ECE276B: Planning & Learning in Robotics
Lecture 10: Bellman Equations

Instructor:

Nikolay Atanasov: natanasov@ucsd.edu

Teaching Assistants:

Zhichao Li: zhl355@eng.ucsd.edu

Ehsan Zobeidi: ezobeidi@eng.ucsd.edu

Ibrahim Akbar: iakbar@eng.ucsd.edu

UC San Diego

JACOBS SCHOOL OF ENGINEERING

Electrical and Computer Engineering

Infinite-Horizon Stochastic Optimal Control

- **Discounted Problem**:

$$
\begin{aligned}
V^*(x) = \min_{\pi} \quad & V^{\pi}(x) := \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \ell(x_t, \pi(x_t)) \;\middle|\; x_0 = x\right] \\
\text{s.t.} \quad & x_{t+1} \sim p_f(\cdot \mid x_t, \pi(x_t)), \\
& x_t \in \mathcal{X}, \\
& \pi(x_t) \in \mathcal{U}(x_t)
\end{aligned}
$$

- The optimal cost of the Discounted problem satisfies the **Bellman Equation** via the equivalence to the SSP problem:

$$
V^*(x) = \min_{u \in \mathcal{U}(x)} \left(\ell(x, u) + \gamma \sum_{x' \in \mathcal{X}} p_f(x' \mid x, u) V^*(x') \right), \quad \forall x \in \mathcal{X}
$$

- There exist several methods to solve the Bellman Equation for the Discounted and SSP problems:
  - Value Iteration (VI)
  - Policy Iteration (PI)
  - Linear Programming (LP)

Value Iteration (VI)

- Applies the Dynamic Programming recursion with an arbitrary initialization $V_0(x)$ to compute $V^*(x)$ for $x \in \mathcal{X}$
- VI requires an infinite iterations for $V_k(x)$ to converge to $V^*(x)$. In practice, define a threshold for $|V_{k+1}(x) - V_k(x)|$ for all $x \in \mathcal{X}$
- **SSP**:

$$V_{k+1}(x) = \min_{u \in \tilde{\mathcal{U}}(x)} \Big[\tilde{\ell}(x,u) + \sum_{x \in \tilde{\mathcal{X}} \setminus \{0\}} \tilde{p}(x' \mid x, u) V_k(x') \Big], \qquad \forall x \in \tilde{\mathcal{X}} \setminus \{0\}$$

- **Discounted Problem**:

$$V_{k+1}(x) = \min_{u \in \mathcal{U}(x)} \Big[\ell(x,u) + \gamma \sum_{x \in \mathcal{X}} p(x' \mid x, u) V_k(x') \Big], \qquad \forall x \in \mathcal{X}$$

Gauss-Seidel Value Iteration

- A regular VI implementation stores the values from a previous iteration and updates them for all states simultaneously:

$$\bar{V}(x) \leftarrow \min_{u \in \mathcal{U}(x)} \left(\ell(x,u) + \gamma \sum_{x' \in \mathcal{X}} p_f(x' \mid x, u) V(x') \right), \qquad \forall x \in \mathcal{X}$$

$$V(x) \leftarrow \bar{V}(x), \qquad \forall x \in \mathcal{X}$$

- **Gauss-Seidel Value Iteration** updates the values in place:

$$V(x) \leftarrow \min_{u \in \mathcal{U}(x)} \left(\ell(x,u) + \gamma \sum_{x' \in \mathcal{X}} p_f(x' \mid x, u) V(x') \right), \qquad \forall x \in \mathcal{X}$$

- Gauss-Seidel VI often leads to faster convergence and requires less memory than VI

Policy Evaluation

- The VI algorithm computes the optimal value function $V^*(x)$ for every state $x \in \mathcal{X}$
- The VI algorithm is the infinite-horizon equivalent of the DP algorithm
- Instead of the optimal value function $V^*(x)$, is it possible to compute the value function $V^\pi(x)$ for a given policy π?

Policy Evaluation Theorem (Discounted Problem)

The cost vector V^π for policy π is the unique solution of:

$$V^\pi(x) = \ell(x, \pi(x)) + \gamma \sum_{x' \in \mathcal{X}} p_f(x' \mid x, \pi(x)) V^\pi(x'), \qquad \forall x \in \mathcal{X}$$

Furthermore, given any initial conditions V_0, the sequence V_k generated by the recursion below converges to V^π:

$$V_{k+1}(x) = \ell(x, \pi(x)) + \gamma \sum_{x' \in \mathcal{X}} p_f(x' \mid x, \pi(x)) V_k(x'), \qquad \forall x \in \mathcal{X}$$

Policy Evaluation

Policy Evaluation Theorem (SSP)

Under the termination state assumption, the cost vector $V^\pi(1), \ldots, V^\pi(n)$ for any proper policy π is the unique solution of:

$$V^\pi(x) = \ell(x, \pi(x)) + \sum_{x' \in \tilde{\mathcal{X}} \setminus \{0\}} \tilde{p}_f(x' \mid x, \pi(x)) V^\pi(x'). \qquad \forall x \in \tilde{\mathcal{X}} \setminus \{0\}$$

Furthermore, given any initial conditions V_0, the sequence V_k generated by the recursion below converges to V^π:

$$V_{k+1}(x) = \ell(x, \pi(x)) + \sum_{x' \in \tilde{\mathcal{X}} \setminus \{0\}} \tilde{p}_f(x' \mid x, \pi(x)) V_k(x'), \qquad \forall x \in \tilde{\mathcal{X}} \setminus \{0\}$$

- **Proof**: This is a special case of the Bellman Equation Theorem (SSP). Consider a modified problem, where the only allowable control at state x is $\pi(x)$. Since the proper policy π is the only policy under consideration, the proper policy assumption is satisfied and the arg min over $u \in \mathcal{U}(x)$ has to be $\pi(x)$.

Policy Evaluation as a Linear System (SSP)

- The Policy Evaluation Theorem requires solving a linear system of equations:

$$\mathbf{v} = \boldsymbol{\ell} + \tilde{P}\mathbf{v} \qquad \Rightarrow \qquad (I - \tilde{P})\mathbf{v} = \boldsymbol{\ell}$$

  where $\mathbf{v}_i := V^\pi(i)$, $\boldsymbol{\ell}_i := \ell(i, \pi(i))$, $\tilde{P}_{ij} := \tilde{p}_f(j \mid i, \pi(i))$ for $i, j = 1, \ldots, n$.

- There exists a unique solution for $\mathbf{v}$, iff $(I - \tilde{P})$ is invertible. This is guaranteed as long as π is a proper policy.

- **Proof**: $(I - \tilde{P})$ is invertible iff $\tilde{P}$ does not have eigenvalues at 1. By the Chapman-Kolmogorov equation, $[\tilde{P}^T]_{ij} = \mathbb{P}(x_T = j \mid x_0 = i)$ and since π is proper, $[\tilde{P}^T]_{ij} \to 0$ as $T \to \infty$ for all $i, j \in \tilde{\mathcal{X}} \setminus \{0\}$. Since $\tilde{P}^T$ vanishes as $T \to \infty$ all eigenvalues of $\tilde{P}$ must have modulus less than 1 and therefore $(I - \tilde{P})$ exists.

Policy Evaluation as a Linear System (SSP)

- The Policy Evaluation Thm is an iterative solution to $(I - \tilde{P})\mathbf{v} = \boldsymbol{\ell}$:

$$
\begin{aligned}
\mathbf{v}_1 &= \boldsymbol{\ell} + \tilde{P}\mathbf{v}_0 \\
\mathbf{v}_2 &= \boldsymbol{\ell} + \tilde{P}\mathbf{v}_1 = \boldsymbol{\ell} + \tilde{P}\boldsymbol{\ell} + \tilde{P}^2\mathbf{v}_0 \\
&\vdots \\
\mathbf{v}_T &= (I + \tilde{P} + \tilde{P}^2 + \tilde{P}^3 + \ldots + \tilde{P}^{T-1})\boldsymbol{\ell} + \tilde{P}^T\mathbf{v}_0 \\
&\vdots \\
\mathbf{v}_\infty &\to (I - \tilde{P})^{-1}\boldsymbol{\ell}
\end{aligned}
$$

Policy Evaluation as a Linear System (Summary)

- The linear system view of the Policy Evaluation Theorem can be extended to the Discounted problem through the SSP equivalence and subsequently to the finite-horizon setting

- Let $\mathbf{v}_i := V^\pi(i)$, $\boldsymbol{\ell}_i := \ell(i, \pi(i))$, $P_{ij} := p_f(j \mid i, \pi(i))$ for $i, j = 1, \ldots, n$

- **SSP (First Exit)**: Let $\mathcal{T} \subseteq \mathcal{X}$ be the set of terminal states with terminal costs $\mathbf{q}$ and $\mathcal{N} \subseteq \mathcal{X}$ be the set of nonterminal states. The value of policy π is:

$$(I - P_{\mathcal{N}\mathcal{N}})\,\mathbf{v}_{\mathcal{N}} = \boldsymbol{\ell} + P_{\mathcal{N}\mathcal{T}}\mathbf{q}$$

- **Discounted Problem**: $(I - \gamma P)\mathbf{v} = \boldsymbol{\ell}$
    - The matrix P has eigenvalues with modulus ≤ 1. All eigenvalues of γP have modulus < 1, so $(\gamma P)^T \to 0$ as $T \to \infty$ and $(I - \gamma P)^{-1}$ exists.

- **Finite Horizon**: $\mathbf{v}_t = \boldsymbol{\ell}_t + P_t \mathbf{v}_{t+1}$ starting from $\mathbf{v}_T = \mathbf{q}$

Policy Iteration (PI)

- An alternative to VI for computing $V^*(x)$, which iterates over policies instead of values
- **SSP**: repeat until $V^{\pi'}(x) = V^{\pi}(x)$ for all $x \in \tilde{\mathcal{X}} \setminus \{0\}$:
  1. **Policy Evaluation**: given a policy π, compute V^{π}:
  $$V^{\pi}(x) = \tilde{\ell}(x, \pi(x)) + \sum_{x' \in \tilde{\mathcal{X}} \setminus \{0\}} \tilde{p}_f(x' \mid x, \pi(x)) V^{\pi}(x'), \qquad \forall x \in \tilde{\mathcal{X}} \setminus \{0\}$$
  2. **Policy Improvement**: given V^{π}, obtain a new stationary policy π':
  $$\pi'(x) = \arg\min_{u \in \tilde{\mathcal{U}}(x)} \Big[\tilde{\ell}(x, u) + \sum_{x' \in \tilde{\mathcal{X}} \setminus \{0\}} \tilde{p}_f(x' \mid x, u) V^{\pi}(x')\Big], \qquad \forall x \in \tilde{\mathcal{X}} \setminus \{0\}$$
- **Discounted Problem**: repeat until $V^{\pi'}(x) = V^{\pi}(x)$ for all $x \in \mathcal{X}$:
  1. **Policy Evaluation**: given a policy π, compute V^{π}:
  $$V^{\pi}(x) = \ell(x, \pi(x)) + \gamma \sum_{x' \in \mathcal{X}} p_f(x' \mid x, \pi(x)) V^{\pi}(x'), \qquad \forall x \in \mathcal{X}$$
  2. **Policy Improvement**: given V^{π}, obtain a new stationary policy π':
  $$\pi'(x) = \arg\min_{u \in \mathcal{U}(x)} \Big[\ell(x, u) + \gamma \sum_{x' \in \mathcal{X}} p_f(x' \mid x, u) V^{\pi}(x')\Big], \qquad \forall x \in \mathcal{X}$$

Policy Improvement Theorem

Let π and π' be deterministic policies such that $V^\pi(x) \geq Q^\pi(x, \pi'(x))$ for all $x \in \mathcal{X}$. Then, π' is at least as good as π, i.e., $V^\pi(x) \geq V^{\pi'}(x)$ for all $x \in \mathcal{X}$

- **Proof**:

$$
\begin{aligned}
V^\pi(x) &\geq Q^\pi(x, \pi'(x)) = \ell(x, \pi'(x)) + \gamma \mathbb{E}_{x' \sim p_f(\cdot x, \pi'(x))} \left[V^\pi(x') \right] \\
&\geq \ell(x, \pi'(x)) + \gamma \mathbb{E}_{x' \sim p_f(\cdot x, \pi'(x))} \left[Q^\pi(x', \pi'(x')) \right] \\
&= \ell(x, \pi'(x)) + \gamma \mathbb{E}_{x' \sim p_f(\cdot x, \pi'(x))} \left\{ \ell(x', \pi'(x')) + \gamma \mathbb{E}_{x'' \sim p_f(\cdot x', \pi'(x'))} V^\pi(x'') \right\} \\
&\geq \cdots \geq \mathbb{E} \left[\sum_{t=0}^{\infty} \gamma^t \ell(x_t, \pi'(x_t)) \middle| x_0 = x \right] = V^{\pi'}(x)
\end{aligned}
$$

Theorem: Optimality of PI

Suppose that:

- $\gamma < 1$ (Discounted Problem)
- there exists a termination state and a proper policy (SSP)

Then, the Policy Iteration algorithm converges to an optimal policy after a finite number of steps.

Proof of Optimality of PI (SSP)

- Let π be a proper policy with value V^π obtained from the Policy Evaluation step.
- Let π' be the policy obtained from the Policy Improvement step.
- By definition of the Policy Improvement step: $V^\pi(x) \geq Q^\pi(x, \pi'(x))$ for all $x \in \tilde{\mathcal{X}} \setminus \{0\}$
- By the Policy Improvement Thm, $V^\pi(x) \geq V^{\pi'}(x)$ for all $x \in \tilde{\mathcal{X}} \setminus \{0\}$
- Since π is proper, $V^\pi(x) < \infty$ for all $x \in \tilde{\mathcal{X}}$, and hence π' is proper
- Since π' is proper, the Policy Evaluation step has a unique solution $V^{\pi'}$
- Since the number of stationary policies is finite, eventually $V^\pi = V^{\pi'}$ after a finite number of steps.
- Once V^π has converged, it follows from the Policy Improvement step:

$$V^{\pi'}(x) = V^\pi(x) = \min_{u \in \tilde{\mathcal{U}}(x)} \left(\tilde{\ell}(x, u) + \sum_{x' \in \tilde{\mathcal{X}} \setminus \{0\}} \tilde{p}_f(x' \mid x, u) V^\pi(x') \right), \quad x \in \tilde{\mathcal{X}} \setminus \{0\}$$

- Since this is the Bellman Equation for the SSP problem, we have converged to an optimal policy $\pi^* = \pi$ with optimal cost $V^* = V^\pi$.

Comparison between VI and PI

- PI and VI actually have a lot in common
- Rewrite VI as follows:
  2. **Policy Improvement**: Given $V_k(x)$ obtain a stationary policy:

$$\pi(x) = \arg\min_{u \in \mathcal{U}(x)} \Big[\ell(x, u) + \gamma \sum_{x' \in \mathcal{X}} p_f(x' \mid x, u) V_k(x') \Big], \qquad \forall x \in \mathcal{X}$$

  1. **Value Update**: Given $\pi(x)$ and $V_k(x)$, compute

$$V_{k+1}(x) = \ell(x, \pi(x)) + \gamma \sum_{x' \in \mathcal{X}} p_f(x' \mid x, \pi(x)) V_k(x'), \qquad \forall x \in \mathcal{X}$$

- The Value Update step of VI is an iterative solution to the linear system of equations in the Policy Evaluation Theorem
- PI solves Policy Evaluation equation, which is equivalent to running the Value Update step of VI an infinite number of times!

Comparison between VI and PI

- **Complexity of VI per Iteration**: $O(|\mathcal{X}|^2|\mathcal{U}|)$: evaluating the expectation (i.e., sum over x') requires $|\mathcal{X}|$ operations and there are $|\mathcal{X}|$ minimizations over $|\mathcal{U}|$ possible control inputs.
- **Complexity of PI per Iteration**: $O(|\mathcal{X}|^2\left(|\mathcal{X}| + |\mathcal{U}|\right))$: the Policy Evaluation step requires solving a system of $|\mathcal{X}|$ equations in $|\mathcal{X}|$ unknowns ($O(|\mathcal{X}|^3)$), while the Policy Improvement step has the same complexity as one iteration of VI.
- PI is more computationally expensive than VI
- Theoretically it takes an infinite number of iterations for VI to converge
- PI converges in $|\mathcal{U}|^{|\mathcal{X}|}$ iterations (all possible policies) in the worst case

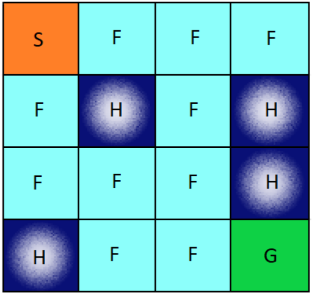
Generalized Policy Iteration

- Assuming that the Value Update and Policy Improvement steps are executed an infinite number of times for all states, all combinations of the following converge:
  - Any number of Value Update steps in between Policy Improvement steps
  - Any number of states updated at each Value Update step
  - Any number of states updated at each Policy Improvement step

Example: Frozen Lake Problem

- Winter is here.
- You and your friends were tossing around a frisbee at the park when you made a wild throw that left the frisbee out in the middle of the lake.
- The water is mostly frozen, but there are a few holes where the ice has melted.
- If you step into one of those holes, you'll fall into the freezing water.
- At this time, there's an international frisbee shortage, so it's absolutely imperative that you navigate across the lake and retrieve the disc.
- However, the ice is slippery, so you won't always move in the direction you intend.

Example: Frozen Lake Problem



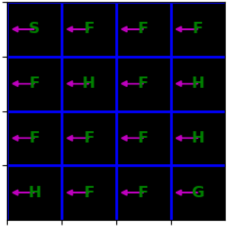
- S : starting point, safe
- F : frozen surface, safe
- H : hole, fall to your doom
- G : goal, where the frisbee is located
- $\mathcal{X} = \{0, 1, \ldots, 15\}$
- $\mathcal{U}(x) = \{\text{Left}(0), \text{Down}(1), \text{Right}(2), \text{Up}(3)\}$
- You receive a reward of 1 if you reach the goal, and zero otherwise

- A requested action $u \in \mathcal{U}(x)$ succeeds 80% of the time. A neighboring action is executed in the other 50% of the time due to slip:

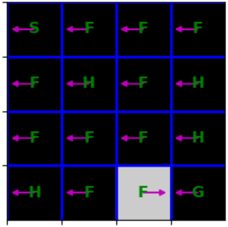
$$x' \mid x = 9, u = 1 = \begin{cases} 13, & \text{with prob. } 0.8 \\ 8, & \text{with prob. } 0.1 \\ 10, & \text{with prob. } 0.1 \end{cases}$$

- The state remains unchanged if a control leads outside of the map
- An episode ends when you reach the goal or fall in a hole.

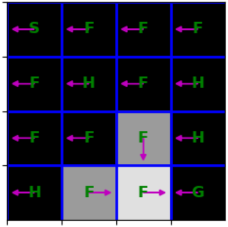
Value Iteration on Frozen Lake



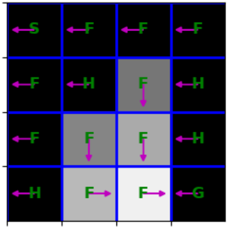
(a) $t = 0$



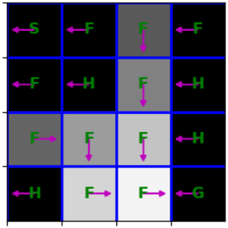
(b) $t = 1$



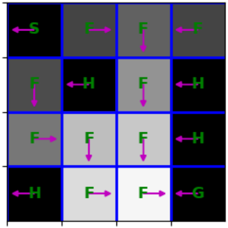
(c) $t = 2$



(d) $t = 3$



(e) $t = 4$

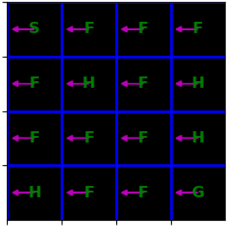


(f) $t = 5$

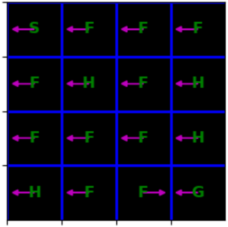
Value Iteration on Frozen Lake

Iteration	$\max_x \lvert V_{t+1}(x) - V_t(x) \rvert$	**# changed actions**	$V(0)$
0	0.80000	0	0.000
1	0.60800	1	0.000
2	0.51984	2	0.000
3	0.39508	2	0.000
4	0.30026	2	0.000
5	0.25355	2	0.254
6	0.10478	1	0.345
7	0.09657	0	0.442
8	0.03656	0	0.478
9	0.02772	0	0.506
10	0.01111	0	0.517
11	0.00735	0	0.524
12	0.00310	0	0.527
13	0.00190	0	0.529
14	0.00083	0	0.530
15	0.00049	0	0.531
16	0.00022	0	0.531

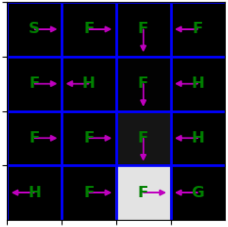
Policy Iteration on Frozen Lake



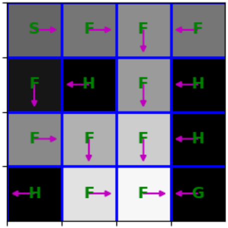
(a) $t = 0$



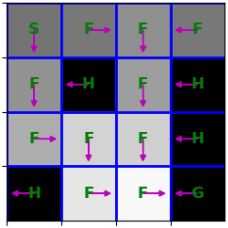
(b) $t = 1$



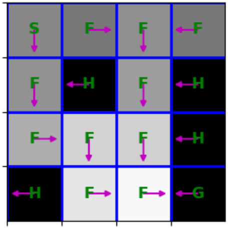
(c) $t = 2$



(d) $t = 3$



(e) $t = 4$

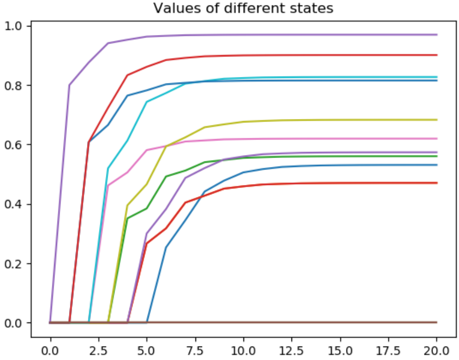


(f) $t = 5$

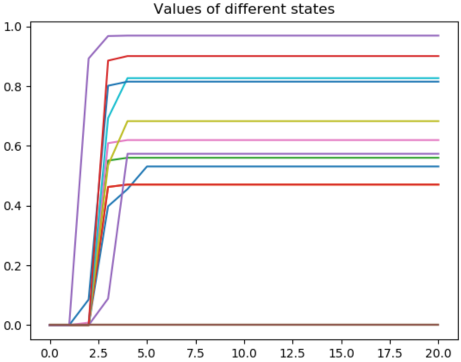
Policy Iteration on Frozen Lake

Iteration	$\max_x \lvert V_{t+1}(x) - V_t(x) \rvert$	**# changed actions**	$V(0)$
0	0.00000	0	0.000
1	0.89296	1	0.000
2	0.88580	9	0.398
3	0.48504	2	0.455
4	0.07573	1	0.531
5	0.00000	0	0.531
6	0.00000	0	0.531
7	0.00000	0	0.531
8	0.00000	0	0.531
9	0.00000	0	0.531
10	0.00000	0	0.531
11	0.00000	0	0.531
12	0.00000	0	0.531
13	0.00000	0	0.531
14	0.00000	0	0.531
15	0.00000	0	0.531
16	0.00000	0	0.531
17	0.00000	0	0.531

Value Iteration vs Policy Iteration

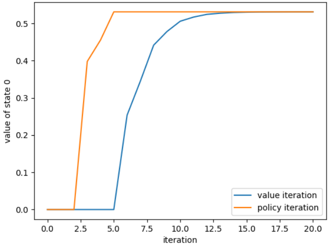


(a) VI

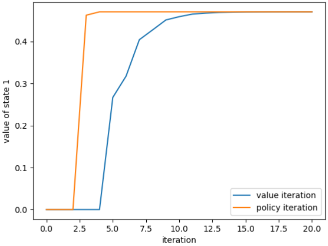


(b) PI

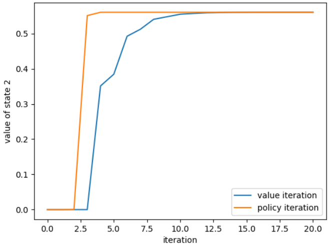
Value Iteration vs Policy Iteration



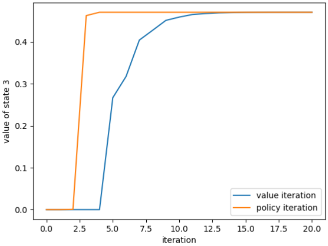
(a) State 0



(b) State 1



(c) State 2



(d) State 3

Example: 100 Games of Rock-Paper-Scissors (POMDP)

- Planning horizon: $T = 100$, $\gamma = 1$
- State:
  - score differential $s \in \mathcal{S} := \{-100, \ldots, 100\}$ (observable)
  - opponent's preference $y \in \mathcal{Y} := \{R, P, S\}$ (unobservable)
- Control: $u \in \mathcal{U} := \{R, P, S\}$
- Cost: $\tilde{\ell}(s, y, u) \equiv 0$, $\tilde{\mathfrak{q}}(s, y) = -s$
- Motion model:
  - $p_f(s' \mid s, y = R, u = R) = \begin{cases} 0.5 & \text{if } s' = s \\ 0.25 & \text{if } s' = s + 1 \\ 0.25 & \text{if } s' = s - 1 \end{cases}$
  - $p_f(s' \mid s, y = R, u = P) = \begin{cases} 0.5 & \text{if } s' = s + 1 \\ 0.25 & \text{if } s' = s \\ 0.25 & \text{if } s' = s - 1 \end{cases}$
  - ...
- Observation: $z \in \mathcal{Z} := \{R, P, S\}$
- Observation model: $p_h(z \mid y) = \begin{cases} 0.5 & \text{if } y = z \\ 0.25 & \text{otherwise} \end{cases}$

Example: 100 Games of Rock-Paper-Scissors (MDP)

- The probability mass function b_t of y_t is a sufficient statistic for y_t
- State:
  - score differential $s \in \mathcal{S} := \{-100, \ldots, 100\}$ (observable)
  - preference pmf $b \in \mathcal{B} = \mathcal{P}(\mathcal{Y}) := \{p \in [0,1]^3 \mid \mathbf{1}^T p = 1\}$ (observable)
- Control: $u \in \mathcal{U} := \{R, P, S\}$
- Cost: $\ell(s, b, u) = \int \tilde{\ell}(s, y, u) b(y) dy = 0$, $\mathfrak{q}(s, b) = \int \tilde{\mathfrak{q}}(s, y) b(y) dy = -s$
- Let $\mathbf{w}(z) := \begin{bmatrix} p_h(z \mid y = R) \\ p_h(z \mid y = P) \\ p_h(z \mid y = S) \end{bmatrix}$ be the vector of observation likelihoods
- Motion model for the preference pmf (Bayes Filter):

$$b_{t+1} \mid b_t = \begin{cases} \frac{\mathbf{w}(S) \odot b_t}{\mathbf{w}(S)^T b_t} & \text{w.p.} \quad \mathbf{w}(S)^T b_t \\ \frac{\mathbf{w}(R) \odot b_t}{\mathbf{w}(R)^T b_t} & \text{w.p.} \quad \mathbf{w}(R)^T b_t \\ \frac{\mathbf{w}(P) \odot b_t}{\mathbf{w}(P)^T b_t} & \text{w.p.} \quad \mathbf{w}(P)^T b_t \end{cases}$$

$\odot$ = elementwise multiplication

Example: 100 Games of Rock-Paper-Scissors (MDP)

- Motion model for the score differential:

$$s_{t+1} \mid s_t, R = \begin{cases} s_t + 1 & \text{w.p. } \mathbf{w}(S)^T b_t \\ s_t & \text{w.p. } \mathbf{w}(R)^T b_t \\ s_t - 1 & \text{w.p. } \mathbf{w}(P)^T b_t \end{cases} \quad s_{t+1} \mid s_t, P = \begin{cases} s_t - 1 & \text{w.p. } \mathbf{w}(S)^T b_t \\ s_t + 1 & \text{w.p. } \mathbf{w}(R)^T b_t \\ s_t & \text{w.p. } \mathbf{w}(P)^T b_t \end{cases}$$

$$s_{t+1} \mid s_t, S = \begin{cases} s_t & \text{w.p. } \mathbf{w}(S)^T b_t \\ s_t - 1 & \text{w.p. } \mathbf{w}(R)^T b_t \\ s_t + 1 & \text{w.p. } \mathbf{w}(P)^T b_t \end{cases}$$

- Discretize $\mathcal{B}$ into a finite set $\mathcal{B}_d$ of pmfs

- Apply the Dynamic Programming algorithm:
  - $V_{100}(s, b) = -s, \; \forall s \in \mathcal{S}, b \in \mathcal{B}_d$
  - $V_{99}(s, b) = \min_{u \in \{R,P,S\}} \sum_{s' \in \mathcal{S}, b' \in \mathcal{B}_d} V_{100}(s', b') p_f(s', b' \mid s, b, u), \; \forall s \in \mathcal{S}, b \in \mathcal{B}_d$
  - ...

Linear Programming Solution to the Bellman Equation

- Suppose we initialize VI with a vector V_0 that satisfies a relaxed Bellman Equation:

$$V_0(x) \le \min_{u \in \mathcal{U}(x)} \left(\ell(x,u) + \gamma \sum_{x' \in \mathcal{X}} p_f(x' \mid x, u) V_0(x') \right), \qquad \forall x \in \mathcal{X}$$

- Applying VI to V_0 leads to:

$$\begin{aligned} V_1(x) &= \min_{u \in \mathcal{U}(x)} \left(\ell(x,u) + \gamma \sum_{x' \in \mathcal{X}} p_f(x' \mid x, u) V_0(x') \right) \ge V_0(x), \quad \forall x \in \mathcal{X} \\ V_2(x) &= \min_{u \in \mathcal{U}(x)} \left(\ell(x,u) + \gamma \sum_{x' \in \mathcal{X}} p_f(x' \mid x, u) V_1(x') \right) \\ &\ge \min_{u \in \mathcal{U}(x)} \left(\ell(x,u) + \gamma \sum_{x' \in \mathcal{X}} p_f(x' \mid x, u) V_0(x') \right) = V_1(x), \quad \forall x \in \mathcal{X} \end{aligned}$$

Linear Programming Solution to the Bellman Equation

- The above shows that $V_{k+1}(x) \geq V_k(x)$ for all k and $x \in \mathcal{X}$
- Since VI guarantees that $V_k(x) \to V^*(x)$ as $k \to \infty$ we also have:

$$V^*(x) \geq V_0(x), \quad \forall x \in \mathcal{X} \quad \Rightarrow \quad \sum_{x \in \mathcal{X}} w(x) V^*(x) \geq \sum_{x \in \mathcal{X}} w(x) V_0(x)$$

  for any $w(x) > 0$ for all $x \in \mathcal{X}$.

- The above holds for **any** V_0 that satisfies:

$$V_0(x) \leq \min_{u \in \mathcal{U}(x)} \left(\ell(x, u) + \gamma \sum_{x' \in \mathcal{X}} p_f(x' \mid x, u) V_0(x') \right), \qquad \forall x \in \mathcal{X}$$

- Note that V^* also satisfies this condition with equality (Bellman Equation) and hence is the maximal V_0 (at each state) that satisfies the condition.

Linear Programming Solution to the Bellman Equation

LP Solution to the Bellman Equation

The solution V^* to the linear program (with $w(x) > 0$):

$$\max_{V} \sum_{x \in \mathcal{X}} w(x) V(x)$$

$$\text{s.t.} \quad V(x) \leq \left(\ell(x,u) + \gamma \sum_{x' \in \mathcal{X}} p_f(x' \mid x,u) V(x') \right), \qquad \forall u \in \mathcal{U}(x), \forall x \in \mathcal{X}$$

also solves the Bellman Equation to yield the optimal value function for a discounted infinite-horizon finite-state stochastic optimal control problem.

- An equivalent result holds for the SSP.

LP Solution to the BE (Proof)

- Let J^* be the solution to the linear program so that:

$$J^*(x) \leq \left(\ell(x,u) + \gamma \sum_{x' \in \mathcal{X}} p_f(x' \mid x, u) J^*(x')\right), \qquad \forall u \in \mathcal{U}(x), \forall x \in \mathcal{X}$$

- Since J^* is feasible, it satisfies $J^*(x) \leq V^*(x)$ for all $x \in \mathcal{X}$
- By contradiction, suppose that $J^* \neq V^*$. Then, there exists a state $y \in \mathcal{X}$ such that:

$$J^*(y) < V^*(y) \quad \Rightarrow \quad \sum_{x \in \mathcal{X}} w(x) J^*(x) < \sum_{x \in \mathcal{X}} w(x) V^*(x)$$

for any positive $w(x)$ but since V^* solves the Bellman Equation:

$$V^*(x) \leq \left(\ell(x,u) + \gamma \sum_{x' \in \mathcal{X}} p_f(x' \mid x, u) V^*(j)\right), \qquad \forall u \in \mathcal{U}(x), \forall x \in \mathcal{X}$$

- Thus, V^* is feasible and has lower cost that J^*, which is a contradiction.

Bellman Equations (Summary)

Finite Horizon Formulation

- Trajectories terminate at $T < \infty$

$$\min_{\pi} V_{\tau}^{\pi}(x) = \mathbb{E}\left[\sum_{t=\tau}^{T-1} \ell_t(x_t, \pi_t(x_t)) + \mathfrak{q}(x_T) \,\middle|\, x_\tau = x\right]$$

- The optimal value $V_t^*(x)$ can be found with a single backward pass through time, initialized from $V_T^*(x) = \mathfrak{q}(x)$ and following the recursion:

Bellman Equations (Finite Horizon Problem)

Hamiltonian:	$H_t[x, u, V(\cdot)] = \ell_t(x, u) + \mathbb{E}_{x' \sim p_f(\cdot \mid x, u)} V(x')$
Policy Evaluation:	$V_t^{\pi}(x) = H_t[x, \pi_t(x), V_{t+1}^{\pi}(\cdot)]$
Bellman Equation:	$V_t^*(x) = \min\limits_{u \in \mathcal{U}(x)} H_t[x, u, V_{t+1}^*(\cdot)]$
Optimal Policy:	$\pi_t^*(x) = \arg\min\limits_{u \in \mathcal{U}(x)} H_t[x, u, V_{t+1}^*(\cdot)]$

First Exit (SSP) Formulation

- Trajectories terminate at T_{first}, when a goal state $x \in \mathcal{T} \subseteq \mathcal{X}$ is reached:

$$\min_{\pi} V^{\pi}(x) = \mathbb{E}\left[\sum_{t=0}^{T_{first}-1} \ell(x_t, \pi(x_t)) + \mathfrak{q}(x_{T_{first}}) \middle| x_0 = x\right]$$

- At terminal states, $V^*(x) = V^{\pi}(x) = \mathfrak{q}(x)$ for all $x \in \mathcal{T}$
- At other states, the following are satisfied:

Bellman Equations (First Exit Problem)

Hamiltonian:	$H[x, u, V(\cdot)] = \ell(x, u) + \mathbb{E}_{x' \sim p_f(\cdot \mid x, u)} V(x')$
Policy Evaluation:	$V^{\pi}(x) = H[x, \pi(x), V^{\pi}(\cdot)]$
Bellman Equation:	$V^*(x) = \min\limits_{u \in \mathcal{U}(x)} H[x, u, V^*(\cdot)]$
Optimal Policy:	$\pi^*(x) = \arg\min\limits_{u \in \mathcal{U}(x)} H[x, u, V^*(\cdot)]$

Discounted Formulation

- Trajectories continue forever but costs are discounted via $\gamma \in [0, 1)$:

$$\min_{\pi} V^{\pi}(x) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \ell(x_t, \pi(x_t)) \middle| x_0 = x\right]$$

Bellman Equations (Discounted Problem)

Hamiltonian:	$H[x, u, V(\cdot)] = \ell(x, u) + \gamma \mathbb{E}_{x' \sim p_f(\cdot \mid x, u)} V(x')$
Policy Evaluation:	$V^{\pi}(x) = H[x, \pi(x), V^{\pi}(\cdot)]$
Bellman Equation:	$V^*(x) = \min\limits_{u \in \mathcal{U}(x)} H[x, u, V^*(\cdot)]$
Optimal Policy:	$\pi^*(x) = \arg\min\limits_{u \in \mathcal{U}(x)} H[x, u, V^*(\cdot)]$

- Every discounted problem can be converted to a first exist problem by scaling the transition probabilities by γ, introducing a terminal state with zero cost, and setting all transition probabilities to that state to $1 - \gamma$

Value Function

- **Value Function**: the expected long-term cost of following policy π starting from state x:

$$\begin{aligned} V^{\pi}(x) :=& \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \ell(x_t, \pi(x_t)) \,\middle|\, x_0 = x\right] \\ =& \ell(x, \pi(x)) + \gamma \mathbb{E}\left[\sum_{t=1}^{\infty} \gamma^{t-1} \ell(x_t, \pi(x_t)) \,\middle|\, x_0 = x\right] \\ =& \ell(x, \pi(x)) + \gamma \mathbb{E}_{x' \sim p_f(\cdot|x, \pi(x))}\left[V^{\pi}(x')\right] \end{aligned}$$

- **Value Iteration**: computes the optimal value function

$$V^*(x) := \min_{\pi} V^{\pi}(x) = \min_{u \in \mathcal{U}(x)} \left\{\ell(x, u) + \gamma \mathbb{E}_{x' \sim p_f(\cdot|x,u)}\left[V^*(x')\right]\right\}$$

Action-Value (Q) Function

- **Q Function**: the expected long-term cost of taking action u in state x and following policy π afterwards:

$$\begin{aligned} Q^\pi(x,u) :=& \ell(x,u) + \mathbb{E}\left[\sum_{t=1}^{\infty} \gamma^t \ell(x_t, \pi(x_t)) \,\middle|\, x_0 = x\right] \\ =& \ell(x,u) + \gamma \mathbb{E}_{x' \sim p_f(\cdot|x,u)}\left[V^\pi(x')\right] \\ =& \ell(x,u) + \gamma \mathbb{E}_{x' \sim p_f(\cdot|x,u)}\left[Q^\pi(x', \pi(x'))\right] \end{aligned}$$

- **Q-Value Iteration**: computes the optimal Q function

$$\begin{aligned} Q^*(x,u) := \min_\pi Q^\pi(x,u) =& \ell(x,u) + \gamma \mathbb{E}_{x' \sim p_f(\cdot|x,u)}\left[\min_\pi V^\pi(x')\right] \\ =& \ell(x,u) + \gamma \mathbb{E}_{x' \sim p_f(\cdot|x,u)}\left[V^*(x')\right] \\ =& \ell(x,u) + \gamma \mathbb{E}_{x' \sim p_f(\cdot|x,u)}\left[\min_{u' \in \mathcal{U}(x')} Q^*(x', u')\right] \end{aligned}$$

- $Q^*(x,u)$ allows us to choose optimal actions **without having to know anything about the dynamics** $p_f(x' \mid x, u)$:

$$\pi^*(x) = \arg\min_{u \in \mathcal{U}(x)} \left\{\ell(x,u) + \gamma \mathbb{E}_{x' \sim p_f(\cdot|x,u)}\left[V^*(x')\right]\right\} = \arg\min_{u \in \mathcal{U}(x)} Q^*(x,u)$$

Backup Operators

- **Policy Evaluation Backup Operator**:

$$\mathcal{T}_\pi[V](x) := H[x, \pi(x), V] = \ell(x, \pi(x)) + \gamma \mathbb{E}_{x' \sim p_f(\cdot|x,\pi(x))}\left[V(x')\right]$$

- **Value Iteration Backup Operator**:

$$\mathcal{T}_*[V](x) := \min_{u \in \mathcal{U}(x)} H[x, u, V] = \min_{u \in \mathcal{U}(x)} \left\{\ell(x, u) + \gamma \mathbb{E}_{x' \sim p_f(\cdot|x,u)}\left[V(x')\right]\right\}$$

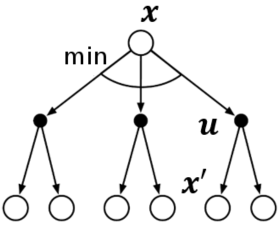
- **Policy Q-Evaluation Backup Operator**:

$$\mathcal{T}_\pi[Q](x, u) := \ell(x, u) + \gamma \mathbb{E}_{x' \sim p_f(\cdot|x,\pi(x))}\left[Q(x', \pi(x'))\right]$$

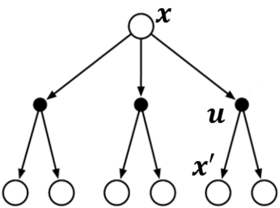
- **Q-Value Iteration Backup Operator**:

$$\mathcal{T}_*[Q](x, u) := \ell(x, u) + \gamma \mathbb{E}_{x' \sim p_f(\cdot|x,u)}\left[\min_{u' \in \mathcal{U}(x')} Q(x', u')\right]$$

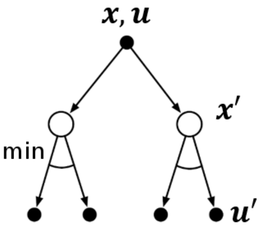
Backup Operators (Stochastic Policy)



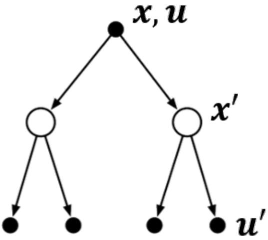
(a) $\mathcal{T}_*[V](x)$



(b) $\mathcal{T}_\pi[V](x)$



(c) $\mathcal{T}_*[Q](x, u)$



(d) $\mathcal{T}_\pi[Q](x, u)$

Contraction in Discounted Problems

Properties of $\mathcal{T}_*[V]$

1. Monotonicity: $V(x) \leq V'(x) \quad \Rightarrow \quad \mathcal{T}_*[V](x) \leq \mathcal{T}_*[V'](x)$
2. γ-Additivity: $\mathcal{T}_*[V+d](x) = \mathcal{T}_*[V](x) + \gamma d$
3. Contraction: $\|\mathcal{T}_*[V](x) - \mathcal{T}_*[V'](x)\|_\infty \leq \gamma \|V(x) - V'(x)\|_\infty$

- **Proof of Contraction**: Let $d = \max_x |V(x) - V'(x)|$. Then:

$$V(x) - d \leq V'(x) \leq V(x) + d, \quad \forall x \in \mathcal{X}$$

Apply $\mathcal{T}_*$ to both sides and use monotonicity and additivity:

$$\mathcal{T}_*[V](x) - \gamma d \leq \mathcal{T}_*[V'](x) \leq \mathcal{T}_*[V](x) + \gamma d, \quad \forall x \in \mathcal{X}$$

VI and PI Revisited

- **Value Iteration**:
  - V^* is the solution to $V = \mathcal{T}_*[V]$ (Bellman Equation)
  - Since $\mathcal{T}_*$ is a contraction, the fixed-point equation has a unique solution (Contraction Mapping Theorem), which can be determined iteratively:

$$V_{k+1} = \mathcal{T}_*[V_k] \qquad \text{(Value Iteration)}$$

  - **Initialization**:
    - Discounted: arbitrary
    - First exit: $V_k(x) = \mathfrak{q}(x)$ for all k and all terminal $x \in \mathcal{T}$

- **Policy Iteration**:
  - **Policy Evaluation**: Given π compute V^π via

$$\mathbf{v} = (I - \gamma P)^{-1}\boldsymbol{\ell} \qquad \text{OR} \qquad V_{k+1} = \mathcal{T}_\pi[V_k] \quad \text{(Policy Evaluation Thm)}$$

  - **Policy Improvement**: choose the action that minimizes the Hamiltonian:

$$\pi'(x) = \underset{u \in \mathcal{U}(x)}{\arg\min}\, H[x, u, V^\pi(\cdot)]$$

  - **Initialization**: arbitrary as long as V^π is finite

Value Iteration

- V^* **is a fixed point of** $\mathcal{T}_*$: $V_0,\ \mathcal{T}_*[V_0],\ \mathcal{T}_*^2[V_0],\ \mathcal{T}_*^3[V_0], \ldots \quad \to V^*$

Algorithm 1 Value Iteration

1: Initialize V_0
2: **for** $k = 0, 1, 2, \ldots$ **do**
3: $\quad V_{k+1} = \mathcal{T}_* [V_k]$

- Q^* **is a fixed point of** $\mathcal{T}_*$: $Q_0,\ \mathcal{T}_*[Q_0],\ \mathcal{T}_*^2[Q_0],\ \mathcal{T}_*^3[Q_0], \ldots \quad \to Q^*$

Algorithm 2 Q-Value Iteration

1: Initialize Q_0
2: **for** $k = 0, 1, 2, \ldots$ **do**
3: $\quad Q_{k+1} = \mathcal{T}_* [Q_k]$

Policy Iteration

- Policy Evaluation: $V_0,\ \mathcal{T}_\pi[V_0],\ \mathcal{T}_\pi^2[V_0],\ \mathcal{T}_\pi^3[V_0],\dots \quad \to V^\pi$

Algorithm 3 Policy Iteration

1: Initialize V_0
2: **for** $k = 0, 1, 2, \dots$ **do**
3: $\quad \pi_{k+1}(x) = \arg\min_{u \in \mathcal{U}(x)} H[x, u, V_k(\cdot)]$ ▷ Policy Improvement
4: $\quad V_{k+1} = \mathcal{T}_{\pi_{k+1}}^\infty [V_k]$ ▷ Policy Evaluation

- Policy Q-Evaluation: $Q_0,\ \mathcal{T}_\pi[Q_0],\ \mathcal{T}_\pi^2[Q_0],\ \mathcal{T}_\pi^3[Q_0],\dots \quad \to Q^\pi$

Algorithm 4 Q-Policy Iteration

1: Initialize Q_0
2: **for** $k = 0, 1, 2 \dots$ **do**
3: $\quad \pi_{k+1}(x) = \arg\min_{u \in \mathcal{U}(x)} Q_k(x, u)$ ▷ Policy Improvement
4: $\quad Q_{k+1} = \mathcal{T}_{\pi_{k+1}}^\infty [Q_k]$ ▷ Policy Evaluation

Generalized Policy Iteration

Algorithm 5 Generalized Policy Iteration

1: Initialize V_0
2: **for** $k = 0, 1, 2, \ldots$ **do**
3: $\quad \pi_{k+1}(x) = \underset{u \in \mathcal{U}(x)}{\arg\min}\, H[x, u, V_k(\cdot)]$ ▷ Policy Improvement
4: $\quad V_{k+1} = \mathcal{T}^n_{\pi_{k+1}}[V_k], \quad \text{for } n \geq 1$ ▷ Policy Evaluation

Algorithm 6 Generalized Q-Policy Iteration

1: Initialize Q_0
2: **for** $k = 0, 1, 2, \ldots$ **do**
3: $\quad \pi_{k+1}(x) = \underset{u \in \mathcal{U}(x)}{\arg\min}\, Q_k(x, u)$ ▷ Policy Improvement
4: $\quad Q_{k+1} = \mathcal{T}^n_{\pi_{k+1}}[Q_k], \quad \text{for } n \geq 1$ ▷ Policy Evaluation