Policy Iteration

Reinforcement Learning

Roberto Capobianco

Sapienza Università di Roma

Recap

Bellman Equation

The value of a certain state is expanded in terms of the current reward and the value of the next states according to the policy

r here is function of s and π(s)

$V^{\pi}(s_t) = \mathbb{E}_{\pi}[r_t+\gamma r_{t+1}+\gamma^2 r_{t+2}+\gamma^3 r_{t+3}+...|s_t] = r_t+\gamma\mathbb{E}_{s'\sim p(.|s,\pi(s))}[V^{\pi}(s')]$

$Q^{\pi}(s_t,a) = r_t+\gamma\mathbb{E}_{s'\sim p(.|s,a)}[V^{\pi}(s')]$

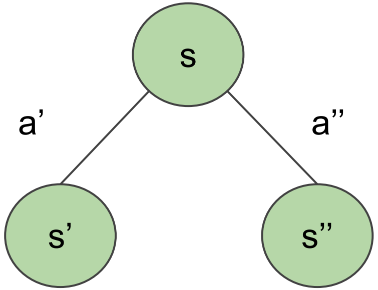
r here is function of s and a

As a result $V(s) = Q(s,\pi(s))$

Bellman Optimality Example

$$V^*(s)=\max_a[r(s,a)+\gamma\mathbb{E}_{s'\sim p(.|s,a)}V^*(s')]$$

- Try a’, get $r(s,a')$, compute $Q^*(s,a')=r(s,a')+\gamma V^*(s')$
- Try a’’, get $r(s,a'')$, compute $Q^*(s,a'')=r(s,a'')+\gamma V^*(s'')$



Assume we know V^* at s’ and s’’

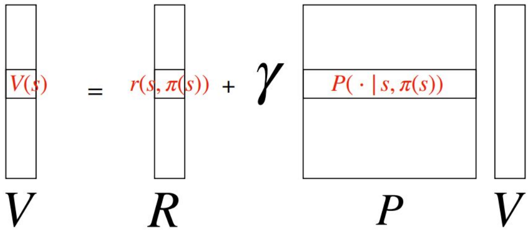
$$V^*(s)=\max_{a',a''}\{Q^*(s,a'),Q^*(s,a'')\}$$

Exact Policy Evaluation

We know that **for ALL states**, Bellman equation holds

$$V^{\pi}(s) = r+\gamma\mathbb{E}_{s'\sim p(.|s,\pi(s))}[V^{\pi}(s')]$$

We can combine all the constraints together:



Since we have this set of constraints

$$V = R + \gamma PV$$

we can solve for V as

$$V = (I - \gamma P)^{-1}R$$

:(Nice but computationally expensive: inverting the matrix is $O(S^3)$

Fixed-Point Iteration & Contractions

What is a fixed-point? A point where holds

$$x = f(x)$$

How can we find such points?

- Initialize x_0
- Repeat $x_{i+1} = f(x_i)$
- Stop at convergence where x is found and does not change anymore

Convergence to a fixed-point is possible thanks to the existence of **contraction mappings**

f: M->M (M is a metric space) is a contraction mapping if:

$$|f(x) - f(x')| \leq k|x-x'| \text{ for } k \text{ in } [0, 1)$$

Iterative Policy Evaluation

- Initialize V_0 in [0, 1/(1-γ)] (typically 0)
- Until convergence:

$$V_{i+1} = R + \gamma P V_i$$

(note: this is using matricial form because it's doing it for all states)

$$\| V^{t+1} - V^{\pi} \|_{\infty} \leq \gamma \| V^{t} - V^{\pi} \|_{\infty} \leq \gamma^{t+1} \| V^{0} - V^{\pi} \|_{\infty}$$

For each iteration it's $O(S^2)$

How to Find the Optimal Policy?

Now, what we're really interested in is finding the optimal policy π^*

Let's use Bellman optimality! ...and the Bellman Operator (which is a contraction)

$$TQ(s,a) = r(s,a)+\gamma\mathbb{E}_{s'\sim p(.|s,a)}\max_{a'}[Q(s',a')])$$

Since Q: S x A -> $\mathbb{R}$, then also TQ: S x A -> $\mathbb{R}$

Value Iteration & Optimal Policy

We can obtain $Q^* = TQ^*$, since Q^* is a fixed-point solution to $Q = TQ$

- Initialize $||Q_0||$ in $[0, 1/(1-\gamma)]$ (typically 0)
- Until convergence, for all states and actions:

$$Q_{i+1} = TQ_i$$

$$||Q_{i+1} - Q^*|| = ||TQ_i - TQ^*|| \leq \gamma||Q_i - Q^*|| \leq \gamma^{i+1}||Q_0 - Q^*||$$

We know that $\pi^*(s) = \text{argmax}_a Q^*(s,a)$, and since $Q_i(s,a) \cong Q^*(s,a)$ we could choose

$$\pi_i(s) = \text{argmax}_a Q_i(s,a)$$

End - Recap

Clarification: V_0 Initialization

We Initialize V_0 in $[0, 1/(1-\gamma)]$ (typically 0)

WHY?

Clarification: V_0 Initialization

We Initialize V_0 in $[0, 1/(1-\gamma)]$ (typically 0)

WHY?

Under the assumption that R(s,a) is in [0, 1]

The maximum possible value of V is $1 + \gamma + \gamma^2 + \ldots = 1/(1-\gamma)$

Clarification: V_0 Initialization

We Initialize V_0 in [0, 1/(1-γ)] (typically 0)

WHY?

Under the assumption that R(s,a) is in [0, 1]

The maximum possible value of V is $1 + \gamma + \gamma^2 + \ldots = 1/(1-\gamma)$

There is no reason to initialize higher than that!

Clarification: Number of Iterations of Value Iteration

If we want an ϵ error on the quality of the policy, to determine the number of iterations i we can just solve for it in this equation

$$2\gamma^i/(1-\gamma)||Q_0-Q^*|| \leq \epsilon$$

Clarification: Number of Iterations of Value Iteration

If we want an ϵ error on the quality of the policy, to determine the number of iterations i we can just solve for it in this equation

$$2\gamma^i/(1-\gamma)\boxed{||Q_0}-Q^*|| \leq \epsilon$$

In range [0, 1/(1-γ)]

SAPIENZA
UNIVERSITÀ DI ROMA

Clarification: Number of Iterations of Value Iteration

If we want an ϵ error on the quality of the policy, to determine the number of iterations i we can just solve for it in this equation

$$2\gamma^i/(1-\gamma)||Q_0\boxed{-Q^*}|| \leq \epsilon$$

In range [0, 1/(1-γ)]

Clarification: Number of Iterations of Value Iteration

If we want an ϵ error on the quality of the policy, to determine the number of iterations i we can just solve for it in this equation

$$2\gamma^i/(1-\gamma)\boxed{||Q_0-Q^*||} \leq \epsilon$$

By the infinity norm, the maximum value is $1/(1-\gamma)$

Clarification: Number of Iterations of Value Iteration

If we want an ϵ error on the quality of the policy, to determine the number of iterations i we can just solve for it in this equation

$$2\gamma^i/(1-\gamma)||Q_0-Q^*|| = 2(1-(1-\gamma))^i/(1-\gamma)||Q^*|| \leq \epsilon$$

Additionally add and subtract 1

Clarification: Number of Iterations of Value Iteration

If we want an ϵ error on the quality of the policy, to determine the number of iterations *i* we can just solve for it in this equation

$$2\gamma^i/(1-\gamma)||Q_0-Q^*|| = 2(1-(1-\gamma))^i/(1-\gamma)||Q^*|| \leq \epsilon$$

$$2(1-(1-\gamma))^i/(1-\gamma)||Q^*|| \leq 2e^{-(1-\gamma)i}/(1-\gamma)^2 \leq \epsilon$$

$1+x \leq e^x$ for all reals

Clarification: Number of Iterations of Value Iteration

If we want an ϵ error on the quality of the policy, to determine the number of iterations i we can just solve for it in this equation

$$2\gamma^i/(1-\gamma)||Q_0-Q^*|| = 2(1-(1-\gamma))^i/(1-\gamma)||Q^*|| \leq \epsilon$$

$$2(1-(1-\gamma))^i/(1-\gamma)||Q^*|| \leq 2e^{-(1-\gamma)i}/(1-\gamma)^2 \leq \epsilon$$

$$e^{-(1-\gamma)i} \leq \epsilon(1-\gamma)^2/2$$

$\log(1/x)=-\log(x)$

$$-i(1-\gamma)\leq-\log(2/(\epsilon(1-\gamma)^2))$$

Clarification: Number of Iterations of Value Iteration

If we want an ϵ error on the quality of the policy, to determine the number of iterations i we can just solve for it in this equation

$$2\gamma^i/(1-\gamma)||Q_0-Q^*|| = 2(1-(1-\gamma))^i/(1-\gamma)||Q^*|| \leq \epsilon$$

$$2(1-(1-\gamma))^i/(1-\gamma)||Q^*|| \leq 2e^{-(1-\gamma)i}/(1-\gamma)^2 \leq \epsilon$$

$$e^{-(1-\gamma)i} \leq \epsilon(1-\gamma)^2/2$$

$$-i(1-\gamma)\leq-\log(2/(\epsilon(1-\gamma)^2))$$

$$i\geq\log(2/(\epsilon(1-\gamma)^2))/(1-\gamma)$$

Clarification: Number of Iterations of Value Iteration

If we want an ϵ error on the quality of the policy, to determine the number of iterations i we can just solve for it in this equation

$$i \geq \frac{\log \frac{2}{\epsilon(1-\gamma)^2}}{1-\gamma}$$

Another Note on Value Iteration

- Q_t is approximating Q^*
- From Q_t we compute a policy π_t

However...

Another Note on Value Iteration

- Q_t is approximating Q^*
- From Q_t we compute a policy π_t

However...

Q_t is generally different from $Q^{\pi t}$ until we converge to approximately Q^*

E.g, Q_0 is just a random initial guess, maybe not corresponding to any policy's Q value

Complexity of Value Iteration

For each iteration it's $O(S^2A)$

Policy Iteration

- Outputs policies at every iteration: $\{\pi_0, \pi_1, \pi_2 ... \pi_T\}$
- Different from Value Iteration that was outputting values

Policy Iteration

- Outputs policies at every iteration: $\{\pi_0, \pi_1, \pi_2...\pi_T\}$
- Different from Value Iteration that was outputting values

Procedure:

1. Start with a random guess π_0 (can be deterministic or stochastic)
2. For t=0,...,T: $Q^\pi(s_t,a) = r_t+\gamma\mathbb{E}_{s'\sim p(.|s,a)}[V^\pi(s')]$
   a. Do **policy evaluation** and compute $Q^{\pi t}$ for all s,a
   b. Do **policy improvement** as $\pi_{t+1}=\text{argmax}_a Q^{\pi t}(s,a)$ for all s

Policy Iteration

- Outputs policies at every iteration: $\{\pi_0, \pi_1, \pi_2 \ldots \pi_T\}$
- Different from Value Iteration that was outputting values

Procedure:

1. Start with a random guess π_0 (can be deterministic or stochastic)

Policy Iteration

- Outputs policies at every iteration: $\{\pi_0, \pi_1, \pi_2 ... \pi_T\}$
- Different from Value Iteration that was outputting values

Procedure:

1. Start with a random guess π_0 (can be deterministic or stochastic)
2. For t=0,...,T:
    a. Do **policy evaluation** and compute $Q^{\pi t}$ for all s,a

Remember that $Q^{\pi}(s_t,a) = r_t+\gamma\mathbb{E}_{s'\sim p(.|s,a)}[V^{\pi}(s')]$!

Policy Iteration

- Outputs policies at every iteration: $\{\pi_0, \pi_1, \pi_2 ... \pi_T\}$
- Different from Value Iteration that was outputting values

Procedure:

1. Start with a random guess π_0 (can be deterministic or stochastic)
2. For t=0,...,T:
   a. Do **policy evaluation** and compute $Q^{\pi t}$ for all s,a

Remember that $Q^{\pi}(s_t,a) = r_t+\gamma\mathbb{E}_{s'\sim p(.|s,a)}[V^{\pi}(s')]$!

We can first compute V, for example, and then get Q from that

Policy Iteration

- Outputs policies at every iteration: $\{\pi_0, \pi_1, \pi_2 ... \pi_T\}$
- Different from Value Iteration that was outputting values

Procedure:

1. Start with a random guess π_0 (can be deterministic or stochastic)
2. For t=0,...,T:
   a. Do **policy evaluation** and compute $Q^{\pi t}$ for all s,a

For simplicity and to forget about approximation errors, let's assume we use the exact policy iteration

Policy Iteration

- Outputs policies at every iteration: $\{\pi_0, \pi_1, \pi_2 ... \pi_T\}$
- Different from Value Iteration that was outputting values

Procedure:

1. Start with a random guess π_0 (can be deterministic or stochastic)
2. For t=0,...,T:
    a. Do **policy evaluation** and compute $Q^{\pi t}$ for all s,a

Differently from Value Iteration, here we are outputting Q values of actual policies!

Policy Iteration

- Outputs policies at every iteration: $\{\pi_0, \pi_1, \pi_2 ... \pi_T\}$
- Different from Value Iteration that was outputting values

Procedure:

1. Start with a random guess π_0 (can be deterministic or stochastic)
2. For t=0,...,T:
    a. Do **policy evaluation** and compute $Q^{\pi t}$ for all s,a
    b. Do **policy improvement** as $\pi_{t+1}=\text{argmax}_a Q^{\pi t}(s,a)$ for all s

Policy Iteration

Procedure:

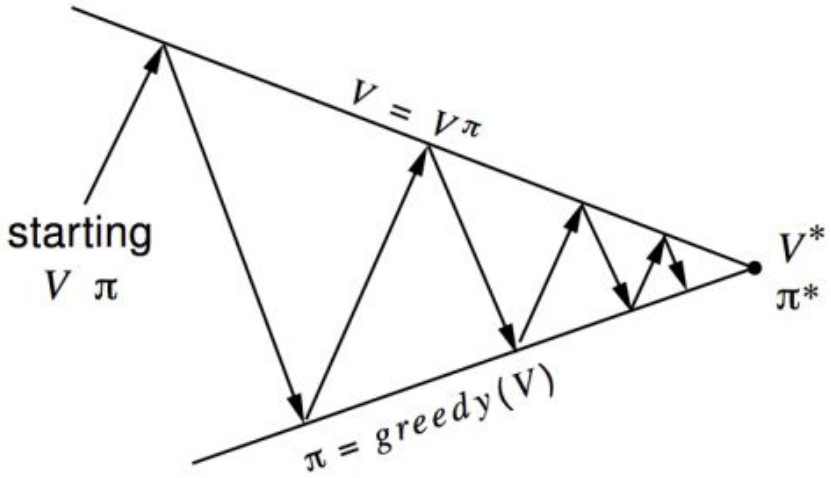
1. Start with a random guess π_0 (can be deterministic or stochastic)
2. For t=0,...,T:
   a. Do **policy evaluation** and compute $Q^{\pi t}$ for all s,a
   b. Do **policy improvement** as π_{t+1}=argmax$_a Q^{\pi t}$(s,a) for all s

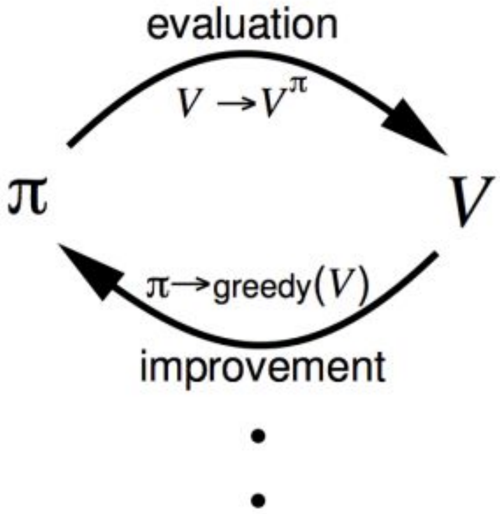
This algorithm only makes progress, and the performance progress of the policy is monotonic

Properties of Policy Iteration

- Monotonic improvement: $Q^{\pi_{t+1}} \geq Q^{\pi_t}$ for all s,a
- Convergence: $\|V^{\pi_i} - V^*\| \leq \gamma^{i+1}\|V^{\pi_0} - V^*\|$

Properties of Policy Iteration



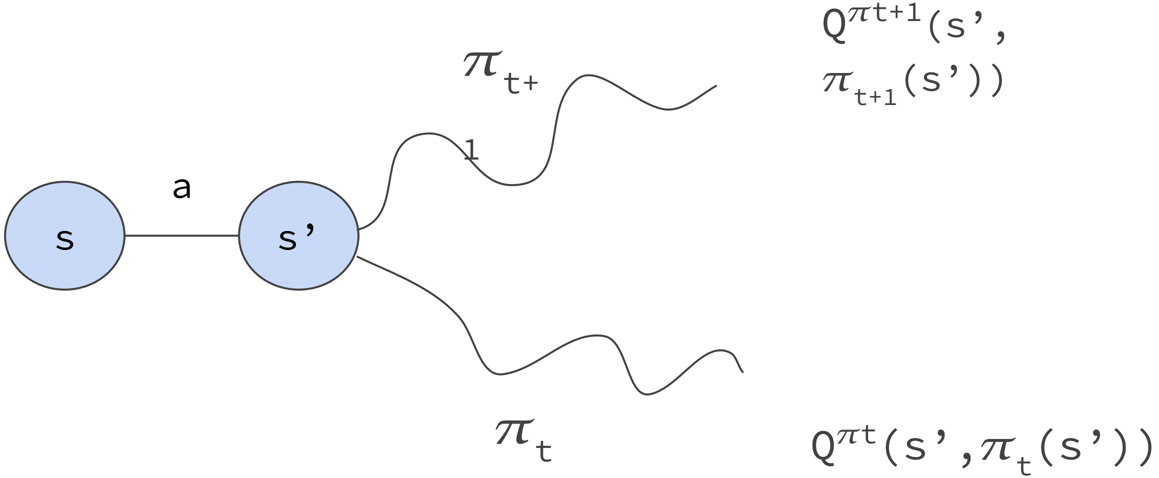


Credits: David Silver

Monotonic Improvement

$\pi_{t+1} = \text{argmax}_a Q^{\pi t}(s,a)$

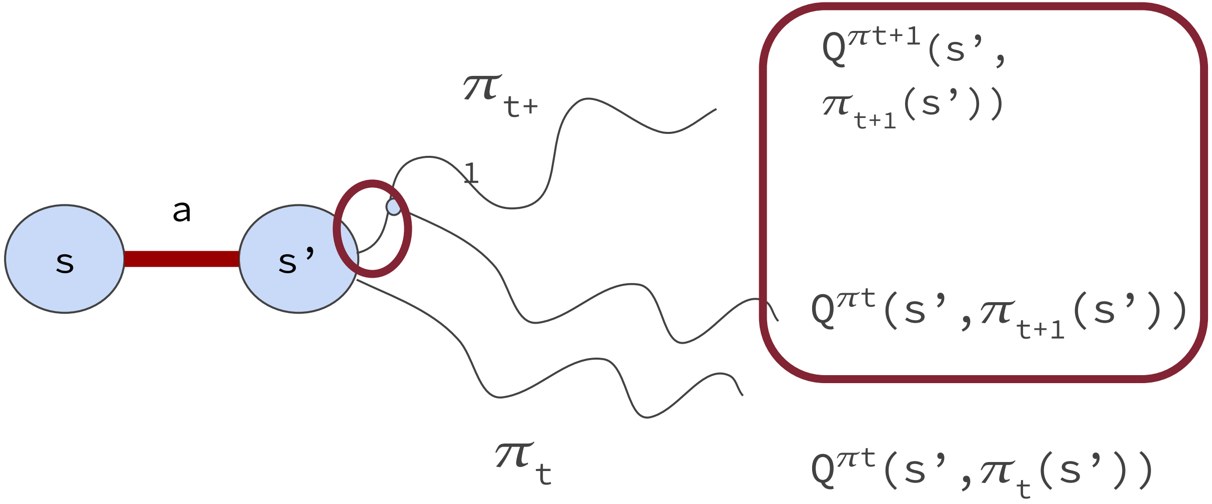
We want to show that $Q^{\pi t+1} \geq Q^{\pi t}$ for all s,a



Monotonic Improvement

$\pi_{t+1} = \text{argmax}_a Q^{\pi t}(s,a)$

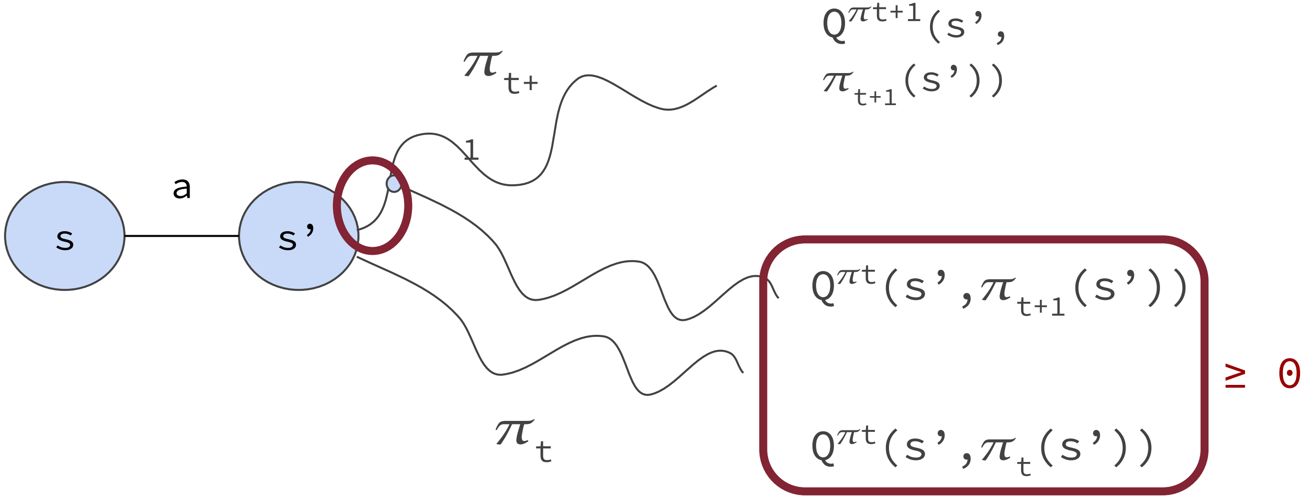
We want to show that $Q^{\pi t+1} \geq Q^{\pi t}$ for all s,a



We are back at the starting point: we can be recursive!

Monotonic Improvement

We want to show that $Q^{\pi t+1} \geq Q^{\pi t}$ for all s,a



since $\pi_{t+1} = \text{argmax}_a Q^{\pi t}(s,a)$

Monotonic Improvement

π_{t+1}=argmax$_a Q^{\pi t}$(s,a
)

We want to show that $Q^{\pi t+1} \geq Q^{\pi t}$ for all s,a

expand definition and simplify r(s,a)

$$Q^{\pi^{t+1}}(s,a) - Q^{\pi^t}(s,a) = \gamma \mathbb{E}_{s' \sim P(s,a)} \left[Q^{\pi^{t+1}}(s', \pi^{t+1}(s')) - Q^{\pi^t}(s', \pi^t(s')) \right]$$

$$= \gamma \mathbb{E}_{s' \sim P(s,a)} \left[Q^{\pi^{t+1}}(s', \pi^{t+1}(s')) - Q^{\pi^t}(s', \pi^{t+1}(s')) + Q^{\pi^t}(s', \pi^{t+1}(s')) - Q^{\pi^t}(s', \pi^t(s')) \right]$$

$$\geq \gamma \mathbb{E}_{s' \sim P(s,a)} \left[Q^{\pi^{t+1}}(s', \pi^{t+1}(s')) - Q^{\pi^t}(s', \pi^{t+1}(s')) \right] \geq \ldots, \geq -\gamma^{\infty}/(1-\gamma) = 0$$

Monotonic Improvement

π_{t+1}=argmax$_a Q^{\pi t}$(s,a
)

We want to show that $Q^{\pi t+1} \geq Q^{\pi t}$ for all s,a

$$Q^{\pi^{t+1}}(s,a) - Q^{\pi^t}(s,a) = \gamma \mathbb{E}_{s' \sim P(s,a)} \left[Q^{\pi^{t+1}}(s', \pi^{t+1}(s')) - Q^{\pi^t}(s', \pi^t(s')) \right]$$

add and subtract the Q of our intermediate policy

$$= \gamma \mathbb{E}_{s' \sim P(s,a)} \left[Q^{\pi^{t+1}}(s', \pi^{t+1}(s')) \underline{- Q^{\pi^t}(s', \pi^{t+1}(s')) + Q^{\pi^t}(s', \pi^{t+1}(s'))} - Q^{\pi^t}(s', \pi^t(s')) \right]$$

$$\geq \gamma \mathbb{E}_{s' \sim P(s,a)} \left[Q^{\pi^{t+1}}(s', \pi^{t+1}(s')) - Q^{\pi^t}(s', \pi^{t+1}(s')) \right] \geq \ldots, \geq -\gamma^{\infty}/(1-\gamma) = 0$$

Monotonic Improvement

π_{t+1}=argmax$_a Q^{\pi t}$(s,a
)

We want to show that $Q^{\pi t+1} \geq Q^{\pi t}$ for all s,a

$$Q^{\pi^{t+1}}(s,a) - Q^{\pi^t}(s,a) = \gamma \mathbb{E}_{s' \sim P(s,a)} \left[Q^{\pi^{t+1}}(s', \pi^{t+1}(s')) - Q^{\pi^t}(s', \pi^t(s')) \right]$$

$$= \gamma \mathbb{E}_{s' \sim P(s,a)} \left[Q^{\pi^{t+1}}(s', \pi^{t+1}(s')) - Q^{\pi^t}(s', \pi^{t+1}(s')) + \underbrace{Q^{\pi^t}(s', \pi^{t+1}(s')) - Q^{\pi^t}(s', \pi^t(s'))}_{\geq 0} \right]$$

$$\geq \gamma \mathbb{E}_{s' \sim P(s,a)} \left[Q^{\pi^{t+1}}(s', \pi^{t+1}(s')) - Q^{\pi^t}(s', \pi^{t+1}(s')) \right] \geq \ldots, \geq -\gamma^{\infty}/(1-\gamma) = 0$$

Monotonic Improvement

$\pi_{t+1}=\text{argmax}_a Q^{\pi t}(s,a)$

We want to show that $Q^{\pi t+1} \geq Q^{\pi t}$ for all s,a

$$Q^{\pi^{t+1}}(s,a) - Q^{\pi^t}(s,a) = \gamma \mathbb{E}_{s' \sim P(s,a)} \left[Q^{\pi^{t+1}}(s', \pi^{t+1}(s')) - Q^{\pi^t}(s', \pi^t(s')) \right]$$

$$= \gamma \mathbb{E}_{s' \sim P(s,a)} \left[Q^{\pi^{t+1}}(s', \pi^{t+1}(s')) - Q^{\pi^t}(s', \pi^{t+1}(s')) + Q^{\pi^t}(s', \pi^{t+1}(s')) - Q^{\pi^t}(s', \pi^t(s')) \right]$$

$$\geq \gamma \mathbb{E}_{s' \sim P(s,a)} \left[Q^{\pi^{t+1}}(s', \pi^{t+1}(s')) - Q^{\pi^t}(s', \pi^{t+1}(s')) \right] \geq \ldots, \geq -\gamma^{\infty}/(1-\gamma) = 0$$

Properties of Policy Iteration

- Monotonic improvement: $Q^{\pi_{t+1}} \geq Q^{\pi_t}$ for all s,a
- Convergence: $||V^{\pi_i} - V^*|| \leq \gamma^{i+1}||V^{\pi_0} - V^*||$

Convergence? Prove it yourselves!

Properties of Policy Iteration

- Monotonic improvement: $Q^{\pi t+1} \geq Q^{\pi t}$ for all s,a
- Convergence: $||V^{\pi i} - V^*|| \leq \gamma^{i+1}||V^{\pi 0} - V^*||$

Complexity $O(S^3+S^2A)$

Properties of Policy Iteration

- Monotonic improvement: $Q^{\pi_{t+1}} \geq Q^{\pi_t}$ for all s,a
- Convergence: $\|V^{\pi_i} - V^*\| \leq \gamma^{i+1}\|V^{\pi_0} - V^*\|$

Is there a max number of iterations of policy iteration?

$|A|^{|S|}$ since that is the maximum number of policies, and as the policy improvement step is monotonically improving, each policy can only appear in one round of policy iteration unless it is an optimal policy

Properties of Policy Iteration

- Monotonic improvement: $Q^{\pi_{t+1}} \geq Q^{\pi_t}$ for all s,a
- Convergence: $||V^{\pi_i} - V^*|| \leq \gamma^{i+1}||V^{\pi_0} - V^*||$

When do we stop?

if the policy does not change anymore for any state

We Did Dynamic Programming!

Dynamic Programming is a method for solving complex problems by breaking them down into subproblems:

- Solve the subproblems
- Combine solutions to subproblems

We Did Dynamic Programming!

Dynamic Programming can be applied if we have:

- *Optimal substructure:* Optimality exists and the optimal solution can be decomposed into subproblems
- *Overlapping subproblems:* Subproblems recur many times and the solutions can be cached and reused

MDPs satisfy both properties: thanks Bellman equation!

We Did Dynamic Programming!

We applied dynamic programming for **planning** as we assumed to know the MDP transition probabilities

Problem	Bellman Equation	Algorithm
Prediction	Bellman Expectation Equation	Iterative Policy Evaluation
Control	Bellman Expectation Equation + Greedy Policy Improvement	Policy Iteration
Control	Bellman Optimality Equation	Value Iteration

Credits: David Silver

Primal Linear Program

As an alternative to VI and PI

Consider the Bellman optimality equation

$$V(s) = \max_a\{r_t+\gamma\mathbb{E}_{s'\sim p(.|s,\pi(s))}[V(s')]\}$$

and write it as a linear program:

$$\min V(s)$$

$$\text{such that } V(s)\geq r_t+\gamma\mathbb{E}_{s'\sim p(.|s,\pi(s))}[V(s')] \text{ for all } s,a$$

Primal Linear Program

$$\min V(s)$$

$$\text{such that } V(s) \geq r_t + \gamma \mathbb{E}_{s' \sim p(.|s,\pi(s))}[V(s')] \text{ for all } s,a$$

Using a LP solver we can get a solution which is V^*

(not used a lot in practice)

Primal Linear Program

$$\min V(s)$$

$$\text{such that } V(s) \geq F(V) \text{ for all } s,a$$

Any feasible solution must satisfy $V \geq F(V) \geq F(F(V)) \geq \ldots \geq F^{\infty}V \geq V^*$

Dual Linear Program

There is also a dual linear program, that finds the solution directly in policy space