**IIIT Hyderabad** **Scribed By: Adithya Avvaru, Dishant Chauhan**
**Topics in Machine Learning (CSE975)** **Lecture #14**
**Instructor: Dr. Naresh Manwani** **September 26, 2017**

---

**Outline.** *Policy Iteration Algorithm followed by the proof of convergence. We discuss the proof that this algorithm generates a sequence of policies with non-decreasing state values. An example is also discussed.*

# 1 Policy Iteration Algorithm

Consider a finite MDP with $N$ states. The Policy Iteration algorithm has two major steps - Policy Evaluation and Policy Improvement. The complete description of policy iteration is given in Algorithm **??**.

**Algorithm 1** Policy Iteration Algorithm

**Initialization:** $\pi_0$: an stationary policy; set $k = 0$

**for** $k = 1, 2, ..., K$ **do**

**Policy Evaluation:** Given $\pi_k$, compute state values $V^{\pi_k} = [v_{\pi_k}(s_1) \ \dots \ v_{\pi_k}(s_N)]^T$ as:

$$v_{\pi_k}(s_i) = \sum_{a \in \mathcal{A}(s_i)} \pi_k(a|s_i) \left[ r(s_i, a) + \gamma \sum_{j=1}^{N} p(s_j|s_i, a) v_{\pi_k}(s_j) \right]$$

**Policy Improvement :** Find the new policy $\pi_{k+1}$ as:

$$\pi_{k+1}(s_i) = \underset{a \in \mathcal{A}(s_i)}{\operatorname{argmax}} \ \ r(s_i, a) + \gamma \sum_{j=1}^{N} p(s_j|s_i, a) v_{\pi_k}(s_j)$$

Return the last policy $\pi_* = \pi_K$

---

We stop iterating when the policies don't change. Let $K$ be the smallest index such that $\pi_K = \pi_{K+1}$. Then we stop after $K$ iteration. This algorithm will essentially generate a sequence of monotonically improving policies and value functions, the proofs of which are explained in the next sections. In the flow diagram below, E refers to Policy Evaluation and I refers to Policy Improvement. In

$$\pi_0 \xrightarrow{\text{E}} v_{\pi_0} \xrightarrow{\text{I}} \pi_1 \xrightarrow{\text{E}} v_{\pi_1} \xrightarrow{\text{I}} \pi_2 \xrightarrow{\text{E}} \cdots \xrightarrow{\text{I}} \pi_* \xrightarrow{\text{E}} v_*$$

the finite MDP case, state space and action space are finite. Then maximum number of possible policies in Policy Iteration algorithm is $|\mathcal{A}|^{|\mathcal{S}|}$ where $|\mathcal{A}|$ is the cardinality of set of actions and $|\mathcal{S}|$ is the cardinality of set of states. In each iteration, we are changing the policy only when we are improving. Otherwise, we stop the algorithm. That means, no policy will be repeated. Hence, the algorithm converges in finite set of steps as we have only finite set of policies. But, the bigger question is 'Will it converge to optimal policy?'

## 2 Convergence and Performance Guarantee of the algorithm

**Theorem 1. Convergence Theorem** *Let $T$ be the $\gamma$-Contraction Mapping $T : \mathbb{R}^N \to \mathbb{R}^N$ and $V_{n+1} = T(V_n)$, then*

$$\lim_{n\to\infty} V_n = V_\gamma^*$$

*where $V_\gamma^*$ is unique fixed point of $T$, which means $T(V_\gamma^*) = V_\gamma^*$.*

**Proof:** To prove this, we first show that the sequence $V_{n+1} = T(V_n)$ is a Cauchy sequence.[1] Now consider,

$$\begin{aligned}
\|V_{n+m} - V_n\|_\infty &= \|\sum_{k=0}^{m-1}(V_{n+k+1} - V_{n+k})\|_\infty \\
&\leq \sum_{k=0}^{m-1}\|V_{n+k+1} - V_{n+k}\|_\infty && \text{using triangle inequality} \\
&= \sum_{k=0}^{m-1}\|T(V_{n+k}) - T(V_{n+k-1})\|_\infty \\
&\leq \sum_{k=0}^{m-1}\gamma\|V_{n+k} - V_{n+k-1}\|_\infty && \because T \text{ is a } \gamma - \text{contraction mapping} \\
&\leq \sum_{k=0}^{m-1}\gamma^{n+k}\|V_1 - V_0\|_\infty \\
&= \frac{\gamma^n(1-\gamma^m)}{1-\gamma}\|V_1 - V_0\|_\infty
\end{aligned}$$

For $\gamma < 1$, $\lim_{n\to\infty}\frac{\gamma^n(1-\gamma^m)}{1-\gamma}\|V_1 - V_0\|_\infty = 0$. Thus, $\forall\epsilon > 0, \exists n_0$ such that $\forall n \geq n_0$,

$$\|V_{n+m} - V_n\|_\infty \leq \epsilon$$

This shows that the sequence $V_{n+1} = T(V_n)$ is a Cauchy Sequence. Thus, the sequence will converge. Let $V_\gamma^*$ be the limit. Thus,

$$\lim_{n\to\infty} T^n(V_0) = V_\gamma^*$$

Now, we show that $V_\gamma^*$ is a fixed point, i.e.

$$T(V_\gamma^*) = V_\gamma^*$$

[1] ***Def: Cauchy Sequence*** : A sequence whose elements become arbitrary close to each other as the sequence progresses. Mathematically,

$$\forall\epsilon > 0 \quad \exists n_0 \quad s.t. \quad \|V_{n+m} - V_n\| \leq \epsilon \quad \forall n \geq n_0$$

Thus,

$$\begin{aligned}\|T(V_\gamma^*) - V_\gamma^*\| &= \|T(V_\gamma^*) - V_{n+1} + V_{n+1} - V_\gamma^*\| \\ &\leq \|T(V_\gamma^*) - V_{n+1}\| + \|V_{n+1} - V_\gamma^*\| \\ &= \|T(V_\gamma^*) - TV_n\|_\infty + \|V_{n+1} - V_\gamma^*\|_\infty \\ &\leq \gamma\|V_\gamma^* - V_n\|_\infty + \|V_{n+1} - V_\gamma^*\|_\infty\end{aligned}$$

But, $\lim_{n\to\infty} V_n = V_\gamma^*$. Thus, for an $\epsilon > 0$, $\exists n_0(\epsilon)$ such that

$$\|V_n - V_\gamma^*\|_\infty \leq \epsilon, \ \ \forall n \ \geq \ n_0(\epsilon)$$

Thus, $\forall n \ \geq \ n_0(\epsilon)$,

$$\|T(V_\gamma^*) - V_\gamma^*\|_\infty \leq \gamma\epsilon + \epsilon = \epsilon(\gamma + 1)$$

But, the above inequality holds for any arbitrary small $\epsilon > 0$. Thus,

$$\begin{aligned}\|T(V_\gamma^*) - V_\gamma^*\|_\infty &= 0 \\ T(V_\gamma^*) &= V_\gamma^*\end{aligned}$$

Thus $V_\gamma^*$ is the fixed point of $T$. Now, we show that $V_\gamma^*$ is unique fixed point of $T$. We show it by contradiction. Let $V_\gamma^*$ and $V_\gamma^{**}$ be two fixed points, then

$$\begin{aligned}\|V_\gamma^* - V_\gamma^{**}\|_\infty &= \|T(V_\gamma^*) - T(V_\gamma^{**})\|_\infty && \text{using the definition of fixed point} \\ &\leq \gamma\|V_\gamma^* - V_\gamma^{**}\|_\infty && \because \ T \text{ is a } \gamma - \text{contraction operator}\end{aligned}$$

This implies that $\gamma > 1$. Which is a contradiction. Thus $V_\gamma^*$ is a unique fixed point of $T$.

### 2.1 Performance Guarantee

The two-step Policy Iteration algorithm calculates state values, $V_{\pi_k}$ in every iteration $k$ and improves on the policy $\pi_k$ to $\pi_{k+1}$ using these state values until the two policies are equal or upto $K$ iterations. But, it is important to prove that there is performance improvement in policies and state values when iterating over $k$. The following theorem proves this performance guarantee.

**Theorem 2.** *The Policy Iteration algorithm generates a sequence of policies with non-decreasing performance i.e.*

$$V^{\pi_{k+1}} \ \geq \ V^{\pi_k}, \ \ \forall k = 1 \ldots K$$

*where $V^{\pi_k} \in \mathbb{R}^N$ is the state value vector corresponding to policy $\pi_k$.*

*Proof.* Let $F^{\pi_k}$ be the Bellman's Expectation Operator corresponding to policy $\pi_k$ and $V^{\pi_k}$ be its fixed point. Thus, $F^{\pi_k}(V^{\pi_k}) = V^{\pi_k}$. Similarly, let $F$ be the Bellman's Optimality Operator.

$$\begin{aligned}V^{\pi_k} &= F^{\pi_k}(V^{\pi_k}) && V^{\pi_k} \text{ is fixed point of } F^{\pi_k} \\ &\leq F(V^{\pi_k}) && \text{by the definition of Optimality Operator} \\ &= F^{\pi_{k+1}}(V^{\pi_k}) && \text{from policy improvement step}\end{aligned}$$

Applying monotonicity property of Bellman Expectation Operator $n-1$ times, we get

$$V^{\pi_k} \leq F^{\pi_{k+1}}(V^{\pi_k}) \leq (F^{\pi_{k+1}})^2(V^{\pi_k}) \leq .... \leq (F^{\pi_{k+1}})^n(V^{\pi_k})$$

Since $F^{\pi_{k+1}}$ is $\gamma$-contraction mapping, $(F^{\pi_{k+1}})^n(V^{\pi_k})$ converges to $V^{\pi_{k+1}}$, fixed point of $F^{\pi_{k+1}}$. Thus, we can conclude that

$$V^{\pi_k} \leq V^{\pi_{k+1}}$$

□

# 3 Example Of Policy Iteration

Consider a finite MDP as described in Figure **??**.

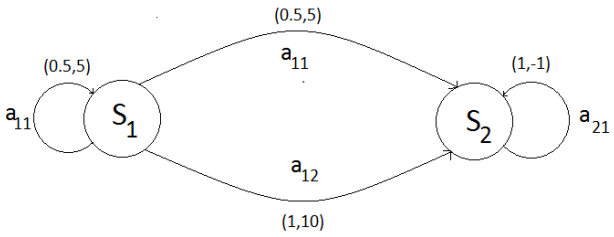


Figure 1: Example of MDP with two states, $s_1$ and $s_2$ i.e. $\mathcal{S} = \{s_1, s_2\}$. The action set is $\mathcal{A} = \{a_{11}, a_{12}, a_{21}\}$. Only transitions with non-zero probabilities are represented. Each transition is labeled with the action taken followed by a pair $(p, r)$, where $p$ is the probability of the transition and $r$ the expected reward for taking that transition.

- Possible Actions at states $s_1$ and $s_2$ are given by

$$\mathcal{A}(s_1) = \{a_{11}, a_{12}\}$$
$$\mathcal{A}(s_2) = \{a_{21}\}$$

- Transition probabilities are

$$p(s_1|s_1, a_{11}) = 0.5$$
$$p(s_2|s_1, a_{11}) = 0.5$$
$$p(s_2|s_1, a_{12}) = 1$$
$$p(s_2|s_2, a_{21}) = 1$$

- Rewards $r(s, a, s')$ are given by

$$r(s_1, a_{11}, s_1) = 5$$
$$r(s_1, a_{11}, s_2) = 5$$
$$r(s_1, a_{12}, s_2) = 10$$
$$r(s_2, a_{21}, s_2) = -1$$

Let $\gamma = 0.95$ and the initial policy $\pi_0$ be as follows.

$$\pi_0(s_1) = a_{12}$$
$$\pi_0(s_2) = a_{21}$$

**Iteration 1**:
***Step 1: Policy Evaluation***

$$v_{\pi_0}(s_1) = 10 + 0.95 * [1 * v_{\pi_0}(s_2)]$$
$$v_{\pi_0}(s_2) = -1 + 0.95 * [1 * v_{\pi_0}(s_2)]$$

Solving these equations, we get

$$v_{\pi_0}(s_1) = -9$$
$$v_{\pi_0}(s_2) = -20$$

***Step 2: Policy Improvement***

$$\begin{aligned} q_{\pi_0}(s_1, a_{11}) &= r(s_1, a_{11}) + \gamma[p(s_1|s_1, a_{11})v_{\pi_0}(s_1) + p(s_2|s_1, a_{11})v_{\pi_0}(s_2)] \\ &= 5 + 0.95 * [0.5 * (-9) + 0.5 * (-20)] = -8.775 \\ q_{\pi_0}(s_1, a_{12}) &= r(s_1, a_{12}) + \gamma[p(s_1|s_1, a_{12})v_{\pi_0}(s_1) + p(s_2|s_1, a_{12})v_{\pi_0}(s_2)] \\ &= 10 + 0.95 * [1 * (-20)] = -9 \\ q_{\pi_0}(s_2, a_{21}) &= r(s_2, a_{21}) + \gamma[p(s_1|s_2, a_{21})v_{\pi_0}(s_1) + p(s_2|s_1, a_{21})v_{\pi_0}(s_2)] \\ &= -1 + 0.95 * [1 * (-20)] = -20 \end{aligned}$$

Thus,

$$\pi_1(s_1) = \arg \max_{a \in \mathcal{A}(s_1)} q_{\pi_0}(s_1, a) = a_{11}$$
$$\pi_1(s_2) = \arg \max_{a \in \mathcal{A}(s_2)} q_{\pi_0}(s_2, a) = a_{21}$$

Using $\pi_1$, we repeat step-1 and step-2 for next iteration. We see that $\pi_2 = \pi_1$. Thus, we stop and $\pi_* = \pi_2$.

# References


[1] Richard S. Sutton and Andrew Barto, Reinforcement Learning: An Introduction, Second Edition, The MIT Press, 2012.

[2] Martin L. Puterman, Markov Decision Processes: Discrete Stochastic Dynamic Programming, 1st Edition, John Wiley & Sons, Inc. New York, NY, USA, 1994.