Exact Methods : Value and Policy Iteration

Easwar Subramanian

TCS Innovation Labs, Hyderabad

Email : easwar.subramanian@tcs.com / cs5500.2020@iith.ac.in

August 26, 2023

Overview

1. Review
2. Value Iteration
3. Proof of Value Iteration Convergence
4. Policy Iteration
5. Possible Extensions
6. Few Remarks

Review

Optimal Policy

Define a partial ordering over policies

$$\pi \geq \pi', \quad \text{if} \quad V^{\pi}(s) \geq V^{\pi'}(s), \quad \forall s \in \mathcal{S}$$

Theorem

- There exists an optimal policy π_* that is better than or equal to all other policies.
- All optimal policies achieve the optimal value function, $V_*(s) = V^{\pi_*}(s)$
- All optimal policies achieve the optimal action-value function, $Q_*(s, a) = Q^{\pi_*}(s, a)$

Solution to an MDP

Solving an MDP <u>means</u> finding a policy π_* as follows

$$\pi_* = \arg\max_{\pi} \left[\mathbb{E}_\pi \left(\sum_{t=0}^{\infty} \gamma^t r_{t+1} \right) \right]$$

is **maximum**

- Denote optimal value function $V_*(s) = V^{\pi_*}(s)$
- Denote optimal action value function $Q_*(s,a) = Q^{\pi_*}(s,a)$
- The main goal in RL or solving an MDP means finding an **optimal value function** V_* or **optimal action value function** Q_* or **optimal policy** π_*

Value Iteration

Value Iteration

Question : Is there a way to arrive at V_* starting from an arbitrary value function V_0 ?

Answer : Value Iteration

Bellman Evaluation Equation

$$V^\pi(s) = \sum_a \pi(a|s) \sum_{s'} \mathcal{P}^a_{ss'} \left[\mathcal{R}^a_{ss'} + \gamma V^\pi(s')\right]$$

- For a MDP with $\mathcal{S} = n$, Bellman Evaluation Equation for $V^\pi(s)$ is a system of $n = |\mathcal{S}|$ (linear) equations with n variables and can be solved if the model is known

Denote,

$$\begin{aligned}
\mathcal{P}^\pi(s'|s) &= \sum_{a \in \mathcal{A}} \pi(a|s) \mathcal{P}^a_{ss'} \\
\mathcal{R}^\pi(s) &= \sum_{a \in \mathcal{A}} \pi(a|s) \sum_{s'} \mathcal{P}^a_{ss'} \mathcal{R}^a_{ss'} = \mathbb{E}(r_{t+1}|s_t = s)
\end{aligned}$$

$$V^\pi = \mathcal{R}^\pi + \gamma \mathcal{P}^\pi V^\pi \implies V^\pi = (I - \gamma \mathcal{P}^\pi)^{-1} \mathcal{R}^\pi$$

Optimality Equation for State Value Function

Question : Can we have a recursive formulation for $V_*(s)$?

$$V_*(s) = \max_a Q_*(s,a) = \max_a \left[\sum_{s' \in \mathcal{S}} \mathcal{P}^a_{ss'} \left(\mathcal{R}^a_{ss'} + \gamma V_*(s') \right) \right]$$

Question : These are also a system of equations with $n = |\mathcal{S}|$ with n variables. Can we solve them ?

Answer : Optimality equations are non-linear system of equations with n unknowns and n non-linear constraints (i.e., the max operator).

Solving the Bellman Optimality Equation

- Bellman optimality equations are non-linear
- In general, there are no closed form solutions
- Iterative methods are typically used

Bellman's Optimality Principle

Principle of Optimality

The tail of an optimal policy must be optimal

- Any optimal policy can be subdivided into two components; an optimal first action, followed by an optimal policy from successor state s'.

Solution Methodology : Dynamic Programming

Bellman optimality equation :

$$V_*(s) = \max_a \left[\sum_{s' \in \mathcal{S}} \mathcal{P}^a_{ss'} \left(\mathcal{R}^a_{ss'} + \gamma V_*(s') \right) \right]$$

Optimal Substructure : Optimal solution can be constructed from optimal solutions to subproblems

Overlapping Subproblems : Problem can be broken down into subproblems and can be reused several times

- Markov Decision Processes, generally, satisfy both these characterstics
- Dynamic Programming is a popular solution method for problems having such properties

Value Iteration : Idea

- Suppose we know the value $V_*(s')$
- Then the solution $V_*(s)$ can be found by one step look ahead

$$V_*(s) \leftarrow \max_a \left[\sum_{s' \in \mathcal{S}} \mathcal{P}^a_{ss'} \left(\mathcal{R}^a_{ss'} + \gamma V_*(s') \right) \right]$$

- Idea of value iteration is to perform the above updates iteratively

Value Iteration : Algorithm

Algorithm Value Iteration

1: Start with an initial value function $V_1(\cdot)$;
2: **for** $k = 1, 2, \cdots, K$ **do**
3: **for** $s \in \mathcal{S}$ **do**
4: Calculate
$$V_{k+1}(s) \leftarrow \max_a \left[\sum_{s' \in \mathcal{S}} \mathcal{P}^a_{ss'} \left(\mathcal{R}^a_{ss'} + \gamma V_k(s') \right) \right]$$
5: **end for**
6: **end for**

Value Iteration : Example

No noise and discount factor $\gamma = 1$

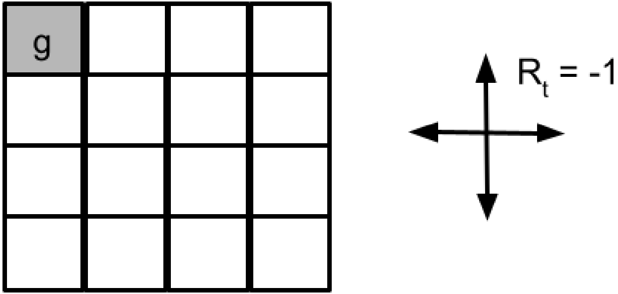


Figure Source: David Silver's UCL Course

Value Iteration : Example

$$V_{k+1}(s) \leftarrow \max_{a} \left[\sum_{s' \in \mathcal{S}} \mathcal{P}^{a}_{ss'} \left(\mathcal{R}^{a}_{ss'} + \gamma V_k(s') \right) \right]$$

g			

Problem

0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

V_1

0	-1	-1	-1
-1	-1	-1	-1
-1	-1	-1	-1
-1	-1	-1	-1

V_2

0	-1	-2	-2
-1	-2	-2	-2
-2	-2	-2	-2
-2	-2	-2	-2

V_3

0	-1	-2	-3
-1	-2	-3	-3
-2	-3	-3	-3
-3	-3	-3	-3

V_4

0	-1	-2	-3
-1	-2	-3	-4
-2	-3	-4	-4
-3	-4	-4	-4

V_5

0	-1	-2	-3
-1	-2	-3	-4
-2	-3	-4	-5
-3	-4	-5	-5

V_6

0	-1	-2	-3
-1	-2	-3	-4
-2	-3	-4	-5
-3	-4	-5	-6

V_7

Value Iteration : Remarks

- The sequence of value functions $\{V_1, V_2, \cdots, \}$ converge
- It converges to V_*
- Convergence is independent of the choice of V_0.
- Intermediate value functions need not correspond to a policy in the sense of satisfying the Bellman Evaluation Equation
- However, for any k, one can come up with a greedy policy as follows

$$\pi_{k+1}(s) \leftarrow \text{greedy} V_k(s)$$

- The crux of proving the above statements lie in **Banach Fixed Point Theorem / Contraction Mapping Theorem**

Optimality Equation for Action-Value Function

There is a recursive formulation for $Q_*(\cdot, \cdot)$

$$Q_*(s,a) = \left[\sum_{s' \in \mathcal{S}} \mathcal{P}^a_{ss'} \left(\mathcal{R}^a_{ss'} + \gamma \max_{a'} Q_*(s', a')\right)\right]$$

One could similarly conceive an iterative algorithm to compute optimal Q_* using the above recursive formulation !!

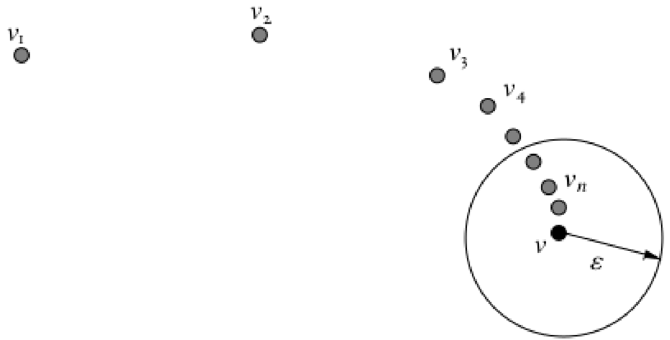
Proof of Value Iteration Convergence

Notion of Convergence

Convergence

Let $\mathcal{V}$ be a vector space. A sequence of vectors $\{v_n\} \in \mathcal{V}$ (with $n \in \mathbb{N}$) is said to converge to v if and only if

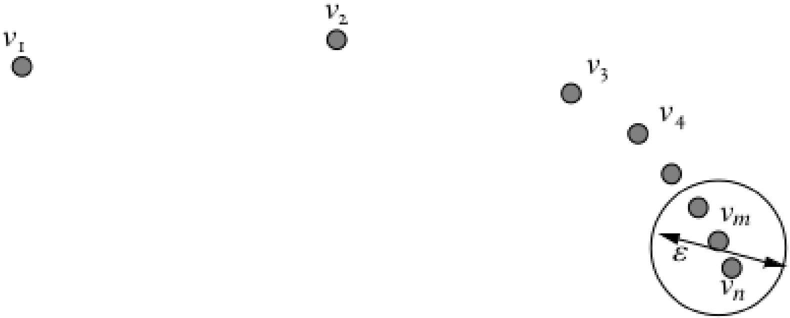
$$\lim_{n\to\infty} \|v_n - v\| = 0$$



Cauchy Sequence

Cauchy Sequence

A sequence of vectors $\{v_n\} \in \mathcal{V}$ (with $n \in \mathbb{N}$) is said to be a Cauchy sequence, if and only if, for each $\varepsilon > 0$, there exists an N_ε such that $\|v_n - v_m\| \leq \varepsilon$ for any $n, m > N_\varepsilon$



ustration

Notion of Completeness

Completeness

A **normed vector space** $(\mathcal{V}, \|\cdot\|)$ is complete, if and only if, every Cauchy sequence in $\mathcal{V}$ converges to a point in $\mathcal{V}$

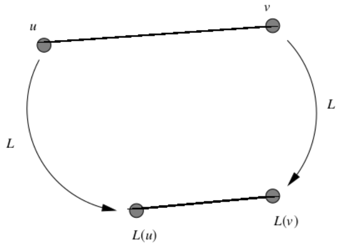
Contractions

Contractions

Let $(\mathcal{V}, \|\cdot\|)$ be a normed vector space and and let $L : \mathcal{V} \to \mathcal{V}$. We say that L is a contraction, or a contraction mapping, if there is a real number $\gamma \in [0, 1)$, such that

$$\|L(v) - L(u)\| \leq \gamma \|v - u\|$$

for all v and u in $\mathcal{V}$, where the term γ is called a Lipschitz coefficient for L.



Notion of Fixed Point

Fixed Point

A vector $v \in \mathcal{V}$ is a fixed point of the map $L : \mathcal{V} \to \mathcal{V}$ if $L(v) = v$

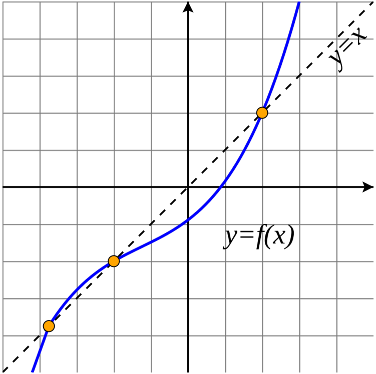
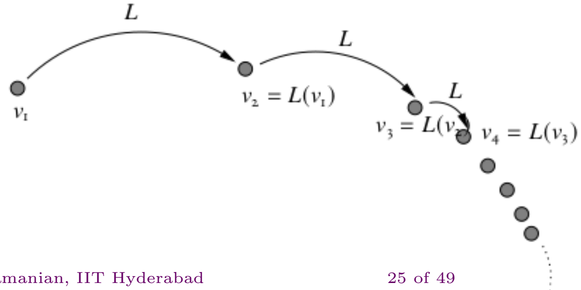


Figure: Fixed Point : Illustration

Banach Fixed Point Theorem

Theorem

Let $<\mathcal{V}, \|\cdot\|)>$ be a complete normed vector space and let $L : \mathcal{V} \to \mathcal{V}$ be a γ-contraction mapping. Then iterative application of L converges to a unique fixed point in $\mathcal{V}$ independent of the starting point



Value Function Space

- $\mathcal{S}$ is a discrete state space with $|\mathcal{S}| = n$
- $\mathcal{A}_s \subseteq \mathcal{A}$ be the non-empty subset of actions allowed from state s
- $\mathcal{V}$ be a vector space of set of all bounded real valued functions from $\mathcal{S}$ to $\mathbb{R}$
- Measure the distance between state value functions $u, v \in \mathcal{V}$ using the max-norm defined as follows

$$\|u - v\| = \|u - v\|_\infty = \max_{s \in \mathcal{S}} |u(s) - v(s)| \quad s \in \mathcal{S}; u, v \in \mathcal{V}$$

  - ★ Largest distance between state values
- The space $\mathcal{V}$ is complete

Bellman Evaluation Operator

$$V_{k+1}^{\pi}(s) = \sum_{a} \pi(a|s) \sum_{s'} \mathcal{P}_{ss'}^{a} \left[\mathcal{R}_{ss'}^{a} + \gamma V_{k}^{\pi}(s')\right]$$

Denote,

$$\begin{aligned} \mathcal{P}^{\pi}(s'|s) &= \sum_{a \in \mathcal{A}} \pi(a|s) \mathcal{P}_{ss'}^{a} \\ \mathcal{R}^{\pi}(s) &= \sum_{a \in \mathcal{A}} \pi(a|s) \sum_{s'} \mathcal{P}_{ss'}^{a} \mathcal{R}_{ss'}^{a} = \mathbb{E}(r_{t+1}|s_t = s) \end{aligned}$$

Then, we can write,

$$V^{\pi} = \mathcal{R}^{\pi} + \gamma \mathcal{P}^{\pi} V^{\pi} \qquad \text{(or)}\ V_{k+1} = \mathcal{R}^{\pi} + \gamma \mathcal{P}^{\pi} V_{k}$$

Define **Bellman Evaluation Operator** $(\mathcal{L}^{\pi} : \mathcal{V} \rightarrow \mathcal{V})$ as,

$$L^{\pi}(v) = \mathcal{R}^{\pi} + \gamma \mathcal{P}^{\pi} v$$

Bellman Optimality Operator

$$V_{k+1}(s) = \max_a \left[\sum_{s' \in \mathcal{S}} \mathcal{P}^a_{ss'} \left(\mathcal{R}^a_{ss'} + \gamma V_k(s') \right) \right]$$

Denote,

$$\begin{aligned} \mathcal{P}^a(s) &= \sum_{s' \in \mathcal{S}} \mathcal{P}^a_{ss'} \\ \mathcal{R}^a(s) &= \sum_{s' \in \mathcal{S}} \mathcal{P}^a_{ss'} \mathcal{R}^a_{ss'} \end{aligned}$$

Then, we can write,

$$V_{k+1} = \max_{a \in \mathcal{A}} \left[\mathcal{R}^a + \gamma \mathcal{P}^a V_k \right]$$

Definte **Bellman Optimality Operator** : $(\mathcal{L} : \mathcal{V} \to \mathcal{V})$ as

$$L(v) = \max_{a \in \mathcal{A}} \left[\mathcal{R}^a + \gamma \mathcal{P}^a v \right]$$

Remark : Note that since value functions are a mapping from state space to real numbers one can also think of $\mathcal{L}^\pi$ and $\mathcal{L}$ as mappings from $\mathbb{R}^d \to \mathbb{R}^d$

Fixed Points of Maps $\mathcal{L}^\pi$ and $\mathcal{L}$

We can see that V^π is a fixed point of function $\mathcal{L}^\pi$

$$\mathcal{L}^\pi V^\pi = V^\pi$$

and V_* is a fixed point of operator $\mathcal{L}$

$$\mathcal{L}V_* = V_*$$

Bellman Evaluation Operator is a Contraction

Recall that Bellman evaluation operator is given by $L^\pi : \mathcal{V} \to \mathcal{V}$

$$L^\pi(v) = \mathcal{R}^\pi + \gamma \mathcal{P}^\pi v$$

- This operator is γ contraction. i.e., it makes value fuctions closer by at least γ.

Proof.

For any two value functions u and v in the space $\mathcal{V}$, we have,

$$\begin{aligned}
\|L^\pi(u) - L^\pi(v)\|_\infty &= \|(\mathcal{R}^\pi + \gamma\mathcal{P}^\pi u) - (\mathcal{R}^\pi + \gamma\mathcal{P}^\pi v)\|_\infty \\
&= \|\gamma\mathcal{P}^\pi(u-v)\|_\infty (\leq \gamma\|P^\pi\|_\infty\|(u-v)\|_\infty = \gamma\|(u-v)\|_\infty) \\
&\leq \|\gamma\mathcal{P}^\pi\|u-v\|_\infty\|_\infty \\
&\leq \gamma\|u-v\|_\infty
\end{aligned}$$

(We used for every $x \in \mathbb{R}^n$, and A is a $m \times n$ matrix, $\|Ax\|_\infty \leq \|A\|_\infty\|x\|_\infty$) $\square$

Convergence of Bellman Updates

- Banach fixed-point theorem guarantees that iteratively applying evaluation operator $\mathcal{L}^\pi$ to any function $V \in \mathcal{V}$ will converge to a unique function $V^\pi \in V$
- Similarly, the Bellman optimality operator $(\mathcal{L} : \mathcal{V} \to \mathcal{V})$

$$L(v) = \max_{a \in \mathcal{A}} \left[\mathcal{R}^a + \gamma \mathcal{P}^a v\right] \quad (\text{ A similar argument as } L^\pi)$$

  is also a γ contraction and hence iteratively applying optimality operator $\mathcal{L}$ to any funciton $V \in \mathcal{V}$ will converge to a unique function $V_* \in V$
- Does $V_* = \max_\pi V^\pi(\cdot)$? (Yes, it does)

Policy Iteration

Policy Iteration

Question : Is there a way to arrive at π_* starting from an arbitrary policy π ?

Answer : Policy Iteration

- Evaluate the policy π
  - ★ Compute $V^{\pi}(s) = \mathbb{E}_{\pi}(r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots | s_t = s)$
- Improve the policy π

$$\pi'(s) = \text{greedy}(V^{\pi}(s))$$

$$\pi_0 \xrightarrow{\text{E}} V^{\pi_0} \xrightarrow{\text{I}} \pi_1 \xrightarrow{\text{E}} V^{\pi_1} \xrightarrow{\text{I}} \pi_2 \xrightarrow{\text{E}} \cdots \xrightarrow{\text{I}} \pi^* \xrightarrow{\text{E}} V^*,$$

Policy Evaluation

- **Problem** : Evaluate a given policy π
- Compute $V^{\pi}(s) = \mathbb{E}_{\pi}(r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots | s_t = s)$
- **Solution 1** : Solve a system of linear equations using any solver
- **Solution 2** : Iterative application of Bellman Evaluation Equation
- Iterative update rule :

$$V_{k+1}^{\pi}(s) \leftarrow \sum_{a} \pi(a|s) \sum_{s'} \mathcal{P}_{ss'}^{a} \left[\mathcal{R}_{ss'}^{a} + \gamma V_{k}^{\pi}(s')\right]$$

- The sequence of value functions $\{V_1^{\pi}, V_2^{\pi}, \cdots,\}$ converge to V^{π}

Policy Improvement

Suppose we know V^π. How to improve policy π ?

The answer lies in the definition of action value function $Q^\pi(s,a)$. Recall that,

$$\begin{aligned}
Q^\pi(s,a) &= \mathbb{E}_\pi\left(\sum_{k=0}^{\infty}\gamma^k r_{t+k+1}|s_t = s, a_t = a\right) \\
&= \mathbb{E}(r_{t+1} + \gamma V^\pi(s_{t+1})|s_t = s, a_t = a) \\
&= \sum_{s' \in \mathcal{S}} \mathcal{P}^a_{ss'}\left[\mathcal{R}^a_{ss'} + \gamma V^\pi(s')\right]
\end{aligned}$$

- If $Q^\pi(s,a) > V^\pi(s) \implies$ Better to select action a in state s and thereafter follow the policy π
- This is a special case of the policy improvement theorem

Policy Improvement Theorem

Theorem

Let π and π' be any pair of deterministic policies such that, for all $s \in \mathcal{S}$,

$$Q^{\pi}(s, \pi'(s)) \geq V^{\pi}(s).$$

Then $V^{\pi'}(s) \geq V^{\pi}(s)$ for all $s \in \mathcal{S}$

Proof.

$$\begin{aligned}
V^{\pi}(s) &\leq Q^{\pi}(s, \pi'(s)) = \mathbb{E}_{\pi'}(r_{t+1} + \gamma V^{\pi}(s_{t+1}) | s_t = s) \\
&\leq \mathbb{E}_{\pi'}(r_{t+1} + \gamma Q^{\pi}(s_{t+1}, \pi'(s_{t+1})) | s_t = s) \\
&= \mathbb{E}_{\pi'}(r_{t+1} + \gamma r_{t+2} + \gamma^2 V^{\pi}(s_{t+2}) | s_t = s) \\
&\leq \mathbb{E}_{\pi'}(r_{t+1} + \gamma r_{t+2} + \gamma^2 Q^{\pi}(s_{t+2}, \pi'(s_{t+2})) | s_t = s) \\
&\leq \mathbb{E}_{\pi'}(r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots | s_t = s) = V^{\pi'}(s)
\end{aligned}$$

□

Policy Improvement

- Now consider the greedy policy $\pi' = \text{greedy}(V^\pi)$.
- Then, $\pi' \geq \pi$. That is, $V^{\pi'}(s) \geq V^\pi(s)$ for all $s \in \mathcal{S}$.
  - ★ By defintion of π', at state s, the action chosen by policy π' is given by the greedy operator
    $$\pi'(s) = \arg\max_a Q^\pi(s, a)$$
  - ★ This improves the value from any state s over one step
    $$Q^\pi(s, \pi'(s)) = \max_a Q^\pi(s, a) \geq Q^\pi(s, \pi(s)) = V^\pi(s)$$
  - ★ It therefore improves the value function, $V^{\pi'}(s) \geq V^\pi(s)$
- Policy π' is at least as good as policy π

Figure Source: Refer to David Silver Lecture 3 slides for a more detailed proof

Policy Improvement

- If improvements stop,

$$Q^{\pi}(s, \pi'(s)) = \max_{a} Q^{\pi}(s, a) = Q^{\pi}(s, \pi(s)) = V^{\pi}(s)$$

- Bellman optimality equation is satisfied as,

$$V^{\pi}(s) = \max_{a} Q^{\pi}(s, a)$$

- The policy π for which the improvement stops is the optimal policy.

$$V^{\pi}(s) = V_{*}(s) \quad \forall s \in \mathcal{S}$$

Policy Iteration : Algorithm

Algorithm Policy Iteration

Start with an initial policy π_1
for $i = 1, 2, \cdots, N$ **do**
    Evaluate $V^{\pi_i}(s) \quad \forall s \in \mathcal{S}$. That is,
    for $k = 1, 2, \cdots, K$ **do**
        For all $s \in \mathcal{S}$ calculate

$$V_{k+1}^{\pi_i}(s) \leftarrow \sum_a \pi(a|s) \sum_{s'} \mathcal{P}_{ss'}^a \left[\mathcal{R}_{ss'}^a + \gamma V_k^{\pi_i}(s')\right]$$

    end for
    Perform policy Improvement

$$\pi_{i+1} = \text{greedy}(V^{\pi_i})$$

end for

Policy Iteration : Example

Update Rule :

$$V_{k+1}^{\pi_i}(s) \leftarrow \sum_a \pi(a|s) \sum_{s'} \mathcal{P}_{ss'}^a \left[\mathcal{R}_{ss'}^a + \gamma V_k^{\pi_i}(s')\right]$$

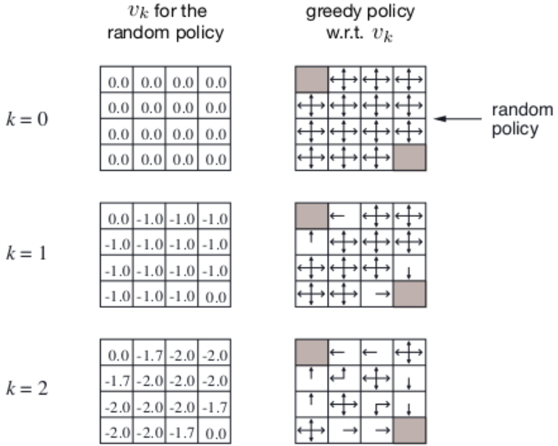


Figure Source: David Silver's UCL course

Policy Iteration : Example

$k = 3$

0.0	-2.4	-2.9	-3.0
-2.4	-2.9	-3.0	-2.9
-2.9	-3.0	-2.9	-2.4
-3.0	-2.9	-2.4	0.0

$k = 10$

0.0	-6.1	-8.4	-9.0
-6.1	-7.7	-8.4	-8.4
-8.4	-8.4	-7.7	-6.1
-9.0	-8.4	-6.1	0.0

$k = \infty$

0.0	-14.	-20.	-22.
-14.	-18.	-20.	-20.
-20.	-20.	-18.	-14.
-22.	-20.	-14.	0.0

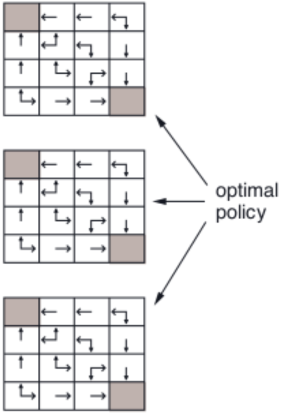
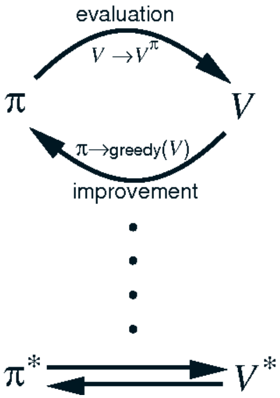
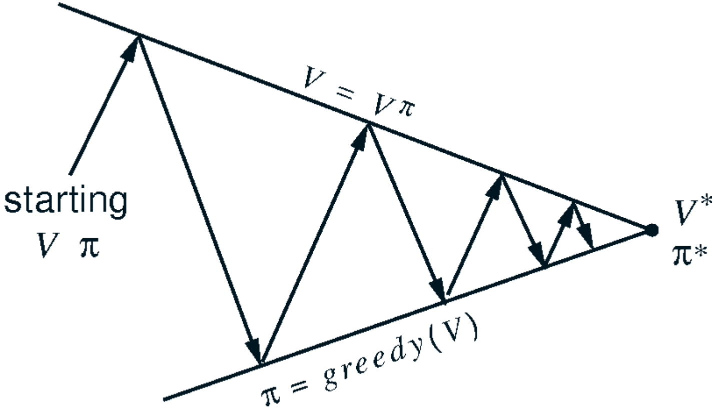


Figure Source: David Silver's UCL course

Policy Iteration : Schematic Representation





- The sequence $\{\pi_1, \pi_2, \cdots, \}$ is guaranteed to converge.
- At convergence, both current policy and the value function associated with the policy are optimal.

Figure Source: David Silver's UCL course

Modified Policy Iteration

Can we computationally simplify policy iteration process ?

- We need not wait for policy evaluation to converge to V^{π}
- We can have a stopping criterion like ϵ-convergence of value function evaluation or K iterations of policy evaluation
- Extreme case of $K = 1$ is **value iteration**. We update the policy every iteration

Possible Extensions

Asynchronous Dynamic Programming

- Updates to states are done individually, in any order
- For each selected state, apply the appropriate backup
- Can significantly reduce computation
- Convergence guarantees exist, if all states are selected sufficient number of times

Real Time Dynamic Programming

- Idea : update only states that are relevant to agent
- After each time step, we get s_t, a_t, r_{t+1}
- Perform the following update

$$V(s_t) \leftarrow \max_a \left[\sum_{s' \in \mathcal{S}} \mathcal{P}^a_{s_t s'} \left(\mathcal{R}^a_{s_t s'} + \gamma V(s') \right) \right]$$

Few Remarks

MDP and RL setting

- **MDP Setting** : The agent has knowledge of the state transition matrices $\mathcal{P}^a_{ss'}$ and the reward function $\mathcal{R}$
- **RL Setting** : The agent does not have knowledge of the state transition matrices $\mathcal{P}^a_{ss'}$ and the reward function $\mathcal{R}$
  - ★ The goal in both cases are same; Determine optimal sequence of actions such that the total discounted future reward is maximum.
  - ★ Although, this course would assume Markovian structure to state transitions, in many (sequential) decision making problems we may have to consider the history as well.

Prediction and Control using Dynamic Programming

- Dynamic Programming assumes full knowledge of MDP
- Used for both **prediction** and **control** in an MDP
- Prediction
  - ★ Input MDP ($< \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma >$) and policy π
  - ★ Output : $V^{\pi}(\cdot)$
- Control
  - ★ Input MDP ($< \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma >$)
  - ★ Output : Optimal value function $V_*(\cdot)$ or optimal policy π_*

Slide Credit: David Silver's UCL course