732A96/TDDE15 ADVANCED MACHINE LEARNING

EXAM 2024-01-03

Teacher

Jose M. Peña. He will visit the rooms for questions.

Grades

- For 732A96 (A-E means pass):
  - A=19-20 points
  - B=17-18 points
  - C=14-16 points
  - D=12-13 points
  - E=10-11 points
  - F=0-9 points
- For TDDE15 (3-5 means pass):
  - 5=18-20 points
  - 4=14-17 points
  - 3=10-13 points
  - U=0-9 points

In each question, full points requires clear and well motivated answers and commented code.

Instructions

- This is an individual exam. No help from others is allowed. No communication with others is allowed. Answers to the exam questions may be sent to Urkund.
- This is an anonymous exam. Do not write your name on it.
- The answers to the exam should be submitted in a single PDF file. You can make a PDF from LibreOffice (similar to Microsoft Word). You can also use Markdown from RStudio (no support is provided though). Include important code needed to grade the exam (inline or at the end of the PDF file).

Allowed help

Everything on the course web page. Your individual and group solutions to the labs. This help is available on the corresponding directories of the exam system.

1. Probabilistic Graphical Models (5 p)

Consider the algorithm below. Recall that $X \perp_G Y|Z$ denotes that X and Y are d-separated given Z in the directed and acyclic graph G, and $X \not\perp_G Y|Z$ that they are not d-separated. What do you think that the algorithm intends to return in Out? Explain your answer.

Input: A directed and acyclic graph G, and a target node Z.
Output: A node set Out.

$Out := \varnothing$
Repeat until Out does not change
    If there exists some node Y such that $Y \neq Z$ and $Y \notin Out$ and $Y \not\perp_G Z|Out$ then
        $Out := Out \cup Y$
Repeat until Out does not change
    If there exists some node $Y \in Out$ such that $Y \perp_G Z|Out \setminus Y$ then
        $Out := Out \setminus Y$
Return Out

---

2. Hidden Markov Models (5 p)

(3 p) Use the `HMM` package to implement the dishonest casino hidden Markov model (HMM). This HMM is included in the package. You can get a description of it by typing `?dishonestCasino`, and you can run it by typing `dishonestCasino()`. Use any transition and emission probabilities that you consider appropriate. Finally, sample the HMM built.

(2 p) Modify your previous implementation so that when a die is chosen, it is used for at least three consecutive throws. In particular, this regime's minimum duration should be implemented implicitly by duplicating hidden states and the emission model, i.e. do not use increasing or decreasing counting variables. Finally, sample the HMM built.

3. Reinforcement Learning (10 p)

(9 p) You are asked to solve the environment A in lab 3 using the following algorithm, a.k.a. Monte Carlo (MC) control. In the algorithm, $\mathcal{S}$ denotes all the non-terminal states, and $\mathcal{A}(s)$ denotes all the actions that can be performed in state s. Moreover, an ϵ-soft policy is a policy for which $\pi(a|s) \geq \epsilon/|\mathcal{A}(s)|$ for every state s and action a, where $|\mathcal{A}(s)|$ denotes the number of actions in $\mathcal{A}(s)$. To reduce the running time, you may want to discard a generated episode if it is of length greater than 50, i.e. abort the episode generation if it has not reached a terminal state after 50 actions and start a new episode generation.

On-policy first-visit MC control (for ε-soft policies), estimates $\pi \approx \pi_*$

Algorithm parameter: small $\varepsilon > 0$

Initialize:

$\pi \leftarrow$ an arbitrary ε-soft policy

$Q(s,a) \in \mathbb{R}$ (arbitrarily), for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$

$Returns(s,a) \leftarrow$ empty list, for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$

Repeat forever (for each episode):

Generate an episode following π: $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$

$G \leftarrow 0$

Loop for each step of episode, $t = T-1, T-2, \ldots, 0$:

$G \leftarrow \gamma G + R_{t+1}$

Unless the pair S_t, A_t appears in $S_0, A_0, S_1, A_1 \ldots, S_{t-1}, A_{t-1}$:

Append G to $Returns(S_t, A_t)$

$Q(S_t, A_t) \leftarrow \text{average}(Returns(S_t, A_t))$

$A^* \leftarrow \arg\max_a Q(S_t, a)$ (with ties broken arbitrarily)

For all $a \in \mathcal{A}(S_t)$:

$$\pi(a|S_t) \leftarrow \begin{cases} 1 - \varepsilon + \varepsilon/|\mathcal{A}(S_t)| & \text{if } a = A^* \\ \varepsilon/|\mathcal{A}(S_t)| & \text{if } a \neq A^* \end{cases}$$

(1 p) Name a disadvantage of MC control over Q-learning. Explain your answer.