EE675A - IITK, 2022-23-II

Lecture 12: Policy and Value Iteration Algorithms in RL

27th February, 2023

Lecturer: Subrahmanya Swamy Peruru *Scribe: Vivek Agrawal & Vikash Kumar*

1 Recap and Overview

In the last few lectures, we have been discussing about how the problems are formulated as Markov Decision Processes (MDP). We also defined various terms used in MDPs like state (S_t), action (A_t), reward (R_t) and return (G_t). These all together help in formulating the Markov Decision Processes. We also derived Bellman Expectation Equations and Bellman Optimality Equations.

1.1 Bellman Expectation Equations

$$V_\pi(s) = \sum_{a,s',r} P(a, s', r|s)(r + \gamma V_\pi(s')) \quad \forall s, s' \in S, a \in A, r \in R$$

1.2 Bellman Optimality Equations

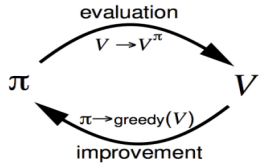
$$V^*(s) = \max_a (R_s^a + \gamma \sum_{s'} V^*(s') P_{ss'}^a) \quad \forall s, s' \in S, a \in A, r \in R$$

where , A and S are action and state sets respectively.

Today, we will discuss the two dynamic programming methods, namely policy iteration and value iteration to find π^*. Both of these methods require the model for MDPs to be given.

2 Policy Iteration

In Policy Iteration, we start with an arbitrary policy (π_0) and use Policy Evaluation (PE) algorithm to find the value function (v_{π_0}) for the chosen policy. Then we use Policy Improvement (PI) algorithm to find a better policy (π_1). We will prove this fact also.



2.1 Policy Evaluation (PE)

From the Bellman Expectation Equations, we derived the following :

$$V_{\pi_0} = R^{\pi_0} + \gamma P^{\pi_0} V_{\pi_0} \qquad ...(1)$$

$$V_{\pi_0} = (1 - \gamma P^{\pi_0})^{-1} R^{\pi_0} \qquad ...(2)$$

Solving these set of equations for all states gives us the value function V_{π_0}. But since solving these equations can become too complex for a MDP with large state space, we use **Iteration** to evaluate the value function.
In this we start with an arbitrary value function $V_0 \in R^n$ and iteratively compute value function(V_k) for the kth iteration using

$$V_{k+1} = R^{\pi_0} + \gamma P^{\pi_0} V_k \quad \forall k \geq 0$$

We compute V_k iteratively until $V_{k+1} = V_k$, which denotes convergence of the value function. This then satisfies the Bellman Expectation Equations which we know have unique solution for a finite MDP. Thus, this gives the value function for policy π_0 as $V_{\pi_0} = V_k$.

2.2 Policy Improvement (PI)

To find the policy from the above evaluated Value Function, we use the following equation.

$$\pi_1(s) = \underset{a}{\operatorname{argmax}}\, q_{\pi_0}(s, a) \quad \forall s \in S$$

Claim: π_1 is a better policy than π_0 i.e. $V_{\pi_1} \geq V_{\pi_0}$ ...(3)

Proof: From equation 3, we have

$$\max_a q_{\pi_0}(s, a) = q_{\pi_0}(s, \pi_1(s)) \qquad ...(4)$$

Suppose at time = t, agent is at state $S_{t+1} = s$. We have the value function for this state in terms of state action value function as follows:

$$V_{\pi_0}(s) = q_{\pi_0}(s, \pi_0(s)) \leq q_{\pi_0}(s, \pi_1(s)) \qquad from\ eqn(4)$$

This tells us that on choosing the next immediate action using policy pi_1 and thereafter following the old policy π_0 gives a higher expected return.
Now suppose the agent on taking the action $a = \pi_1(s)$ goes to state $S_{t+1} = s'$. Again at this step, from the same explanation as above, if we take the immediate action according to policy pi_1 and then follow policy π_0, then this will give higher expected return.
Thus, arguing similarly recursively for all the future states must follow policy π_1 to get a higher expected return. Hence, proved $V_{\pi_1} \geq V_{\pi_0}$ and π_1 is a better policy compared to π_0. This motivates us to know a general theorem (stated below) concerning any two deterministic policies.

Policy Improvement Theorem:

Let π and π' be two deterministic policies of a finite MDP such that

$$q_\pi(s, \pi'(s)) \geq V_\pi(s) \quad \forall s \in S$$

Then,

$$V_{\pi'} \geq V_\pi$$

Proof: Since

$$V_\pi(s) \leq q_\pi(s, \pi'(s)) \quad \forall s \in S$$

Putting expression for state value function q_π in terms of value function, we get

$$V_\pi(s) \leq R^{\pi'} + \gamma P^{\pi'} V_\pi \quad \forall s \in S$$

$$(1 - \gamma P^{\pi'}) V_\pi(s) \leq R^{\pi'} \quad \forall s \in S$$

$$V_\pi(s) \leq (1 - \gamma P^{\pi'})^{-1} R^{\pi'} \quad \forall s \in S$$

$$V_\pi(s) \leq V_{\pi'}(s) \quad \forall s \in S$$

On iteratively solving, if $V_{\pi'} = V_\pi$, then V_π is the optimal value function (from Bellman Optimality Equations). Since there are total $|A|$ actions possible for each state, we have a total number of possible policies equal to $|A|^{|S|}$ resulting in time complexity of $O(|A|^{|S|})$.

3 Generalised Policy Iteration (GPI)

An optimal policy can be attained using Generalized Policy Iteration (GPI), which combines algorithms like Value Iteration or Policy Iteration.
The algorithm starts with a random policy and value function. From here, algorithm evaluates the policy to search for an improved value function, avoiding finding the optimal value function which might take very large number of iterations. From this improved value function, we improve the policy and repeat the previous steps and once the optimal value function is reached, a single iteration of Policy Improvement is performed to get an optimal policy.

4 Value Iteration : GPI with only one step of PE

The value Iteration starts with a random value function based on Bellmans equation to find a policy that obtains high rewards in the long term. Being an iterative process, value iteration keeps searching for an improved value function to find the optimal value function. After the optimal value function is obtained, a single iteration of Policy Improvement(PE) is performed based on the optimal value function resulting in an optimal policy.

Algorithm: Let Initial policy be π_0 and value function be V_0
1. Perform PI step to find the improved policy π_1 using the following equation,

$$\pi_1(s) = \underset{a}{\operatorname{argmax}} \left(R_s^a + \gamma \sum_{s'} P_{ss'}^a V_0(s') \right)$$

2. Perform the PE step to find the improved value function V_1 using the following equation,

$$V_1(s) = \underset{a}{\operatorname{argmax}} \left(R_s^{\pi_1(s)} + \gamma \sum_{s'} P_{ss'}^{\pi_1(s)} V_0(s') \right)$$

3. Combine both steps 1 and 2 to eliminate the use of policy completely and get the equation to find the improved value function just from the previous value function,

$$V_1(s) = \max_a R_s^a + \gamma \sum_{s'} P_{ss'}^a V_0(s')$$

4. Perform step 3 iteratively, until $V_1 = V_0$ Since this equation is same as Bellman Optimality Equation, we can argue that if we get $V_1 = V_0$, we have an optimal value function.
5. Find the optimal policy from this using PI.

5 Convergence Proof for Policy Evaluation

- Contraction Mapping
- Complete Metric Space
- Banach Fixed Point Theorem.

These three concepts will help us understand the convergence proof for policy iteration.

5.1 Contraction Mapping

Let V be a vector space (R^n)
Let d(x,y) be a distant metric for any two vectors $x, y \in V$
For example, consider this

$$d(x, y) = \max_s ||x(s) - y(s)||$$

Then f is a Contraction Mapping if,

$$d(f(x), f(y)) \leq cd(x, y) \quad c \in (0, 1)$$

5.2 Complete Metric Space

Consider a metric space (V, d) where V is a vector Space and d is a distance metric.
If every Convergent sequence possible in V has its limit also in V, then the metric space is called Complete Metric Space.
For example,

$$x_n = \frac{1}{n+1} \forall n \in N$$

is a sequence in vector space $V = (0, 1)$ converges to 0, which doesn't belong to V, therefore the metric space with this vector space is not complete.

5.3 Banach Fixed Point theorem

Let (V,d(x,y)) be a complete matrix space and let $f : V \to V$ be a contraction mapping, then

- There exists a fixed point for f, i.e. $\exists x^* \; s.t. \; f(x^*) = x^*$
- The fixed point is unique
- $X_{k+1} = f(X_k)$ will converge to a fixed point from any arbitrary $X_0 \in V$

We know that $V_{k+1} = R^\pi + \gamma P^\pi V_k$ and we have to prove that $f_\pi(V) = R^\pi + \gamma P^\pi V$.
Proof: Let $u, v \in R^n$

$$d(f_\pi(v), f_\pi(u) \leq cd(u, v), c \in (0, 1)$$

This gives

$$\max_s |f_\pi(v(s)) - f_\pi(u(s))| \leq c \max_{s'} |v(s') - u(s')|$$

Consider the term $|f_\pi(v(s)) - f_\pi(u(s))|$. Simplifying it more below,

$$= \gamma | \sum_{s'} P^\pi_{ss'}(v(s') - u(s'))|$$

$$= \gamma \sum_{s'} P^\pi_{ss'} |(v(s') - u(s'))|$$

Since d is defined as max over all states

$$\leq \gamma \sum_{s'} P^\pi_{ss'} d(u, v)$$

$$= \gamma d(u, v)$$

Putting this simplified expression, we get

$$\max_s |f_\pi(v(s)) - f_\pi(u(s))| \leq \gamma d(u, v) \leq c \max_{s'} |v(s') - u(s')|$$

This gives

$$c = \gamma$$

.