Assignment 3 (Sol.)

Reinforcement Learning

Prof. B. Ravindran

1. In solving a multi-arm bandit problem using the policy gradient method, are we assured of converging to the optimal solution?

   (a) no

   (b) yes

   Sol. (a)
   Depending upon the properties of the function whose gradient is being ascended, the policy gradient approach may converge to a local optimum.

2. In many supervised machine learning algorithms, such as neural networks, we rely on the gradient descent technique. However, in the policy gradient approach to bandit problems, we made use of gradient ascent. This discrepancy can mainly be attributed to the differences in

   (a) the objectives of the learning tasks

   (b) the parameters of the functions whose gradient are being calculated

   (c) the nature of the feedback received by the algorithms

   Sol. (c)
   The feedback in most supervised learning algorithms is an error signal which we wish to minimise. Hence, we would look to perform gradient descent. In policy gradient, we are trying to maximise the reward signal, hence gradient ascent.

3. Consider a bandit problem in which the parameters on which the policy depends are the preferences of the actions and the action selection probabilities are determined by the softmax relationship as $\pi(a_i) = \frac{e^{\theta_i}}{\sum_{j=1}^{k} e^{\theta_j}}$, where k is the total number of actions and θ_i is the preference value of action a_i. Derive the parameter update conditions according to the REINFORCE procedure considering the above described parameters and where the baseline is the reference reward defined as the average of the rewards received for all arms.

   (a) $\Delta\theta_i = \alpha(R - b)(1 - \pi(a_i; \theta))$

   (b) $\Delta\theta_i = \alpha(R - b)\frac{e^{\theta_i}}{\sum_{j=1}^{k} e^{\theta_j}}$

   (c) $\Delta\theta_i = \alpha(R - b)(\pi(a_i; \theta) - 1)$

   (d) $\Delta\theta_i = \alpha(R - b)\frac{1 - e^{\theta_i}}{\sum_{j=1}^{k} e^{\theta_j}}$

Sol. (a)
According to the REINFORCE method, at each step, we update the parameter as

$$\Delta\theta_n = \alpha(R_n - b_n)\frac{\partial ln\pi(a_n;\theta)}{\partial\theta_n}$$

Considering the softmax preferences, the characteristic eligibility is

$$\frac{\partial ln\pi(a_n;\theta)}{\partial\theta_n} = \frac{\partial}{\partial\theta_n} ln \frac{e^{\theta_i}}{\sum_{j=1}^{k} e^{\theta_j}} = \frac{\partial}{\partial\theta_n}(\theta_n - ln(\sum_{j=1}^{k} e^{\theta_j})) = 1 - \pi(a_i;\theta)$$

Considering the baseline, b as the average of all rewards seen so far, and α as the step size parameter, the update conditions are

$$\Delta\theta_i = \alpha(R - b)(1 - \pi(a_i;\theta))$$

4. Repeat the above problem for the case where the parameters are the mean and variance of the normal distribution according to which the actions are selected and the baseline is zero.

(a) $\Delta\mu_n = \alpha_n r_n \left(\frac{a_n-\mu_n}{\sigma_n}\right); \Delta\sigma_n = \alpha_n r_n \left\{\frac{(a_n-\mu_n)^2}{\sigma_n^2} - 1\right\}$

(b) $\Delta\mu_n = \alpha_n r_n \left(\frac{a_n-\mu_n}{\sigma_n^2}\right); \Delta\sigma_n = \alpha_n r_n \left\{\frac{(a_n-\mu_n)^2-\sigma_n^2}{\sigma_n^2}\right\}$

(c) $\Delta\mu_n = \alpha_n r_n \left(\frac{a_n-\mu_n}{\sigma_n^2}\right); \Delta\sigma_n = \alpha_n r_n \left\{\frac{(a_n-\mu_n)^2-\sigma_n^2}{\sigma_n^3}\right\}$

(d) $\Delta\mu_n = \alpha_n r_n \left(\frac{a_n-\mu_n}{\sigma_n}\right); \Delta\sigma_n = \alpha_n r_n \left\{\frac{(a_n-\mu_n)^2-\sigma}{\sigma_n^2}\right\}$

Sol. (c)
Assuming parameters to be μ and σ, we have policy,

$$\pi(a;\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(a-\mu)^2}{2\sigma}}$$

For the mean:

$$\frac{\partial \ln\pi(a_n;\mu_n,\sigma_n)}{\partial\mu_n} = \frac{\partial}{\partial\mu_n}\left\{-\frac{(a_n-\mu_n)^2}{2\sigma_n^2}\right\} = \frac{a_n-\mu_n}{\sigma_n^2}$$

For the variance:

$$\frac{\partial \ln\pi(a_n;\mu_n,\sigma_n)}{\partial\sigma_n} = \frac{\partial}{\partial\sigma_n}\left\{-\ln(\sqrt{2\pi\sigma_n})\right\} + \frac{\partial}{\partial\sigma_n}\left\{-\frac{(a_n-\mu_n)^2}{2\sigma_n^2}\right\}$$

Solving, we have:

$$\frac{\partial \ln\pi(a_n;\mu_n,\sigma_n)}{\partial\sigma_n} = -\frac{1}{\sigma_n} + \frac{(a_n-\mu_n)^2}{\sigma_n^3} = \frac{1}{\sigma_n}\{\left(\frac{a_n-\mu_n}{\sigma_n}\right)^2 - 1\}$$

Thus, the updates are:

$$\mu_{n+1} = \mu_n + \alpha_n r_n \left(\frac{a_n-\mu_n}{\sigma_n^2}\right)$$

$$\sigma_{n+1} = \sigma_n + \alpha_n r_n \frac{1}{\sigma_n}\left\{\left(\frac{a_n-\mu_n}{\sigma_n}\right)^2 - 1\right\}$$

5. Which among the following is/are differences between contextual bandits and full RL problems?

   (a) the actions and states in contextual bandits share features, but not in full RL problems

   (b) the actions in contextual bandits do not determine the next state, but typically do in full RL problems

   (c) full RL problems can be modelled as MDPs whereas contextual bandit problems cannot

   (d) no difference

   Sol. (b)
   Option (a) is not true because we can think of representations where actions and states share features. Similarly, option (c) is false because contextual bandit problems can also be modelled as MDPs.

6. Given a stationary policy, is it possible that if the agent is in the same state at two different time steps, it can choose two different actions?

   (a) no

   (b) yes

   Sol. (b)
   A stationary policy does not mean that the policy is deterministic. For example, for state s_1 and actions a_1 and a_2, a stationary policy π may have $\pi(a_1|s_1) = 0.4$ and $\pi(a_2|s_1) = 0.6$. Thus, at different time steps, if the agent is in the same state s_1, it may perform either of the two actions.

7. In class we saw that it is possible to learn via a sequence of stationary policies, i.e., during an episode, the policy does not change, but we move to a different stationary policy before the next episode begins. Does the temporal difference method encountered when discussing the tic-tac-toe example follow this pattern of learning?

   (a) no

   (b) yes

   Sol. (a)
   Recall that within an episode, at each step, after observing a reward, we modify the value function (the probability values in the tic-tac-toe example). Since the policy used to select actions is derived from the value function, in effect we are modifying the policy at each step.

8. We saw the following definition of the action-value function for policy π: $q_\pi(s, a) = E_\pi[G_t|S_t = s, A_t = a]$. Suppose that the action selected according to the policy π in state s is a_1. For the same state, will the function be defined for actions other than a_1?

   (a) no

   (b) yes

   Sol. (b)
   The interpretation of the action-value function $q_\pi(s, a) = E_\pi[G_t|S_t = s, A_t = a]$ is the expectation of the return given that we take action a in state s at time t and thereafter select actions based on policy π. Thus, the only constraint for the actions is that they must be applicable in state s.

9. Consider a 100x100 grid world domain where the agent starts each episode in the bottom-left corner, and the goal is to reach the top-right corner in the least number of steps. To learn an optimal policy to solve this problem you decide on a reward formulation in which the agent receives a reward of +1 on reaching the goal state and 0 for all other transitions. Suppose you try two variants of this reward formulation, (P_1), where you use discounted returns with $\gamma \in (0, 1)$, and (P_2), where no discounting is used. Which among the following would you expect to observe?

   (a) the same policy is learned in (P_1) and (P_2)

   (b) no learning in (P_1)

   (c) no learning in (P_2)

   (d) policy learned in (P_2) is better than the policy learned in (P_1)

   Sol. (c)
   In (P_2), since there is no discounting, the return for each episode regardless of the number of steps is +1. This prevents the agent from learning a policy which tries to minimise the number of steps to reach the goal state. In (P_1), the discount factor ensures that the longer the agent takes to reach the goal state, the lesser reward it gets. This motivates the agent to find take the shortest path to the goal state.

10. Given an MDP with finite state set, S, and an arbitrary function, f, that maps S to the real numbers, the MDP has a policy π such that $f = V^{\pi}$.

    (a) false

    (b) true

    Sol. (a)
    Consider an MDP where all rewards are 0.