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.

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 any code you use in the appendix.

Exercise 1 [15pt] 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 $l_m(p_t, z_t) = -\log(\sum_{i=1}^{d} p_t^i e^{-z_t^i})$.

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

	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 $L_{t-1}^i = \sum_{s=1}^{t-1} z_s^i$ 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) [5pt] Compute the selected actions p_t for $t = 1, \ldots, 4$ for strategy A and B. Put the values p_t in a table, where rows correspond to experts e and columns correspond to adversary moves z, and round the probabilities to two decimals (for example, 0.33) — this is the same style as the table above. If you use code, make sure to give it. If you do the calculations by hand (or calculator), show at least one calculation (formulas and all steps) for one p_t^i for $t > 1$.

(b) [2pt] Compute the total mix loss obtained by all strategies.

(c) [3pt] Compute the corresponding expert regrets of all strategies after $n = 4$ rounds. Show how you got to your answer.

(d) [2pt] For this example, we have shown in the lecture that for AA we have the theoretical guarantee[1] $R_n^E \leq \log(d)$. Use this guarantee to derive an upperbound on the cumulative mix loss of AA for $n = 4$: $\sum_{t=1}^{4} l_m(p_t, z_t) \leq C$. Give the formula for C.

(e) [1pt] Compute the numerical value for C for this example.

(f) [2pt] Are the mix loss of strategy A and B smaller than the constant C for this example? Why (or why not)?

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

Exercise 2 **[10pt]** In this exercise we will show that the theoretical guarantee for the expert setting: $R_n^E \leq \log(d)$ cannot be improved (in other words, this bound is 'tight'). Assume again we are in the online learning setting with d experts and n rounds. We evaluate using the mix loss.

(a) **[3pt]** From now on, assume $n = 1$ and $d = 2$. Say $p_1 = (a, 1-a)$, where $0 \leq a \leq 1$. What is the best adversary move $z_1 \in (-\infty, \infty]^2$ (that maximizes the regret)? The adversary move can depend on the value a. Hint: you may try some values of a and write down some adversary moves z to get a 'feeling' for what the adversary should do.

(b) **[3pt]** Show that for all possible values of a there is an adversary move such that the regret is larger or equal to $log(d)$. You may still take $d = 2$.

(c) **[3pt]** Now we go to $d > 2$. So $p_1 = (p_1^1, p_1^2, \ldots, p_1^d)$ where $\sum_i p_1^i = 1$. Show again that for any p_1 there is an adversary move $z_1 \in (-\infty, \infty]^d$ such that the regret is larger or equal to $log(d)$ for $n = 1$.

(d) **[1pt]** Now we go to $n > 1$. So now we are given multiple moves p_t for $t = 1, \ldots, n$. Show that, given a set of moves $p_1, \ldots, p_n$, we can find a set of moves $z_1, \ldots, z_n$ such that the regret is larger or equal to $log(d)$. Hint: you can re-use the approach of the previous questions to get a regret of $log(d)$ for $t = 1$.

Exercise 3 **[20pt]** 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) **[3pt]** For both strategies, compute the theoretical guarantee[2] 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. Simplify C_n^A and C_n^B as much as possible.

(b) **[1pt]** Compute $\frac{C_n^A}{C_n^B}$ numerically for $n = 2$.

(c) **[1pt]** Which bound is tighter for $n = 2$? Or with other words, is C_n^A or C_n^B smaller?

(d) **[3pt]** Compute C_n^A and C_n^B for $n = 4$. Again simplify as much as possible.

(e) **[1pt]** Compute $\frac{C_n^A}{C_n^B}$ numerically for $n = 4$.

(f) **[1pt]** Which bound is tighter for $n = 4$? Or with other words, is C_n^A or C_n^B smaller?

(g) **[5pt]** Lets assume that strategy B has a tighter bound for some n, meaning $C_n^B < C_n^A$. Does that necessarily mean that for any adversary moves z_t for $t = 1, \ldots, n$, 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).

(h) **[2pt]** How does the strategy p_t change when we increase the learning rate?

(i) **[3pt]** What happens to the strategy if we set the learning rate extremely high ($\eta = \infty$)? Do we still have a guarantee on the regret?

[2] See the lecture (slides), or see [Bubeck, 2011, Chapter 2.2].

Exercise 4 [20pt] We will do portfolio selection in the expert setting. You will first show that in the expert setting we should use the mix-loss. Then you will download bitcoin data and develop an algorithm that automatically trades 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).

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 $p_t = (p_t^1, ..., p_t^d) \in \Delta_d$ on the d different assets[3]. Given the asset prices x_t^i and x_{t+1}^i, we define $r_t^i = x_{t+1}^i / x_t^i$. Then the value of our wealth W_{t+1} on the next day is given by:

$$W_{t+1} = p_t^T r_t W_t$$

If we now choose $z_t^i = -\log(r_t^i)$ as the adversary moves, the mix loss is given by:

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

.

(a) [2pt] Show the above equality.

(b) [1pt] Why is the mix loss appropriate to minimize in the investment setting?

In this exercise we use some bitcoin data collected by Jesse Vent [4]. 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 $t = 2, \ldots, 213$ 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 to trade coins. This homework in no way constitutes investment advice, and note that investing in bitcoin and altcoin carries a high risk.

We will now consider coin trading in the expert setting as follows. We have five experts, $e_1, \ldots, e_5$. 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 $t = 1, \ldots, 213$, where p_t is computed by AA.

(c) [0pt] For this exercise we have provided some code. Inspect the given files.

(d) [0pt] Compute the adversary moves $z_t^i = -\log(r_t^i)$. You don't have to put the

[3] Example: 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.

[4] `https://www.kaggle.com/jessevent/all-crypto-currencies`

values in the report.

(e) **[1pt]** Compute the loss of each expert numerically and report them in a table.

(f) **[4pt]** Implement the AA algorithm (`AA.m`). Please include code in the appendix.

(g) **[2pt]** Compute the loss of AA at $n = 213$ and report it.

(h) **[2pt]** What is the regret of AA? Report it. Show how you got to your answer, using formulas if necessary.

(i) **[1pt]** How does the loss of AA compare with the loss of each expert?

(j) **[1pt]** How does the expert regret of AA compare with the guaranteed expert regret?

(k) **[1pt]** Is the adversary generating 'difficult' data? Explain why (or why not)?

(l) **[1pt]** Visualize the strategy of AA using a plot of p_t vs t. Also visualize the values of the coins, x_t vs t.

(m) **[2pt]** Generally, if coins raise in value AA will invest in them. However, if you look carefully at the plot, you may observe that a coin might raise in value, however the AA strategy will invest less in that coin in the next time step. Show such an example using a plot (zoom in on the behaviour and mark it) and explain why this may happen.

(n) **[1pt]** If we would have invested according to the AA strategy, how much would our wealth have increased?

(o) **[1pt]** Do you think in reality we could also make such a large profit? Why (or why not)? To address this question you need to have some knowledge about investment and / or bitcoin trading.

Exercise 5 **[15pt]** 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 $l : \mathcal{A} \times \mathcal{Z} \to \mathbb{R}$ and in each round t we take an action $a_t \in \mathcal{A}$. Our goal is to minimize the OCO regret $R_n = \sum_{t=1}^{n} l(a_t, z_t) - \min\limits_{a \in \mathcal{A}} l(a, z_t)$, 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 $\max\limits_{a \in \mathcal{A}} ||a|| \leq R$. 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) **[5pt]** 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 that $R_n \leq RG\sqrt{n}$.

We now look at the portfolio selection in the OCO setting.

(b) **[5pt]** In this exercise we show that the 'constant rebalanced portfolio' can already be quite powerful. Show that a constant rebalanced portfolio can have exponential wealth increase, while investing in any single stock leads to exponential wealth decrease. More precisely; Construct two series of numbers with values in $(0, \infty)$, which represent the returns of two stocks, such that:

- The W_t decreases exponentially in the number of rounds t, when we invest in any individual stock.

- The wealth W_t increases exponentially in the number of rounds t, when we distribute our wealth in each round new, according to a fixed distribution.

(c) [5pt] Write pseudo-code for a portfolio selection algorithm based on online gradient descent. That is, given a set of return vectors, spell out the exact constants and updates based upon the gradients (calculate!) of the payoff functions. How would you chose G and R for the optimal learning rate as derived in **5(a)**?

Exercise 6 [Bonus 10pt, this exercise is optional, you can achieve full score for the homework without it]

To set the learning rate η according to **5(a)** 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 the learning rate η dependent on the time t.

Show that if we set $\eta_t = \frac{R}{G\sqrt{t}}$ 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 proof*[5]*. 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 $\sum_{t=1}^{n} \frac{1}{\sqrt{t}} \leq 2\sqrt{n}$.*

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

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.