AI 3000 / CS 5500 : Reinforcement Learning

Assignment № 2

Due Date : 17/09/2022

Course Instructor : Easwar Subramanian 07/09/2022

Problem 1 : Value Iteration

Let M be an MDP given by $< \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma >$ with $|\mathcal{S}| < \infty$ and $|\mathcal{A}| < \infty$ and $\gamma \in [0, 1)$. We are given a policy π and the task is to evalute $V^\pi(s)$ for every state $s \in \mathcal{S}$ of the MDP. To this end, we use the iterative policy evalution algorithm. It is the analog of the algorithm described in (**Refer : Slides 15, 20 of Lecture 7**) for the policy evaluation case. We start the iterative policy evaluation algorithm with an initial guess V_1 and let V_{k+1} be the $k+1$-th iterate of the value function corresponding to policy π. Our constraint on compute infrastructure does not allow us to wait for the successive iterates of the value function to converge to the true value function V^π given by $V^\pi = (I - \gamma P)^{-1} R$. Instead, we let the algorithm terminate at time step $k+1$ when the distance between the sucessive iterates given by $\|V_{k+1} - V_k\|_\infty \leq \varepsilon$ for a given $\varepsilon > 0$.

(a) Prove that the error estimate between the obtained value function estimate V_{k+1} and true value function V^π is given by

$$\|V_{k+1} - V^\pi\|_\infty \leq \frac{\varepsilon\gamma}{1-\gamma}$$

(5 Points)

The last iterate of the algorithm is V_{k+1} and we know that $\|V_{k+1} - V_k\|_\infty \leq \varepsilon$. By using the triangular inequality (of norms) and by using the fact $BV_k = V_{k+1}$ where B is the Bellman evaluaiton backup, we have,

$$\begin{aligned} \|V_k - V\|_\infty &\leq \|V_k - V_{k+1}\|_\infty + \|V_{k+1} - V\|_\infty = \|V_k - V_{k+1}\|_\infty + \|BV_k - BV\|_\infty \\ &\leq \|V_k - V_{k+1}\|_\infty + \gamma\|V_k - V\|_\infty = \varepsilon + \gamma\|V_k - V\|_\infty \end{aligned}$$

Therefore, $\|V_k - V\|_\infty \leq \frac{\varepsilon}{1-\gamma}$. This allows us to conclude that,

$$\|V_{k+1} - V\|_\infty = \|BV_k - BV\|_\infty \leq \gamma\|V_k - V\|_\infty \leq \frac{\gamma\varepsilon}{1-\gamma}$$

(b) Prove that the iterative policy evaluation algorithm converges geometrically, i.e.

$$\|V_{k+1} - V^\pi\|_\infty \leq \gamma^k \|V_1 - V^\pi_\infty\|$$

(2 Points)

We already proved that,

$$\|V_{k+1} - V\|_\infty \leq \gamma \|V_k - V\|_\infty$$

,

Applying this inequality recursively, we get the desired result

(c) Let v denote a value function and consider the Bellman optimallity operator given by,

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

Prove that the Bellman optimality operator ($\mathcal{L}$) satisfies the monotoniciity property. That is, for any two value functions u and v such that $u \leq v$ (this means, $u(s) \leq v(s)$ for all $s \in \mathcal{S}$), we have $\mathcal{L}(u) \leq \mathcal{L}(v)$ (3 Points)

By the defintion of Bellman optimality operator $\mathcal{L}$, for a fixed state $s \in \mathcal{S}$ and for action set $\mathcal{A}$ finite, one can conlude that there exists $a_1, a_2 \in \mathcal{A}$ (with a_1 and a_2 possibly different) such that,

$$L(u(s)) = \left[\mathcal{R}(s, a_1) + \gamma \sum_{s'} P(s'|s, a_1) u(s)\right]$$

and

$$L(v(s)) = \left[\mathcal{R}(s, a_2) + \gamma \sum_{s'} P(s'|s, a_2) v(s)\right]$$

It is then easy to observe (using the defintion of optimality operator) that,

$$L(v(s)) \geq \left[\mathcal{R}(s, a_1) + \gamma \sum_{s'} P(s'|s, a_1) v(s)\right]$$

Now, we have,

$$L(u(s)) - L(v(s)) \leq \left[\gamma \sum_{s'} P(s'|s, a_1) \left(u(s) -)v(s)\right)\right]$$

Since, $u(s) \leq v(s)$, we have,

$$L(u(s)) - L(v(s)) \leq \left[\gamma \sum_{s'} P(s'|s, a_1) \left(u(s) -)v(s)\right)\right] \leq \left[\gamma \sum_{s'} P(s'|s, a_1) \left(v(s) - v(s)\right)\right] = 0$$

Since s was choosen arbitarily, we have the desired result.

Problem 2 : On Contractions

(a) Let P and Q be two contractions defined on a normed vector space $< \mathcal{V}, \|\cdot\| >$. Prove that the compositions $P \circ Q$ and $Q \circ P$ are contractions on the same normed vector space. (3 Points) Indeed, for all , $v, u \in \mathcal{V}$, we have,

$$\|P \circ Q(v) - P \circ Q(u)\| = \|P(Q(v)) - P(Q(u))\| \leq \gamma_p \|Q(v) - Q(u)\| \leq \gamma_p \gamma_q \|v - u\|$$

It is importrant to note that since γ_p and γ_q belong to $[0, 1)$, hence the product $\gamma_p \gamma_q$ also belong to $[0, 1)$.

(b) What can be suitable contraction (or LIpschitz) coeffecients for the contractions $P \circ Q$ and $Q \circ P$? (1 point)

$\gamma_p \gamma_q$ is the contraction co-efficient

(c) Define operator $\mathcal{B}$ as $\mathcal{F} \circ \mathcal{L}$ where $\mathcal{L}$ is the Bellman optimality operator and $\mathcal{F}$ is any other suitable operator. For example, $\mathcal{F}$ could play the role of a function approximator to the Bellman backup $\mathcal{L}$. Under what conditions would the value iteration algorithm converge to a unique solution if operator $\mathcal{B}$ is used in place of $\mathcal{L}$ (in the value iteration algorithm) ? Explain your answer. (2 Points)

For unique solution to exist, the operator $\mathcal{F} \circ \mathcal{L}$ must be a contraction. This composition can be contraction only when both $\mathcal{F}$ and $\mathcal{L}$ is a contraction under the max-norm.

Problem 3 : Monte Carlo Methods

Consider a Markov process with 2 states $\mathcal{S} = \{S, A\}$ with transition probablities as shown in the table below, where $p \in (0, 1)$ is a non-zero probablity. To generate a MRP from this Markov chain, assume that the rewards for being in states S and A are 1 and 0, respectively. In addition, let the discount factor of the MRP be $\gamma = 1$.

	S	A
S	$1-p$	p
A	0	1

(a) Provide a generic form for a typical trajectory starting at state S. (1 Point)

(b) Estimate $V(S)$ using first visit MC. (2 Points)

(c) Estimate $V(S)$ using every visit MC. (2 Points)

(d) What is true value of $V(S)$? (3 Points)

(e) Explain if the every vist MC estimate is biased. (2 Points)

(f) In general for a MRP, comment on the convergence properties of the first visit MC and every visit MC algorithms (2 Points)

(a) All trajectories starting in state S look like $SSSS \cdots A$ where there are l occurences of S.

(b) First time estimate of $V(S)$ would be l.

(c) There are l every time estimates of $V(S)$ for every trajectory and their sum is $l(l+1)/2$. So, everytime estimate per trajectory is given by $(l+1)/2$

(d) The true value of $V(S)$ is $1/p$ as it is the average length of the trajectory.

(e) Yes. Expected value of the everytime estimate $((1/p)+1)/2$ which is erroneous by a factor of 2, if p were small

(f) FVMC convergence rely just on law of large numbers; whereas EVMC convergence may need to more sophisticated care since the samples for mean calculation are independent.

Problem 4 : Temporal Difference Methods

Given a policy π of an MDP M, consider the one step TD error given by

$$\delta_t = r_{t+1} + \gamma V^\pi(s_{t+1}) - V^\pi(s_t)$$

where (s_t, r_{t+1}, s_{t+1}) is a transition observed under policy π at time t (**Refer : Slide 11 Lecture 10**).

(a) Compute $\mathbb{E}_\pi(\delta_t | s_t = s)$ if δ_t uses the true state value function V^π (2 Points)

$$\begin{aligned}
\mathbb{E}_\pi(\delta_t | s_t = s) &= \mathbb{E}_\pi\left[r_{t+1} + \gamma V^\pi(s_{t+1}) - V^\pi(s_t) | s_t = s\right] \\
&= \mathbb{E}_\pi\left[r_{t+1} + \gamma V^\pi(s_{t+1}) | s_t = s\right] - V^\pi(s_t) \\
&= V^\pi(s_t) - V^\pi(s_t) = 0
\end{aligned}$$

We have used the fact that δ_t uses the true state value function V^π in the penultimate step

(b) Compute $\mathbb{E}_\pi(\delta_t | S_t = s, A_t = a)$, for an arbitrary action a taken at time t, if δ_t uses the true state value function V^π (2 Points)

$$\begin{aligned}
\mathbb{E}_\pi(\delta_t | s_t = s) &= \mathbb{E}_\pi\left[r_{t+1} + \gamma V^\pi(s_{t+1}) - V^\pi(s_t) | s_t = s, a_t = a\right] \\
&= \mathbb{E}_\pi\left[r_{t+1} + \gamma V^\pi(s_{t+1}) | s_t = s, a_t = a\right] - V^\pi(s_t) \\
&= Q^\pi(s_t, a_t) - V^\pi(s_t)
\end{aligned}$$

(c) In the TD(λ) algorithm, we use λ returns as the target. The λ return target is given by,

$$G_t^\lambda = (1-\lambda)\sum_{n=1}^{\infty} \lambda^{n-1} G_t^{(n)}$$

where $G_t^{(n)}$ is the n-step return defined as,

$$G_t^{(n)} = r_{t+1} + \gamma r_{t+2} + \cdots + \gamma^{n-1} r_{t+n} + \gamma^n V(s_{t+n}).$$

The parameter λ is used to determine the weights corresponding to each of the n-step returns in the λ-return target. We know that the weights decay exponentially with n. Therefore, in the G_t^λ sequence, after some terms, the weights of subsequent terms would have fallen by more than half as compared to the weight of the first term. Let $\eta(\lambda)$ denote the time by which

the weighting sequence would have fallen to half of its initial value. Derive an expression that relates the parameter λ to $\eta(\lambda)$. Use the expression derived to compute the value of λ for which the weights would drop to half after 3 step returns. (3 Points)

The λ-return is defined as

$$G_t^\lambda = (1-\lambda)\sum_{n=1}^{\infty}\lambda^{n-1}G_t^{(n)}$$

Defining $\tau = \eta(\lambda)$ to be the value of n such that

$$\lambda^\tau = \frac{1}{2} \implies \tau \approx \frac{\ln(\frac{1}{2})}{\ln(\lambda)}$$

Thus given λ, the half life τ is given by this expression. For example if $\lambda \approx \frac{1}{\sqrt{2}}$, then we compute that $\tau = 3$ so we are looking that many steps ahead before our weighting drops by one half.

(Pls. Check if this is correct)

Problem 5 : On Learning Rates

In any TD based algorithm, the update rule is of the following form

$$V(s) \leftarrow V(s) + \alpha_t[r + \gamma V(s') - V(s)]$$

where α_t is the learning rate at the t-th time step. In here, the time step t refers to the t-th time we are updating the value of the state s. Among other conditions, the learning rate α_t has to obey the Robbins-Monroe condition given by,

$$\begin{aligned}\sum_{t=0}^{\infty}\alpha_t &= \infty \\ \sum_{t=0}^{\infty}\alpha_t^2 &< \infty\end{aligned}$$

for convergence to true $V(s)$. Other conditions being same, reason out if the following values for α_t would result in convergence. (5 Points)

(1) $\alpha_t = \frac{1}{t}$

(2) $\alpha_t = \frac{1}{t^2}$

(3) $\alpha_t = \frac{1}{t^{\frac{2}{3}}}$

(4) $\alpha_t = \frac{1}{t^{\frac{1}{2}}}$

Generalize the above result for $\alpha_t = \frac{1}{t^p}$ for any positive real number p (i.e. $p \in \mathbb{R}^+$)

The series $\sum_{i=1}^{\infty}\frac{1}{t}$ is harmonic series and it does not converge. In fact, one can rewrite the series in the following way (by re-grouping terms)

$$\begin{aligned}\sum_{i=1}^{\infty}\frac{1}{t} &= 1+\frac{1}{2}+\frac{1}{3}+\frac{1}{4}+\cdots \\ &> 1+\frac{1}{2}+\frac{1}{2}+\frac{1}{4}+\frac{1}{4}+\frac{1}{8}+\frac{1}{8}+\frac{1}{8}+\frac{1}{8}\cdots \\ &= 1+\frac{1}{2}+\frac{1}{2}+\frac{1}{2}+\cdots = \infty\end{aligned}$$

A generalization of the Harmonic series is the p-series (Hyperharmonic series) defined as $\sum_{i=1}^{\infty} \frac{1}{t^p}$ for any +ve real number p. The p-series converges for all $p > 1$ (overharmonic series) and diverges for all $p \leq 1$. So, one can now use the above property to get the following results.

α_t	$\sum \alpha_t$	$\sum \alpha_t^2$	Algo converges
$\frac{1}{t^2}$	$< \infty$	$< \infty$	No
$\frac{1}{t}$	∞	$< \infty$	Yes
$\frac{1}{t^{\frac{2}{3}}}$	∞	$< \infty$	Yes
$\frac{1}{t^{0.5}}$	∞	∞	No

Problem 6 : Programming Value and Policy Iteration

Implement value and policy iteration algorithm and test it on **'Frozen Lake'** environment in openAI gym. **'Frozen Lake'** is a grid-world like environment available in gym. The purpose of this exercise is to to help you get hands on with using gym and to understand the implementation details of value and policy iteration algorithm(s)

This question will not be graded but **will** still come in handy for future assignments.

ALL THE BEST