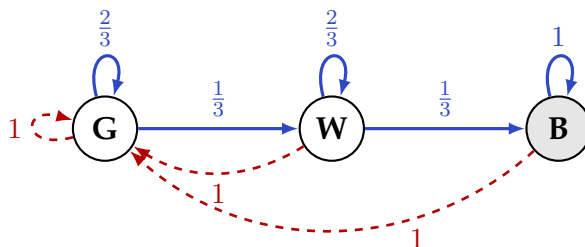


Lecture 5: Value Iteration

How to use these notes. Board content in board order: figures, equations, tables, one-line statements. Today's model is Lecture 3's machine, and Lecture 4's results are taken as known: $V^* = (44.30, 32.41, 29.87)$, $\pi^* = \text{"repair when broken"}$, the optimality equation, the operator $\mathcal{T}$, greedy extraction, and the optimality test. Reading: [Sutton and Barto \(2018, Section 4.4\)](#): value iteration, its pseudocode (which updates in place where this note computes each V_{k+1} from the whole of V_k ; its last line is today's stopping rule), and Example 4.3, the gambler, for value iteration on a problem with a hundred states. Optional: [Puterman \(1994, Sections 6.2–6.3\)](#) for the contraction, value iteration, and the stopping rule with every proof; [Agarwal et al. \(2022, Section 1.3.1\)](#) for the same in Q form (Lemma 1.10, Theorem 1.12); [Jiang \(2024, Section 2.2\)](#) for the two readings of the iterates and the residual check; and [Puterman \(1994, Chapter 4\)](#) for the finite horizon. 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.

Lecture 4 Left an Equation

The machine again: good, worn, or broken; run or repair; $\gamma = 0.9$; it starts good.



$r(s, a)$	run	repair
G	8	-10
W	4	-10
B	0	-10

Where each action leads. Solid: run. Dashed: repair, always back to G.

This week's profit, fixed once (s, a) is.

- Lecture 4 found $V^* = (44.30, 32.41, 29.87)$ and $\pi^* = (\text{run}, \text{run}, \text{repair})$ in two ways: eight linear solves, one per deterministic policy; or one guess, one solve, and the optimality test. Neither scales: there are $|\mathcal{A}|^{|\mathcal{S}|}$ guesses, and nothing in a failed test says which to try next.
- Lecture 3 turned the equation $V^\pi = r^\pi + \gamma P^\pi V^\pi$ into an iteration, $V_{k+1} = r^\pi + \gamma P^\pi V_k$, and never guessed anything. Under "never repair":

$$(8, 4, 0) \rightarrow (14, 6.4, 0) \rightarrow (18.32, 7.84, 0) \rightarrow \cdots \rightarrow (27.5, 10, 0).$$

Known from Lecture 4. For every s ,

$$V^*(s) = \max_a \left[r(s, a) + \gamma \sum_{s'} P(s' | s, a) V^*(s') \right], \quad \text{i.e.} \quad V^* = \mathcal{T}V^*$$

for the operator $(\mathcal{T}V)(s) := \max_a [r(s, a) + \gamma \sum_{s'} P(s' | s, a) V(s')]$. Nonlinear because of the max: no linear solve. Lecture 4 stated that V^* is its only solution; today proves it.

Can Lecture 3's iteration find V^* with the max inside, without ever guessing a policy, and why does it converge?

Today's targets.

1. Run value iteration by hand on the machine: Lecture 3's iteration with the max inside, from $V_0 = 0$.
2. Say what the k -th iterate is: the best expected output of k weeks, the machine switched off afterwards. On a short chain, watch a prize four weeks away take four iterations to reach the start.
3. Prove that $\mathcal{T}$ is a γ -contraction in the max norm: the one proof of the hour.
4. Draw what follows: the optimality equation has exactly one solution, the iterates converge from any start at rate γ^k , and about $\log(1/\varepsilon)/(1 - \gamma)$ iterations buy accuracy ε .
5. Turn the tolerance θ into a certificate: a change of at most θ means an error of at most $\gamma\theta/(1 - \gamma)$.
6. Read a policy off the iterates by one lookahead, and see it lock in long before the numbers settle.

Value Iteration

Value iteration (Sutton and Barto, 2018, Section 4.4)

- Input: the model (P, r, γ) ; a tolerance $\theta > 0$.
- $V_0(s) \leftarrow 0$ for all s (any start works, see below).
- For $k = 0, 1, 2, \dots$: for every $s \in \mathcal{S}$,

$$V_{k+1}(s) \leftarrow \max_a \left[r(s, a) + \gamma \sum_{s'} P(s' | s, a) V_k(s') \right], \quad \text{i.e.} \quad V_{k+1} = \mathcal{T}V_k.$$

- Stop when $\max_s |V_{k+1}(s) - V_k(s)| \leq \theta$; return V_{k+1} and the policy greedy with respect to it, $\pi(s) \leftarrow \operatorname{argmax}_a [r(s, a) + \gamma \sum_{s'} P(s' | s, a) V_{k+1}(s')]$.

- Cost: one lookahead per pair (s, a) , so $O(|\mathcal{S}|^2 |\mathcal{A}|)$ per iteration for a dense P , $|\mathcal{A}|$ times Lecture 3's; two arrays of length $|\mathcal{S}|$.
- The one change from Lecture 3's update: $\max_a$ where the average $\sum_a \pi(a | s)$ was. What θ buys: below, once the contraction is in hand.

The machine from $V_0 = 0$ at $\gamma = 0.9$, by hand:

- $V_1(s) = \max_a r(s, a) = (8, 4, 0)$: the best single week.
- V_2 at G: $\max\{8 + 0.9(\frac{2}{3} \cdot 8 + \frac{1}{3} \cdot 4), -10 + 0.9 \cdot 8\} = \max\{14, -2.8\} = 14$; at W: $\max\{4 + 0.9 \cdot \frac{2}{3} \cdot 4, -2.8\} = 6.4$; at B: $\max\{0, -2.8\} = 0$.
- Lecture 3's numbers so far, under "never repair": the repair column of the lookahead is one number, $-10 + 0.9 V_k(G) = -10$, then -2.8 , negative, so the max picked run everywhere.

Example. At B in iteration 3, run or repair?

Run keeps 0; repair pays $-10 + 0.9 V_2(G) = -10 + 12.6 = 2.6$. Repair: the max bites for the first time, and $V_3(B) = 2.6$ where Lecture 3's "never repair" kept 0. From here the two tables part.

- Iteration 3 in full. G: $\max\{8 + 0.9(\frac{2}{3} \cdot 14 + \frac{1}{3} \cdot 6.4), -10 + 0.9 \cdot 14\} = \max\{18.32, 2.6\} = 18.32$; W: $\max\{4 + 0.9 \cdot \frac{2}{3} \cdot 6.4, 2.6\} = 7.84$; B: $\max\{0, 2.6\} = 2.6$.
- Iteration 4. G: $8 + 0.9(\frac{2}{3} \cdot 18.32 + \frac{1}{3} \cdot 7.84) = 21.34$; W: $4 + 0.9(\frac{2}{3} \cdot 7.84 + \frac{1}{3} \cdot 2.6) = 9.48$; B: $-10 + 0.9 \cdot 18.32 = 6.49$.
- No policy was guessed. The max chooses an action in every state in every iteration, and it may choose differently in the next.
- Every entry climbs: run pays at least 0, so $V_1 \geq V_0 = 0$, and $\mathcal{T}$ keeps $\leq$ entry by entry. From $(40, 40, 40)$ one iteration gives $(44, 40, 36)$: G up, B down, the same limit (below).

k	$V_k(\text{G})$	$V_k(\text{W})$	$V_k(\text{B})$	$\ V_k - V^*\ _\infty$	ratio
0	0	0	0	44.30	
1	8	4	0	36.30	0.82
2	14	6.4	0	30.30	0.83
3	18.32	7.84	2.6	27.27	0.90
4	21.34	9.48	6.49	23.39	0.86
5	23.65	11.64	9.21	20.77	0.89
10	32.08	20.18	17.65	12.23	0.90
20	40.04	28.14	25.61	4.26	0.90
40	43.79	31.89	29.36	0.52	0.90
78	44.29	32.40	29.86	0.0095	0.90
∞	44.30	32.41	29.87	0	

Value iteration from $V_0 = 0$, $\gamma = 0.9$, by computer. The last column is $\|V_k - V^*\|_\infty / \|V_{k-1} - V^*\|_\infty$; it settles at exactly 0.90. Within 0.01 of V^* after 78 iterations.

Example. The same iteration at $\gamma = 0.5$: does the max ever bite?

$V_1 = (8, 4, 0)$, $V_2 = (11.33, 5.33, 0)$, $V_3 = (12.67, 5.78, 0)$, toward $(13.5, 6, 0)$.

Repair pays $-10 + 0.5 V_k(\text{G}) \leq -10 + 0.5 \cdot 13.5 = -3.25 < 0$ in every iteration. No: at $\gamma = 0.5$ value iteration is Lecture 3's evaluation of "never repair", optimal there (Lecture 4).

	Lecture 3: iterative policy evaluation	Today: value iteration
Update	$V_{k+1} = r^\pi + \gamma P^\pi V_k$: an average over π 's action	$V_{k+1} = \mathcal{T}V_k$: a max over actions
V_k from $V_0 = 0$	the value of running π for k weeks	the best value of any k -week plan
Limit	V^π	V^*
Rate	$\ V_k - V^\pi\ _\infty \leq \gamma^k \ V_0 - V^\pi\ _\infty$	the same, with the same γ
Output	numbers; the policy never moves	numbers, and by one lookahead a policy
Cost per iteration	$O(\mathcal{S} ^2)$	$O(\mathcal{S} ^2 \mathcal{A})$

What the Iterates Mean

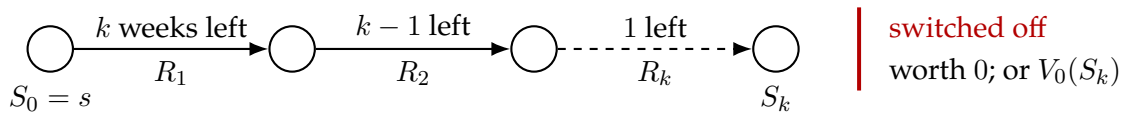
Lecture 3's V_k under a fixed π was " k weeks of π , then switched off". With the max inside, the k weeks are planned.

From $V_0 = 0$, V_k is the best k -week value. For every s ,

$$V_k(s) = \max_{\pi_0, \dots, \pi_{k-1}} \mathbb{E} \left[\sum_{t=0}^{k-1} \gamma^t R_{t+1} \mid S_0 = s \right],$$

the most any k -week plan can collect in expectation, the machine switched off afterwards; the plan may use a different rule π_t each week. Why, in two lines:

- One week: $V_1(s) = \max_a r(s, a)$.
- k to $k + 1$ weeks: take a now, earn $r(s, a)$, land in s' ; the best k weeks from s' are worth $V_k(s')$, however s' was reached (Markov). So the best $(k + 1)$ -week plan opens with the best a : $\max_a [r(s, a) + \gamma \sum_{s'} P(s' \mid s, a) V_k(s')] = (\mathcal{T}V_k)(s) = V_{k+1}(s)$.



The k -week plan behind $V_k(s)$: with j weeks left, act greedily with respect to V_{j-1} (Lecture 2's fixed-horizon convention). After the last week nothing counts, or, from a general start, the machine is sold for $V_0(S_k)$.

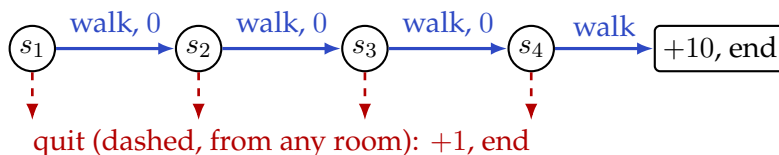
- The plan watches the clock: j weeks left, act greedily with respect to V_{j-1} . Machine: one or two weeks left, never repair; three or more, repair when broken. A repair pays only if two weeks follow it: $-10 + 0.9 \cdot 8 = -2.8$ with one, $-10 + 0.9 \cdot 14 = 2.6$ with two.
- General start V_0 : the machine is sold at week k for $V_0(S_k)$, a terminal payout. From $(40, 40, 40)$ at B: run then sell, $0.9 \cdot 40 = 36$; repair then sell, $-10 + 36 = 26$; so $V_1(B) = 36$.
- Lecture 3's k -step value: what π earns in k weeks. Today's V_k : the best any plan earns in k weeks. So Lecture 3's is never larger; under "never repair" equal for $k \leq 2$, apart at $k = 3$: 0 against 2.6 at B.

Example. Two weeks from B: which plan is best, and what is it worth? Three weeks?

Two weeks: run, run earns 0; repair, run earns $-10 + 0.9 \cdot 8 = -2.8$. Best: $0 = V_2(B)$.

Three weeks: repair, run, run earns $-10 + 0.9(8 + 0.9(\frac{2}{3} \cdot 8 + \frac{1}{3} \cdot 4)) = -10 + 0.9 \cdot 14 = 2.6 = V_3(B)$; run, run, run still 0. None of the 8^3 three-week plans does better: the iterations enumerated them, one week per iteration, last week first.

Example. A prize four weeks away. Four rooms in a row. *Walk* moves one room on for nothing, and from the last room collects 10 and ends the game; *quit* collects 1 and ends the game, from any room. $\gamma = 0.9$. From $V_0 = 0$: when does $V_k(s_1)$ see the prize, and when does the policy read off V_k walk from s_1 ?



The chain. $V^* = (7.29, 8.1, 9, 10)$: walk everywhere, since $10 \cdot 0.9^3 > 1$.

$V_1 = (1, 1, 1, 10)$: only the last room sees the prize. $V_2 = (1, 1, 9, 10)$, $V_3 = (1, 8.1, 9, 10)$, $V_4 = (7.29, 8.1, 9, 10) = V^*$: the prize walks back one room per iteration and reaches s_1 in iteration 4; then nothing is left dark, since no game lasts more than four weeks.

Policy at s_1 : off V_0, V_1, V_2 , walking pays $0.9 V_k(s_2) \leq 0.9 < 1$: quit. Off V_3 : $0.9 \cdot 8.1 = 7.29 > 1$: walk. Three iterations of the wrong move, for any γ above 0.47: a reward n weeks away needs n iterations to be seen and $n - 1$ to be acted on, however small $1 - \gamma$ is.

A deadline: backward induction (Bellman, 1957). With exactly H weeks (Lecture 2's fixed-horizon convention) the k -week reading is the whole answer. $V_h^* := V_h$ from $V_0 = 0$ is the best h -week value; with h weeks left the rule is $\pi_h^*(s) = \operatorname{argmax}_a Q_{V_{h-1}}(s, a)$; at time t there are $H - t$ weeks left, so $\pi_t = \pi_{H-t}^*$.

- Compute $V_1, \dots, V_H$, last week first, and keep the policy of every h : H iterations, exact, no tolerance, $\gamma = 1$ allowed, since nothing here is a limit.
- Machine, five weeks from B: repair with 5, 4, 3 weeks left, run with 2 and 1, worth $V_5(B) = 9.21$; "repair when broken" all five weeks 8.01 (it also repairs with two weeks left, -2.8 , and with one, -10); "never repair" 0. From G: 23.65 against 22.38 and 23.42: over five weeks never repairing beats π^* , and the plan that watches the clock beats both.
- Fold the clock into the state, (s, h) : a stationary MDP with $H|\mathcal{S}| + |\mathcal{S}|$ states in which every week lowers h by one; backward induction is value iteration on it, one layer per iteration (Agarwal et al., 2022, Chapter 1).

Convergence without the contraction (Jiang, 2024, Section 2.2). Lecture 3: any policy's value from s is its k -week value plus a tail, and $|\text{tail}| \leq \gamma^k R_{\max}/(1 - \gamma)$ ($R_{\max} = 10$ here). Two sandwich lines:

- π^* for k weeks is a k -week plan: $V^*(s) \leq V_k(s) + \gamma^k R_{\max}/(1 - \gamma)$.
 - The best k -week plan, continued anyhow, is a policy: $V_k(s) - \gamma^k R_{\max}/(1 - \gamma) \leq V^*(s)$.
- So $|V_k(s) - V^*(s)| \leq \gamma^k R_{\max}/(1 - \gamma) = 100 \cdot 0.9^k$ from $V_0 = 0$: convergence, by counting weeks. (Signed rewards: both lines need the tail bound.) The next section gets the same rate from any start, and what counting cannot give: no other solution.

Why Value Iteration Converges: $\mathcal{T}$ Is a Contraction

Two tables of values; apply $\mathcal{T}$ to each. How far apart are they afterwards?

The max norm and the theorem. For $V \in \mathbb{R}^{|S|}$, $\|V\|_\infty := \max_s |V(s)|$; the distance between two tables is their largest entry gap. The error column of the table above is $\|V_k - V^*\|_\infty$.

Theorem. For all $V, V' \in \mathbb{R}^{|S|}$,

$$\|\mathcal{T}V - \mathcal{T}V'\|_\infty \leq \gamma \|V - V'\|_\infty$$

$\mathcal{T}$ is a γ -contraction: one iteration shrinks the gap between any two tables by the factor γ at least.

Proof. Two steps.

Step 1: two maxima differ by at most the largest gap. For any $f, g : \mathcal{A} \rightarrow \mathbb{R}$,

$$\left| \max_a f(a) - \max_a g(a) \right| \leq \max_a |f(a) - g(a)|.$$

Say $\max_a f \geq \max_a g$ and let a^* attain $\max_a f$. Then

$$0 \leq \max_a f(a) - \max_a g(a) = f(a^*) - \max_a g(a) \leq f(a^*) - g(a^*) \leq \max_a |f(a) - g(a)|,$$

since $g(a^*)$ is one of the terms in $\max_a g$. The other case is the same with f and g exchanged.

Step 2: the update at one state. Fix s and write $Q_V(s, a) := r(s, a) + \gamma \sum_{s'} P(s' | s, a) V(s')$ for the lookahead table of V (Lecture 4), so $(\mathcal{T}V)(s) = \max_a Q_V(s, a)$.

$$\begin{aligned} |(\mathcal{T}V)(s) - (\mathcal{T}V')(s)| &= \left| \max_a Q_V(s, a) - \max_a Q_{V'}(s, a) \right| \\ &\leq \max_a |Q_V(s, a) - Q_{V'}(s, a)| && \text{Step 1} \\ &= \max_a \gamma \left| \sum_{s'} P(s' | s, a) (V(s') - V'(s')) \right| && \text{the rewards cancel} \\ &\leq \gamma \max_a \sum_{s'} P(s' | s, a) |V(s') - V'(s')| \\ &\leq \gamma \|V - V'\|_\infty. && \text{a row of } P \text{ averages} \end{aligned}$$

The last step: $P(\cdot | s, a)$ is nonnegative and sums to one, so the sum is an average of the gaps $|V(s') - V'(s')|$, and an average is at most the largest term. The bound holds at every s ; take the max over s . $\square$

-

Step 2: repair's row is the point mass on G , and G carries the largest gap of V_2 , 30.30: the average is its largest term.

- A worst case over every finite MDP with this γ and $R_{\max}$; the machine nearly attains it, 78 against 88, ratio exactly 0.9 in the end: under π^* the machine runs forever, so a reward k weeks ahead is missing from V_k and worth 0.9^k . Lecture 3's "never repair" drained into B: 0.6 per iteration at W.
- A better start saves iterations, not rate: from $(40, 40, 40)$ the gap is 10.13 instead of 44.30; guarantee 66, needed 59.
- The run sees the change $\|V_{k+1} - V_k\|_\infty$, never the error $\|V_k - V^*\|_\infty$: after 78 iterations, 0.001 against 0.0095. What the first number says about the second: two boxes on.

Banach's fixed-point theorem (Banach, 1922). A contraction of a complete space has exactly one fixed point, and iterating it from anywhere converges there at rate γ^k .

- Here: existence came from Lecture 4's derivation ($V^* = \mathcal{T}V^*$); the contraction adds uniqueness and the rate.
- It can do more: the iterates are Cauchy, so a fixed point exists without Lecture 4's derivation; with Lecture 4's "greedy table is optimal" argument and one monotonicity step, a second route to an optimal deterministic stationary policy, no supremum over all rules (Puterman, 1994, Section 6.2) (Agarwal et al., 2022, Theorem 1.7). Not part of this course.

Where $\gamma < 1$ is used.

- Not in the inequality: at $\gamma = 1$, $\mathcal{T}$ is *nonexpansive*: the gap between two tables does not grow, and need not shrink.
- In every line that follows from it: each divides by $1 - \gamma$.
- Lecture 3's failure mode, seen from here: at $\gamma = 1$ the iterates of "repair when broken", and value iteration's, grow by about 3.7 a week forever; no table solves $V = \mathcal{T}V$, and no theorem promised one.

Stopping: What the Change Certifies

The run watches one number, the change of the last iteration. Here is what it certifies.

Stopping rule. If an iteration changes no entry by more than θ ,

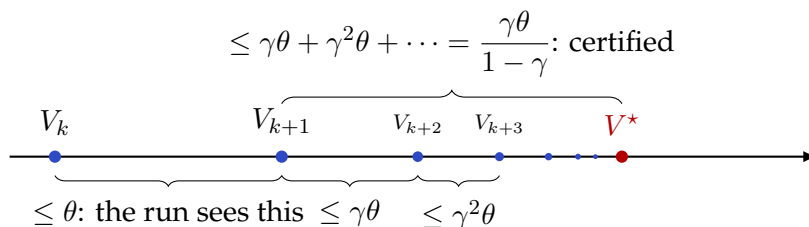
$$\|V_{k+1} - V_k\|_\infty \leq \theta \quad \implies \quad \|V_{k+1} - V^*\|_\infty \leq \frac{\gamma\theta}{1-\gamma}.$$

Why, in three lines.

- The next iteration moves at most $\gamma\theta$: $\|V_{k+2} - V_{k+1}\|_\infty = \|\mathcal{T}V_{k+1} - \mathcal{T}V_k\|_\infty \leq \gamma\|V_{k+1} - V_k\|_\infty$. The one after, at most $\gamma^2\theta$; and so on.

- The iterates end at V^* (from any start, above), so $V^* - V_{k+1}$ is the sum of all the later moves.
- Triangle inequality, then a geometric series:

$$\|V^* - V_{k+1}\|_\infty \leq \|V_{k+2} - V_{k+1}\|_\infty + \|V_{k+3} - V_{k+2}\|_\infty + \dots \leq \gamma\theta + \gamma^2\theta + \dots = \gamma\theta/(1-\gamma).$$



The remaining movement is a geometric series: every later iteration moves at most γ times the one before, so the run knows how far the limit can be without ever seeing it.

- The factor $\gamma/(1-\gamma)$ is 9 at $\gamma = 0.9$ and 99 at $\gamma = 0.99$: the tolerance is not the accuracy. To certify 0.01 the machine needs $\theta = 0.0011$; the run then stops after 78 iterations, the count above, certifying 0.0099 with a true error of 0.0095.
- The change is the *residual* of V_k : $\|V_{k+1} - V_k\|_\infty = \|\mathcal{T}V_k - V_k\|_\infty$, how badly V_k fails the optimality equation. For any table V , however it was produced (Williams and Baird, 1993):

$$\|V - V^*\|_\infty \leq \|V - \mathcal{T}V\|_\infty + \|\mathcal{T}V - \mathcal{T}V^*\|_\infty \leq \|\mathcal{T}V - V\|_\infty + \gamma\|V - V^*\|_\infty,$$

so
$$\|V - V^*\|_\infty \leq \frac{\|\mathcal{T}V - V\|_\infty}{1-\gamma},$$

a certificate computable from V and the model alone.

θ	iterations until the change is $\leq \theta$	certified $\gamma\theta/(1-\gamma)$	true error	true/certified
1	13	9	8.91	0.99
0.1	35	0.9	0.878	0.98
0.01	57	0.09	0.086	0.96
0.0011	78	0.0099	0.0095	0.96
0.001	79	0.009	0.0085	0.95

The machine from $V_0 = 0$ with the stopping rule, by computer. The certificate is nearly exact here: from iteration 15 on, the true error is exactly 9 times the last change.

Example. A run with $\theta = 0.01$ stops after 57 iterations. What is certified about V_{57} ? Is an error of 0.01 certified?

Certified: $\|V_{57} - V^*\|_\infty \leq 0.9 \cdot 0.01/0.1 = 0.09$; the truth is 0.086.

No: a tolerance of 0.01 certifies 0.09. For 0.01 the tolerance must be 0.0011, and the run must go on to iteration 78.

- A small value error guarantees a small performance loss for the greedy policy, with a bound depending on γ . It does not necessarily certify exact optimality. We will derive this performance bound later in the unit.

Why the certificate is nearly exact on the machine. From iteration 2 on the greedy policy of every iterate is π^* (next section), so both maxima in $\mathcal{TV}_{k+1} - \mathcal{TV}_k$ are attained by π^* and the change obeys $V_{k+2} - V_{k+1} = \gamma P^{\pi^*}(V_{k+1} - V_k)$, the same recursion as the error. Powers of a stochastic matrix average every vector toward a constant one, and under π^* the machine visits all three states forever, so by iteration 15 the change is the same number in every state to three decimals; on a constant vector P^{π^*} is the identity, and the remaining movement is $(\gamma + \gamma^2 + \dots)$ times the change: 9 times. The rule is attained for the reason the rate was: under π^* the machine never stops.

Reading the Policy Off the Iterates

The greedy policy of the k -th iterate. For every s ,

$$\pi_k(s) \leftarrow \operatorname{argmax}_a \left[r(s, a) + \gamma \sum_{s'} P(s' | s, a) V_k(s') \right] :$$

Lecture 4's extraction with V_k in place of V^* .

Cost: nothing extra; Q_{V_k} is the table the next iteration computes anyway.

Meaning: by the k -week reading, π_k is the best first move of a $(k + 1)$ -week plan.

k	V_k	$Q_{V_k}(s, \text{repair}), \text{ any } s$	$Q_{V_k}(\text{B}, \text{run})$	greedy w.r.t. V_k
0	(0, 0, 0)	-10	0	never repair
1	(8, 4, 0)	-2.8	0	never repair
2	(14, 6.4, 0)	2.6	0	repair when broken
3	(18.32, 7.84, 2.6)	6.49	2.34	repair when broken
4	(21.34, 9.48, 6.49)	9.21	5.84	repair when broken
78	(44.29, 32.40, 29.86)	29.86	26.88	repair when broken
∞	(44.30, 32.41, 29.87)	29.87	26.89	repair when broken

The repair column is $-10 + 0.9 V_k(\text{G})$; at G and W, run always wins. The policy is π^* from $k = 2$ on; the values need 78 iterations.

- Off V_0 or V_1 : “never repair” (Lecture 4's myopic row), worth 27.5 at G, not 44.30. Off V_2 : π^* , while $\|V_2 - V^*\|_\infty = 30.30$: the policy locks in 76 iterations ahead of the values.

- Nothing the run computes says so: the error column needs V^* , and the stopping rule certifies value accuracy without establishing exact optimality of the greedy policy. What such a policy is worth, and how to iterate on policies so that a run ends exactly: the rest of the unit.
- Two questions this leaves: does π_k reach π^* , and is it good meanwhile? Here yes and yes from $k = 2$; in general, a bound, not an iteration count.

The same iteration on Q tables. Start from $Q_0 = 0$ and iterate, one entry per pair,

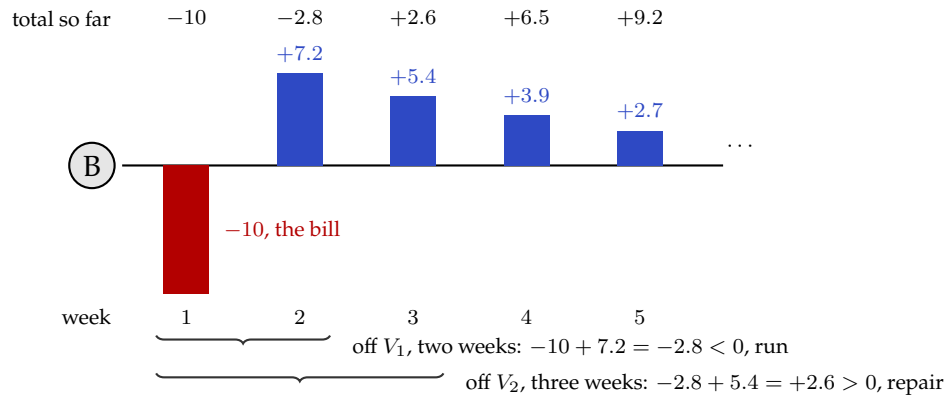
$$Q_{k+1}(s, a) \leftarrow r(s, a) + \gamma \sum_{s'} P(s' \mid s, a) \max_{a'} Q_k(s', a').$$

- With $V_k(s) := \max_a Q_k(s, a)$: $Q_{k+1} = Q_{V_k}$, the lookahead table of the V iteration. The same numbers, $|\mathcal{S}||\mathcal{A}|$ of them; the policy by a lookup, no model at that step.
- The contraction proof carries over word for word.
- Survives sampled transitions in place of P and r : Q-learning, later.

	Brute force (Lecture 4)	Guess and test (Lecture 4)	Value iteration (today)
Computes	V^π for every table π	V^π for one π , then $\mathcal{T}V^\pi$	$V_k = \mathcal{T}^k V_0$, no policy fixed
Cost	$ \mathcal{A} ^{ \mathcal{S} }$ linear solves	one solve and one lookahead per guess	$ \mathcal{S} ^2 \mathcal{A} $ per iteration, about $\log(1/\varepsilon)/(1 - \gamma)$ iterations
Answer	exact V^*, π^*	exact, if the guess passes; otherwise a failed guess	within $\gamma^k R_{\max}/(1 - \gamma)$, certified within $\gamma\theta/(1 - \gamma)$ at the stop; a policy by one lookahead
Needs	the model	the model and a good guess	the model

Failure Mode: Blind Past the Horizon

Stopped after k iterations, V_k counts weeks 1 to k ; later weeks are dark. The machine's repair shows what the dark can hide.

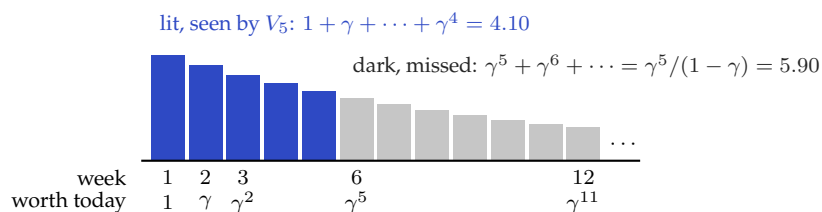


Repair at B, then run: the bill now, the payoff after, each week worth less ($\times 0.9$, and wear). Repaid in week 3. The policy read off V_k sees $k + 1$ weeks: two see a loss, three see the repayment; the rest is dark.

What V_k sees, and what it misses.

- From $V_0 = 0$: V_k counts weeks 1 to k ; the policy read off it is the first move of the best $(k + 1)$ -week plan. Later weeks: dark.
- A reward t weeks ahead counts γ^t of itself, so the whole dark part is worth at most $(\gamma^k + \gamma^{k+1} + \dots)R_{\max} = \gamma^k R_{\max} / (1 - \gamma)$: the bound from zero, above.
- Iterations needed for accuracy ε , from the bound: $\frac{1}{1-\gamma} \log \frac{R_{\max}}{(1-\gamma)\varepsilon}$, proportional to the horizon $1/(1 - \gamma)$: 2, 10, 100 weeks at $\gamma = 0.5, 0.9, 0.99$. Set by γ alone.
- The chain above: the prize was dark to $V_k(s_1)$ for $k < 4$ and to the policy for $k < 3$, at any γ . The bound counts weeks by γ ; a model can put its reward exactly n weeks out.

Example. Can the dark part be as large as the bound says? Take the simplest machine: one state, it never breaks, it earns 1 a week. V^* ? V_k ? The error?



One bar per week; each week's dollar is worth γ of the one before. $\gamma = 0.9$, $k = 5$.

$V^* = 1 + \gamma + \gamma^2 + \dots = \frac{1}{1-\gamma}$: all the bars. $V_k = 1 + \gamma + \dots + \gamma^{k-1} = \frac{1-\gamma^k}{1-\gamma}$: the lit ones.
 $V^* - V_k = \gamma^k + \gamma^{k+1} + \dots = \frac{\gamma^k}{1-\gamma}$: the dark ones. Yes: this is the bound $\gamma^k R_{\max} / (1 - \gamma)$ itself ($R_{\max} = 1$), and the theorem's $\gamma^k \|V_0 - V^*\|_\infty$ with equality at every iteration; no

better rate holds for all MDPs.

$\gamma = 0.99$: $V^* = 100$, $V_1 = 1$, $V_{100} = 63.4$; within 1 of V^* takes 459 iterations, within 0.01, 917.

γ	$\frac{1}{1-\gamma}$	iterations to $\ V_k - V^*\ _\infty \leq 0.01$ needed	bound (worst case)	$\pi_k = \pi^*$ from $k =$	π^*	$V^*(G)$
0.5	2	8	11	0	never repair	13.5
0.9	10	78	88	2	repair when broken	44.30
0.99	100	1047	1146	2	repair when broken	378.9

Back to the machine, $V_0 = 0$, by computer: the iterations scale with the horizon; the policy locks in at once.

- $\gamma = 0.99$: the policy is right at iteration 2; the values take 1047 iterations, thirteen times $\gamma = 0.9$'s 78, each $|\mathcal{S}|^2|\mathcal{A}|$ operations. The horizon is paid for in iterations.
- $\gamma = 0.5$: a two-week window; the max never bites; 8 iterations, fewer than the bound because the chain drains into B.
- The remedy is not to iterate less: the stopping rule says when the values are certified. What the policy read off an approximate table is worth, and whether iterating on policies can end exactly: next.

Two more limits.

- Size. One iteration touches every (s, a, s') . Millions of states, or a continuum: neither the table nor the iteration fits. Later: sampled transitions for the update, a function with parameters for the table.
- Exactness. Value iteration never arrives: $V_k < V^*$ for every k in the one-state example, and in general the iterates only approach V^* . Iterating on policies instead, evaluate then improve, stops after finitely many rounds with π^* exactly: policy iteration, next time. Lecture 4's failure-mode table showed one such step.

Summary

Lecture 5 on one board

- Value iteration: $V_0 = 0$; $V_{k+1} = \mathcal{T}V_k$, i.e. $V_{k+1}(s) = \max_a [r(s, a) + \gamma \sum_{s'} P(s' | s, a) V_k(s')]$; stop when no entry moves by more than θ ; read off the greedy policy. One lookahead per pair per iteration, $O(|\mathcal{S}|^2|\mathcal{A}|)$. Machine: $(8, 4, 0) \rightarrow (14, 6.4, 0) \rightarrow (18.32, 7.84, 2.6) \rightarrow (21.34, 9.48, 6.49) \rightarrow \dots \rightarrow (44.30, 32.41, 29.87)$; the max first bites at B in iteration 3 ($2.6 > 0$).
- From $V_0 = 0$, $V_k(s)$ is the best expected output of k weeks over all plans, the machine switched off afterwards; the plan acts greedily with respect to V_{j-1} with

- j weeks left. From a general V_0 , the machine is sold at week k for $V_0(S_k)$. The chain: a prize four rooms away reaches $V_k(s_1)$ at $k = 4$ and the policy at $k = 3$, $(1, 1, 1, 10) \rightarrow (1, 1, 9, 10) \rightarrow (1, 8.1, 9, 10) \rightarrow (7.29, 8.1, 9, 10)$.
- Theorem (proved): $\|\mathcal{T}V - \mathcal{T}V'\|_\infty \leq \gamma \|V - V'\|_\infty$. Step 1, $|\max f - \max g| \leq \max |f - g|$; Step 2, the rewards cancel and a row of P averages. $\gamma < 1$ unused in the proof, used in every line after it. Iteration 3 of the machine attains it: $27.27 = 0.9 \cdot 30.30$.
 - From the contraction: $V = \mathcal{T}V$ has exactly one solution (Lecture 4's test stands); $\|V_k - V^*\|_\infty \leq \gamma^k \|V_0 - V^*\|_\infty \leq \gamma^k R_{\max}/(1 - \gamma)$ from zero; $\log(R_{\max}/((1 - \gamma)\epsilon)) / \log(1/\gamma)$ iterations guarantee ϵ : 88 for the machine at 0.01, 78 needed, ratio exactly 0.9 in the end.
 - Stopping rule: $\|V_{k+1} - V_k\|_\infty \leq \theta$ certifies $\|V_{k+1} - V^*\|_\infty \leq \gamma\theta/(1 - \gamma)$; the later moves are a geometric series. The change is the residual $\|\mathcal{T}V_k - V_k\|_\infty$, and any table is within $\|\mathcal{T}V - V\|_\infty/(1 - \gamma)$ of V^* . Machine: $\theta = 0.01$ stops at iteration 57 certifying 0.09 (0.086 true); 0.01 needs $\theta = 0.0011$ and iteration 78.
 - Policy off the iterates: π_k greedy w.r.t. V_k . Machine: never repair off V_0, V_1 ; π^* from V_2 on, 76 iterations before the values settle. The rule bounds value error and hence policy loss, without certifying exact optimality.
 - Failure mode: V_k sees k weeks, the policy off it $k + 1$; the rest is dark, worth at most $\gamma^k R_{\max}/(1 - \gamma)$. Machine: a repair's bill is repaid in week 3, so off V_1 never repair (27.5), off V_2 repair. One state, reward 1: the dark part is exactly $\gamma^k/(1 - \gamma)$. Iterations scale with $1/(1 - \gamma)$: 8, 78, 1047 at $\gamma = 0.5, 0.9, 0.99$.
 - Next: iterate on policies instead of values. Evaluate a policy exactly, take one greedy step, repeat: why every step improves, and why the run ends exactly at π^* (policy iteration).

References

- Alekh Agarwal, Nan Jiang, Sham M. Kakade, and Wen Sun. Reinforcement learning: Theory and algorithms. Book draft, 2022. URL <https://rltheorybook.github.io/>.
- Stefan Banach. Sur les opérations dans les ensembles abstraits et leur application aux équations intégrales. *Fundamenta Mathematicae*, 3:133–181, 1922.
- Richard Bellman. *Dynamic Programming*. Princeton University Press, 1957.
- Emma Brunskill. CS 234: Reinforcement learning. Stanford University course materials, 2026. URL <https://web.stanford.edu/class/cs234/>. Winter 2026 offering.
- Nan Jiang. MDP preliminaries. Lecture notes, University of Illinois Urbana-Champaign, 2024. URL <https://nanjiang.cs.illinois.edu/files/cs542f22/note1.pdf>. Dated October 5, 2019; updated September 27, 2024. Accessed 2026-09-13.
- Nan Jiang. CS 443: Reinforcement learning. University of Illinois Urbana-Champaign course materials, 2026. URL <https://nanjiang.cs.illinois.edu/cs443/>. Spring 2026 offering.
- Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 1994.
- 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 and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 2 edition, 2018. URL <http://incompleteideas.net/book/the-book-2nd.html>.
- Ronald J. Williams and Leemon C. Baird. Tight performance bounds on greedy policies based on imperfect value functions. Technical Report NU-CCS-93-14, College of Computer Science, Northeastern University, 1993.