DAT440 / DIT471 Advanced topics in machine learning - Solutions Final Exam

Course Responsible and Examiner: Morteza Haghir Chehreghani

May 31, 2022, 2:00 pm

1. Multiple choice questions
   (a) i
   (b) ii, iii
   (c) ii, iii
   (d) ii
   (e) ii

2. UCB
   (a) $a = 2$
   (b) $\hat{\mu}_{t+1}(1) = \frac{3}{10} = 0.3, \quad \hat{\mu}_{t+1}(2) = \frac{1}{3} \approx 0.33, \quad \hat{\mu}_{t+1}(3) = \frac{7}{10} = 0.7, \quad \hat{\mu}_{t+1}(4) = \frac{1}{5} = 0.2$
   (c) The first term and second term of the upper confidence bound can be seen as an exploitation term and exploration term, respectively. Hence, decreasing the second term will result in less exploration. As α decreases, UCB will become more greedy.
   (d) $a = 3$
   (e) By inserting the values from the scenario in Table 1 into the provided inequality, we get (for $\alpha = 2$ here, but a value calculated using another positive α would also be correct) for $t = 27$: $\mathbb{E}[R(t)] \leq \frac{1}{54} + 8\sqrt{54} \approx 58.8$. Alternatively, since $\Delta(a) \leq 1$ for all $a \in \mathcal{A}$, and $\mathbb{E}[R(t)] \leq t \max_{a \in \mathcal{A}} \Delta(a)$, then $\mathbb{E}[R(t)] \leq 27$.

3. Regret bounds
   (a) It is false. If it was true, the upper bound would be lower than the lower bound for stochastic K-armed bandit problems (see e.g., slide 61 of the lecture notes for stochastic bandit problems: Bandits 2 - new.pdf).
   (b) It is true. By e.g., Theorem 1.14 in the [Bandits] book, we know that

$$\mathbb{E}[R(T)] \leq O\left(\sum_{\text{arms } a \text{ where } \mu(a) < \mu(a^*)} \frac{\log T}{\Delta(a)}\right).$$

   Since $\mu(a^*) - \mu(a^{(2)}) \geq \Delta(a)$ for all arms a where $\mu(a) < \mu(a^*)$, the statement holds. This is an *instance-dependent* bound, i.e., it contains constants (the gaps $\Delta(a)$) which depend on the problem instance (characterized by the unknown mean vector).

   (c) It is true. By e.g., Theorem 1.14 in the [Bandits] book (or by the bound given in question (2.e) of this exam), we know that

$$\mathbb{E}[R(t)] \leq O\left(\sqrt{Kt \log T}\right),$$

   for $t \leq T$, from which the statement follows. This is an *instance-independent* bound, i.e., the constants are universal for all problem instances (characterized by the unknown mean vector).

4. Batch algorithms For more information, see slides on batch RL in lecture on TD learning.

	MCMC	TD
A	(1+0.8)/2=0.9	1*V(B)=1.36
B	$(1 + 0.8 + 0.4)/3 = 2.2/3 \approx 0.73$	2/3*V(C)+1/3*V(D)=41/30 $\approx$ 1.36
C	0.7	(1+0.4+0.8+0.6)/4=0.7
D	2.7	(5+0.4)/2=2.7

5. RL algorithms

   (a) Sarsa

   i. Sarsa is on-policy since we both evaluate and improve the policy that is being followed. Meaning that the behavior and target policy is the same.

   ii. The policy improvement is done at line 5 and 8 since this is where we use the action-value function to improve the policy with respect to our current action-value function.

   iii. Policy evaluation is done at line 9 since this is where we update the action-value function with respect to the current policy.

   iv. To ensure convergence to an optimal policy (and action-value function), all state-action pairs need to be visited an infinite number of times and the policy converges in the limit to the greedy policy. Hence, the ϵ-greedy policy makes sure that all state-action pairs are visited an infinite number of times, and converges to the greedy policy by using a carefully chosen ϵ, e.g, $\epsilon = 1/t$.

   (b) Actor-Critic

   i. It is on-policy since the critic must learn about and critique the policy that the actor currently follows.

   ii. The actor is the policy structure because it is used to select actions. Hence, line 11 corresponds to the improving the actor since this is where the policy parameterization is updated.

   iii. The estimated state-value function is the critic, because it criticizes the actions made by the actor. At line 10, the critic is improved since the parameterization of the value function is updated.

   iv. The critique takes the form of a TD error and this scalar signal is the only output of the critic. This critique is used to evaluate that action a_t that was just selected by the actor. If the TD error is positive, it suggests that the actor should strengthen the tendency to select action a_t, whereas if the TD error is negative, it suggests the tendency should be weakened.

6. Value iteration

   (a) The value iteration update for each state s is given by

$$v_{k+1}(s) = \max_a \sum_{s',r} p(s', r|s, a) \left[r + \gamma v_k(s')\right].$$

   We have a deterministic environment where $s'_{s,a}$ is the (deterministic) new state given our current state s and action a. Also, transition results in a -1 reward and $\gamma = 1$. This gives us the following value iteration update

$$v_{k+1}(s) = \max_a \left[-1 + v_k(s'_{s,a})\right].$$

   Using this formula, for iteration 1 and 2 we obtain the value function and greedy policy shown in Figure 2 and 3, respectively. All greedy actions are shown.

→	↑↓	←	**Z**
←↑→	■	■	←→
↑	■	←↑	↑
→	↑↓	←↑	←↓→

Figure 1: Initial greedy policy

2	2	2	0
1	■	■	3
1	■	2	3
2	2	2	1

(a) Value function

←↑→	←↓↑→	←↑↓	**Z**
↑	■	■	←↓→
↓	■	→	↑→
←↓→	←↓↑→	←↓↑	↑

(b) Greedy policy

Figure 2: Iteration 1

1	1	1	0
1	■	■	2
1	■	2	2
1	1	1	2

(a) Value function

←↓↑→	←↓↑→	←↑↓	**Z**
←↓↑→	■	■	←↓→
←↓↑→	■	←↑→	←↓↑→
←↓↑→	←↓↑→	↑→	↑↓→

(b) Greedy policy

Figure 3: Iteration 2

(b) See the slides/book. Some differences are that Value iteration just performs one sweep of policy evaluation and (possibly) updates the policy directly. Policy iteration performs policy evaluation until some stopping criteria before updating the policy. This means that each step in policy iteration actually improves the policy (in value iteration you might just update the value) but takes longer as you have to do multiple steps of policy evaluation (although the value will often not change much between one update of policy.