DAT440 / DIT470 / DIT471 Advanced topics in machine learning - Final Exam

Course Responsible and Examiner: Morteza Haghir Chehreghani

May 31, 2022, 2:00 pm

- You explain your solutions, for any question that the calculations are needed you must show the steps.
- You may have a cheat sheet of two pages (either one double-sided sheet or two one-sided sheets) in A4 format (font size 11 if typed, and similar font size if it is handwritten). You can include whatever you want in the cheat sheet. Besides that, you may not use any other resource or reference to answer the questions.
- You may have a simple standard calculator commonly used at Chalmers, though the questions do not need use of a calculator.
- Read the questions carefully such that you do not miss any question and ensure you clearly give the answer required for each (sub)question.
- The exam grade and the final grade will be computed according to the formula mentioned on the course webpage. Accordingly, your grade will follow the this distribution: 28 out of 60 (3,G), 36 out of 60 (4), 48 out of 60 (5, VG).
- Teaching assistants will visit the exam premises on two occasions to answer the clarification questions: one hour after the start of the exam and when an hour of the exam remains.

1. (9 points) Multiple choice questions: For each question you have to select **all options that are true for full score (and none if all are false)**. Fill in your answers in the form on the final page. **Make sure to include it with your exam.** The points do not necessarily reflect the number of correct answers.

   For the RL-related questions, consider an arbitrary action A_t taken according to policy π at state S_t which results in reward R_{t+1} and new state S_{t+1}. Then, $V_\pi(S_t)$ indicates the state values of state S_t under π and γ is the respective discount factor as introduced in the course.

   (a) (2 points) Which paradigm explains the following equation best? $V_\pi(S_t) = \sum_{k=1}^{\infty} \gamma^{k-1} R_{t+k}$
      i. Monte Carlo
      ii. Dynamic Programming
      iii. Temporal Difference

   (b) (3 points) Initializing the Q values in Q-learning to a larger value will:
      i. Makes the algorithm converge to a better policy
      ii. Increases the amount of exploration done by the algorithm
      iii. May improve the convergence speed in practice.
      iv. Makes $\epsilon-$greedy exploration unnecessary.

   (c) (2 points) What holds for $V_\pi(S_t) = R_{t+1} + \gamma V_\pi(S_{t+1})$?
      i. It is an unbiased estimate of the expected return $v_\pi(S_t)$

ii. It is a biased estimate of the expected return $v_\pi(S_t)$
iii. It has lower variance than the return
iv. It has higher variance than the return

(d) (1 point) A small discount factor γ in the return will make the agent care more about future rewards.
i. True
ii. False

(e) (1 point) The ϵ_t-greedy algorithm for K-armed bandit problems, with exploration probability $\epsilon_t = t^{-1/3} \cdot (K \log t)^{1/3}$ at time step t, *adapts* the exploration to the observed rewards.
i. True
ii. False

2. (10 points) UCB

Consider a stochastic K-armed bandit problem with a set of arms $\mathcal{A}$, and $|\mathcal{A}| = K$. We define, at time t, the arm $a_t \in \mathcal{A}$ played by the agent, the reward $x_t(a) \in \{0, 1\}$ received by an agent if it plays arm $a \in \mathcal{A}$, the cumulative reward $s_t(a) = \sum_{n=1}^{t} \mathbb{1}\{a_n = a\} x_t(a)$ of arm a until time t, the number of times $n_t(a) = \sum_{n=1}^{t} \mathbb{1}\{a_n = a\}$ that arm a has been played until time t, the average reward $\hat{\mu}_t(a) = s_t(a)/n_t(a)$ of arm a until time t.

Algorithm 1 UCB

1: Play each arm $a \in \mathcal{A}$ once.
2: In each time step t, play $\arg\max_{a \in \mathcal{A}} \hat{\mu}_{t-1}(a) + \sqrt{\frac{\alpha \ln T}{n_{t-1}(a)}}$.

Consider the scenario in Table 1, where an agent has played in a multi-armed bandit environment with $K = 4$ arms until time $t = 27$ (with horizon $T = 54$). Note that the reward $x_{t+1}(a)$ is revealed to the agent *if and only if* arm $a \in \mathcal{A}$ is played at time $t + 1$.

Table 1: Multi-armed bandit scenario

a	1	2	3	4
$s_t(a)$	3	1	7	1
$n_t(a)$	10	2	10	5
$x_{t+1}(a)$	1	0	1	0

(a) (2 points) Which arm will be played by UCB with $\alpha = 2$ at time $t + 1$ (note that $\ln T \approx 4$)?

(b) (2 points) Calculate $\hat{\mu}_{t+1}(a)$ for each arm $a \in \mathcal{A}$ (under UCB with $\alpha = 2$).

(c) (2 points) Explain what happens if α decreases.

(d) (1 point) Which arm would be played at time $t + 1$ by UCB with $\alpha = 0$?

(e) (3 points) From the analysis in the assignment, we know that the expected regret $\mathbb{E}[R(t)]$ of UCB can be bounded such that, in this setting, for $t \leq T$ (with $\alpha > 0$): $\mathbb{E}[R(t)] \leq \frac{2t}{T^{2\alpha-2}} + 2\sqrt{\alpha K t \ln T}$. For the scenario outlined in Table 1, calculate a constant which is an upper bound for the expected regret, at time $t = 27$.

3. (9 points) Consider a stochastic K-armed bandit problem with a set of arms $\mathcal{A}$, where $|\mathcal{A}| = K$, T is the horizon, $\mu(a)$ is the fixed but unknown mean reward of arm a, $n_t(a)$ is the number of times arm a has been played until the end of round $t \leq T$, the arm with the highest expected reward is $a^* := \max_{a \in \mathcal{A}} \mu(a)$, the arm played at round t is a_t, and the gap of arm a is $\Delta(a) := \mu(a^*) - \mu(a)$. Answer the following questions about upper bounds on the expected regret $\mathbb{E}[R(T)]$ of the UCB algorithm:

(a) (3 points) Is it true that $\mathbb{E}[R(T)] \leq O\left(\sum_{\text{arms } a \text{ where } \mu(a) < \mu(a^*)} \frac{\sqrt{\log T}}{\Delta(a)}\right)$? Briefly explain your answer.

(b) (3 points) Is it true that $\mathbb{E}[R(T)] \leq O\left(\frac{K \log T}{\mu(a^*) - \mu(a^{(2)})}\right)$ where $a^{(2)}$ is the arm with the second highest mean reward? Briefly explain your answer.

(c) (3 points) Is it true that $\mathbb{E}[R(T)] \leq O\left(\sqrt{KT \log T}\right)$? Briefly explain your answer.

4. (12 points) The data in Table 2 has been generated from the model shown in Figure 1. For each state, compute the value of each state (A-D) using both batch Temporal Difference-learning and batch Monte Carlo and present in a table. Let the discount $\gamma = 1$ and let T indicate a terminal state with no reward (or value). The agent can start an episode at any state (except T) and will terminate upon reaching state T.

Table 2: Batch data obtained from the MDP, each row is a new episode

S_1	R_1	S_2	R_2	S_3	R_3	S_4
A	0	B	0	C	1	T
S_1	R_1	S_2	R_2	S_3		
B	0	D	0.4	T		
S_1	R_1	S_2				
C	0.4	T				
S_1	R_1	S_2	R_2	S_3	R_3	S_4
A	0	B	0	C	0.8	T
S_1	R_1	S_2				
C	0.6	T				
S_1	R_1	S_2				
D	5	T				

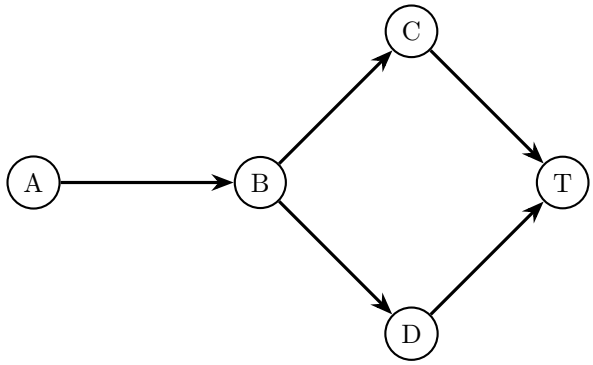

Figure 1: Overview of the model

5. (13 points) RL algorithms

(a) (7 points) Algorithm 2 shows the Sarsa algorithm. Answer the following questions about this algorithm:

i. (1 point) Is it on-policy or off-policy? Briefly explain your answer.

ii. (2 points) What part(s) of the algorithm corresponds to the *policy improvement* step(s)? Briefly explain your answer.

iii. (2 points) What part(s) of the algorithm corresponds to the *policy evaluation* step(s)? Briefly explain your answer.

iv. (2 points) Why is it important to use a ϵ-greedy policy instead of greedy policy as the behavior policy?

(b) (6 points) Algorithm 3 shows the (one-step) Actor-Critic algorithm. Answer the following questions about this algorithm:

i. (1 point) Is it on-policy or off-policy? Briefly explain your answer.

ii. (1 point) What part(s) of the algorithm corresponds to improving the *actor*? Briefly explain your answer.

iii. (1 point) What part(s) of the algorithm corresponds to improving the *critic*? Briefly explain your answer.

iv. (3 points) Explain the output of the *critic* and how it is used to influence the *actor*.

Algorithm 2 Sarsa

Inputs:
step size $\alpha \in (0, 1]$
small $\epsilon > 0$
discount factor $\gamma \in [0, 1]$
Initialize:
$Q(s, a)$, for all $s \in \mathcal{S}^+$, $a \in \mathcal{A}(s)$, arbitrarily except that $Q(terminal, \cdot) = 0$
for each episode **do**
  Initialize state S (first state of episode)
  Choose action A from S using policy derived from Q (e.g., ϵ-greedy)
  for each step of episode (end if S is terminal) **do**
    Take action A, observe reward R and new state S'
    Choose A' from S' using policy derived from Q (e.g., ϵ-greedy)
    $Q(S, A) \leftarrow Q(S, A) - \alpha \left[R + \gamma Q(S', A') - Q(S, A)\right]$
    $S \leftarrow S'$
    $A \leftarrow A'$
  end for
end for

Algorithm 3 One-step Actor-Critic

Inputs:
a differential policy parameterization $\pi(\alpha|s, \boldsymbol{\theta})$
a differentiable state-value function parameterization $\hat{v}(s, \boldsymbol{w})$
step sizes $\alpha^{\boldsymbol{\theta}} > 0$, $\alpha^{\boldsymbol{w}} > 0$
discount factor $\gamma \in [0, 1]$
Initialize:
policy parameters $\boldsymbol{\theta} \in \mathbb{R}^{d'}$
state-value weights $\boldsymbol{w} \in \mathbb{R}^{d}$
for each episode **do**
  Initialize state S (first state of episode)
  $I \leftarrow 1$
  for each step of episode (end if S is terminal) **do**
    $A \sim \pi(\cdot|S, \boldsymbol{\theta})$
    Take action A, observe reward R and new state S'
    $\delta \leftarrow R + \gamma \hat{v}(S', \boldsymbol{w}) - \hat{v}(S, \boldsymbol{w})$ ▷ if S' is terminal, then $\hat{v}(S', \boldsymbol{w}) \doteq 0$
    $\boldsymbol{w} \leftarrow \boldsymbol{w} + \alpha^{\boldsymbol{w}} \delta \nabla_{\boldsymbol{w}} \hat{v}(S, \boldsymbol{w})$
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha^{\boldsymbol{\theta}} I \delta \nabla_{\boldsymbol{\theta}} \ln \pi(A|S, \boldsymbol{\theta})$
    $I \leftarrow \gamma I$
    $S \leftarrow S'$
  end for
end for

6. (7 points) Value iteration

   (a) (6 points) Figure 2 shows a deterministic Grid World environment. The state space is $\mathcal{S} = \{A, B, \ldots, L\}$, as seen in Figure 2a, and the action space $\mathcal{A} = \{\text{up}, \text{down}, \text{right}, \text{left}\}$. The terminal state is marked with **Z**. An action taking the agent off the grid or onto a black cell (a cell not in the state space) leaves the state unchanged. Each transition results in a reward of -1. Each transition is deterministic, meaning that the new state fully determined by the current state and action taken.

   Perform two iterations of (synchronous) value iteration on the Grid World environment. Use no discounting ($\gamma = 1$) and initialize $V_0(s)$ as shown in Figure 2b. For each iteration, show the value function and corresponding greedy policy (display all greedy actions in each state) with respect to the value function. Show your calculations.

   (b) (1 point) Explain the difference between value iteration and policy iteration.

A	B	C	**Z**
D			E
F		G	H
I	J	K	L

(a) State space

2	3	1	0
2			4
1		3	1
1	3	2	2

(b) Initial value function $V_0(s)$.

Figure 2: Grid World environment

Hand this page in with your solutions

Anonymous code:

Put a cross in each cell that you select.

	i	ii	iii	iiii
Question				
a				
b				
c				
d				
e				