732A96/TDDE15 ADVANCED MACHINE LEARNING

EXAM 2023-01-04

Teacher

Jose M. Peña. Available for questions by phone kl 10-11.

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. Hidden Markov Models (7 p)

Implement the forward-backward algorithm as it appears in the course slides or in the book by Bishop. Run it on the data that you used in the lab on hidden Markov models. Compute the accuracy of the filtered and smoothed distributions. Show that you obtain the same accuracy when using the `HMM` package.

2. Reinforcement Learning (7 p)

- (3 p) The Q-learning algorithm is based on the following updating rule:

$$q(s,a) \leftarrow q_*(s,a) + \alpha(r + \gamma \max_{a'} q(s',a') - q(s,a))$$

  which implies updating $q(s,a)$ to make it closer to $r+\gamma \max_{a'} q(s',a')$. Some may object that this does not make sense since it assumes that the agent acts greedily in the next state s' due to $\max_{a'}$, but in reality the agent acts ϵ-greedily. Then, they may argue that it would be better to use the following updating rule:

$$q(s,a) \leftarrow q_*(s,a) + \alpha(r + \gamma q(s',a') - q(s,a))$$

  which implies updating $q(s,a)$ as a function of the next state s' and **next action** a'. In other words, one has to produce the next state and action in order to update the current state and action. You are asked to implement this algorithm. **Include your code commented**.

- (4 p) You are asked to run Q-learning and the new algorithm on a modified version of what we called environment C in the lab. The modified version differs from the original in that the reward for positions (1,2:5) is now -10 instead of -1, the reward for position (1,6) is still +10, and the reward for the rest of the positions is -1. **An episode ends now when the agent receives a reward of +10 och -10**, i.e. the episode does not end when the agent receives a reward of -1. Moreover, now $\epsilon = 0.5$, $\gamma = 1$, $\beta = 0$ and $\alpha = 0.1$. The rest of the environment is the same as the original environment C. You are asked to run Q-learning and the new algorithm for 5000 episodes and report the final q-table and policy. You should also plot the reward obtained in each episode (episode in the X-axis and reward in the Y-axis) for both algorithms. Use the function `MovingAverage` provided in the lab. Which algorithm performs best and why ?

3. Gaussian Processes (6 p)

The file `KernelCode.R` distributed with the exam contains code to construct a `kernlab` function for the Matern covariance function with $\nu = 3/2$:

$$k(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \left(1 + \frac{\sqrt{3}r}{\ell}\right) \exp\left(-\frac{\sqrt{3}r}{\ell}\right)$$

where $r = |\mathbf{x} - \mathbf{x}'|$.

- (2 p) Let $f \sim \mathcal{GP}(0, k(\mathbf{x}, \mathbf{x}'))$ a priori and let $\sigma_f^2 = 1$ and $\ell = 0.5$. Plot $k(0, z)$ as a function of z. You can use the grid `zGrid = seq(0.01,1,by=0.01)` for the plotting. Interpret the plot. Connect your discussion to the smoothness of f. Finally, repeat this exercise with $\sigma_f^2 = 0.5$ and discuss the effect this change has on the distribution of f.
- (4 p) The file `lidar.RData` distributed with the exam contains two variables *logratio* and *distance*. Load the variables into memory with the R command `load("lidar.RData")`. Compute the posterior distribution of f in the model

  $$logratio = f(distance) + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, 0.05^2).$$

  You should do this for both length scales $\ell = 1$ and $\ell = 5$. Set $\sigma_f = 1$. Your answer should be in the form of a scatter plot of the data overlayed with curves for (a) the posterior mean of f, (b) 95 % probability intervals for f, and (c) 95 % prediction intervals for y. Use the `gausspr` function in the `kernlab` package for (a), but not for (b) and (c) since the function seems to contain a bug. For (b) and (c) instead, find the appropriate expression in the course slides or in the book by Rasmussen and Williams and implement it. You are not allowed to use Algorithm 2.1. Discuss the differences in results from using the two length scales.