# IN 4320 Machine Learning Assignment 3

Xiangwei Shi
4614909

**Exercise 1**

(a)

**Strategy A**: Since we adopt the mix loss $l_m$ as evaluation, $I_m(e_i, z_s) = z_t^i$ and the cumulative loss becomes $L_{t-1}^i = \sum_{s=1}^{t-1} z_s^i$. For $t = 1$, the selected action $p_1$ is given, which is $p_1 = (1/3, 1/3, 1/3)$. For $t = 2, 3, 4$, the selected action $p_t$ can be decided by the smallest cumulative loss, $\sum_{s=1}^{t-1} z_s^i$. According to the adversary moves $z_t$ for $t = 1, \cdots, 4$, the cumulative loss of different $t$s and experts are shown in Table 1 and the selected actions $p_t$ for $t = 1, \cdots, 4$ are shown in Table 2.

| t | $e_1$ | $e_2$ | $e_3$ |
|---|---|---|---|
| 2 | 0 | 0.1 | 0.2 |
| 3 | 0 | 0.1 | 0.3 |
| 4 | 1 | 0.1 | 0.3 |

Table 1: The cumulative loss of different $t$s and experts for strategy A

| t | $p_t$ |
|---|---|
| 1 | $p_1 = (1/3, 1/3, 1/3)$ |
| 2 | $p_2 = (1, 0, 0)$ |
| 3 | $p_3 = (1, 0, 0)$ |
| 4 | $p_4 = (0, 1, 0)$ |

Table 2: The selected actions $p_t$ for $t = 1, \cdots, 4$ for strategy A

**Strategy B**: Since the strategy has become Aggregating Algorithm, the definition of cumulative loss of different experts up to different $t$s is $L_t^i = \sum_{s=1}^{t} z_s^i$. In terms of AA strategy, the selected actions are defined as $p_t^i = \frac{\exp^{-L_{t-1}^i}}{C_{t-1}}$, where $C_{t-1} = \sum_{j=1}^{i} \exp^{-L_{t-1}^j}$. Therefore, the selected actions $p_t$ for $t = 1, \cdots, 4$ can be calculated and shown in Table 3.

(b)

**Strategy A**: According to the definition of mix loss, $l_m(p_t, z_t) = -In(\sum_{i=1}^{d} p_t^i \exp^{-z_t^i})$, the mix losses for $t = 1, \cdots, 4$ are listed in Table 4. The total mix loss of strategy A is 1.9967. As for the expert regret after 4 rounds, it can be calculated by $R_n^E = \sum_{t=1}^{n} l(p_t, z_t) - \min_i \sum_{t=1}^{n} l(e_i, z_t) = \sum_{t=1}^{n} l_m(p_t, z_t) - \min_i \sum_{t=1}^{n} z_t^i$ for mix loss. From the table of adversary moves, we can conclude that expert $e_3$ is the best expert, whose total loss

| t | $p_t$ |
|---|---|
| 1 | $p_1 = (1/3, 1/3, 1/3)$ |
| 2 | $p_2 = (\frac{\exp^{0}}{\exp^{0}+\exp^{-0.1}+\exp^{-0.2}}, \frac{\exp^{-0.1}}{\exp^{0}+\exp^{-0.1}+\exp^{-0.2}}, \frac{\exp^{-0.2}}{\exp^{0}+\exp^{-0.1}+\exp^{-0.2}}) = (0.3672, 0.3322, 0.3006)$ |
| 3 | $p_3 = (\frac{\exp^{0}}{\exp^{0}+\exp^{-0.1}+\exp^{-0.3}}, \frac{\exp^{-0.1}}{\exp^{0}+\exp^{-0.1}+\exp^{-0.3}}, \frac{\exp^{-0.3}}{\exp^{0}+\exp^{-0.1}+\exp^{-0.3}}) = (0.3780, 0.3420, 0.2800)$ |
| 4 | $p_4 = (\frac{\exp^{-1}}{\exp^{-1}+\exp^{-0.1}+\exp^{-0.3}}, \frac{\exp^{-0.1}}{\exp^{-1}+\exp^{-0.1}+\exp^{-0.3}}, \frac{\exp^{-0.3}}{\exp^{-1}+\exp^{-0.1}+\exp^{-0.3}}) = (0.1827, 0.4494, 0.3679)$ |

Table 3: The selected actions $p_t$ for $t = 1, \cdots, 4$ for strategy B

is 0.3, smaller than those of $e_1(1)$ and $e_2(1)$. The expert regret of strategy A after 4 rounds is $1.9967 - 0.3 = 1.6967$.

| t | $l_m$ |
|---|---|
| 1 | $l_m(p_1, z_1) = -In(\exp^0/3 + \exp^{-0.1}/3 + \exp^{-0.2}/3) = 0.0967$ |
| 2 | $l_m(p_2, z_2) = -In(\exp^0 * 1) = 0$ |
| 3 | $l_m(p_3, z_3) = -In(\exp^{-1} * 1) = 1$ |
| 4 | $l_m(p_4, z_4) = -In(\exp^{-0.9} * 1) = 0.9$ |

Table 4: The selected actions $p_t$ for $t = 1, \cdots, 4$ for strategy A

**Strategy B**: The mix losses for $t = 1, \cdots, 4$ are listed in Table 5. The total mix loss of strategy B is 0.7089. Same as strategy A, the best expert is $e_3$. The expert regret of strategy B after 4 rounds is $0.7089 - 0.3 = 0.4089$.

| t | $l_m$ |
|---|---|
| 1 | $l_m(p_1, z_1) = -In(\exp^0/3 + \exp^{-0.1}/3 + \exp^{-0.2}/3) = 0.0967$ |
| 2 | $l_m(p_2, z_2) = -In(\exp^0 * 0.3672 + \exp^0 * 0.3322 + \exp^{-0.1} * 0.3006) = 0.0290$ |
| 3 | $l_m(p_3, z_3) = -In(\exp^{-1} * 0.3780 + \exp^0 * 0.3420 + \exp^0 * 0.2800) = 0.2730$ |
| 4 | $l_m(p_4, z_4) = -In(\exp^0 * 0.1827 + \exp^{-0.9} * 0.4494 + \exp^0 * 0.3679) = 0.3102$ |

Table 5: The selected actions $p_t$ for $t = 1, \cdots, 4$ for strategy B

(c)
In terms of the theorem, for any adversary and any $n > 0$, if $l$ is the mix loss $l_m$, we have for Aggregating Algorithm that $R_n^E \leq In(d)$. We know that for Aggregating Algorithm and mix loss, the expert regret $R_n^E$ can be written as $\sum_{t=1}^n l_m(p_t, z_t) - \min_i \sum_{t=1}^n z_t^i$, which means $\sum_{t=1}^n l_m(p_t, z_t) \leq \min_i \sum_{t=1}^n z_t^i + In(d)$. Therefore, for this example, the cumulative mix loss of AA for $n = 4$ has an upper bound, which means $\sum_{t=1}^4 l_m(p_t, z_t) \leq \min_i \sum_{t=1}^n z_t^i + In(3)$. The numerical value for $C$ for this example is $0.3 + 1.0986 = 1.3986$. Compared these two strategies, clearly the cumulative mix loss obtained by strategy B (Aggregating Algorithm) meets this theoretical guarantee, while the one of strategy A does not.
(d)
Firstly, let us consider $n = 1$ and with $d$ experts. The expert regrets should be

$$R_n^E = -In(p_1 e^{-z_1} + p_2 e^{-z_2} + \cdots + p_d e^{-z_d}) - \min(z_1, z_2, \cdots, z_d)$$

Since, we want to find an adversary that makes the expert regrets $R_n^E \geq \log(d)$, which means

$$-In(p_1e^{-z_1} + p_2e^{-z_2} + \cdots + p_de^{-z_d}) - \min(z_1, z_2, \cdots, z_d) \geq \log(d)$$

$$In(p_1e^{-z_1} + p_2e^{-z_2} + \cdots + p_de^{-z_d}) + \min(z_1, z_2, \cdots, z_d) \leq -\log(d)$$

$$In(p_1e^{-z_1} + p_2e^{-z_2} + \cdots + p_de^{-z_d}) + \min(z_1, z_2, \cdots, z_d) + \log(d) \leq 0$$

Assuming $\min(z_1, \cdots, z_d) = z_1$ and $\min(p_1, \cdots, p_d) = p_1$, the inequality will become

$$In(p_1e^{-z_1} + p_2e^{-z_2} + \cdots + p_de^{-z_d}) + z_1 + \log(d) \leq 0$$

$$In(p_1e^{-z_1} + p_2e^{-z_2} + \cdots + p_de^{-z_d}) + Ine^{z_1} + Ind \leq 0$$

$$In((p_1e^{-z_1} + p_2e^{-z_2} + \cdots + p_de^{-z_d}) \times e^{z_1} \times d) \leq 0$$

$$In(dp_1 + dp_2e^{z_1-z_2} + \cdots + dp_de^{z_1-z_d}) \leq 0$$

$$dp_1 + dp_2e^{z_1-z_2} + \cdots + dp_de^{z_1-z_d} \leq 1$$

Because of the assumption, when the rest adversaries are as $\infty$, the left sider will be only $dp_1$. Since $p_1$ is the smallest among any actions of $d$ experts, we can get $p_1 \leq \frac{1}{d}$, which makes $0 \leq dp_1 \leq 1$. The inequality above is established and we can find an adversary makes $R_n^E \geq \log(d)$ when $n = 1$.
Similarly, when $n > 1$, we have

$$-In(p_{11}e^{-z_{11}} + \cdots + p_{1d}e^{-z_{1d}} + p_{21}e^{-z_{21}} + \cdots + p_{nd}e^{-z_{nd}}) - \min(z_{11} + \cdots + z_{n1}, \cdots, z_{1d} + \cdots + z_{nd}) \geq lod(d)$$

Assuming $\min(z_{11} + \cdots + z_{n1}, \cdots, z_{1d} + \cdots + z_{nd}) = z_{11} + \cdots + z_{n1}$, and the inequality becomes

$$In(p_{11}e^{-z_{11}} + \cdots + p_{1d}e^{-z_{1d}} + p_{21}e^{-z_{21}} + \cdots + p_{nd}e^{-z_{nd}}) + z_{11} + \cdots + z_{n1} + Ind \leq 0$$

$$In(dp_{11}e^{-z_{11}} + \cdots + dp_{1d}e^{-z_{1d}} + dp_{21}e^{-z_{21}} + \cdots + dp_{nd}e^{-z_{nd}}) + z_{11} + \cdots + z_{n1} \leq 0$$

We can find when $z_{12}, \cdots, z_{n2}, \cdots, z_{nd}$ are set as $\infty$, we get

$$In(dp_{11}e^{-z_{11}} + \cdots + dp_{1d}e^{-z_{1d}}) + z_{11} + \cdots + z_{n1} \leq 0$$

$$dp_{11}e^{z_{21}+\cdots+z_{n1}} + dp_{21}e^{z_{11}+z_{31}+\cdots+z_{n1}} + \cdots + dp_{n1}e^{z_{11}+z_{21}+\cdots+z_{(n-1)1}} \leq 1$$

As long as we set $z_{11}, \cdots, z_{n1}$ all as $-\infty$, the inequality above could be established. And When $n > 1$, we can find an adversary makes $R_n^E \geq \log(d)$.
Combining two conditions, we can find an adversary that makes $R_n^E \geq \log(d)$, which means the theoretical guarantee for AA strategy is tight.

**Exercise 2**

(a)

Since both strategies are Exp strategy and using dot loss, the theoretical guarantee is $R_n^E \leq n\frac{\eta}{8} + \frac{In(d)}{\eta}$. For strategy A, the learning rate is $\eta_A = \sqrt{4\log(d)}$ and the theoretical guarantee for $n = 2$ is $R_2^E \leq \sqrt{\log(d)}$. Therefore, $C_n^A = \sqrt{\log(d)}$. Similarly, for strategy B, the learning rate is $\eta_B = \sqrt{2\log(d)}$ and the theoretical guarantee for $n = 2$ is $R_2^E \leq \frac{3}{2}\sqrt{\frac{\log(d)}{2}}$, which means

$C_n^B = \frac{3}{2}\sqrt{\frac{\log(d)}{2}}$. Clearly, $C_n^A$ is smaller than $C_n^B$. In other words, the bound of strategy A is tighter for $n = 2$.
(b)
For $n = 4$, $C_n^A$ becomes $\frac{3}{2}\sqrt{\log(d)}$ and $C_n^B$ becomes $\sqrt{2\log(d)}$. This time, $C_n^A$ is larger than $C_n^B$, which means the bound of strategy B for $n = 4$ is tighter.
(c)
No, it does not. Here is the informal argument: in this case, Exp strategy and dot loss are adopted. We can know $z_t \in [0, 1]$. Assuming, the adversaries for all experts are always set as 0. The actions will be always the same as the origin, $(\frac{1}{d}, \cdots, \frac{1}{d})$. In this case, the expert regrets for both strategy A and B will be the same. Therefore, in spite of the tighter bound of strategy B for some n, I can find a condition when both expert regrets are the same.
(d)
Since the Exp strategy and dot loss are adopted in this exercise, the actions will be

$$p_t^i = \frac{e^{-\eta \sum_{s=1}^{t-1} z_s^i}}{\sum_{j=1}^{d} e^{-\eta \sum_{s=1}^{t-1} z_s^i}}$$

$$p_t^i = \frac{e^{-\eta \sum_{s=1}^{t-1} z_s^i}}{e^{-\eta \sum_{s=1}^{t-1} z_s^1} + \cdots + e^{-\eta \sum_{s=1}^{t-1} z_s^i} + \cdots + e^{-\eta \sum_{s=1}^{t-1} z_s^d}}$$
$$p_t^i = \frac{1}{e^{-\eta \sum_{s=1}^{t-1} (z_s^1 - z_s^i)} + \cdots + 1 + \cdots + e^{-\eta \sum_{s=1}^{t-1} (z_s^d - z_s^i)}}$$

When we increase the learning rate $\eta$, the action $p_t^i$ will change which depends on the values of $\sum_{s=1}^{t-1}$. Two conditions are certain. For the expert with the smallest value of the sum of adversaries before, the denominator of the equation above will decrease and the value of action $p_t^i$ will increase when the learning rate increases. The other condition is for the expert with the largest value of the sum of adversaries before, the denominator of the equation above will increase and the value of action $p_t^i$ will decrease when the learning rate increases. As for the rest experts, the values of their actions can increase or decrease, which is not certain.
When the learning rate $\eta$ is extremely high ($\eta = \infty$), for the expert with the smallest value of the sum of adversaries before, the value of the action $p_t^i$ will become 1. Since $\sum_{i=1}^{d} p_i = 1$, the rest actions of other experts will become 0.
**Exercise 3**
(a)
According to the lecture and the exercise, a OCO regret of $R_n \leq \frac{R^2}{2\eta} + \frac{\eta G^2 n}{2}$ for any adversary. We derive that inequality w.r.t the learning rate $\eta$,

$$\frac{\mathrm{d}\eta}{\mathrm{d}R_n} = -\frac{R^2}{2\eta^2} + \frac{G^2 n}{2}$$

Let us make the equation equals to 0, which means,

$$\frac{R^2}{2\eta^2} = \frac{G^2 n}{2}$$

$$\eta^2 = \frac{R^2}{G^2 n}$$
$$\eta = \frac{R}{G\sqrt{n}}$$

Therefore, we can get the optimal learning rate of OCO is $\eta_m = \frac{R}{G\sqrt{n}}$. To verify this optimal learning rate minimize or maximize the right side of the inequality, we can substitute half of the optimal learning rate and twice of the optimal learning rate into the derivative equation above, we can get the values are negative and positive, respectively, which means the value of the right side bound decreases first and increases after the optimal learning rate. At last, we substitute the optimal learning rate $\eta_m = \frac{R}{G\sqrt{n}}$ into the right side bound, we can get $\frac{R^2}{2\eta_m} + \frac{\eta_m G^2 n}{2} = RG\sqrt{n}$. Therefore, the optimal choice that $R_n \leq RG\sqrt{n}$.

(b)

Firstly, we use the Lemma 1 from the lecture, differentiable function, $l(a) - l(b) \leq \nabla^T l(a)(a - b)$, we can get

$$l(a_t, z_t) - \min l(a, z_t) \leq \nabla^T l(a_t, z_t)(a_t - a).$$

And then sum up all the rounds, which we can get

$$\sum_{t=1}^{n} l(a_t, z_t) - \min l(a, z_t) \leq \sum_{t=1}^{n} \nabla^T l(a_t, z_t)(a_t - a).$$

The left side of the inequality above is the OCO regret $R_n$. And then based on the definition 4 from the lecture, $w_{t+1} = a_t - \eta \nabla_a l(a_t, z_t)$ and $a_{t+1} = \Pi_A(w_{t+1})$. We can get

$$a_t = \Pi_A(w_t) = \Pi_A(a_{t-1} - \eta \nabla_a l(a_{t-1}, z_{t-1})).$$

Next, we use the theorem 1 (Pythagorans) from the lecture, we can get an inequality,

$$||a_t - a||_2 = ||\Pi_A(w_t) - a||_2 \leq ||w_t - a||_2$$

$$||a_t - a||_2 = ||\Pi_A(a_{t-1} - \eta_t \nabla_a l(a_{t-1}, z_{t-1})) - a||_2 \leq ||a_{t-1} - \eta_t \nabla_a l(a_{t-1}, z_{t-1}) - a||_2$$

Substitute the equation, $t = t + 1$, into the inequality above,

$$||a_{t+1} - a||_2 = ||\Pi_A(a_t - \eta_t \nabla_a l(a_t, z_t)) - a||_2 \leq ||a_t - \eta_t \nabla_a l(a_t, z_t) - a||_2$$

$$||a_{t+1} - a||_2 \leq ||a_t - \eta_t \nabla_a l(a_t, z_t) - a||_2$$
$$(a_{t+1} - a)^2 \leq (a_t - \eta_t \nabla_a l(a_t, z_t) - a)^2$$
$$(a_{t+1} - a)^2 \leq (a_t - a)^2 - 2\eta_t \nabla_a l(a_t, z_t)(a_t - a) + (\eta_t \nabla_a l(a_t, z_t))^2$$
$$2\eta_t \nabla_a l(a_t, z_t)(a_t - a) \leq (a_t - a)^2 - (a_{t+1} - a)^2 + (\eta_t \nabla_a l(a_t, z_t))^2$$
$$\nabla_a l(a_t, z_t)(a_t - a) \leq \frac{(a_t - a)^2 - (a_{t+1} - a)^2}{2\eta_t} + \frac{\eta_t (\nabla_a l(a_t, z_t)^2}{2}$$

According to the exercise, $||\nabla_a l(a, z)|| \leq G$, the inequality above will become

$$\nabla_a l(a_t, z_t)(a_t - a) \leq \frac{(a_t - a)^2 - (a_{t+1} - a)^2}{2\eta_t} + \frac{\eta_t (\nabla_a l(a_t, z_t)^2}{2} \leq \frac{(a_t - a)^2 - (a_{t+1} - a)^2}{2\eta_t} + \frac{\eta_t G^2}{2}$$

When we sum the inequality form $t = 1$ to $n$ and combine the second inequality of this exercise, we can get

$$R_n \leq \sum_{t=1}^{n} \frac{(a_t - a)^2 - (a_{t+1} - a)^2}{2\eta_t} + \frac{\eta_t G^2}{2} = \sum_{t=1}^{n} \frac{(a_t - a)^2}{2\eta_t} - \frac{(a_{t+1} - a)^2}{2\eta_t} + \frac{\eta_t G^2}{2}$$

$$\leq \sum_{t=1}^{n} \frac{(a_t - a)^2}{2\eta_t} - \frac{(a_t - a)^2}{2\eta_{t-1}} + \frac{\eta_t G^2}{2} = \sum_{t=1}^{n} \frac{(a_t - a)^2}{2}(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}}) + \frac{\eta_t G^2}{2}$$

Because of the assumption that $\max_{a \in A} ||a|| \leq R$, the inequality becomes

$$R_n \leq \sum_{t=1}^{n} \frac{R^2}{2}(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}}) + \frac{\eta_t G^2}{2} = \frac{R^2}{2}(\frac{1}{\eta_t} - \frac{1}{\eta_0}) + \sum_{t=1}^{n} \frac{\eta_t G^2}{2}.$$

Since $\frac{1}{\eta_0} = 0$ and $\eta_t = \frac{R}{G\sqrt{t}}$, we can get

$$R_n \leq \frac{R^2}{2}\frac{1}{\eta_t} + \sum_{t=1}^{n} \frac{\eta_t G^2}{2} = \frac{RG\sqrt{n}}{2} + \sum_{t=1}^{n} \frac{RG}{\sqrt{t}}$$

According to the hints, $\sum_{t=1}^{n} \frac{1}{\sqrt{t}} \leq 2\sqrt{n}$, we finally get

$$R_n \leq \frac{RG\sqrt{t}}{2} + \sum_{t=1}^{n} \frac{RG}{\sqrt{t}} \leq \frac{RG\sqrt{n}}{2} + RG\sqrt{n} = \frac{3}{2}RG\sqrt{n}.$$

**Exercise 4**

(a)

Since $W_t$ is the total wealth, $W_t * p_t^i$ represents on time $t$ the value of the amount of asset $i$. The amount of asset $i$ that we buy/sell can be represented as $\frac{W_t * p_t^i}{x_t^i}$. Therefore, on time $t+1$, we can buy/sell asset $i$ will be $\frac{W_t * p_t^i}{x_t^i} * x_{t+1}^i$. We can get the relation that

$$\sum_{i=0}^{d} p_t^i \frac{x_{t+1}^i}{x_t^i} = \frac{W_{t+1}}{W_t}.$$

If we use the ratio $r_t^i$, we can get

$$\sum_{i=0}^{d} p_t^i r_t^i = \frac{W_{t+1}}{W_t}.$$

(b)

When the adversary moves $z_t^i = -\log(r_t^i)$, we can get

$$\sum_{t=1}^{n} -\log(\sum_{i=1}^{d} p_t^i \exp^{-z_t^i}) = \sum_{t=1}^{n} -\log(\sum_{i=1}^{d} p_t^i r_t^i).$$

Use the conclusion of 4a, we can get

$$\sum_{t=1}^{n} -\log(\sum_{i=1}^{d} p_t^i r_t^i) = \sum_{t=1}^{n} -\log(\frac{W_{t+1}}{W_t}) = -\log(\prod_{t=1}^{n} \frac{W_{t+1}}{W_t}) = -\log(W_n/W_1).$$

Therefore, we can get $\sum_{t=1}^{n} -\log(\sum_{i=1}^{d} p_t^i \exp^{-z_t^i}) = -\log(W_n/W_1)$.
If we set the adversary moves as such, the mix loss will stand for the ratio of the wealth after $n$ days and the original wealth, which reflects how much wealth that we earn or loss after $n$ days' investment. Therefore, the mix loss is appropriate for this setting. And we want that we earn wealth instead of losing some, which means $\frac{W_n}{W_1}$ is larger the better. If we want $\frac{W_n}{W_1}$ to be as large as possible, then we want $-\log(W_n/W_1)$ as small as possible. Therefore, we want the total mix loss to be as small as possible.
(c)
2.
In terms of adversary moves $z_t^i = -\log(r_t^i)$ and the conclusion of (a) and (b), we can get the adversary moves $z_t^i = -\log(r_t^i) = -\log(x_{t+1}^i/x_t^i)$ and the total loss of each expert is $\sum_{t=1}^{n} -\log(p_t^i \exp^{-z_t^i}) = \sum_{t=1}^{n} -\log(p_t^i r_t^i)$.
3.
Based on AA, the expert regret of AA using 213 days is 0.2232. The code of AA.m will be in the appendix.
4.
The total loss of AA is -1.4311, and the total loss of the experts is -1.6543. The total loss of AA is larger than the one of the experts. The expert regret is smaller than the theoretical guarantee $\ln 5$. No, the adversary is not generating 'difficult' data.
5.
The plot of $p_t$ vs. $t$ and $x_t$ vs. $t$ is shown below, as Figure 1.
From Figure 1, we can get the strategy AA affects **BTC**, **BCH** and **XRP** intensively. And

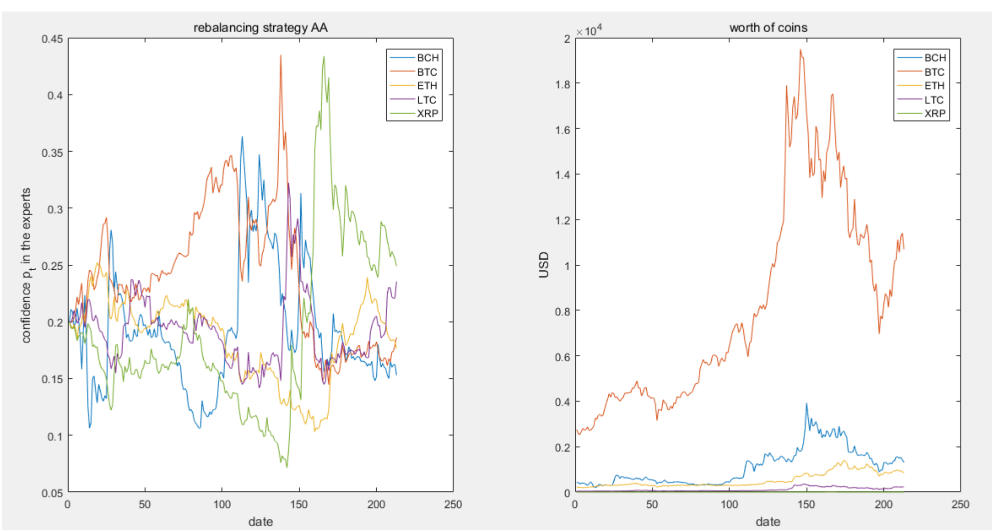


Figure 1: Plot of strategy of AA

the confidence of these coins vary strongly. Combining the worth of coins (right figure of

Figure 1), we can get when the price of coins increase, the confidence $p_t$ increases. We can also get the trend of wealth of these coins have similar trend of the confidence of them, which makes sense. When the wealth of the coin rise up, the AA strategy forces the confidence of this coin to increase.
6.
The wealth has increased $1.6969 \times 10^3$ USD, if we use AA strategy.
(d)
1.
The OCO regret can be computed by $R_n = \sum_{t=1}^n l_m(a_t, z_t) - \min_{a \in A} \sum_{t=1}^n l_m(a, z_t)$. And the expert regret can be computed by $R_n^E = \sum_{t=1}^n l_m(p_t, z_t) - \min_i \sum_{t=1}^n l_m(e_i, z_t)$. If the adversary moves are the same for both regrets, only difference will be $p_t$ and $a_t$. The $p_t$ of expert regret above was determined by AA strategy. The $a_t$ of OCO regret here will be determined by online gradient descent strategy.
2.
The mix loss of this online convex optimization setting is given as $l_m(a_t, z_t) = -\log(a_t^T r_t)$. We derive the mix loss function w.r.t a, and we will get

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

$$= -\frac{1}{ar_t}\frac{\mathrm{d}ar_t}{\mathrm{da}} = -\frac{1}{ar_t}r_t = -\frac{r_t}{ar_t}$$

3.
The code of mix_loss.m will be presented in the appendix.
4.
The code of OGD.m will be presented in the appendix. One thing needs to be mentioned here is how to determine the best fixed action. What I implemented in OGD.m is firstly using the actions from matrix A and then randomly generated actions out of matrix A, which satisfies that the sum of the actions is 1 and each action $a \in [0, 1]$.
5.
Because $a_t^i \in [0, 1]$ and $\max_{a \in A} ||a|| \leq R$, the largest Euclidean distance of a is 1, which makes $R = 1$. As for G, we have the inequality as $||\nabla_a l(a, z)|| \leq G$ and in this exercise it becomes $|| - \frac{r_t}{ar_t}|| \leq G$. G is upper bound of $|| - \frac{r_t}{ar_t}||$, which can be calculated

$$||\nabla_a l(a, z)|| \leq \frac{\max\sqrt{\sum r_t^2}}{a_t^T r_t} \leq \frac{\max\sqrt{\sum r_t^2}}{\min r_t},$$

which is consistent with G in the code.
6.
The plot of strategy of OGD is shown below, as Figure 2. The wealth has increased $1.8856 \times 10^3$ USD, if we use OGD to invest.
7.
The optimal 'fixed' action is $[0.2375; 0.0034; 0.0082; 0.2843; 0.4665]$, and the regret is 0.1710.
8.
The total loss of OGD is -1.8153 and the loss of the best fixed action is -1.9862. The loss of OGD is larger than the loss of the best fixed action. And the regret of OGD is smaller than

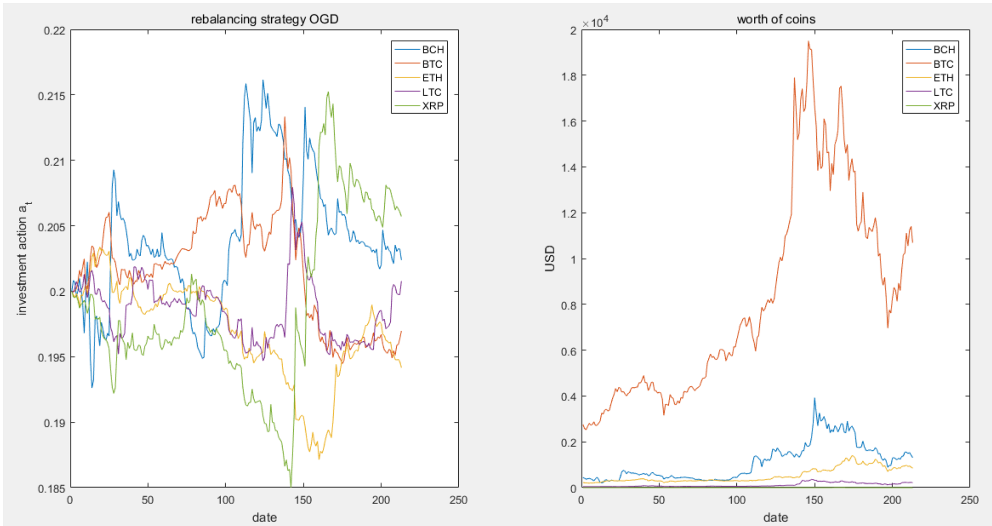


Figure 2: Plot of strategy of OGD

the guaranteed OCO regret. No, the adversary is not generating 'difficult' data.
9.
The assumption of $||\nabla l(a,z)||_2 \leq G$ will be violated when this happens. The upper bound from the OCO regret will become $\frac{R^2}{2\eta}$.

**Appendix**:
Code of AA.m

```
% Exercise: Aggregating Algorithm (AA)

clear all;
load coin_data;

d_expert = 5;
n = 213;

% compute adversary movez z_t
z = -log(r);

% compute strategy p_t (see slides)
p = zeros(size(s));
p(1,:) = [0.2, 0.2, 0.2, 0.2, 0.2];
p(2,:) = exp(-z(1,:))./sum(exp(-z(1,:)));
for i = 3:n
    numerator = exp(-sum(z(1:i-1,:)));
    denominator = sum(exp(-sum(z(1:i-1,:))));
    p(i,:) = numerator./denominator;
end

% compute loss of strategy p_t
tmp = p.*exp(-z);
tmp = sum(tmp,2);
for i = 1:n
    loss(i,:) = -log(tmp(i,:));
end
loss_p = sum(loss);

% compute losses of experts
loss = sum(z);
loss_e = min(loss);

% compute regret
R = loss_p-loss_e;

% compute total gain of investing with strategy p_t
gain = sum(p(n,:).*s(n,:)-p(1,:).*s(1,:));

%% plot of the strategy p and the coin data
```

```
% if you store the strategy in the matrix p (size n * d)
% this piece of code will visualize your strategy

figure
subplot(1,2,1);
plot(p)
legend(symbols_str)
title('rebalancing strategy AA')
xlabel('date')
ylabel('confidence p_t in the experts')

subplot(1,2,2);
plot(s)
legend(symbols_str)
title('worth of coins')
xlabel('date')
ylabel('USD')
```

Code of mix_loss.m

```
function [l, g] = mix_loss(a, r)
%   [l, g] = MIX_LOSS(a, r)
%   Input:
%         a (column vector), the investment strategy (note a should be normaliz
%         r (column vector), stock changes on day t (compared with t-1)
%   Output:
%         l (number), the mix loss
%         g (column vector), the gradient of the mix loss (with respect to acti
l = -log(a'*r);
g = -r./(a'*r);
end
```

Code of OGD.m

```
% Exercise: Online Gradient Descent (OGD)

clear all;
load coin_data;

a_init = [0.2, 0.2, 0.2, 0.2, 0.2]'; % initial action

n = 213; % is the number of days
d = 5; % number of coins

% we provide you with values R and G.
alpha = sqrt(max(sum(r.^2,2)));
epsilon = min(min(r));
```

```
G = alpha/epsilon;
R = 1;

% set eta:
eta = R/(G*sqrt(n));

a = a_init; % initialize action. a is always a column vector

L = nan(n,1); % keep track of all incurred losses
A = nan(d,n); % keep track of all our previous actions

for t = 1:n

    % we play action a
    [l,g] = mix_loss(a,r(t,:)'); % incur loss l, compute gradient g

    A(:,t) = a; % store played action
    L(t) = l; % store incurred loss

    % update our action, make sure it is a column vector
    a = a-eta.*g;

    % after the update, the action might not be anymore in the action
    % set A (for example, we may not have sum(a) = 1 anymore).
    % therefore we should always project action back to the action set:
    a = project_to_simplex(a')'; % project back (a = \Pi_A(w) from lecture)

end

% compute total loss
loss = sum(L);

% compute total gain in wealth
gain = s(n,:)*A(:,n)-s(1,:)*A(:,1);

% compute best fixed strategy (you may make use of loss_fixed_action.m and op
loss_fixed = zeros(size(L));
% looking for the best fixed action in matrix A
for i = 1:n
    [loss_fixed(i),gradient] = loss_fixed_action(A(:,i));
end
% looking for the best fixed action out of matrix A
y = zeros(d,15000);
y(:,1:n) = A;
for i = n+1:15000
```

```
    x = rand(d,1);
    y(:,i) = x./sum(x);
    [loss_fixed(i),gradient] = loss_fixed_action(y(:,i));
end
loss_f = min(loss_fixed);
[bestx,best_y] = find(loss_fixed == min(loss_fixed));
best = y(:,bestx);

% compute regret
R = loss-loss_f;

%% plot of the strategy A and the coin data

% if you store the strategy in the matrix A (size d * n)
% this piece of code will visualize your strategy

figure
subplot(1,2,1);
plot(A')
legend(symbols_str)
title('rebalancing strategy OGD')
xlabel('date')
ylabel('investment action a_t')

subplot(1,2,2);
plot(s)
legend(symbols_str)
title('worth of coins')
xlabel('date')
ylabel('USD')
```