

IN4320 Machine Learning Homework Exercise 3

Online Learning

In online learning we observe a stream of data and we make predictions for new incoming data. After each prediction we observe some performance measure and we adjust our prediction scheme accordingly. Well known applications are spam classification and investment.

Exercise 1 serves as an introductory exercise into the online learning with expert advice setting (mix loss). Exercise 1d may take some time. Exercise 2 again considers the expert setting, but with the dot loss. Exercise 3 deals with online convex optimization (OCO). Exercise 3b is tricky, here we ask you to do an adapted proof for online gradient descent (OGD). In exercise 4 you apply the online learning theory to bitcoin / altcoin trading. You will implement the aggregating algorithm (AA) and OGD to trade bitcoins / altcoins automatically.

In the exercises we adopt the notation of Bubeck [2011], which we also used in the lecture.

We would prefer you to typeset your answers using latex and that you hand in your assignment by PDF on brightspace. Please include your (matlab) code in the appendix.

Exercise 1 Assume we are in the online learning setting with $d = 3$ experts e_1, e_2 and e_3 and we evaluate using the mix loss $l_m(p_t, z_t) = -\log(\sum_{i=1}^d p_t e^{-z_t^i})$.

Consider the following adversary moves z_t for $t = 1, \dots, 4$:

	z_1	z_2	z_3	z_4
e_1	0	0	1	0
e_2	0.1	0	0	0.9
e_3	0.2	0.1	0	0

Consider the following strategies:

A in round t choose $p_t = e_{b(t)}$, where the expert $b(t)$ is the expert with the smallest cumulative loss $L_{t-1}^i = \sum_{s=1}^{t-1} l(e_i, z_s)$ up to time t . In other words $b(t) = \arg \min_i L_{t-1}^i$. For $t = 1$ choose $p_1 = (1/3, 1/3, 1/3)$.

B the strategy of the Aggregating Algorithm (AA).

- (a) Compute the selected actions p_t for $t = 1, \dots, 4$ for strategy A and B.
- (b) Compute the total mix loss obtained by all strategies and compute the corresponding expert regrets of all strategies after $n = 4$ rounds.
- (c) For this example, we have shown in the lecture that for AA we have the theoretical guarantee¹ $R_E^n \leq \log(d)$. Use this guarantee to derive an upperbound on the cumulative mix loss of AA for $n = 4$: $\sum_{t=1}^4 l_m(p_t, z_t) \leq C$. Compute the numerical value for C for this example. How does the theoretical guarantee compare with the cumulative mix loss obtained by both strategies?

(d) In this exercise we will show that this theoretical guarantee $R_E^n \leq \log(d)$ cannot be improved (in other words, this bound is ‘tight’). To show this, show that for any strategy p_t we can find an adversary $z_t \in (\infty, \infty]^d$ such that $R_n \geq \log d$. *Hint: first consider the case $n = 1$.*

¹See the slides and lecture for a proof. Cesa-Bianchi and Lugosi [2006, Chapter 3.5] also give a proof.

Exercise 2 Now we consider the online learning with expert advice setting with d experts and the dot loss $l_d(p_t, z_t) = p_t^T z_t$. Consider the strategies:

- A the Exp strategy with learning rate $\eta_A = \sqrt{4 \log(d)}$.
- B the Exp strategy with learning rate $\eta_B = \sqrt{2 \log(d)}$.

Define R_n^A as the expert regret of strategy A and R_n^B as the expert regret of strategy B. We do not consider any specific adversary this time.

(a) For both strategies, compute the theoretical guarantee² on the expert regret for $n = 2$. With other words, compute C_n^A for which $R_n^A \leq C_n^A$ for A and compute the value C_n^B for which $R_n^B \leq C_n^B$ for B. Which bound is tighter for $n = 2$?

(b) Now do the same for $n = 4$. Which bound is tighter this time?

(c) Lets assume that strategy B has a tighter bound for some n . Does that necessarily mean that for any adversary moves z_t for $t = 1, \dots, n$, that $R_n^B \leq R_n^A$? You may give an informal argument to explain your answer (an example or proof is not required).

(d) Generally, how does the strategy p_t change when we increase the learning rate? What do you expect to happen if we set the learning rate extremely high ($\eta = \infty$)?

Exercise 3 Assume we are in the online convex optimization (OCO) setting. That means we are given a closed convex set $\mathcal{A} \subset \mathbb{R}^d$ and a convex loss function $l : \mathcal{A} \times \mathcal{Z} \rightarrow \mathbb{R}$ and in each round t we take an action $a_t \in \mathcal{A}$. Our goal is to minimize the OCO regret $R_n = \sum_{t=1}^n l(a_t, z_t) - \min_{a \in \mathcal{A}} l(a, z_t)$, where z_t is a parameter chosen by an adversary in round t . Assume that for all $(a, z) \in (\mathcal{A} \times \mathcal{Z})$ the loss $l(a, z)$ is differentiable with respect to a and that the gradient with respect to a is bounded: $\|\nabla_a l(a, z)\| \leq G$. Assume further that $\max_{a \in \mathcal{A}} \|a\| \leq R$. In this setting we can use the online gradient descent method with a fixed learning rate $\eta \in (0, \infty)$. In the lecture we showed that this procedure achieves a OCO regret of $R_n \leq \frac{R^2}{2\eta} + \frac{\eta G^2 n}{2}$ for any adversary.

(a) Derive that the optimal learning rate is given by $\eta_n = \frac{R}{G\sqrt{n}}$. This means that η_n minimizes the right hand side of the OCO regret bound for R_n from above. Show for this optimal choice $R_n \leq RG\sqrt{n}$.

To set the learning rate η according to **3a** we need to know the number of rounds n in advance. Sometimes we do not know n , and we want to perform well for any n . To get a guarantee on the OCO regret for all n , we can make η dependent on time t .

(b) Show that if we set $\eta_t = \frac{R}{G\sqrt{t}}$ and $\frac{1}{\eta_0} = 0$ we can bound for all $n \geq 1$ the OCO regret by $R_n \leq \frac{3}{2}GR\sqrt{n}$. Explain every step you do as detailed as possible.

Hints: The exercise might be a challenge, but you will manage. You can follow for most parts the online gradient descent proof³. The step of the telescoping sum, however, will be more difficult. Write out the sum for the first few terms of t and try to re-order it in a way that allows you to use the bound $\|a\| \leq R$ for all $a \in \mathcal{A}$, then telescope. At some point in the proof you might want to use the inequality $\sum_{t=1}^n \frac{1}{\sqrt{t}} \leq 2\sqrt{n}$.

²See the lecture (slides), or see [Bubeck, 2011, Chapter 2.2]. Note: you don't have to give a proof!

³See the lecture, or see [Bubeck, 2011, Chapter 4.1]

Exercise 4 We now turn our focus to an application in investment called universal portfolio theory, where we model investment as a repeated decision making scenario. You will first show that in this setting we should use the **mix-loss**. Then you will download bitcoin data and implement two algorithms that automatically trade bitcoins / altcoins.

The setting of the portfolio theory is the following. Assume that on a given day t we can distribute our wealth on **d different assets** (for example stocks or several types of bitcoin). Each asset i has a given price $x_t^i \in (0, \infty)$ on day t (for simplicity we assume that the price of the asset cannot drop to 0).

(a) Assume that at time t we have a wealth of $W_t \in \mathbb{R}$. Assume we distribute our wealth W_t at time t according to $p_t = (p_t^1, \dots, p_t^d) \in \Delta_d$ on the d different assets⁴. Given the asset prices x_t^i and x_{t+1}^i , derive the value of our wealth W_{t+1} on the next day.

Hint: Introduce $r_t^i = x_{t+1}^i / x_t^i$.

Now you show why we should optimize the mix loss in investment.

(b) Using 4a, show that with the adversary moves $z_t^i = -\log(r_t^i)$ that

$$\sum_{t=1}^n -\log \left(\sum_{i=1}^d p_t^i e^{-z_t^i} \right) = -\log(W_n/W_1).$$

Explain why the **mix loss is thus appropriate** for this setting and explain why we want the total mix loss to be as small as possible.

In this exercise we use some bitcoin data collected by Jesse Vent ⁵. We have prepared the data and provide you with some example code. Download and inspect the code. We consider the **5 most popular bitcoins** / altcoins: BCH (Bitcoin Cash), BTC (Bitcoin), LTC (Litecoin), XRP (Ripple), ETH (Ethereum). Each coin is worth a certain amount of US dollars each day. Our algorithm will in the first round $t = 1$ buy an equal amount of coins (so our wealth will be distributed as 20% BCH, 20% BTC, 20% ETH, etc...). Then afterward, each day $t = 2, \dots, 213$ our algorithm will sell some coins and buy new ones (redistribute our wealth), depending on past observed behavior of the coins. We will use the AA algorithm in 4c and the OGD algorithm in 4d. In both cases we will use the **mix loss**. Please note that this homework in no way constitutes investment advice, and that investing in bitcoin and altcoin carries a high risk.

⁴Example: if we have $W_t = 100$ euro, and $p_t^1 = 0.2$, this means on timestep t we buy / sell an amount of asset 1 such that afterward the amount we own of asset 1 is worth 20 euro.

⁵<https://www.kaggle.com/jessevent/all-crypto-currencies>

(c) We will now consider coin trading in the expert setting as follows. We have five experts, e_1, \dots, e_5 . Expert e_1 in each round will choose move $e_1 = (1, 0, 0, 0, 0)$, or with other words, expert i will always invest all its money in asset i . We will re-invest our wealth according to $p_t \in \Delta_5$ on each day $t = 1, \dots, 213$, where p_t is computed by AA.

1. For this exercise we have provided some code. Inspect the given files.
2. Compute the adversary moves $z_t^i = -\log(r_t^i)$ and the total loss of each expert.
3. Implement AA (`AA.m`) and compute the expert regret of AA (use $n = 213$).
4. How does the total loss of AA compare with the total loss of the experts? How does the expert regret of AA compare with the guaranteed expert regret? Is the adversary generating ‘difficult’ data?
5. Visualize the strategy of AA using a plot of p_t vs t . Also visualize the values of the coins, x_t vs t . What do you observe? Can you explain the strategy?
6. If we would have invested according to the AA strategy, how much would our wealth have increased?

(d) We will now consider coin trading in the online convex optimization setting. Now the action space $\mathcal{A} = \Delta_5$ (recall this means that for all $a_t \in \mathcal{A}$ we have that $\sum_{i=1}^d a_t^i = 1$ and $a_t^i \in [0, 1]$). The action determines how we invest our money in bitcoin on each day $t = 1, \dots, 213$. The loss function is again the mix loss $l(a_t, z_t) = l_m(a_t, z_t)$.

1. For this example, how is the OCO regret different from the expert regret considered above?
2. Show that the gradient of the mix loss is equal to

$$\nabla_a l_m(a, z_t) = \frac{-r_t}{a_t^T r_t}$$

3. Implement `mix_loss.m`.
4. Implement online gradient descent (OGD) in `OGD.m`. Use the learning rate $\eta = \frac{R}{G\sqrt{n}}$. For the OCO regret guarantee of this value of η see also exercise **3a**.
5. We have provided values R and G for you to use for this data (in `OGD.m`). Can you explain why these values are OK to for these particular adversary moves z_t ?
6. Run OGD for $n = 213$. Visualize the strategy of OGD as in **4c**. How much would our wealth have increased if we would have used OGD to invest?
7. Compute the optimal ‘fixed’ action $a \in \mathcal{A}$ (you may use an optimization toolbox, and you may use `loss_fixed_action.m`). Compute the OCO regret of OGD.
8. How does the loss of OGD compare with the loss of the best fixed action, and how does the OCO regret of OGD compare with the guaranteed OCO regret (see **3a**)? Is the adversary generating ‘difficult’ data?
9. Which part of the OGD strategy described here is unrealistic?

Bibliography

Sébastien Bubeck. Introduction to online optimization. *Lecture Notes*, pages 1–86, 2011.

Nicolo Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge university press, 2006.