E1 277 Reinforcement Learning : Assignment - 2 (Theory)

February 14, 2022

Instructions

- Total Points: 20
- All questions are compulsory.
- Deadline: 21/02/2022 11:59 PM
- When using any result proven in the class you are required to provide a reference to the lecture number. If you are using a general version of such a result or a result not proven in the class, provide a complete proof or a reference.

Q1. (10 pts.) Consider the following definition of Bellman operators acting on a value function V and Q-function Q for a stochastic policy π.

$$(T^\pi V)(s) = r^\pi(s) + \gamma \int P(s'|s,a)\pi(a|s)V(s')\mathrm{d}s'\mathrm{d}a, \quad \forall s \in \mathcal{S}$$

$$(T^\pi Q)(s,a) = r(s,a) + \gamma \int P(s'|s,a)\pi(a'|s')Q(s',a')\mathrm{d}s'\mathrm{d}a' \quad \forall (s,a) \in \mathcal{S}\times\mathcal{A}$$

Similarly, consider the definitions of Bellman optimal operators.

$$(T^* V)(s) = \max_{a\in\mathcal{A}} \left\{ r(s,a) + \gamma \int P(s'|s,a)V(s')\mathrm{d}s' \right\}, \quad \forall s \in \mathcal{S}$$

$$(T^* Q)(s,a) = r(s,a) + \gamma \int P(s'|s,a) \max_{a'\in\mathcal{A}} Q(s',a')\mathrm{d}s' \quad \forall (s,a) \in \mathcal{S}\times\mathcal{A}$$

Here, the state space $\mathcal{S}$ and the action space $\mathcal{A}$ could be continuous.

(a) (3 pts) Consider the *Q-value iteration* algorithm:

$$Q_{k+1} = T^* Q_k$$

The sample complexity of the above algorithm is defined as the number of iterations k required to ensure that the error $\|Q_k - Q^*\|_\infty \leq \epsilon$, where $\epsilon > 0$ is the desired accuracy. Assume that the rewards $r(s,a) \in [0,1]$, $\forall (s,a) \in \mathcal{S}\times\mathcal{A}$ and calculate the sample complexity of *Q-value iteration*.

(b) (2 pts) Next consider the policy iteration algorithm:

Policy Evaluation: Compute Q^{π_k} for the current policy π_k.

Policy Improvement: Update the policy $\pi_{k+1} \leftarrow \arg\max_{a \in \mathcal{A}} Q^{\pi_k(s,a)}$

As in (i) compute the sample complexity of the algorithm to ensure that $\|Q^{\pi_k} - Q^*\|_\infty \leq \epsilon$

(c) (3 pts) In a non-stationary policy $\tilde{\pi} = (\pi_1, \pi_2, \pi_3, \ldots)$, at each time step, a different policy π_t is used. Define the value function associated with $\tilde{\pi}$ as

$$V^{\tilde{\pi}} = \liminf_{t \to \infty} (T^{\pi_1} T^{\pi_2} \ldots T^{\pi_t} V_0)$$

Define the optimal value function over non-stationary policies as

$$V^\dagger = \sup_{\tilde{\pi}} V^{\tilde{\pi}},$$

where the sup is taken over all non-stationary policies. Recall that V^* is defined as the fixed point of T^* and show that $V^\dagger = V^*$. What can you conclude from this?

(d) (2 pts) Suppose we define the Bellman operator as

$$(\hat{T}^\pi V)(s) = r^\pi(s) \int P(s'|s) V(s') \mathrm{d}s'.$$

Is the operator monotonic? If yes, prove it. If not, provide a condition under which it is monotonic. Is the operator a contraction? If yes, compute the contraction factor. If no, then provide a condition under which it is a contraction.

Q2. (5 pts.) There are N items in a bag (with N fixed and known). A user picks items sequentially from the bag, examines it and either accepts the item (in which case the entire process ends) or discards the item into the dustbin and picks the next item. Once an item has been discarded, it cannot be re-selected. The objective is to maximize the probability of selecting the best product. The state space consists of three states: $\mathcal{S} = \{s_0, s_1, s_T\}$ and the action space consists of two actions: $\mathcal{A} = \{a_S, a_C\}$. The state s_0 denotes that the current item is not the best among all the *previously* examined items, while s_1 denotes that the current item is the best among all the *previously* examined items. At each of these states, the user can either choose action a_S, in which case she **selects** the current item and stops the process or chooses action a_C in which case she **continues** with the process. State s_T denotes a terminal state where the entire process stops.

(a) (1 pt.) Determine the transition probabilities for the above MDP.

(b) (1 pt.) Design a reward/cost structure keeping in mind the objective of the problem at hand.

(c) (3 pts.) Determine the structure of optimal value function using dynamic programming (You need not compute the exact value).

Q3. (5 pts.) Consider a stochastic multi armed bandit setting with $k = 2$ actions.

- X_{ti}: is a Random variable representing reward of arm $1 \leq i \leq k$ at time $t \geq 0$.
- All $\{X_{ti}\}_{t,i}$ are independent and $\forall 1 \leq i \leq k$, $\{X_{ti}\}_t$ are identically distributed with mean reward μ_i.
- A best arm is one that attains the largest possible expected reward: $i^* = \max_{1 \leq i \leq k} \mu_i$. We denote the corresponding mean reward by μ^*
- A bandit algorithm plays arm I_t at each time t and observes the reward X_{t,I_t}.

We define the *regret* of a bandit algorithm at time T, that plays a sequence of arms $\{I_t\}_{1 \leq t \leq T}$ by:

$$\text{Regret}(T) = T\mu^* - \mathbb{E}\left[\sum_{i=1}^{k} \mu_i N_i(T)\right],$$

where, $N_i(T) = \sum_{t=1}^{T} \mathbb{1}_{\{I_{t=i}\}}$ is the number of times arm i has been played in T iterations. Here the first term is the expected reward obtained if the agent plays the best arm i^* at each iteration and the second term is the expected reward obtained by the algorithm. Consider the following algorithm to choose arms:

Algorithm 1

1: **Input:** A multi armed bandit with k arms, time horizon T and a parameter $m \in \mathbb{N}$.
2: **for** $t = 1, 2, 3, \ldots, mk$ **do**
3: Play arms in round robin: $I_t = (t \bmod k) + 1$
4: **end for**
5: **for** $t = mk + 1, mk + 2, \ldots, T$ **do**
6: Play the arm with the best sample mean:

$$I_t = arg \max_{1 \leq i \leq k} \hat{\mu}_i(mk),$$

$$\text{where, } \hat{\mu}_i(t) = \frac{\sum_{s=1}^{t} X_{si} \mathbb{1}_{\{I_s = i\}}}{\sum_{s=1}^{t} \mathbb{1}_{\{I_s = i\}}}$$

7: **end for**

(a) (1 pt.) Show that maximizing the expected total sum of rewards $\mathbb{E}[\sum_{t=1}^{T} X_{t,I_t}]$ is equivalent to minimizing the regret.

(b) Suppose $k = 2$ and all rewards are bounded in $[0, 1]$ and w.l.o.g., assume that arm 1 is optimal. Show that the above algorithm attains a regret of $\mathcal{O}\left(\frac{1}{\Delta}\log T\right)$, where $\Delta = \mu_1 - \mu_2$ is called the arm gap.

(i) (1 pts) By analyzing the two phases of the algorithm separately, show that the regret can be written as follows:

$$\text{Regret}(T) \leq m\Delta + \Delta(T - 2m)\mathbb{E}[\mathbb{1}_{\hat{\mu}_1(2m) \leq \hat{\mu}_2(2m)}]$$

(ii) (2 pts) Chernoff-Hoefding's bound: Consider $\{X_i\}_i$ iid and $X_i \in [a, b]$ $\forall i$. Then,

$$P\left(\frac{1}{n}\sum_{i=1}^{n}(X_i - \mathbb{E}[X_1]) \geq \varepsilon\right) \leq e^{-2n\varepsilon^2/(b-a)^2}$$

Using this, show that the regret can be written as:

$$\text{Regret}(T) \leq m\Delta + \Delta T e^{-cm\Delta^2}, \text{ for some } c > 0$$

Now, choose an m such that the above epression is minimized and show that $\text{Regret}(T) = \mathcal{O}\left(\frac{1}{\Delta}\log T\right)$

(iii) (1 pt) The regret of the algorithm given above is sub-linear in the time horizon T (which is a good thing). However, can you point out a difficulty in implementing the algorithm to obtain such a sub-linear regret?