Mathematical Foundations of Machine Learning
Final Exam · 2022-2023

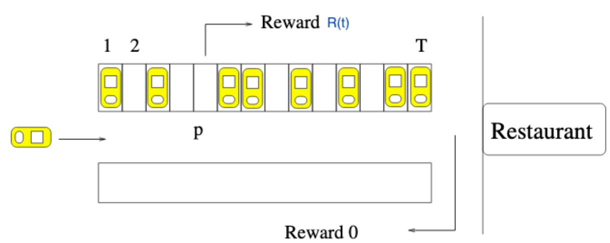
Please treat each problem on a separate copy

Duration: 2 hours, authorized documents: lecture slides and notes

Problem 1 is mandatory, and you have the <u>choice</u> of treating Problem 2 or Problem 3.

Problem 1 - Mandatory (over 5 points)
<u>Markov Decision Processes and Reinforcement Learning</u>

A driver wants to park her car as close as possible to the restaurant. There are T slots. The driver visits the slots one by one (starting from slot 1 to slot T). Each slot is either available or taken. The driver cannot see if a slot is available unless she is in front of the slot. When she sees a slot, she can decide to park now or to continue driving and hope to find a slot that is closer to the restaurant. If she chooses slot t, she earns a reward $R(t)$. If she reaches slot T and this slot is not available, she must go home and her reward is 0.



We assume that the probability for each slot to be available is $p \in [0, 1]$.

Question 1. [3 pts] We first consider that the driver knows p and wants to maximize her expected reward.

1.1 **[1 pt]** Formulate the problem as a Markov decision process. Explain what is the state space that you consider and the actions available in each of the states. If it helps, you can use a drawing.

1.2 **[1 pt]** Write down Bellman's equation, explain its signification, and write an algorithm that computes the optimal decision rule.

1.3 **[1 pt]** Assume that $p = 0.5$, $R = t$ and $T = 10$. Compute the optimal policy and its value.

Question 2. [2 pts] We now assume that p is unknown to the driver and needs to be learnt while parking. Your goal is to propose a learning algorithm to solves the optimal parking problem.

Indication: You can use one of the two approaches below. You are also welcome to propose your own solution.

- Use a Bayesian model and assume that the prior distribution on p is uniform on $[0, 1]$. Can you compute the probability of a slot to be free if k out of the first n slots where empty? Can you use this to formulate the problem as a Markov decision process with an extended state space? Describe then how you use this to compute an optimal decision rule?

- Assume that you have access to a simulator of the model and use a Q-learning approach (with any kind of value function approximation). Hint: for this you would need a MDP with an extended state space. What state space will you use? Describe an algorithm that use this to compute an optimal solution. Will it be efficient or would you need some approximation method?

You have to choose between Problem 2 or Problem 3

Problem 2 (over 15 points)
Nesterov acceleration

The point of this problem is to provide a regret-based analysis of the *Nesterov accelerated gradient (NAG)* algorithm, a famous method for minimizing smooth convex functions.

Let $f\colon \mathbb{R}^d \to \mathbb{R}$ be a differentiable convex function, and let $x^* \in \arg\min f$ be a global minimizer of f (not necessarily unique). We assume throughout that f is *Lipschitz smooth*, i.e., there exist some $L \geq 0$ such that

$$\|\nabla f(x') - \nabla f(x)\| \leq L\|x' - x\| \qquad \text{for all } x, x' \in \mathbb{R}^d. \tag{1}$$

You can take as given the following *descent inequality* that we studied in class: if $x^+ = x - \gamma \nabla f(x)$ for some $x \in \mathbb{R}^d$, $\gamma > 0$, then

$$f(x^+) \leq f(x) - \gamma(1 - L\gamma/2)\|\nabla f(x)\|^2 \tag{Descent}$$

The version of the *NAG* algorithm that we will consider proceeds as

$$\begin{aligned} y_{t+1} &= x_t - \frac{1}{L}\nabla f(x_t) \\ z_{t+1} &= z_t - \gamma_t \nabla f(x_t) \\ x_{t+1} &= \lambda_{t+1} y_{t+1} + (1-\lambda_{t+1}) z_{t+1} \end{aligned} \tag{NAG}$$

where $\gamma_t > 0$ and $\lambda_{t+1} \in [0,1]$ are parameters to be determined. The point of the problem is to derive the rate with which the algorithm converges to the minimum value of f.

Question 1. [1 pt] Show that

$$f(y_{t+1}) - f(x_t) \leq -\frac{1}{2L}\|\nabla f(x_t)\|^2$$

Question 2. [2 pts] Show that Let $w_t = \gamma_t \nabla f(x_t)$, and consider the sequence of loss functions $\ell_t(x) = \langle w_t, x\rangle$. Using the regret analysis for FTRL/OGD (or otherwise), show that z_t enjoys the regret guarantee

$$\sum_{t=1}^{T} \gamma_t \langle \nabla f(x_t), z_t - x^* \rangle \leq \frac{\|x_1 - x^*\|^2}{2} + \frac{1}{2}\sum_{t=1}^{T} \gamma_t^2 \|\nabla f(x_t)\|^2$$

Question 3. [3 pts] For all $t \geq 1$, show that

$$f(x_t) - f(x^*) \leq \frac{\lambda_t}{1-\lambda_t}[f(y_t) - f(x_t)] + \langle \nabla f(x_t), z_t - x^* \rangle$$

Hint: use the convexity of f to upper bound $f(x_t) - f(x^*)$, and show as an interim step that $x_t - z_t = \frac{\lambda_t}{1-\lambda_t}(y_t - x_t)$.

Question 4. [1 pt] Combining the above steps (or otherwise), show that

$$\sum_{t=1}^{T} \gamma_t [f(x_t) - f(x^*)] - \sum_{t=1}^{T} \frac{\lambda_t \gamma_t}{1-\lambda_t}[f(y_t) - f(x_t)] \leq \frac{\|x_1 - x^*\|^2}{2} + \sum_{t=1}^{T} L\gamma_t^2 [f(x_t) - f(y_{t+1})]$$

Question 5. [1 pt] Refactor the above to show that

$$\sum_{t=1}^{T} A_t f(y_{t+1}) - \sum_{t=1}^{T} B_t f(y_t) + \sum_{t=1}^{T} C_t f(x_t) \leq f(x^*) \cdot \Gamma_t + \frac{\|x_1 - x^*\|^2}{2}$$

where

$$A_t = L\gamma_t^2 \qquad B_t = \frac{\lambda_t \gamma_t}{1-\lambda_t} \qquad C_t = \frac{\gamma_t}{1-\lambda_t} - L\gamma_t^2 \qquad \text{and} \qquad \Gamma_t = \sum_{t=1}^{T} \gamma_t$$

Question 6. [2 pts] Show that

$$\frac{1}{\Gamma_t}\left[A_T f(y_{T+1}) - B_1 f(y_1) + \sum_{t=2}^{T}[A_{t-1} - B_t] f(y_t) + \sum_{t=1}^{T} C_t f(x_t)\right] \leq f(x^*) + \frac{\|x_1 - x^*\|^2}{2\Gamma_t}$$

To move forward, we will choose γ_t and λ_t so that only the term $f(y_{T+1})$ survives above.

Question 7. [2 pts] Show that $A_{t-1} - B_t = 0 = C_t$ if and only if

$$\lambda_t = 1 - \frac{1}{L\gamma_t} \qquad \text{and} \qquad \gamma_{t+1} = \frac{1 + \sqrt{1 + 4L^2\gamma_t^2}}{2L}$$

Question 8. [3 pts] Show that the choice above gives

$$\Gamma_T \geq \frac{T^2}{4L}$$

and conclude that the initialization $\lambda_1 = 0$ yields

$$f(y_{T+1}) - f(x^*) \leq \frac{2L\|x_1 - x^*\|^2}{T^2}$$

If you have treated Problem 2, don't treat Problem 3

Problem 3 (over 15 points)
RankBoost Algorithm

In this part we are interested in the analysis of a learning-to-rank algorithm called "RankBoost". Learning-to-rank is involved in the design of modern applications like search engines, information extraction platforms, or movie recommendation systems. The ordering in which the documents or movies are returned is an important component of the system in these applications. Due to resource constraints, ranking is preferred over binary classification in such cases; since it may be inconvenient or even impossible to display or process all elements that a classifier has labeled as relevant. For example, a user will only look at the top 10 or so results of a typical search engine, and not all the pertinent documents that are provided in answer to a query. A ranking problem can be specified as follows:

Observations (documents, movies, etc.) are described in an input space $\mathcal{X} \subseteq \mathbb{R}^d$; we suppose that there exists an unknown distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{X}$ which generates the pairs of examples, and, we denote by $f : \mathcal{X} \times \mathcal{X} \to \{-1, 0, +1\}$ a *preference function* defined as:

$$\forall (x, x') \in \mathcal{X} \times \mathcal{X}; f(x, x') = \begin{cases} +1, & \text{if } x' \text{ is preferred over } x; \\ 0, & \text{if } x' \text{ and } x \text{ have the same preference;} \\ -1, & \text{if } x \text{ is preferred over } x'. \end{cases}$$

A ranking algorithm learns a scoring function $h : \mathcal{X} \to \mathbb{R}$ in a way such that for a pair (x, x'), the score of a preferred example is higher than the score of an no-preferred example. In the contrary case, the pair is said to be misranked by h.

We suppose that we have a labeled training set $S = ((x_i, x'_i, y_i))_{1 \leqslant i \leqslant m} \in (\mathcal{X} \times \mathcal{X} \times \{-1, +1\})^m$, with $y_i = f(x_i, x'_i) \neq 0, \forall i \in \{1, \dots, m\}$. For a a scoring function $h : \mathcal{X} \to \mathbb{R}$, and we are interested in bounding the empirical error of the ranking loss on S:

$$\hat{R}(h) = \frac{1}{m} \sum_{i=1}^{m} \mathbb{1}_{y_i (h(x'_i) - h(x_i)) \leqslant 0} \tag{2}$$

where $\mathbb{1}_\pi = 1$ if the predicate π is true and 0 otherwise.

The pseudo-code of the Rankboost algorithm is given below. This algorithm combines prediction functions in the classe $\mathcal{H} = \{h : \mathcal{X} \to \{0, 1\}\}$ in order to create a better ranking function in terms of empirical ranking error. As the Adaboost algorithm, Ranboost trains a base ranker $h_t : \mathcal{X} \to \{0, 1\}$ at each iteration $t \in \{1, \dots, T\}$ and maintains a distribution D_t over training examples in $S = ((x_i, x'_i, y_i))_{1 \leqslant i \leqslant m} \in (\mathcal{X} \times \mathcal{X} \times \{-1, +1\})^m$. Each base ranker h_t is chosen with respect to the smallest value of $\epsilon_t^- - \epsilon_t^+$, with

$$\epsilon_t^s = \sum_{i=1}^{m} D_t(i) \mathbb{1}_{y_i (h_t(x'_i) - h_t(x_i)) = s}; \text{ for any } s \in \{-1, 0, +1\}. \tag{3}$$

We simplify the notation ϵ_t^{-1} and ϵ_t^{+1} with respectively ϵ_t^- and ϵ_t^+.

Algorithm 1 RankBoost($S = ((x_i, x'_i, y_i))_{1 \leqslant i \leqslant m}$)

1: **for** $i \leftarrow 1$ to m **do**

2: $\quad D_1(i) \leftarrow \frac{1}{m}$

3: **end for**

4: **for** $t \leftarrow 1$ to T **do**

5: $\quad$ Choose $h_t \leftarrow$ base ranker in $\mathcal{H}$ with smallest $\epsilon_t^- - \epsilon_t^+$

6: $\quad$ Set $\alpha_t \leftarrow \frac{1}{2} \log \frac{\epsilon_t^+}{\epsilon_t^-}$

7: $\quad$ Set $Z_t \leftarrow \sum_{i=1}^m D_t(i) e^{-\alpha_t y_i \left(h_t(x'_i) - h_t(x_i)\right)}$ ▷ normalization factor

8: $\quad$ **for** $i \leftarrow 1$ to m **do**

9:

$$D_{t+1}(i) \leftarrow \frac{D_t(i) e^{-\alpha_t y_i \left(h_t(x'_i) - h_t(x_i)\right)}}{Z_t} \tag{4}$$

10: $\quad$ **end for**

11: **end for**

12: **return** $g \leftarrow \sum_{t=1}^T \alpha_t h_t$

Question 1. [2 pts] Explain each of the quantities ϵ_t^+, ϵ_t^- and ϵ_t^0? What is the meaning of "smallest value of $\epsilon_t^- - \epsilon_t^+$"?

We can show that the following equality holds at each iteration t:

$$\epsilon_t^+ + \epsilon_t^- + \epsilon_t^0 = 1. \tag{5}$$

The distribution over the examples is updated with respect to the equation (4) (lines 7-10).

Question 2. [2 pts] After T iterations, the algorithm outputs a scoring function $g : x \mapsto \sum_{t=1}^T \alpha_t h_t(x)$. Show that the ranking loss for g, $\hat{R}(g)$, is bounded by:

$$\hat{R}(g) \leqslant \frac{1}{m} \sum_{i=1}^m e^{-y_i (g(x'_i) - g(x_i))}$$

Question 3. [2 pts] With respect to the definitions of lines 2 and 7 of Algorithm 1; show that

$$\frac{1}{m} \sum_{i=1}^m e^{-y_i (g(x'_i) - g(x_i))} = \prod_{t=1}^T Z_t$$

Question 4. [1 pt] Following the definition of $\epsilon^s; s \in \{-1, 0, +1\}$ (3) show that $\forall t \in \{1, \dots, T\}, Z_t$ can be rewritten as:

$$Z_t = \epsilon_t^+ e^{-\alpha_t} + \epsilon_t^- e^{\alpha_t} + \epsilon_t^0 \tag{6}$$

Question 5. [1 pt] For which value of α_t, Z_t is minimized. Is this coherent with the choice of α_t in the algorithm 1 (line 6)?

Question 6. [1 pt] By plugging back the value of α_t obtained previously in Z_t; show that

$$Z_t = 2\sqrt{\epsilon_t^+ \epsilon_t^-} + \epsilon_t^0.$$

Question 7. [2 pts] From Equation (5), show that

$$4\epsilon_t^+ \epsilon_t^- = (1 - \epsilon_t^0)^2 - (\epsilon_t^+ - \epsilon_t^-)^2$$

Deduce that

$$Z_t = (1 - \epsilon_t^0)\sqrt{1 - \frac{(\epsilon_t^+ - \epsilon_t^-)^2}{(1 - \epsilon_t^0)^2}} + \epsilon_t^0$$

Question 8. [4 pts] We now suppose that $\epsilon_t^0 = 0; \forall t$, using the inequality $\forall z \in \mathbb{R}$; $1 - z \leqslant e^{-z}$ show that

$$Z_t \leqslant \exp\left(-\frac{(\epsilon_t^+ - \epsilon_t^-)^2}{2}\right)$$

Deduce that if there exists γ, such that $\forall t \in \{1, \dots, T\}, 0 < \gamma \leqslant (\epsilon_t^+ - \epsilon_t^-)$, then:

$$\hat{R}(g) \leqslant \exp\left(-\frac{\gamma^2 T}{2}\right)$$

What is the interpretation of this result?