IN4320 Machine Learning — Assignment 3

Francisca Mestre - 4930142

March 2019

Exercise 1

Some of the following exercises were solved using MATLAB. The code used is shown in Appendix A.

(a)

The application of strategy A results in the values of p_t summarized in Table 1. When using

p_1	p_2	p_3	p_4
0.33	1	1	0
0.33	0	0	1
0.33	0	0	0

Table 1: p_t values obtained using strategy A

strategy B (Aggregating Algorithm) the values obtained are as shown in Table 2.

p_1	p_2	p_3	p_4
0.33	0.37	0.38	0.18
0.33	0.33	0.34	0.45
0.33	0.30	0.28	0.37

Table 2: p_t values obtained using strategy B

(b)

The total mix loss, given by

$$\sum_{t=1}^{n} - \ln \left(\sum_{i=1}^{d} p_t^i e^{-z_t^i} \right)$$

obtained for both strategies A and B is, respectively, 1.9967 and 0.7089.

(c)

The expert regret using mix loss is given by

$$R_n^E = \sum_{t=1}^{4} l_m(p_t, z_t) - \min_i \sum_{t=1}^{4} z_t^i. \tag{1}$$

For strategy A its value is 1.6967 and for strategy B the expert regret is 0.4089. These values are obtained by subtracting to the total mix loss of each strategy the minimum of the vector $\sum_{s=1}^{t} z_t^i = \begin{bmatrix} 1 & 1 & 0.3 \end{bmatrix}^T$ which is 0.3. This was done in MATLAB as shown in Appendix A.

(d)

From the lecture we have that for the Aggregating Algorithm the expert regret R_n^E is bounded by $\ln(d)$, $R_n^E \leq \ln(d)$. From the expression for the expert regret (1) we get

$$\sum_{t=1}^{4} l_m(p_t, z_t) \leq \ln(d) + \min_i \sum_{t=1}^{4} z_t^i = C.$$

(e)

For this example we have

$$C = \ln 3 + \min \begin{bmatrix} 1 \\ 1 \\ 0.3 \end{bmatrix} = \ln 3 + 0.3 = 1.3986$$

(f)

The total mix loss for strategy B (AA) is indeed smaller than the constant C which is what we aimed for when deriving C. However, for strategy A this is not verified as the cumulative loss gives much higher results, violating the bound.

Exercise 2

(a)

For $p_1 = (a, 1-a)$ the best adversary move is

$$z_1 = \begin{cases} (-\infty, 0), & \text{if } a = 0 \\ (0, +\infty), & \text{if } 0 < a \leq 0.5 \\ (+\infty, 0), & \text{if } 0.5 < a < 1 \\ (0, -\infty), & \text{if } a = 1. \end{cases}$$

(b)

For $n = 1$ and $d = 2$ we can rewrite (1) as

$$R_1^E = l_m(p_1, z_1) - \min_{1,2}(z_1^1, z_1^2).$$

For all the possible values of $a \in [0, 1]$ and the corresponding adversary moves as described above we have:

- $a = 0$
  $p_1 = (0, 1) = e^2 \implies l_m(p_1, z_1) = z_2$

  $$R_1^E = z_2 - z_1 = +\infty - 0 = +\infty,$$

  with

  $$R_1^E > \ln(2).$$

- $0 < a \leq 0.5$
  $p_1 = (a, 1-a) \implies l_m(p_1, z_1) = -\ln(ae^0 + (1-a)e^{-\infty}) = -\ln(a)$

  $$R_1^E = -\ln(a) - z_1 = -\ln(a),$$

  with $-\ln(0.5) \leq -\ln(a) < +\infty$, thus

  $$\ln(2) \leq R_1^E < +\infty.$$

- $0.5 < a < 1$
  $p_1 = (a, 1-a) \implies l_m(p_1, z_1) = -\ln(ae^{-\infty} + (1-a)e^0) = -\ln(1-a)$

  $$R_1^E = -\ln(1-a) - z_2 = -\ln(1-a),$$

  with $-\ln(0.5) < -\ln(1-a) < +\infty$, thus

  $$\ln(2) < R_1^E < +\infty.$$

- $a = 1$
  $p_1 = (1, 0) = e^1 \implies l_m(p_1, z_1) = z_1$

  $$R_1^E = z_1 - z_2 = 0 - (-\infty) = +\infty,$$

  with

  $$R_1^E > \ln(2).$$

(c)

Assume the adversary move z_1 is infinity except for the m^{th} entry corresponding to the smallest non-zero value of p_1, with $z_1^m = 0$, i.e., $z_1 = (+\infty, \dots, +\infty, 0, +\infty, \dots, +\infty)$. Then, since $e^0 = 1$, $e^{-\infty} = 0$ and $\min_i z_1^m = 0$, the expert regret will be

$$R_1^E = -\ln p_1^m$$

We have that $0 < p_1^m \leq {}^1/_d$ once p_1^m is higher than ${}^1/_d$ it is no longer the smallest value from p_i. Thus, we have

$$-\ln ({}^1/_d) \leq -\ln (p_1^m) < +\infty$$

which is equivalent to having

$$\ln (d) \leq R_1^E < +\infty.$$

Now assume that the smallest value of p_1 is $p_1^m = 0$. If the adversary plays z_1 equal to zero except for $z_1^m = -\infty$, i.e., $z_1 = (0, \dots, 0, -\infty, 0, \dots, 0)$ the expert regret will be

$$R_1^E = -\ln (\sum_{i=1}^{d} p_1^i) - (-\infty) = +\infty > \ln (d).$$

(d)

Take $p_1 = ({}^1/_d, \dots, {}^1/_d)$. As mentioned in the questions above, if the adversary move is, for instance, $z_1 = (0, +\infty, \dots, +\infty)$ the regret at $t = 1$ is $R_1^E = \ln (d)$. In the next move p_2 will be different from p_1 but if we apply the adversary move as described above where z_2 is infinity except for the entry corresponding to that of p_2 with the smallest value then the regret R_2^E equals $\ln (d) - \ln (p_2^m)$. Since $p_2^m < 1 \Rightarrow -\ln(p_2^m) > 0$ and $R_2^E > R_1^E = \ln (d)$. By continuing this application we can always find a set of moves $z_1, \dots, z_n$ such that the regret is larger or equal to $\ln (d)$.

Exercise 3

(a)

Using the results from the slides we know that expert regret when using the Exp Strategy is bounded as follows

$$R_n^E \leq n\frac{\eta}{8} + \frac{\ln (d)}{\eta}. \tag{2}$$

For strategy A we have $\eta_A = \sqrt{4 \ln (d)}$ and $R_n^A \leq C_n^A$ with

$$C_2^A = \sqrt{\ln (d)}.$$

For strategy B we have $\eta_B = \sqrt{2 \ln (d)}$ and $R_n^B \leq C_n^B$ such that

$$C_2^B = \frac{3}{2\sqrt{2}} \ln (d).$$

(b)

With $n = 2$ the ratio ${}^{C_2^A}/_{C_2^B}$ is

$$\frac{C_2^A}{C_2^B} = \frac{\sqrt{\ln(d)}}{{}^3/_{2\sqrt{2}}\sqrt{\ln(d)}} = \frac{2\sqrt{2}}{3} \approx 0.9428. \tag{3}$$

(c)

The bound C_2^A is, then, smaller than C_2^B as verified in the question above.

(d)

Now for $n = 4$, C_4^A and C_4^B are

$$C_4^A = \frac{3}{2}\sqrt{\ln(d)}$$
$$C_4^B = \frac{4}{2\sqrt{2}}\sqrt{\ln(d)}.$$

(e)

The ratio ${}^{C_4^A}/_{C_4^B}$ is now

$$\frac{C_4^A}{C_4^B} = \frac{{}^3/_2\sqrt{\ln(d)}}{{}^4/_{2\sqrt{2}}\sqrt{\ln(d)}} = \frac{3}{2\sqrt{2}} \approx 1.0607. \tag{4}$$

(f)

The tighter bound is C_n^B.

(g)

We have shown that for some n we have

$$C_n^B < C_n^A \tag{5}$$

with C_n^A and C_n^B upper bounds on the regrets R_n^A and R_n^B respectively. As these are bounds and not equalities, (5) does not imply that $R_n^B < R_n^A$ since at some point we can have R_n^A lower than the bound on the regret with strategy B.

(h)

For the Exp strategy we have

$$p_t^i = \frac{e^{-\eta L_{t-1}^i}}{\sum_{j=1}^d e^{-\eta L_{t-1}^j}}. \tag{6}$$

As we increase the learning rate η the strategy at time t puts less trust on the experts with a higher cumulative loss until time $t - 1$.

(i)

If the learning rate is very high ($\eta = +\infty$) the strategy will also explode to infinity as seen in (6). Furthermore, from (2) we see that there is no longer a bound on the regret, i.e. $R_n^E < \infty$.

Exercise 4

(a)

We have that

$$p_t^i e^{-z_t^i} = p_t^i e^{-(-\ln(r_t^i))} = p_t^i r_t^i \tag{7}$$

$$W_{t+1} = p_t^T r_t W_t \Longrightarrow \sum_{i=1}^{d} p_t^i r_t^i = p_t^T r_t = \frac{W_{t+1}}{W_t} \tag{8}$$

By combining (7) and (8) with the left-hand side of the equality we want to show we get

$$\sum_{t=1}^{n} -\ln\Big(\sum_{i=1}^{d} p_t^i e^{-z_t^i}\Big) = \sum_{t=1}^{n} -\ln(\frac{W_{t+1}}{W_t}) =$$

$$= -\ln(W_2) + \ln(1) - \ln(W_3) + \ln(2) - \cdots - \ln(W_{n+1}) + \ln(W_n)$$

which is a telescopic sum that ultimately results in

$$\ln(W_1) - \ln(W_{n+1}) = -\ln(\frac{W_{n+1}}{W_1}).$$

(b)

The mix loss is better because it is always less or equal than the dot loss.

(e)

Table 3 contains the cumulative loss suffered by each expert after the first move and after the last move.

	L_1^1	L_1^2	L_1^3	L_1^4	L_1^5
$t = 1$	-0.0648	-0.0089	0.0055	-0.0079	0.0158
$t = 213$	-1.1530	-1.3649	-1.3249	-1.5774	-1.6543

Table 3: Expert losses at $t = 1$ and $t = 213$

(f)

The code is shown in Appendix B.

(g)

The loss of the Aggregating Algorithm at $n = 213$ is computed as

$$-\ln\Big(\sum_{t=1}^{d} p^i_{213} e^{-z^i_{213}}\Big)$$

and it equals 0.0706.

(h)

The regret of the Aggregating Algorithm is equivalent to the regret in the expert setting using mix loss as given by (1). At the end of the 213 days, the regret of the Aggregating Algorithm is 0.2232.

(i)

The loss of the algorithm relates to the loss of each expert through the expert regret equation as in (1). Since the regret is positive we have that

$$\sum_{t=1}^{n} l_m(p_t, z_t) - \min_i \sum_{t=1}^{n} z^i_t \geq 0 \Longrightarrow \sum_{t=1}^{n} l_m(p_t, z_t) \geq \min_i \sum_{t=1}^{n} z^i_t$$

i.e. the AA loss is greater or equal that the smallest expert loss at each time step.

(j)

For the Aggregating Algorithm we have that the expert regret is upper bounded by $\ln(d) = \ln 5 \approx 1.6094$. Using the code shown below in MATLAB we find that no value of the regret over the time instances $1, \ldots, n$ violates the bound since the vector `auxiliar` has all its entries equal to zero at the end of the test.

```
d = 5;
auxiliar = zeros(length(regret),1);
auxiliar(regret>=log(d)) = 1;
verify = sum(auxiliar);
```

(k)

The adversary is not generating much difficult data as the regret satisfies the bound.

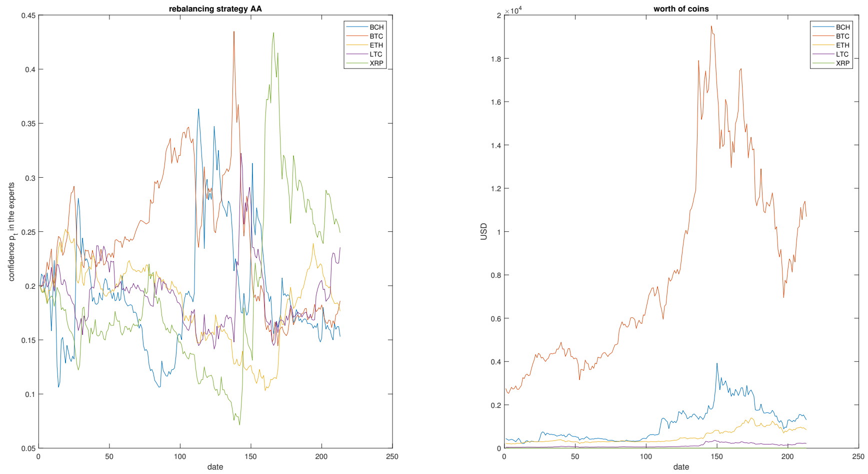

Figure 1: Visualization of strategies p_t and coin value

(l)

Figure 1 shows the strategies p_t over time as well as the bitcoin values.

(m)

In Figure 2 we show a detail of a certain period in time when the value of the coin increases but the algorithm decides not to invest on it so much. This kind of behaviour might occur when the value of the coin dropped in the past so the algorithm will not trust on it to invest. As the algorithm does not know ahead that the value will increase again, it invests less on the coin in question.

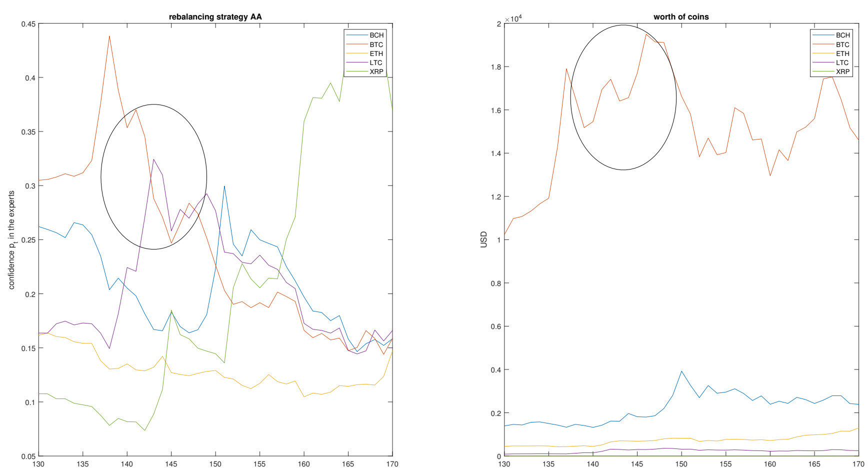

Figure 2: Detail in the behaviour

(n)

If we had invested according to the Aggregating Algorithm our wealth would be 4.5085 times higher than when we started, i.e., ${}^{W_{end}}/_{W_1} = 4.1833$.

(o)

In a real setting the return vector is influenced by the investments of all companies or individuals involved in the trading of the stocks. Thus this algorithm is somewhat idealistic as it cannot know beforehand how the competitors will invest their money. It is, then, not possible to make as much profit as in the simulation.

Exercise 5

(a)

If we take the derivative of the right-hand side regret bound with respect to η and set it to zero we find η_n that minimizes that bound.

$$\frac{du}{dt}\left(\frac{R^2}{2\eta} + \frac{\eta G^2 n}{2}\right) = 0 \Leftrightarrow -\frac{2R^2}{4\eta_n^2} + \frac{G^2 n}{2} = 0 \Leftrightarrow$$
$$\eta_n = \frac{R}{G\sqrt{n}}.$$

(b)

(c)

The pseudo-code for a portfolio selection algorithm based on the online gradient descent method is presented below.

Algorithm 1 Portfolio selection

```
initialize p_1
for t do
    receive return vector of strategy p_t, r_t(i) = (price of asset i at t+1)/(price of asset i at t)
    compute the gradient of the payoff function, ∇_a l(a_t, r_t) = r_t / (r_t^T a_t)
    compute next action, a_{t+1} = Π_A(a_t − η/(r_t^T a_t) r_t)
    compute the regret after t rounds, R_t = Σ_{n=1}^t log(r_n^T a_n) − min_{a∈Δ_d} Σ_{n=1}^t log(r_n^T a)
    compute wealth at time t+1, W_{t+1} = r_t^T a_t W_t
end for
```

initialize p_1

for t **do**

receive **return vector** of strategy p_t, $r_t(i) = \frac{\text{price of asset } i \text{ at } t+1}{\text{price of asset } i \text{ at } t}$

compute the **gradient** of the payoff function, $\nabla_a l(a_t, r_t) = \frac{r_t}{r_t^T a_t}$

compute **next action**, $a_{t+1} = \Pi_{\mathcal{A}}(a_t - \frac{\eta}{r_t^T a_t} r_t)$

compute the **regret** after t rounds, $R_t = \sum_{n=1}^{t} \log(r_n^{\mathrm{T}} a_n) - \min_{a \in \Delta_d} \sum_{n=1}^{t} \log(r_n^{\mathrm{T}} a)$

compute **wealth** at time $t+1$, $W_{t+1} = r_t^{\mathrm{T}} a_t W_t$

end for

The constants G and R are such that

$$\max_{a\in\mathcal{A}} \|a\|_2 \leq R, \quad \|\nabla_a l(a,z)\|_2 \leq G.$$

Then we can select $R = 1$

$$R = 1, \quad G = \frac{\sqrt{d \max_i (r_t^2(i))}}{r_t^{\mathrm{T}} a_t}$$

since the 2-norm of the vectors $a \in \mathcal{A}$ is at most 1 and the 2-norm of the return vector is at most $\sqrt{d \max_i (r_t^2(i))}$.

Appendix

A Exercise 1

```
%% Exercise 1
%% (a)
%Strategy A
e = 1:3;
t = 1:4;
z = [0 0 1 0; 0.1 0 0 0.9; 0.2 0.1 0 0];
pA = zeros(length(e), length(t)); pA(:, 1) = 1/3;
L = zeros(length(e), 1);
for s = 2:length(t)
    for i = 1:length(e)
        L(i) = L(i) + z(i, s-1);
    end
    [val, b] = min(L);
    pA(:, s) = zeros(length(e), 1); pA(b, s) = 1;
end

%Strategy B
pB = zeros(length(e), length(t)); pB(:, 1) = 1/3;
L = zeros(length(e), 1);
for s = 2:length(t)
    for i = 1:length(e)
        L(i) = L(i) + z(i, s-1);
    end
    C = sum(exp(-L));
    pB(:, s) = exp(-L)/C;
end

%% (b)
t_mixlossA = 0; t_mixlossB = 0;
for s = 1:length(t)
   t_mixlossA = t_mixlossA - log(sum(pA(:,s).*exp(-z(:,s))));
   t_mixlossB = t_mixlossB - log(sum(pB(:,s).*exp(-z(:,s))));
end

%% (c)
e_regretA = t_mixlossA - min(sum(z,2));
e_regretB = t_mixlossB - min(sum(z,2));
```

B Exercise 4

```
% Exercise: Aggregating Algorithm (AA)
% First: read readme.m

clear
load coin_data;

d = 5;
n = 213;

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

p = zeros(n,d); W = zeros(n,1); L = p; regret = W; loss_t = W;
   gain = W;
p(1,:) = 1/d;
W_0 = 564 ;
for t = 1:n
    % compute strategy p_t (see slides)
    if t == 1
        p(t,:) = 1/d;
    else
        C = sum(exp(-L(t-1,:)));
        p(t,:) = (exp(-L(t-1,:)))/(C);
    end

    % compute loss of strategy p_t
    loss_t(t) = -log(p(t,:)*exp(-z(t,:))');

    % compute losses of experts
    if t == 1
        L(t,:) = z(t,:);
    else
        L(t,:) = L(t-1,:) + z(t,:);
    end

    % compute regret
    regret(t) = sum(loss_t(1:t)) - min(sum(z(1:t,:),1));

    % compute total gain of investing with strategy p_t
    W(t+1) = (p(t,:)*r(t,:)')*W(t);
    gain(t) = W(t+1)/W(t);
end
```

```
%% 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'), %xlim([130 170])
ylabel('confidence p_t in the experts')

subplot(1,2,2);
plot(s)
legend(symbols_str)
title('worth of coins')
xlabel('date'), %xlim([130 170])
ylabel('USD')

% for exercise 4 (m) it is not sufficient to show this plot,
  please show
% a zoomed in example where the behaviour occurs.
```