_{t+1} \sim \pi(\cdot | S_{t+1})$, call $R_{t+1} + \gamma Q(S_{t+1}, A_{t+1})$ the *target*:

$$\mathbb{E}[R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) | s, a] = r(s, a) + \gamma \sum_{s'} P(s' | s, a) \sum_{a'} \pi(a' | s') Q(s', a')$$

given $S_t = s, A_t = a$: R_{t+1} has mean $r(s, a)$, $S_{t+1} \sim P(\cdot | s, a)$, $A_{t+1} \sim \pi(\cdot | S_{t+1})$

$$= (\mathcal{T}^\pi Q)(s, a)$$

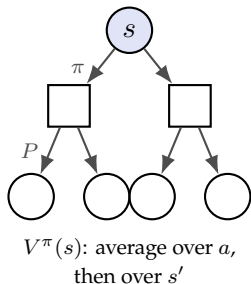
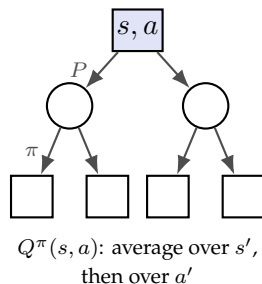
the backup on pairs, from the first box: the expected target is the backup

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

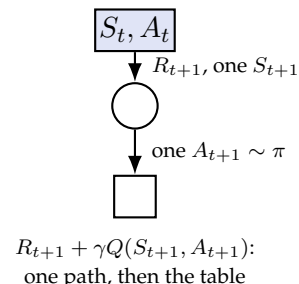
the template: move the entry a fraction α toward the sampled backup

Theorem (stated): for a fixed π , if every pair is visited forever and the step sizes at each

pair satisfy $\sum \alpha = \infty$, $\sum \alpha^2 < \infty$ (Robbins and Monro, 1951), then $Q \rightarrow Q^\pi$ w.p. 1. TD(0) on pairs (Szepesvári, 2010, Section 4.4.1).

 V^π : state to state Q^π : pair to pair

Today: sampled



Three backups. Circles are states, squares are pairs. The value of a pair averages over P and then over π ; the stream supplies one draw of each and the table supplies the rest: bootstrapping, a guess from a guess.

Example: what a deterministic policy never learns. π = never repair, $Q = 0$ at the start, the template run on π 's stream forever. Which of the six entries ever move? (G, run), (W, run), (B, run): the three pairs π visits. The repair column keeps its starting value, 0, forever. At B the lookup compares $Q(\text{B, run}) \rightarrow 0$ with an untouched 0: a tie, and the true answer is 29.87 against 26.89. The data never price repair. The loop needs a policy that takes every action sometimes.

ϵ -soft and ϵ -greedy (Sutton and Barto, 2018, Section 5.4).

π is ϵ -soft: $\pi(a | s) \geq \frac{\epsilon}{|\mathcal{A}|}$ for every s and a every action keeps at least $\epsilon/|\mathcal{A}|$

π' is ϵ -greedy w.r.t. Q : $\pi'(a | s) = \begin{cases} 1 - \epsilon + \epsilon/|\mathcal{A}| & a = \operatorname{argmax}_{a'} Q(s, a') \\ \epsilon/|\mathcal{A}| & \text{otherwise} \end{cases}$

a coin with probability ϵ : tails, the greedy action; heads, a uniform draw

Machine, $\epsilon = 0.1$: the greedy action w.p. 0.95, the other w.p. 0.05. At a tie the greedy mass is split between the maximizers. An ϵ -greedy policy is ϵ -soft.

The loop	With the model	Without the model (today)
evaluation	solve $V^\pi = r^\pi + \gamma P^\pi V^\pi$ exactly	move $Q(S_t, A_t)$ toward $R_{t+1} + \gamma Q(S_{t+1}, A_{t+1})$, one transition at a time
improvement	greedy at every state: a lookahead per pair with P and r	ε -greedy w.r.t. Q : a lookup, the other action kept alive
the policy	deterministic	ε -soft; the stream is the policy's own
the grain	a full solve, then a full improvement: one round	one entry, then the next action: one week
the stop	unchanged means optimal	never stops; what it settles toward is the question below

Generalized policy iteration (Sutton and Barto, 2018, Section 4.6): evaluation and improvement at any grain. Today's grain is one week, and the improvement is softened so that the evaluation has something to learn from.

Improvement Without the Model: ε -Greedy Improves

Policy improvement theorem (stated) (Sutton and Barto, 2018, Section 4.2). For any two policies π and π' , stochastic allowed:

$$\sum_a \pi'(a | s) Q^\pi(s, a) \geq V^\pi(s) \text{ for every } s \implies V^{\pi'}(s) \geq V^\pi(s) \text{ for every } s,$$

one week of π' , then π forever, is no worse than π ; repeating the swap never hurts; the limit is π' forever

strict at every s where the hypothesis is strict.

Theorem: ε -greedy improvement (Sutton and Barto, 2018, Section 5.4). Let π be ε -soft and π' be ε -greedy w.r.t. Q^π . Then $V^{\pi'} \geq V^\pi$.

$$\sum_a \pi'(a | s) Q^\pi(s, a) = \frac{\varepsilon}{|\mathcal{A}|} \sum_a Q^\pi(s, a) + (1 - \varepsilon) \max_a Q^\pi(s, a)$$

π' puts $\varepsilon/|\mathcal{A}|$ on every action and the remaining $1 - \varepsilon$ on a maximizer

$$\geq \frac{\varepsilon}{|\mathcal{A}|} \sum_a Q^\pi(s, a) + (1 - \varepsilon) \sum_a \frac{\pi(a | s) - \varepsilon/|\mathcal{A}|}{1 - \varepsilon} Q^\pi(s, a)$$

the weights are ≥ 0 since π is ε -soft, and sum to 1: a max beats a weighted average

$$= \sum_a \pi(a | s) Q^\pi(s, a) = V^\pi(s)$$

the two sums combine; then the improvement theorem $\square$

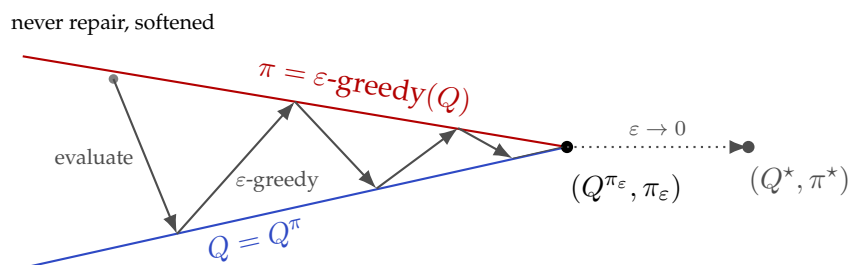
If $\pi' = \pi$, that is, π is ε -greedy w.r.t. its own Q^π , then no ε -soft policy beats π (stated; the reading box below).

Reading: why the fixed point is the best ε -soft policy (Sutton and Barto, 2018, Section 5.4). Fold the coin into the machine: a new machine that, w.p. ε , replaces the chosen action by a uniform draw before moving. Every policy of the new machine is an ε -soft policy of the old one, and conversely; the values agree. The new machine has its own optimality equation: with $\tilde{Q}(s, a) = r(s, a) + \gamma \sum_{s'} P(s' | s, a) \tilde{V}(s')$, $\tilde{V}(s) = (1 - \varepsilon) \max_a \tilde{Q}(s, a) + \frac{\varepsilon}{|A|} \sum_a \tilde{Q}(s, a)$; a policy that is ε -greedy w.r.t. its own values solves it, so it is optimal there. As $\varepsilon \rightarrow 0$ the new machine becomes the old one and the fixed point becomes π^* .

$\varepsilon = 0.1$, every evaluation exact	round 0	round 1	round 2	π^* , exact
π : the named action w.p. 0.95	never repair	repair when worn	repair when broken	repair when broken
V^π (G, W, B)	(29.12, 14.43, 5.59)	(33.30, 20.07, 19.86)	(39.82, 28.87, 25.70)	(44.30, 32.41, 29.87)
$Q^\pi(\cdot, \text{run})$	(29.80, 14.33, 5.03)	(34.00, 22.00, 17.88)	(40.55, 29.03, 23.13)	(44.30, 32.41, 26.89)
$Q^\pi(\cdot, \text{repair})$, every state	16.21	19.97	25.83	29.87
π' : ε -greedy w.r.t. Q^π	repair when worn	repair when broken	unchanged	
hypothesis $\sum_a \pi'(a s) Q^\pi(s, a)$	(29.12, 16.11, 15.65)	(33.30, 21.90, 19.86)	(39.82, 28.87, 25.70)	

The theorem on the machine. In each round the hypothesis row is at least the V^π row, and the next round's V^π is at least the hypothesis. The repair entry is the same in every state: repair always buys a good machine. Same path as exact policy iteration from never repair, softened; it ends (4.49, 3.53, 4.17) below V^* .

Example: what does exploring cost? The fixed point, repair when broken at 0.95, is worth (39.82, 28.87, 25.70) against $V^* = (44.30, 32.41, 29.87)$: 4.49 at G, a tenth of the value. Per week, in the long run: π^* earns $26/7 = 3.71$, the softened one 3.35. Where does it go? One week in twenty the coin says the other action: a good or worn machine is repaired (−10 for 8 or 4), or a broken one is run (0 for −10 and a good machine next week).



Sutton and Barto's two lines, with the improvement softened. Evaluation reaches the blue line, ε -greedy improvement the red one; the only point on both is the ε -greedy fixed point $(Q^{\pi_\varepsilon}, \pi_\varepsilon)$, π_ε being ε -greedy w.r.t. its own Q^{π_ε} ; it is not (Q^*, π^*) . Shrinking ε moves it there.

SARSA: the Loop One Transition at a Time

SARSA (Rummery and Niranjan, 1994; Sutton, 1996), (Sutton and Barto, 2018, Section 6.4).

- Input: the machine; a step size $\alpha \in (0, 1]$; an exploration rate $\varepsilon > 0$. Output: a table Q , read as the policy $\arg\max_a Q(s, a)$.
- Start: $Q_0(s, a) = 0$ for every pair (if episodes end, $Q(\text{terminal}, \cdot) = 0$ always); observe S_0 ; draw A_0 ε -greedy w.r.t. Q_0 at S_0 .
- Each week t : take A_t ; observe R_{t+1}, S_{t+1} ; draw A_{t+1} ε -greedy w.r.t. Q_t at S_{t+1} ; then

$$\delta_t = R_{t+1} + \gamma Q_t(S_{t+1}, A_{t+1}) - Q_t(S_t, A_t) \quad \text{the TD error on pairs}$$

$$Q_{t+1}(S_t, A_t) = Q_t(S_t, A_t) + \alpha \delta_t \quad \text{the template at one entry; the other five unchanged}$$

- *On-policy*: the action inside the target, A_{t+1} , is the action taken next week. The quintuple S, A, R, S', A' is the name.
- Nothing stored but Q : memory $|\mathcal{S}||\mathcal{A}|$, work $O(1)$ a week. The policy is never written down.

Example. Control always has a max somewhere. Where is SARSA's?

In the draw of A_{t+1} : ε -greedy w.r.t. Q_t , so $Q_t(S_{t+1}, A_{t+1})$ is $\max_a Q_t(S_{t+1}, a)$ w.p. 0.95. Improvement at every week, folded into the data; the policy evaluated keeps changing.

Example: SARSA by hand, six weeks. $\alpha = \frac{1}{2}, \varepsilon = 0.1, Q = 0$, the stream of the figure with its actions. Fill the table one week at a time: the target and the error first, then the one entry that moves.

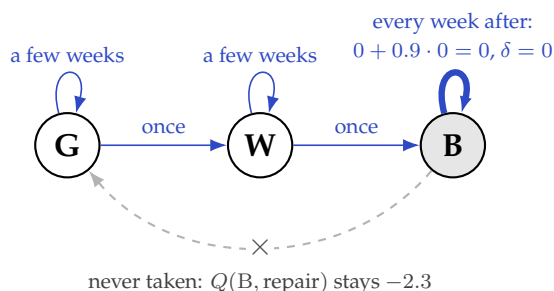
week t	$(S_t, A_t, R_{t+1}, S_{t+1}, A_{t+1})$	target $R_{t+1} + 0.9 Q(S_{t+1}, A_{t+1})$	δ_t	$Q(\cdot, \text{run})$ after	$Q(\cdot, \text{repair})$ after
0	(G, run, 8, G, run)	$8 + 0.9 \cdot 0 = 8$	$8 - 0 = 8$	(4 , 0, 0)	(0, 0, 0)
1	(G, run, 8, W, run)	$8 + 0.9 \cdot 0 = 8$	$8 - 4 = 4$	(6 , 0, 0)	(0, 0, 0)
2	(W, run, 4, W, run)	$4 + 0.9 \cdot 0 = 4$	$4 - 0 = 4$	(6, 2 , 0)	(0, 0, 0)
3	(W, run, 4, B, repair)	$4 + 0.9 \cdot 0 = 4$	$4 - 2 = 2$	(6, 3 , 0)	(0, 0, 0)
4	(B, repair, -10, G, run)	$-10 + 0.9 \cdot 6 = -4.6$	$-4.6 - 0 = -4.6$	(6, 3, 0)	(0, 0, -2.3)
5	(G, run, 8, G, run)	$8 + 0.9 \cdot 6 = 13.4$	$13.4 - 6 = 7.4$	(9.7 , 3, 0)	(0, 0, -2.3)
truth Q^* , for comparison				(44.30, 32.41, 26.89)	(29.87, 29.87, 29.87)

Six updates, one entry each (bold). Week 3 reads the untouched $Q(\text{B, repair}) = 0$ into its target; week 4 reads the guess $Q(\text{G, run}) = 6$ into the price of a repair. The policy read off the table after week 5: run, run, run.

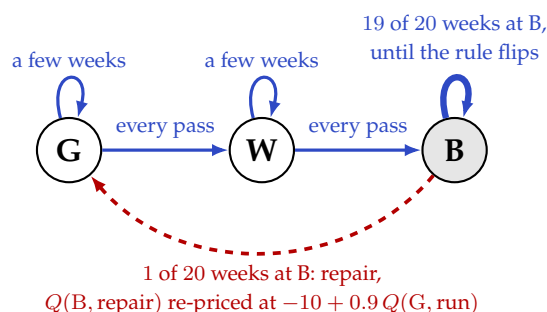
- The table's verdict at B: repair -2.3, run 0. The data priced a repair as a 10 bill plus a good machine the table valued at 6.
- Greedy w.r.t. this table is never repair, the policy worth (27.5, 10, 0). The truth at B is 29.87 against 26.89, the other way.

Check: switch to $\varepsilon = 0$ after week 5. Does the learner ever repair again?

No. The machine is good at week 6; run wears it, then breaks it, and never fixes it, so it reaches B and stays. Every week at B is then the thick loop of the left panel below. Colored arrows there are the ones the learner walks, thicker for more often; gray, never; repair at G and W is left out.

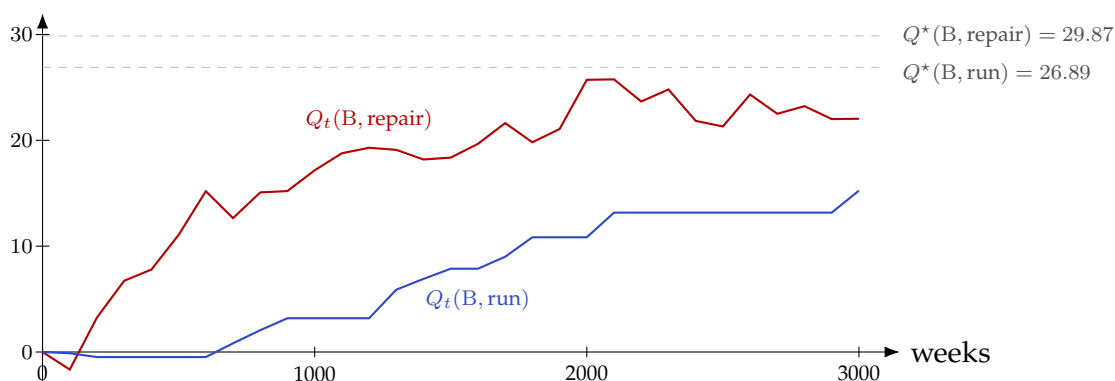


$\varepsilon = 0$: the loop closes. Down the chain once, then run at B every week: target 0, table 0, no error. Repair at B is never taken, so -2.3 is never re-priced.



$\varepsilon = 0.1$: the loop opens. One week in 20 at B the coin repairs (at G and W too): the entry is re-priced, and the learner goes round again, so $Q(G, \text{run})$ grows every pass.

- At B the rule compares two entries: run, checked every week and confirmed (0 under never repair), and repair, never taken, so never re-priced; the truth is 29.87.
- The bill: 0 a week at B for good, against 29.87. The table is settled, $\delta = 0$ every week, and wrong: the next box asks for every pair visited forever, and (B, repair) is not.
- With $\varepsilon = 0.1$ the price of a repair, $-10 + 0.9 Q(G, \text{run})$, is positive once $Q(G, \text{run}) > 10/0.9 = 11.1$ (the truth is 44.30), and the rule at B flips. By computer, a fresh stream from $Q = 0$ at $\alpha = 0.1$: the figure.



One stream of SARSA, $\varepsilon = 0.1$, $\alpha = 0.1$, from $Q = 0$: the two entries at B every 100 weeks. The repair entry starts below zero, crosses the run entry by week 200 and stays above it through the 3,000 weeks shown; it dips back below at 3 of the 970 later checkpoints: SARSA improves against an estimate, not the theorem's exact Q^π , so a week can move the rule the wrong way. The run entry moves only when the coin runs a broken machine.

The expected update, and where SARSA can stop (informal) (Sutton and Barto, 2018, Section 6.4), (Singh et al., 2000). Write π_t for the ε -greedy policy of Q_t : the policy that draws A_{t+1} .

$$\mathbb{E}[R_{t+1} + \gamma Q_t(S_{t+1}, A_{t+1}) \mid Q_t, S_t = s, A_t = a] = (\mathcal{T}^{\pi_t} Q_t)(s, a)$$

the sampled-backup line, with today's policy in the expectation

$$\mathbb{E}[Q_{t+1}(s, a) \mid Q_t, S_t = s, A_t = a] = (1 - \alpha) Q_t(s, a) + \alpha (\mathcal{T}^{\pi_t} Q_t)(s, a)$$

the evaluation step of the loop, toward Q^{π_t} ; the improvement is in π_t itself

$$\text{settled at } \hat{Q}, \text{ every pair visited forever: } \mathbb{E}[\delta_t \mid s, a] = (\mathcal{T}^{\hat{\pi}} \hat{Q})(s, a) - \hat{Q}(s, a) = 0$$

$\hat{\pi} = \varepsilon$ -greedy($\hat{Q}$); a nonzero mean increment repeated forever would move the entry

$$\hat{Q} = \mathcal{T}^{\hat{\pi}} \hat{Q} \Rightarrow \hat{Q} = Q^{\hat{\pi}} : \hat{\pi} \text{ is } \varepsilon\text{-greedy w.r.t. its own values}$$

*the fixed point of the theorem's loop, $\hat{\pi} = \pi_\varepsilon$: the best ε -soft policy, not π^**

Machine, $\varepsilon = 0.1$: $\hat{Q}(\cdot, \text{run}) = (40.55, 29.03, 23.13)$, $\hat{Q}(\cdot, \text{repair}) = 25.83$; $\|\hat{Q} - Q^*\|_\infty = 4.04$, and greedy w.r.t. $\hat{Q}$ is π^* . The table is off by 4 and the policy read off it is right.

Toward Q^* . A constant ε stops at Q^{π_ε} . To reach Q^* the coin must die out, $\varepsilon_t \rightarrow 0$, yet slowly enough that every pair keeps being tried, and α must shrink too; the conditions and the theorem are the reading box below.

Reading (optional): GLIE and the convergence theorem (stated) (Singh et al., 2000), (Sutton and Barto, 2018, Section 6.4), (Szepesvári, 2010, Section 4.4.1). Nothing below depends on this box. A sequence of policies is *greedy in the limit with infinite exploration* if

- (i) every pair (s, a) is visited infinitely often; *keep trying everything*
- (ii) $\pi_t(a \mid s) \rightarrow \mathbb{I}[a = \arg\max_{a'} Q_t(s, a')]$ w.p. 1. *try the other action less and less*

Theorem: SARSA with a GLIE sequence of ε -greedy policies and step sizes with $\sum \alpha = \infty$, $\sum \alpha^2 < \infty$ at every pair gives $Q_t \rightarrow Q^*$ and $\pi_t \rightarrow \pi^*$ w.p. 1. Proof by stochastic approximation; not given.

- $\varepsilon_t \rightarrow 0$ gives (ii). (i) is about visits, which a schedule on a global clock cannot promise; the theorem's schedules count visits per state, $\varepsilon_t(s) = c/N_t(s)$.
- A global $\varepsilon_t = 1/t$ on the machine: (ii) holds, (i) is not guaranteed. While the greedy rule is never repair the chain sits at B, and the expected number of tries of (G, repair) is finite.
- A constant ε breaks (ii): on the machine the table tracks Q^{π_ε} , 4 from Q^* , and the policy read off it is π^* .
- The runs below use a constant α : they show the mechanisms, not the theorem.

Expected SARSA

Expected SARSA (van Seijen et al., 2009), (Sutton and Barto, 2018, Section 6.6). Behavior stays ε -greedy; the target averages the next-action values with their probabilities:

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \underbrace{\left(R_{t+1} + \gamma \sum_a \pi(a | S_{t+1}) Q(S_{t+1}, a) - Q(S_t, A_t) \right)}_{\text{target}}$$

π the ε -greedy policy of the current table; A_{t+1} is drawn but not used in the target

$\mathbb{E}[\text{target} | S_t = s, A_t = a] = (\mathcal{T}^\pi Q)(s, a)$ as before; the noise from A_{t+1} is gone

the same mean, one coin flip less; $|A|$ lookups per week instead of one

- SARSA uses the next action actually drawn. Expected SARSA uses all next actions, weighted by their probabilities: the same expected target, without the next-action noise.
- Fix the current table: repair always returns to G, so its Expected SARSA target is fixed. Random state transitions still make run's target vary.
- Greedy in the sum, the target is $R_{t+1} + \gamma \max_a Q(S_{t+1}, a)$: the optimality backup, sampled. That is Q-learning, *off-policy*: it evaluates a policy other than the one acting.

Example: one repair, three targets. Week 4: repair at B costs 10 and returns the machine to G. Freeze the current table:

$$Q(G, \text{run}) = 6, \quad Q(G, \text{repair}) = 0, \quad \gamma = 0.9, \quad \varepsilon = 0.1.$$

At G, the next action is run with probability 95%, repair with 5%. If the draw switches from run to repair, which targets change?

Method	How it forms the target	Next: run (95%)	Next: repair (5%)
SARSA	Use the action drawn	-4.6	-10
Expected SARSA	Average with weights 95%, 5%	-4.87	-4.87
Q-learning	Use the best-valued action	-4.6	-4.6

Targets for the same repair and the same table. Expected SARSA stays close to the common outcome -4.6; the rare -10 pulls its average down. Q-learning always uses the larger value.

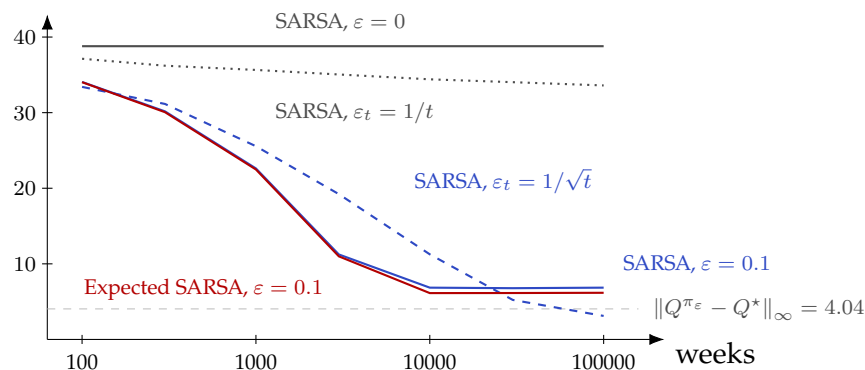
By Computer: SARSA on the Machine

The experiment.

1. Start the machine at G with $Q_0 = 0$, $\alpha = 0.1$, ties broken uniformly at random; run SARSA for 100,000 weeks: one *stream*.
2. After t weeks read the greedy policy off Q_t and the error $\|Q_t - Q^*\|_\infty = \max_{s,a} |Q_t(s,a) - Q^*(s,a)|$; record the earnings per week.
3. Repeat with 200 fresh streams; report the fraction of streams whose greedy policy is π^* , and the mean error, at $t = 100, 1,000, 10,000, 100,000$.

week $t =$	greedy policy of Q_t is π^* : share of streams				mean error $\ Q_t - Q^*\ _\infty$				earned per week
	100	1,000	10,000	100,000	100	1,000	10,000	100,000	
SARSA, $\varepsilon = 0$	0.005	0.005	0.005	0.005	38.79	38.79	38.79	38.79	0.02
SARSA, $\varepsilon = 0.1$	0.055	1.000	0.990	1.000	34.06	22.62	6.85	6.85	3.34
SARSA, $\varepsilon_t = 1/t$	0.010	0.030	0.065	0.110	37.13	35.65	34.41	33.60	0.35
SARSA, $\varepsilon_t = 1/\sqrt{t}$	0.075	1.000	1.000	1.000	33.41	25.56	11.26	3.11	3.69
Expected SARSA, $\varepsilon = 0.1$	0.015	1.000	1.000	1.000	34.03	22.48	6.13	6.17	3.35
π^* in the long run; the softened π^* ($\varepsilon = 0.1$) in the long run									3.71; 3.35

Five learners, 200 streams each. $\varepsilon = 0$ never recovers from its first repair; $\varepsilon_t = 1/t$ explores too little too soon; $\varepsilon = 0.1$ finds π^* and settles near Q^{π_ε} ; $\varepsilon_t = 1/\sqrt{t}$ goes below the fixed- ε floor, still falling at 100,000 weeks, and earns almost what π^* earns; Expected SARSA removes next-action noise from the target, while state-transition noise remains.



The error columns as curves, the week axis logarithmic. Two learners never leave the top; the constant- ε learners level off above 4.04, the distance from Q^{π_ε} to Q^* ; the shrinking ε keeps falling.

- $\varepsilon = 0$: the first repair at B is priced below 0 whenever $Q(G, \text{run})$ is still under 11.1, and never retried; 199 of 200 streams sit at B for good, earning nothing.
- $\varepsilon = 0.1$: π^* read off the table in all 200 streams at weeks 1,000 and 100,000, in 198 at 10,000 (median first week 215); the error levels at 6.85: Q^{π_ε} is 4.04 away and the constant step adds the rest.

- Earnings: 3.34 a week against 3.71 for π^* and 3.35 for the softened one; a tenth of the weeks are spent on the coin.
- Expected SARSA at week 10,000: error 6.13 against 6.85; mean distance to Q^{π_ϵ} 2.33 against 2.99, about 22% lower. Averaging helps; state-transition noise remains.
- Shrinking ϵ : with $1/t$ the coin dies too fast, 180 of 200 streams never leave B; with $1/\sqrt{t}$, π^* at every checkpoint from week 1,000, error 3.11 at 100,000 and falling, earnings 3.69.

Summary

Control without a model, on one board

- Task: from one stream whose actions the learner chooses, with no P and no r , a table whose greedy policy is π^* . Truth: Q^* run (44.30, 32.41, 26.89), repair 29.87; repair when broken.
- From V to Q : greedy w.r.t. V needs a lookahead with P and r ; greedy w.r.t. Q is $\operatorname{argmax}_a Q(s, a)$. Sampled backup $R_{t+1} + \gamma Q(S_{t+1}, A_{t+1})$, mean $(\mathcal{T}^\pi Q)(s, a)$.
- A deterministic policy never prices its other action: ϵ -greedy, 0.95 and 0.05 on the machine. Theorem, three lines: ϵ -greedy w.r.t. Q^π improves any ϵ -soft π . Fixed point: the best ϵ -soft policy, (39.82, 28.87, 25.70), 4.49 below V^* at G.
- SARSA: $\delta_t = R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t)$, $Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \delta_t$, A_{t+1} ϵ -greedy w.r.t. Q . By hand: errors 8, 4, 4, 2, -4.6, 7.4; run (9.7, 3, 0), repair (0, 0, -2.3); greedy says never repair; $\epsilon = 0$ is stuck there.
- Where it can stop: $\hat{Q} = Q^{\hat{\pi}}$ with $\hat{\pi}$ ϵ -greedy w.r.t. $\hat{Q}$; machine (40.55, 29.03, 23.13) and 25.83, off by 4.04, policy right. To reach Q^* : $\epsilon_t \rightarrow 0$ slowly, $\alpha \rightarrow 0$ (the theorem is reading).
- Expected SARSA: $\sum_a \pi(a | S_{t+1}) Q(S_{t+1}, a)$ in the target; same expected target, no next-action noise at a fixed table; greedy in the sum is Q-learning.
- By computer: $\epsilon = 0$ stuck (0.005); $\epsilon = 0.1$ finds π^* by week 1,000, error 6.85; $1/t$ stuck (0.110); $1/\sqrt{t}$ error 3.11; Expected SARSA 6.13; earnings 0.02, 3.34, 0.35, 3.69, 3.35 against 3.71.

References

- Stefan Banach. Sur les opérations dans les ensembles abstraits et leur application aux équations intégrales. *Fundamenta Mathematicae*, 3:133–181, 1922.
- Emma Brunskill. CS 234: Reinforcement learning. Stanford University course materials, 2026. URL <https://web.stanford.edu/class/cs234/>. Winter 2026 offering.
- Nan Jiang. CS 443: Reinforcement learning. University of Illinois Urbana-Champaign course materials, 2026. URL <https://nanjiang.cs.illinois.edu/cs443/>. Spring 2026 offering.
- Herbert Robbins and Sutton Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3):400–407, 1951.
- Gavin A. Rummery and Mahesan Niranjana. On-line Q-learning using connectionist systems. Technical Report CUED/F-INFENG/TR 166, Cambridge University Engineering Department, 1994.
- Satinder Singh, Tommi Jaakkola, Michael L. Littman, and Csaba Szepesvári. Convergence results for single-step on-policy reinforcement-learning algorithms. *Machine Learning*, 38(3):287–308, 2000.
- Wen Sun. CS 4/5789: Introduction to reinforcement learning. Cornell University course materials, 2025. URL https://wensun.github.io/CS4789_spring_2025.html. Spring 2025 offering.
- Richard S. Sutton. Generalization in reinforcement learning: Successful examples using sparse coarse coding. In *Advances in Neural Information Processing Systems 8*, pages 1038–1044. MIT Press, 1996.
- Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 2 edition, 2018. URL <http://incompleteideas.net/book/the-book-2nd.html>.
- Csaba Szepesvári. *Algorithms for Reinforcement Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool, 2010. URL <https://sites.ualberta.ca/~szepesva/papers/RLAlgsInMDPs.pdf>.
- Harm van Seijen, Hado van Hasselt, Shimon Whiteson, and Marco Wiering. A theoretical and empirical analysis of Expected Sarsa. In *IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning*, pages 177–184, 2009.