IN4320 Machine Learning: Assignment 3
Online Learning

Dimitropoulos Georgios: 4727657

March 27, 2018

Exercise 1

(a)

Strategy A

$\boxed{t = 1}$

$\boxed{p_1 = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})}$

$\boxed{t = 2}$

$p_2 = e_{b(2)}$
$b(2) = argmin_i L_1^i$

- Expert 1: $L_1^1 = \sum_{s=1}^{1} z_s^1 = z_1^1 = \boxed{0}$
- Expert 2: $L_1^2 = \sum_{s=1}^{1} z_s^2 = z_1^2 = 0.1$
- Expert 3: $L_1^3 = \sum_{s=1}^{1} z_s^3 = z_1^3 = 0.2$

Thus, $\boxed{p_2 = e_1 = (1, 0, 0)}$

$\boxed{t = 3}$

$p_3 = e_{b(3)}$
$b(3) = argmin_i L_1^i$

- Expert 1: $L_2^1 = \sum_{s=1}^{2} z_s^1 = z_1^1 + z_2^1 = 0 + 0 = \boxed{0}$
- Expert 2: $L_2^2 = \sum_{s=1}^{2} z_s^2 = z_1^2 + z_2^2 = 0 + 0.1 = 0.1$
- Expert 3: $L_2^3 = \sum_{s=1}^{2} z_s^3 = z_1^3 + z_2^3 = 0.2 + 0.1 = 0.3$

Thus, $\boxed{p_3 = e_1 = (1, 0, 0)}$

$\boxed{t = 4}$

$p_4 = e_{b(4)}$
$b(4) = argmin_i L_1^i$

- Expert 1: $L_3^1 = \sum_{s=1}^{3} z_s^1 = z_1^1 + z_2^1 + z_3^1 = 0 + 0 + 1 = 1$
- Expert 2: $L_3^2 = \sum_{s=1}^{3} z_s^2 = z_1^2 + z_2^2 + z_3^2 = 0 + 0 + 0.1 = \boxed{0.1}$
- Expert 3: $L_3^3 = \sum_{s=1}^{3} z_s^3 = z_1^3 + z_2^3 + z_3^3 = 0.2 + 0.1 + 0 = 0.3$

Thus, $\boxed{p_4 = e_2 = (0, 1, 0)}$

Strategy B

$\boxed{t = 1}$

$p_t^i = \frac{1}{d} \leftrightarrow p_1^1 = p_1^2 = p_1^3 = \frac{1}{3} \leftrightarrow \boxed{p_1 = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})}$

$\boxed{t = 2}$

$c_1 = \sum_{j=1}^{3} = e^{-l_1^j} = e^{-l_1^1} + e^{-l_1^2} + e^{-l_1^3} = e^0 + e^{-0.1} + e^{-0.2} = 2.72$

$p_2^1 = \frac{e^{-l_1^1}}{c_1} = \frac{e^0}{2.72} = \boxed{0.36}$

$p_2^2 = \frac{e^{-l_1^2}}{c_1} = \frac{e^{-0.7}}{2.72} = \boxed{0.34}$

$p_2^3 = \frac{e^{-l_1^3}}{c_1} = \frac{e^{-0.2}}{2.72} = \boxed{0.30}$

$\leftrightarrow \boxed{p_2 = (0.36, 0.34, 0.3)}$

$\boxed{t = 3}$

$c_2 = \sum_{j=1}^{3} = e^{-l_2^j} = e^{-l_2^1} + e^{-l_2^2} + e^{-l_2^3} = 1 + e^{-0.1} + e^{-0.3} = 2.64$

$p_3^1 = \frac{e^{-l_2^1}}{c_2} = \frac{e^0}{2.64} = \boxed{0.38}$

$p_3^2 = \frac{e^{-l_2^2}}{c_2} = \frac{e^{-0.1}}{2.64} = \boxed{0.34}$

$p_3^3 = \frac{e^{-l_2^3}}{c_2} = \frac{e^{-0.3}}{2.64} = \boxed{0.28}$

$\leftrightarrow \boxed{p_3 = (0.38, 0.34, 0.28)}$

$\boxed{t = 4}$

$c_3 = \sum_{j=1}^{3} = e^{-l_3^j} = e^{-l_3^1} + e^{-l_3^2} + e^{-l_3^3} = e^{-1} + e^{-0.1} + e^{-0.3} = 2.01$

$p_4^1 = \frac{e^{-l_3^1}}{c_3} = \frac{e^{-1}}{2.01} = \boxed{0.18}$

$p_4^2 = \frac{e^{-l_3^2}}{c_3} = \frac{e^{-0.1}}{2.01} = \boxed{0.45}$

$p_4^3 = \frac{e^{-l_3^3}}{c_3} = \frac{e^{-0.3}}{2.01} = \boxed{0.37}$

$\leftrightarrow \boxed{p_4 = (0.18, 0.45, 0.37)}$

(b)

Total Mix Loss

We have that: $l_m(p_t, z_t) = -log(\sum_{i=1}^{d} p_t^i e^{-z_t^i})$

Strategy A

$\boxed{t = 1}$

$l_m(p_1, z_1) = -log(p_1^1 e^{-z_1^1} + p_1^2 e^{-z_1^2} + p_1^3 e^{-z_1^3}) = -\log(\frac{1}{3}e^0 + \frac{1}{3}e^{-0.1} + \frac{1}{3}e^{-0.2}) = \boxed{0.1}$

$\boxed{t = 2}$

$l_m(p_2, z_2) = -log(p_2^1 e^{-z_2^1} + p_2^2 e^{-z_2^2} + p_2^3 e^{-z_2^3}) = -\log(e^0) = \boxed{0}$

$\boxed{t = 3}$

$l_m(p_3, z_3) = -log(p_3^1 e^{-z_3^1} + p_3^2 e^{-z_3^2} + p_3^3 e^{-z_3^3}) = -\log(e^{-1}) = \boxed{1}$

$\boxed{t = 4}$

$l_m(p_4, z_4) = -log(p_4^1 e^{-z_4^1} + p_4^2 e^{-z_4^2} + p_4^3 e^{-z_4^3}) = -\log(e^{-0.9}) = \boxed{0.9}$

Hence, we have that for the strategy A:

$$\boxed{TotalMixLoss = \sum_{t=1}^{4} l_m(p_t, z_t) = 0.1 + 0 + 1 + 0.9 = 2}$$

Strategy B

$\boxed{t = 1}$

$l_m(p_1, z_1) = -log(p_1^1 e^{-z_1^1} + p_1^2 e^{-z_1^2} + p_1^3 e^{-z_1^3}) = -\log(\frac{1}{3}e^0 + \frac{1}{3}e^{-0.1} + \frac{1}{3}e^{-0.2}) = \boxed{0.1}$

$\boxed{t = 2}$

$l_m(p_2, z_2) = -log(p_2^1 e^{-z_2^1} + p_2^2 e^{-z_2^2} + p_2^3 e^{-z_2^3}) = -\log(0.36e^0 + 0.34e^0 + 0.3e^{-0.1}) = -\log(0.7 + 0.27) = \boxed{0.03}$

$\boxed{t = 3}$

$l_m(p_3, z_3) = -log(p_3^1 e^{-z_3^1} + p_3^2 e^{-z_3^2} + p_3^3 e^{-z_3^3}) = -\log(0.38e^{-1} + 0.34e^0 + 0.28e^0) = -\log(0.14 + 0.62) = \boxed{0.27}$

$\boxed{t = 4}$

$l_m(p_4, z_4) = -log(p_4^1 e^{-z_4^1} + p_4^2 e^{-z_4^2} + p_4^3 e^{-z_4^3}) = -\log(0.18e^0 + 0.45e^{-0.9} + 0.37e^0) = -\log(0.55 + 0.18) = \boxed{0.31}$

Hence, we have that for the strategy B:

$$\boxed{TotalMixLoss = \sum_{t=1}^{4} l_m(p_t, z_t) = 0.1 + 0.03 + 0.27 + 0.31 = 0.71}$$

Expert Regret

We have that: $R_n^E = \sum_{t=1}^{n} l_m(p_t, z_t) - min_i \sum_{t=1}^{n} z_t^i$
Also, for the experts we have:

- Expert 1: $\sum_{t=1}^{4} z_t^1 = z_1^1 + z_2^1 + z_3^1 + z_4^1 = 0 + 0 + 0 + 1 = 1$
- Expert 2: $\sum_{t=1}^{4} z_t^2 = z_1^2 + z_2^2 + z_3^2 + z_4^2 = 0.1 + 0 + 0 + 0.9 = 1$
- Expert 3: $\sum_{t=1}^{4} z_t^3 = z_1^3 + z_2^3 + z_3^3 + z_4^3 = 0.2 + 0.1 + 0 + 0 = \boxed{0.3}$

Thus, $min_i \sum_{t=1}^{n} z_t^i = \boxed{0.3}$

Strategy A

Hence, $\boxed{R_4^A = 2 - min_i \sum_{t=1}^{4} z_t^i = 2 - 0.3 = 1.7}$

Strategy B

And, $\boxed{R_4^B = 0.71 - min_i \sum_{t=1}^{4} z_t^i = 0.71 - 0.3 = 0.41}$

(c)

We know that for AA we have the theoretical guarantee $R_n^E \leq log(d)$
We also know that the expert regret is given by:
$R_4^E = \sum_{t=1}^4 l_m(p_t, z_t) - min_i \sum_{t=1}^4 z_t^i \leftrightarrow \sum_{t=1}^4 l_m(p_t, z_t) = R_4^E + min_i \sum_{t=1}^4 z_t^i \leftrightarrow$
$\sum_{t=1}^4 l_m(p_t, z_t) \leq log(d) + min_i \sum_{t=1}^4 z_t^i$
Thus, $C = log(d) + min_i \sum_{t=1}^4 z_t^i$
<u>Computation:</u>
$d = 3$
$log(3) = 1.09$
$min_i \sum_{t=1}^4 z_t^i = 0.3$
$Thus, \sum_{t=1}^4 l_m(p_t, z_t) \leq 1.09 + 0.3 = 1.39$
So, we have that for this example that:

$$\boxed{\sum_{t=1}^{4} l_m(p_t, z_t) \leq 1.39}$$

We can conclude that, the value of mix loss of Strategy A (2) is much higher than the boundary provided by the theoretical guarantee of AA (1.39). In contrast, the value of mix loss of Strategy B (0.71) is much less than the boundary provided by the theoretical guarantee of AA (1.39).

(d)

Initially we look on the $i_{min} = argmin_i p_t^i$. For this index i_{min} we know that $p_i^{min} \leq \frac{1}{d} \leftrightarrow log(p_i^{min}) \leq log(\frac{1}{d}) = log1 - log(d) = -log(d)$. Thus $\boxed{-log(p_i^{min}) \geq logd}$

So, the adversary could choose $z_t^i = 0$, if $i = i_{min}$ and $z_t^i = \infty$ if $i \neq i_{min}$. Thus, in other words, the adversary chooses ∞ for all experts and 0 for the expert with the lowest probability e_i, for each time t.So we have for a time t: $l_m(p_t, z_t) = -log(\sum_{i=1}^{d} p_t^i e^{-z_t^i}) = -log(p_t^1 e^{-\infty} + ... + p_t^{min} e^0 + ... + p_t^d e^{-\infty}) = -log(p_t^{min}) \geq log(d)$

Also we have that $min_i z_t^i = 0$. Thus, since the expert regret is given by the formula $R_n^E = \sum_{t=1}^{n} l_m(p_t, z_t) - min_i \sum_{t=1}^{n} z_t^i$, we have that $R_n^E \geq log(d)$.

In case that we have equal probabilities on a time for all experts, the adversary chooses ∞ for all experts and we reach the same conclusion.

Exercise 2

(a)

We have the theoretical guarantee: $R_n^E \leq n\frac{\eta}{8} + \frac{\log(d)}{\eta}$

$\boxed{n = 2}$

- Strategy A with $\eta_A = \sqrt{4\log(d)}$
  $R_2^A \leq n\frac{\eta_A}{8} + \frac{\log(d)}{\eta_A} = 2\frac{\sqrt{4\log(d)}}{8} + \frac{\log(d)}{\sqrt{4\log(d)}} = \frac{log(d)}{\sqrt{4\log(d)}} + \frac{log(d)}{\sqrt{4\log(d)}} = \frac{2log(d)}{2\sqrt{log(d)}} = \sqrt{\log(d)}$
  $\Rightarrow \boxed{R_2^A \leq \sqrt{\log(d)}}$ and $\boxed{C_n^A = \sqrt{\log(d)}}$

- Strategy B with $\eta_B = \sqrt{2\log(d)}$
  $R_2^B \leq n\frac{\eta_B}{8} + \frac{\log(d)}{\eta_B} = \frac{2\sqrt{2\log(d)}}{8} + \frac{\log(d)}{\sqrt{2\log(d)}} = \frac{3log(d)}{2\sqrt{2\log(d)}} = \frac{3}{4}\sqrt{2\log(d)}$
  $\Rightarrow \boxed{R_2^B \leq \frac{3}{4}\sqrt{2\log(d)}}$ and $\boxed{C_n^B = \frac{3}{4}\sqrt{2\log(d)}}$

We can notice here that $\frac{3}{4}\sqrt{2} > 1$, and we can conclude that $C_n^B > C_n^A$
Hence, the bound C_n^A is tighter.

(b)

We also have here the theoretical guarantee: $R_n^E \leq n\frac{\eta}{8} + \frac{\log(d)}{\eta}$

$\boxed{n = 4}$

- Strategy A with $\eta_A = \sqrt{4\log(d)}$
  $R_2^A \leq n\frac{\eta_A}{8} + \frac{\log(d)}{\eta_A} = 4\frac{\sqrt{4\log(d)}}{8} + \frac{\log(d)}{\sqrt{4\log(d)}} = \frac{24logd}{8\sqrt{4logd}} = \frac{3logd}{2\sqrt{logd}}$
  $\Rightarrow \boxed{R_4^A \leq \frac{3}{2}\sqrt{\log(d)}}$ and $\boxed{C_n^A = \frac{3}{2}\sqrt{\log(d)}}$

- Strategy B with $\eta_B = \sqrt{2\log(d)}$
  $R_2^B \leq n\frac{\eta_B}{8} + \frac{\log(d)}{\eta_B} = 4\frac{\sqrt{2\log(d)}}{8} + \frac{\log(d)}{\sqrt{2\log(d)}} = \frac{16logd}{8\sqrt{2logd}} = \frac{2logd}{2\sqrt{2logd}} = \sqrt{2\log(d)}$
  $\boxed{\Rightarrow R_4^B \leq \sqrt{2\log(d)}}$ and $\boxed{C_n^B = \sqrt{2\log(d)}}$

We can notice here that $\frac{3}{2} > \sqrt{2}$, and we can conclude that $C_n^A > C_n^B$
Hence, the bound C_n^B is tighter.

(c)

We assume here that strategy B has a tighter bound for some n, meaning that $C_n^A > C_n^B$
In the previous question we found that $R_n^A \leq C_n^A$ and $R_n^B \leq C_n^B$
We are going to use a counterexample in order to prove our consideration. Let for $n = k$, $C_k^A > C_k^B$ and assume that $C_k^B = 5$ and $C_k^A = 10$ which of course fulfill the inequality $C_k^A > C_k^B$. Then we have that $R_k^A \leq 10$ and $R_k^B \leq 5$. Let also assume that $R_k^B = 3$ and $R_k^A = 2$ then we would have that $R_k^B \leq C_k^B(3 < 5)$ and $R_k^A \leq C_k^A(2 < 10)$. However, $R_k^B > R_k^A(3 > 2)$. So, if we have that $C_n^A > C_n^B$, then this does not necessarily imply that $R_n^A > R_n^B$ for any adversary moves z_t for $t = 1, ..., n$.

(d)

We have that the exp strategy is given by the formula: $p_t^i = \frac{e^{-\eta L_{t-1}^i}}{\sum_{j=1}^d e^{-\eta L_{t-1}^j}} = \frac{e^{-\eta L_{t-1}^i}}{e^{-\eta L_{t-1}^1} + e^{-\eta L_{t-1}^2} + ... + e^{-\eta L_{t-1}^i} + ... e^{-\eta L_{t-1}^d}}$
Looking at this equation we can distinguish 2 cases. Namely we have:
<u>Case 1</u>
If $L_{t-1}^i = L_{t-1}^1 = L_{t-1}^2 = ...L_{t-1}^d = c$ then the above formula can be simplified as: $p_t^i = \frac{a}{ad} = \frac{1}{d}(constant)$. Thus we can conclude in this case that the strategy p_t^i is independent of the learning rate η. The same also holds for the case that $L_{t-1}^1 = L_{t-1}^2 = ...L_{t-1}^d = 0$ and $L_{t-1}^i \neq 0$. Again here the strategy p_t^i is independent of the learning rate η.
<u>Case 2</u> In any other case, the denominator would be greater than the numerator, so it dominates. Thus, if we increase the learning η, then the strategy p_t^i is going to be increased. In the extreme case that we set the the learning η rate high (∞) then strategy p_t^i is going to go to unity by taking the limit $\lim_{\eta \to \infty} p_t^i$. So, we could say that as $\lim_{\eta \to \infty}$ we converge to the following the leader technique. In contrast, the lower η, we resemble a uniform distribution (more conservative).

Exercise 3

(a)

Firstly, we are going to derive that the optimal learning rate is given by $\eta_n = \frac{R}{G\sqrt{n}}$. This means that this η_n minimizes the right hand side of the OCO regret bound, namely the part $\frac{R^2}{2\eta} + \frac{\eta G^2 n}{2}$.
In order to prove this we define the function $f(\eta) = \frac{R^2}{2\eta} + \frac{\eta G^2 n}{2}, \eta \neq 0$.
We are going to take the derivative of this function with respect to η and we will show that derivative equals to zero for this value of η and that the second order derivative is positive.

$\frac{\partial f(\eta)}{\partial \eta} = 0 \Leftrightarrow \frac{R^2}{2}(\frac{-1}{\eta^2}) + \frac{G^2 n}{2} = 0 \Leftrightarrow \frac{R^2}{2\eta^2} = \frac{G^2 \eta}{2} \Leftrightarrow \eta^2 = \frac{R^2}{G^2 n} \Leftrightarrow \eta = \frac{R}{G\sqrt{n}}$

Also we have that $\frac{\partial^2 f(\eta)}{\partial \eta^2} = \frac{R^2}{2\eta^4} > 0$

Hence the optimal learning rate is given by $\boxed{\eta = \frac{R}{G\sqrt{n}}}$

It has been shown in the lecture that in the online convex optimization (OCO) setting we have a OCO regret of $R_n \leq \frac{R^2}{2\eta} + \frac{\eta G^2 n}{2}$ for any adversary.
Hence, for $\eta_n = \frac{R}{G\sqrt{n}}$ we have:
$R_n \leq \frac{\frac{R^2}{1}}{\frac{2R}{G\sqrt{n}}} + \frac{RG^2 n}{G\sqrt{n}2} = \frac{R^2 G\sqrt{n}}{2R} + \frac{RG^2 n}{G\sqrt{n}2} = \frac{RG^2 n}{2G\sqrt{n}} + \frac{RG^2 n}{2G\sqrt{n}} = \frac{2RG^2 n}{2G\sqrt{n}} = \frac{GRn}{\sqrt{n}} = \frac{GRn\sqrt{n}}{n} = RG\sqrt{n}$ ■
Hence, for this optimal choice of the learning rate $\eta_n = \frac{R}{G\sqrt{n}}$ we can thus bound $\boxed{R_n \leq RG\sqrt{n}}$

(b)

Let $a \in A$ be an arbitrary action. We are going to show that the OCO regret bound holds for this a. If indeed the bound holds for an arbitrary a, then it will also hold for the minimizing value of a as it defined in the OCO regret. Thus this would be enough. First we use a *Lemma* that says: If $l : R^d \to R$ is a convex and differentiable function then the following inequality holds:

$$l(a_t, z_t) - l(a, z_t) \leq \nabla a_t^T l(a_t, z_t)(a_t - a) \tag{1}$$

Let $\nabla a_t^T l(a_t, z_t) = g_t$ for convenience. Thus we can written (1) as:

$$l(a_t, z_t) - l(a, z_t) \leq g_t(a_t - a) \tag{2}$$

We have that (2) holds for all t, thus we can sum this over n rounds:

$$R_n^a := \sum_{t=1}^{n} l(a_t, z_t) - l(a, z_t) \leq \sum_{t=1}^{n} g_t(a_t - a) \tag{3}$$

We can notice that the left hand side of (3) is the OCO regret, except that we compare to a and not to the best a. So, in order to be able to reach our goal, we are going to bound the right hand side of (3). We know from the following $Theorem(Pythagoras)$ that if $A \subset R^d$ is a closed and convex set, then for all points $a \in A$ and $w \in R^d$ the following inequality holds:

$$||\Pi_A(w) - a||_2 \leq ||w - a||_2 \tag{4}$$

Also, based on the strategy given by the Online Gradient Descent with learning rate η we have that:

$$a_{t+1} = \Pi_A(a_t - \eta_t g_t) \tag{5}$$

Hence by combining (4),(5) we can derive that:

$$||a_{t+1} - a||^2 = ||\Pi_A(a_t - \eta_t g_t) - a||^2 \leq ||a_t - \eta_t g_t - a||^2 \tag{6}$$

After that we expand the squared norm in (6). So:

$||a_{t+1} - a||^2 = ||\Pi_A(a_t - \eta_t g_t) - a||^2 \leq ||a_t - \eta_t g_t - a||^2 = ||a_t - a||^2 - 2\eta_t g_t(a_t - a) + ||\eta_t g_t||^2$

We can then rewritten this as follows:

$2\eta_t g_t(a_t - a) \leq ||a_t - a||^2 - ||a_{t+1} - a||^2 + ||\eta_t g_t||^2$

After that we can sum over n turns and we can divide with $2\eta_t$ and we will get:

$$\sum_{t=1}^{n} g_t(a_t - a) \leq \sum_{t=1}^{n} \frac{1}{2\eta_t}||a_t - a||^2 - \frac{1}{2\eta_t}||a_{t+1} - a||^2 + \frac{\eta_t}{2}||g_t||^2 \tag{7}$$

Thus, by inspecting (3),(7) it suffices to show that:

$$\sum_{t=1}^{n} \frac{1}{2\eta_t}||a_t - a||^2 - \frac{1}{2\eta_t}||a_{t+1} - a||^2 + \frac{\eta_t}{2}||g_t||^2 = \frac{3}{2}GR\sqrt{n} \tag{8}$$

Thus, lets focus on the left hand side of (8). We have that:

$$\sum_{t=1}^{n} \tfrac{1}{2\eta_t}||a_t - a||^2 - \tfrac{1}{2\eta_t}||a_{t+1} - a||^2 + \tfrac{\eta_t}{2}||g_t||^2 = \sum_{t=1}^{n} \tfrac{1}{2\eta_t}(||a_t - a||^2 - ||a_{t+1} - a||^2) + \sum_{t=1}^{n} \tfrac{\eta_t}{2}||g_t||^2 =$$

$$\sum_{t=1}^{n} \frac{1}{2\eta_t}(||a_t - a||^2 - ||a_{t+1} - a||^2) + \frac{G^2}{2}\sum_{t=1}^{n} \eta_t \tag{9}$$

In (9) we used the fact that $||\nabla_a l(a, z)|| \leq G$. Lets focus now on the second term of (9), namely on $\frac{G^2}{2}\sum_{t=1}^{n} \eta_t$. We have that:

$$\frac{G^2}{2}\sum_{t=1}^{n} \eta_t = \frac{G^2}{2}\sum_{t=1}^{n} \frac{R}{G}\frac{1}{\sqrt{t}} \leq \frac{GR}{2}2\sqrt{n} = GR\sqrt{n} \tag{10}$$

In (10) we used that $\eta_t = \frac{R}{G\sqrt{t}}$ and the inequality $\sum_{t=1}^{n} \frac{1}{\sqrt{t}} \leq 2\sqrt{n}$

Hence, (9) can be written as:

$$\sum_{t=1}^{n} \frac{1}{2\eta_t}(||a_t - a||^2 - ||a_{t+1} - a||^2) + GR\sqrt{n} \tag{11}$$

Lets focus now on the first term of (11), namely on $\sum_{t=1}^{n} \frac{1}{2\eta_t}(||a_t - a||^2 - ||a_{t+1} - a||^2)$

We are going to write out the sum for the first few terms of t now:

t=1: $\frac{1}{2\eta_1}(||a_1 - a||^2 - ||a_2 - a||^2)+$
t=2: $\frac{1}{2\eta_2}(||a_2 - a||^2 - ||a_3 - a||^2)+$
...
t=n-1: $\frac{1}{2\eta_{n-1}}(||a_{n-1} - a||^2 - ||a_n - a||^2)+$
t=n: $\frac{1}{2\eta_n}(||a_n - a||^2 - ||a_{n+1} - a||^2)$

We can re-order them now as:

$\frac{1}{2\eta_1}||a_1 - a||^2 - \frac{1}{2\eta_n}||a_{n+1} - a||^2 \; -\frac{1}{2\eta_1}||a_2 - a||^2 + \frac{1}{2\eta_2}||a_2 - a||^2 \; -\frac{1}{2\eta_2}||a_3 - a||^2 + \frac{1}{2\eta_3}||a_3 - a||^2 + ...+$
$-\frac{1}{2\eta_{n-1}}||a_n - a||^2 + \frac{1}{2\eta_n}||a_n - a||^2 = \frac{1}{2\eta_1}||a_1 - a||^2 - \frac{1}{2\eta_n}||a_{n+1} - a||^2 + \frac{1}{2}\sum_{t=2}^{n}(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}})||a_t - a||^2$

Hence, (11) can be re-written as:

$$\frac{1}{2\eta_1}||a_1 - a||^2 - \frac{1}{2\eta_n}||a_{n+1} - a||^2 + \frac{1}{2}\sum_{t=2}^{n}(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}})||a_t - a||^2 + GR\sqrt{n} \tag{12}$$

Now, we can use the bound $||a|| \leq R$ for all $a \in A$ and (12) can be re-written as:

$$R^2(\frac{1}{2\eta_1} + \frac{1}{2}\sum_{t=2}^{n}(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}})) + GR\sqrt{n} \tag{13}$$

Here, in (13) we can telescope and we have for $\eta_n = \frac{R}{G\sqrt{n}}$:

$R^2(\frac{1}{2\eta_1} + \frac{1}{2\eta_2} - \frac{1}{2\eta_1} + \frac{1}{2\eta_3} - \frac{1}{2\eta_2} + ... + \frac{1}{2\eta_{n-1}} - \frac{1}{2\eta_{n-2}} + \frac{1}{2\eta_n} - \frac{1}{2\eta_{n-1}}) + GR\sqrt{n} =$

$= R^2\frac{1}{2\eta_n} + GR\sqrt{n} = \frac{R^2}{\frac{1}{\frac{2R}{G\sqrt{n}}}} + GR\sqrt{n} = \frac{R^2 G\sqrt{n}}{2R} + GR\sqrt{n} = \frac{GR\sqrt{n}}{2} + GR\sqrt{n} = \frac{3}{2}GR\sqrt{n}$

Thus we shown that:

$$\sum_{t=1}^{n} \frac{1}{2\eta_t}||a_t - a||^2 - \frac{1}{2\eta_t}||a_{t+1} - a||^2 + \frac{\eta_t}{2}||g_t||^2 = \frac{3}{2}GR\sqrt{n} \tag{14}$$

So, by using (14), we have that (7) can be written as:

$$\sum_{t=1}^{n} g_t(a_t - a) \leq \frac{3}{2}GR\sqrt{n} \tag{15}$$

Finally, by using (15), we have that (3) can be written as:

$$R_n^a := \sum_{t=1}^{n} l(a_t, z_t) - l(a, z_t) \leq \frac{3}{2}GR\sqrt{n} \;\blacksquare \tag{16}$$

Hence, we shown 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 $\boxed{R_n^a \leq \frac{3}{2}GR\sqrt{n}}$

Exercise 4

(a)

Based on the description, we derive that the value of our wealth on the next day should be given:

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

(b)

Based on 4a we have that:

$\frac{W_{t+1}}{W_t} = \sum_{i=1}^{d} p_t^i r_t^i$ and also we have $z_t^i = -log(r_t^i)$

Thus, $\sum_{t=1}^{n} -log(\sum_{i=1}^{d} p_t^i e^{-z_t^i}) = \sum_{t=1}^{n} -log(\sum_{i=1}^{d} p_t^i e^{log r_t^i}) = \sum_{t=1}^{n} -log(\sum_{i=1}^{d} p_t^i r_t^i) = \sum_{t=1}^{n} -log(\frac{W_{t+1}}{W_t}) = -log(\frac{W_2}{W_1}) - log(\frac{W_3}{W_2}) - ... - log(\frac{W_{n+1}}{W_n}) = -log(W_2) + log(W_1) - log(W_3) + log(W_2) - ... - log(W_{n+1}) + log(W_n) = log(W_1) - log(Wn+1) = -log(\frac{Wn+1}{W_1})$

Hence, we have that $\boxed{\sum_{t=1}^{n} -log(\sum_{i=1}^{d} p_t^i e^{-z_t^i}) = -log(\frac{W_{n+1}}{W_1})}$

Thus, based on the form of the above formula and on the definition of the cumulative miss loss, we conclude that the mix loss is appropriate for this setting. Also, on the other hand in the dot loss we have the assumption that $z_t^i \in [0,1]$ which is not the case here. Finally, we want the total mix loss to be as small as possible, as in this case our wealth would be greater, because smaller total mix loss means greater wealth.

(c)

1.

Nothing to do here. Just to inspect the the given files.

2.

The solution is in the AA.m file. Namely the adversary moves are in the matrix Zt and the total loss of each expert in the matrix ExpertLoss.

3.

The solution is in the AA.m file. The expert regret of AA has been bound to be 0.2232.

4.

The total loss of AA has been found to be -1.4311(sum LossPt). The total losses of the five experts are -1.1530, -1.3649, -1.3249, -1.5774 and -1.6543 (lossesofExperts). So we conclude, that the loss of AA is smaller than the losses of the first three experts and larger than the loss of the two last experts. Also, as we said the expert regret of AA has been bound to be 0.2232 which is smaller than the guaranteed expert regret which equals 1.6094 ($log(5)$). Thus we could say that the adversary does not generate so ‘difficult' data since the difference of the expert regret of AA (0.2232) is large enough from the guaranteed expert regret (1.6094)

5.

It can be depicted in figure 1 that in general the confidence of BTC is higher that the other coins. Also, the worth of BTC is higher in comparison to the other coins. So, we could say from this that the strategy which we follow is good. Also we could notice that after its peak, the confidence in BTC has a great fall. Finally, a lot of fluctuations in the confidence of the coins can be depicted and it is something which we expect since the strategy of AA is to guarantee small regret against any adversary and thus we spread our chances in order to be conservative.

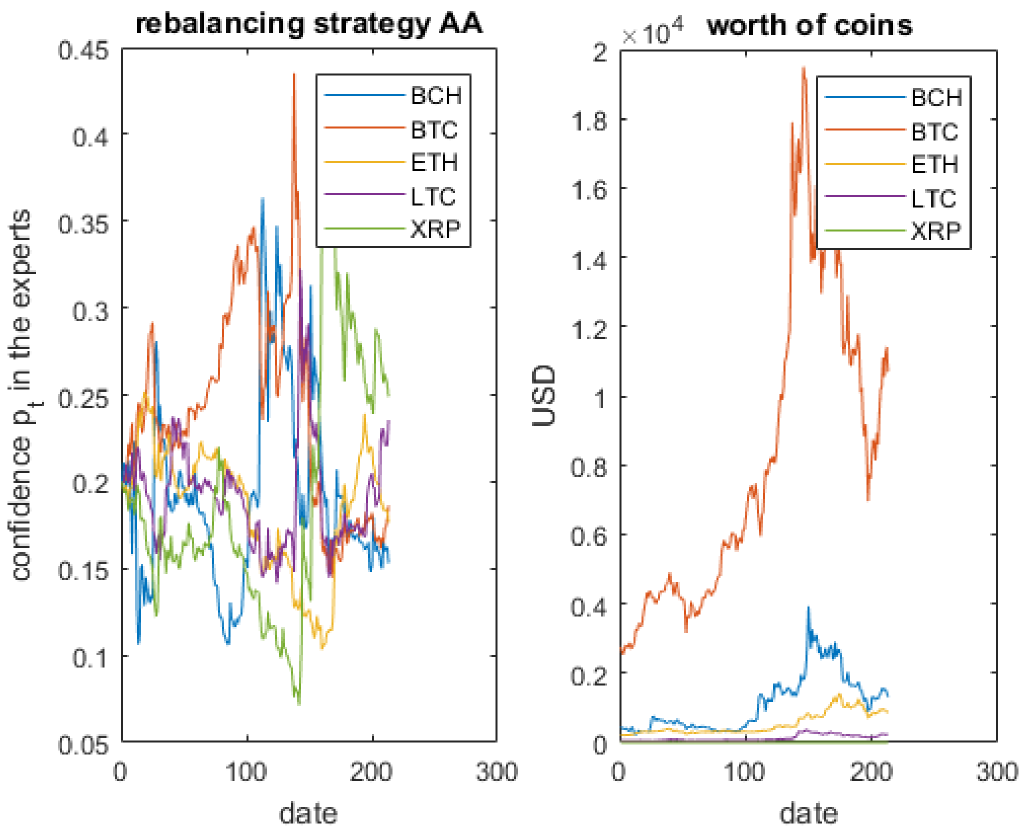

Figure 1: Confidence p_t in the experts (Left) and worth of coins (Right)

6.

If we would have invested according to the AA strategy,our wealth would have been increased 4.1833 times (gainTotal variable in AA.m). Hence, if we had invested for example 100 euro, our wealth would have been increased to 418.33 euro.

(d)

1.

We have that the OCO regret is given by the formula: $R_n = \sum_{t=1}^{n} l_m(a_t, z_t) - min_{a\epsilon A} \sum_{t=1}^{n} l(a, z_t)$ where the expert regret is given by the formula $R_n^E = \sum_{t=1}^{n} l_m(p_t, z_t) - min_i \sum_{t=1}^{n} z_t^i$. Hence, we have a different performance measure. In the expert setting we measured the performance of a strategy a_t by the expert regret. Since in the OCO setting, we do not have experts any more, we compare our strategy now to the best fixed action instead to the best expert. In this particular example, in the previous question, in the expert setting, we had five experts, $e_1, ..., e_5$ where expert e_1 in each round will choose move $e_1 = (1, 0, 0, 0, 0)$ and in general expert i will always invest all its money in asset i. In contrast here, in the OCO setup we got rid the notion of experts since we might not have access to those.

2.

$\nabla_a L_m(a, z_t) = \nabla_a(-log(a_t^T r_t)) = \frac{-1}{a_t^T r_t} \nabla_a(a_t^T r_t) = \frac{-r_t}{a_t^T r_t}$

Hence, we have that $\boxed{\nabla_a L_m(a, z_t) = \dfrac{-r_t}{a_t^T r_t}}$

3.

The solution is in the mixLoss.m file.

4.

The solution is in the OGD.m file.

5.

We have been provided with some values for R and G. Namely we have that R=1 and G=4.5072. It should be $||a||_2 \leq R$. So, for these particular adversary moves z_t this value of R is OK, since the actions $a_t^i \in [0, 1]$ for all $a_t \in A$. Also, it should be $||\nabla l(a, z)||_2 \leq G$ for all $(a, z) \in A \times Z$. Hence, also the value of G is OK since we calculated and found that $max||\nabla l(a, z)||_2 = 2.3359$.

6.

If we would have used OGD to invest, our wealth would have been increased 6.1428 times (gainTotal variable in OGD.m). Hence, if we had invested for example 100 euro, our wealth would have been increased to 614.28 euro.

7.

The solution is in the OGD.m file. We found that the optimal 'fixed' action $a \in A$ is the action which corresponds to day 165 and namely it is $a_{165} = (0.20663, 0.1960, 0.1886, 0.1968, 0.2120)$ and gives and a loss of -1.8584. Also, we found that the OCO regret of OGD equals 0.0431.

8.

The loss of OGD has been found to be -1.8153 and the loss of the best fixed action -1.8584, which is smaller. The OCO regret of OGD has been found to be 0.0431, where the guaranteed OCO regret equals 65.7803, which is larger.Thus we could say that the adversary does not generate so 'difficult' data since the difference of the OCO regret of OGD (0.0431) is very large enough from the guaranteed OCO regret (65.7803).

9.

In case that a coin i and a time t crashes, that is $r_t^i = 0$, then we have that $G -> \infty$. In this case, the assumption $||\nabla l(a, z)||_2 \leq G$ of Theorem 2 is violated as it becomes unbounded. Thus, the the bound from the OCO regret also goes to ∞ and hence the OCO regret also becomes unbounded.

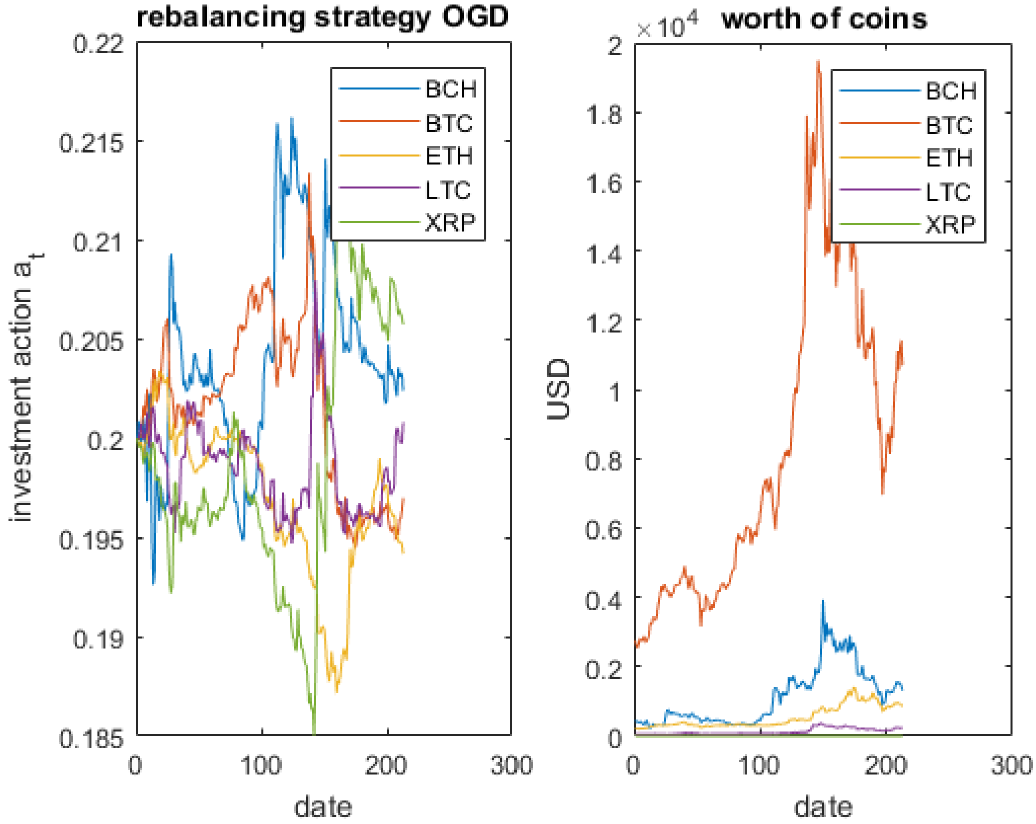

Figure 2: strategy of OGD

```
%Appendix
% Exercise: Aggregating Algorithm (AA)

clear all;
load coin_data;

d = 5;
n = 213;

% compute adversary movez z_t
%%% your code here %%%
z_t=-log(r);

%Total loss of each expert
%%% your code here %%%
for i=1:n
    for j=1:d
        expert_loss(i,j) = sum(z_t(1:i,j));
    end
end

% compute strategy p_t (see slides)
%%% your code here %%%
p(1,1)=0.2;
p(1,2)=0.2;
p(1,3)=0.2;
p(1,4)=0.2;
```

```
p(1,5)=0.2;
for i=2:n
    for j=1:d

            p(i,j)=exp(-expert_loss(i-1,j))/ sum(exp(-expert_loss(i-1,:)));

    end
end

% compute loss of strategy p_t
%%% your code here %%%
for i=1:n
    Loss_pt(i) = -log(sum(p(i,:).*r(i,:)));
end

% compute losses of experts
%%% your code here %%%
for i=1:d
    losses_of_experts(i) = expert_loss(213,i);
end

% compute regret
%%% your code here %%%
Expert_Regret = sum(Loss_pt) - min(losses_of_experts);

% compute total gain of investing with strategy p_t
%%% your code here %%%
for i=1:n
    daily_gain(i)= sum(r(i,:).*p(i,:));
end

%total gain
gain_total= prod(daily_gain);

% compare total loss of AA and total loss of expert
comparison=sum(Loss_pt)<losses_of_experts(1)
comparison1=sum(Loss_pt)<losses_of_experts(2)
comparison2=sum(Loss_pt)<losses_of_experts(3)
comparison3=sum(Loss_pt)<losses_of_experts(4)
comparison4=sum(Loss_pt)<losses_of_experts(5)

% compare expert regret of AA and theoretical guarantee
comparison5=Expert_Regret<log(d)


%% 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')
```

```
% Exercise: Mix Loss
function [l, g] = mix_loss(a, r)
%    [l, g] = MIX_LOSS(a, r)
%    Input:
%        a (column vector), the investment strategy (note a should be normalized so tha
%        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 action a)

    %%% your code here %%%
l=-log(a'*r);
g=-r/(a'*r);
end
```

```
% 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:
%%% your code here %%%
eta=R/(sqrt(n)*G);

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
    %%% your code here %%%
    a=A(:,t)-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
%%% your code here %%%
Total_Loss=sum(L,1)

% compute total gain in wealth
%%% your code here %%%
B=A'
for i=1:n
    daily_gain(i)= sum(r(i,:).*B(i,:));
end

%total gain
gain_total= prod(daily_gain);

% compute best fixed strategy (you may make use of loss_fixed_action.m and optimizatic
```

```
%%% your code here %%%
for i=1:213
    best_fixed(i)=loss_fixed_action(A(:,i));
end
[C,I] = min(best_fixed);

% compute regret
%%% your code here %%%
OCO_Regret = Total_Loss - best_fixed(I);

% compare total loss of OGD and  loss of best fixed action
comparison=Total_Loss<C

% compare OCO regret of OGD and theoretical guarantee OCO regret
comparison1=OCO_Regret<R*G*sqrt(n)

%% 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')
```