DIPARTIMENTO DI ELETTRONICA, INFORMAZIONE E BIOINGEGNERIA

Politecnico di Milano

Machine Learning (Code: 097683)
September 7, 2021

Surname:	
Name:	
Student ID:	
Row:	Column:

Time: 2 hours Maximum Marks: 33

- The following exam is composed of **8 exercises** (one per page). The first page needs to be filled with your **name, surname and student ID**. The following pages should be used **only in the large squares** present on each page. Any solution provided either outside these spaces or **without a motivation** will not be considered for the final mark.
- During this exam you are **not allowed to use electronic devices**, such as laptops, smartphones, tablets and/or similar. As well, you are not allowed to bring with you any kind of note, book, written scheme, and/or similar. You are also not allowed to communicate with other students during the exam.
- The first reported violation of the above-mentioned rules will be annotated on the exam and will be considered for the final mark decision. The second reported violation of the above-mentioned rules will imply the immediate expulsion of the student from the exam room and the **annulment of the exam**.
- You are allowed to write the exam either with a pen (black or blue) or a pencil. It is your responsibility to provide a readable solution. We will not be held accountable for accidental partial or total cancellation of the exam.
- The exam can be written either in **English** or **Italian**.
- You are allowed to withdraw from the exam at any time without any penalty. You are allowed to leave the room not early than half the time of the duration of the exam. You are not allowed to keep the text of the exam with you while leaving the room.
- **Three of the points will be given on the basis on how quick you are in solving the exam.** If you finish earlier than 45 min before the end of the exam you will get 3 points, if you finish earlier than 30 min you will get 2 points and if you finish earlier than 15 min you will get 1 point (the points cannot be accumulated).
- The box on Page 10 can only be used to complete the Exercises 7, and/or 8.

Ex. 1	Ex. 2	Ex. 3	Ex. 4	Ex. 5	Ex. 6	Ex. 7	Ex. 8	Time	Tot.
/ 7	/ 7	/ 2	/ 2	/ 2	/ 2	/ 4	/ 4	/ 3	/ 33

Exercise 1 (7 marks)

Describe and compare Ridge regression and LASSO algorithms to solve linear regression problems.

Exercise 2 (7 marks)

Describe and compare Monte Carlo and Temporal Difference for model-free policy evaluation.

Exercise 3 (2 marks)

Consider the following snippet of code and answers to the questions below providing adequate motivations.

```
1 while m < M:
2   ns, r = env.transition_model(a)
3   na = eps_greedy(s, Q, eps)
4   Q[s, a] = Q[s, a] + alpha * (r + env.gamma * Q[ns, na] - Q[s, a])
5   m = m + 1
6   s = ns
7   a = na
```

1. What algorithm is this code implementing? What kind of problem is it addressing?
2. Explain the operations performed by the `eps_greedy` function.
3. What conditions do we need on `alpha` and `eps` to make the algorithm converge to a desirable solution?
4. How can we modify Line 4 to make the algorithm work off-policy?

1. This snippet of code is implementing the main loop of the `SARSA` algorithm, which tackles the Reinforcement Learning control problem.
2. The `eps_greedy` function is implementing an epsilon greedy policy. Thus, it returns the action that maximizes the `Q` value in `s` with probability 1 - `eps`, or a random action with probability `eps`.
3. To ensure that the algorithm will eventually converge to the optimal policy, one should take `eps` that goes to zero, and a learning rate `alpha` that follows Robbins-Monro conditions.
4. We could change the update rule to implement the `Q-learning` algorithm, i.e.,
   `Q[s, a] = Q[s, a] + alpha * (r + env.gamma * np.max(Q[ns, :]) - Q[s, a])`,
   or apply the importance sampling correction to the samples we have.

Exercise 4 (2 marks)

State whether the following claims about Bagging and Boosting are true or false, motivating your answers:

1. Since Boosting and Bagging are ensemble methods, they can be both parallelized.
2. Bagging should be applied with weak learners.
3. The central idea of Boosting consists in using bootstrapping.
4. It is not a good idea to use Boosting with a deep neural network as a base learner.

1. False, only Bagging can be parallelized, since training is done on different datasets, while Boosting is sequential by nature.
2. False, weak learners are good candidate for Boosting, since they have low variance. Tipically one uses instead bagging when more complex and unstable learners are needed, to reduce their variance.
3. False, bootstrapping is used in bagging, whose name derived indeed from "boosting aggregation".
4. True, it is not a good idea to do that, since deep neural networks are very complex predictor, which can have large variance. Therefore, you may not succeed in lowering bias without increasing variance. Moreover, since you need to train the network multiple times, the procedure may require a lot of time.

Exercise 5 (2 marks)

Vertical farming is the practice of growing crops in vertically stacked layers. A vertical farming company is faced with the problem of properly supplying its plants with the right amount of nutrients. Each plant lives in a separate piece of soil and is individually supplied with water and other nutritious substances. While the conditions in which plants live are similar, they are not identical, thus, different plants may need a tailored income of nutrients. Based on a historical dataset with measurements (including water and nutrition provided) of properly fed plants at different times, we would like to feed new plants in a proper way.

1. Can the above problem be solved using a machine learning model? Which one? Provide and motivate either the features/target or action/states/reward for the chosen model.
2. Provide an algorithm to solve the above problem, commenting on the pros and cons of its usage
3. Suppose now that the company wants to refine the current approach. They provide you with a new dataset that contains plants measurements (including water and nutrition provided) together with a health index, which gives you information on how much the plant is "healthy" at sampling time. They ask you to use this dataset to learn how to feed plants in order to maximize their health in the long run. How would you change your approach?

1. The problem can be thought as a supervised learning regression problem, where, based on the data coming from the sensor monitoring plants, one has to predict the correct amount of each nutrient. To see it in another way, we are learning to imitate an expert policy for feeding plants (imitation learning).
2. The problem could be solved for example using linear regression. The main advantage is the simplicity of the approach, while the main drawback is that it needs proper feature to works, hence, the complexity is moved to the feature engineering phase. One could instead resort to more complex approaches involving neural networks, in order to learn this high level feature mappings.
3. In this case, we can instead think the problem as an RL setting, in which states are the features we built from input sensors, actions are the nutrient values we want to supply, and the reward is given by the health index, which we want to maximize in the long run. Without having such an index, the problem cannot be framed in this way, since it lacks a proper reward function, hence, we can only imitate a policy we know to work well (fitting healthy plants data).

Exercise 6 (2 marks)

Are the following statements about Support Vector Machines (SVMs) True or False? Motivate your answers.

1. When using an SVM, the computational cost of computing predictions scales with the size of training samples.
2. When training a soft-margin SVM, the noisier is the data the larger should be set the value of hyperparameter C.
3. Hard-margin SVMs can be successfully applied also to datasets that appear to be not linearly separable.
4. The aim of the Kernel Trick is to limit the computational cost of the SVM training on datasets with a very large number of samples.

1. FALSE: SVM are sparse model and the prediction cost scales with the number of support vectors.
2. FALSE: the larger is C the lower is the bias (and conversely the higher the variance). This is exactly the opposite of what we should expect to do on a noisy dataset.
3. TRUE: applying the kernel function we could find the problem to bel linearly separable in the feature space.
4. FALSE: the computational cost of training SVM is unfortunately cubic in the number of training samples. Kernel trick allows to limit the cost of computing a large number of features.

Exercise 7 (4 marks)

Consider a binary perceptron classifier defined by parameters $w = [2, 1, 1]^\top$ with features vector $\phi([a, b]) = [a, b, ab]^\top$). Answer to the following questions related to the perceptron algorithm. Provide full calculations and clear motivations.

1. Given a new data point $(x, t) = ([a, b], t)$, explain the procedure you would follow to decide whether the classifier should be retrained.
2. Consider the data point $(x_1, t_1) = ([1, 2], +1)$. Update the classifier with the perceptron algorithm ($\alpha = 1$).
3. Consider the data point $(x_2, t_2) = ([-1, -2], +1)$. Update the classifier with the perceptron algorithm ($\alpha = 1$).
4. After the previous updates to the classifier, can we say the retrain procedure is completed?

1. First, we should check if the new data point is correctly classified, i.e., whether

$$sign(w^\top \phi(x)) = sign([2, 1, 1] \cdot [a, b, ab]^\top) = t$$

   If the answer is positive we do not need to retrain, otherwise we should retrain the model, i.e., use the stochastic gradient descent method until convergence.

2.
$$sign(w^\top \phi(x_1)) = sign([2, 1, 1] \cdot [1, 2, 2]^\top) = sign(6) = +1$$

   The data point x_1 is correctly classified, so we do not need to update the model.

3.
$$sign(w^\top \phi(x_2)) = sign([2, 1, 1] \cdot [-1, -2, 2]^\top) = sign(-2) = -1$$

   The data point x_2 is misclassified, so we have to update the model:

$$w \leftarrow w + \alpha\phi(x_2)t_2 = [2, 1, 1]^\top + [-1, -2, 2]^\top = [1, -1, 3]^\top$$

4. No, since we have updated the classifier, we should check if all the other data points in the dataset are still correctly classified.

Exercise 8 (4 marks)

Consider a MAB algorithm choosing the arm a_t and providing the following rewards R_t over a time horizon of $T = 10$ rounds.

$R_{1,t}$	**0**	1	1	0	**0**	1	0	1	**0**	0
$R_{2,t}$	1	**0**	1	1	1	1	**1**	1	1	**1**
$R_{3,t}$	0	1	**0**	**0**	0	**0**	1	**0**	1	1
a_t	1	2	3	3	1	3	2	3	1	2

(only the reward for the chosen arm is revealed to the algorithm) Moreover, assume that the expected value for the three arms' reward are $\mu_1 = 0.3$, $\mu_2 = 0.8$, and $\mu_3 = 0.6$.

1. Compute the cumulated reward, the regret and the pseudo-regret for the algorithm over the time horizon T. (Recall that the regret is computed over realizations, while the pseudo-regret is computed over expected values);

2. Compute the values for the UCB1 bounds for at time $t = 5$ for the three arms. Do you think that the algorithm used in this setting can be the UCB1?

3. Do you think that the algorithm used in this setting might be the Thompson sampling?

1.
   - The cumulated reward is: $0 + 0 + 0 + 0 + 0 + 1 + 0 + 0 + 1 = 2$
   - The regret is: $9 - 2 = 7$
   - The pseudo-regret is: $0.5 + 0 + 0.2 + 0.2 + 0.5 + 0.2 + 0 + 0.2 + 0.5 + 0 = 2.3$ ($\Delta_1 = 0.5$, $\Delta_2 = 0$, and $\Delta_3 = 0.2$)

2. $U_{1,5} = \frac{0}{2} + \sqrt{\frac{\log(5)}{2}}$, $U_{2,5} = \frac{0}{1} + \sqrt{\frac{\log(5)}{1}}$, and $U_{1,5} = \frac{0}{2} + \sqrt{\frac{\log(5)}{2}}$, therefore, the arm chosen for the next round would be $a_t = 2$. Consequently, the algorithm run above cannot be the UCB1.

3. Since the TS has a stochastic nature, there is the chance that any possible sequence of the arms is chosen, even if with a small probability. Therefore, the algorithm run above might be TS.