IN4320 Machine Learning
Exercise 3

25 February 2019

TU Delft
Coordinated by Alexander Mey and Tom Viering

Matthijs Bekendam 4725751 J.M.Bekendam@student.tudelft.nl

Contents

1 Question 1

1.1 1a

	z_1	z_2	z_3	z_4
e_1	0.33	1	1	0
e_2	0.33	0	0	1
e_3	0.33	0	0	0

Table 1: P_t actions for strategy A

	z_1	z_2	z_3	z_4
e_1	0.33	0.37	0.38	0.18
e_2	0.33	0.33	0.34	0.45
e_3	0.33	0.30	0.28	0.37

Table 2: P_t actions for strategy B

Listing 1: Code exercise 1a strategy A

```
Z = [0  0 1 0;
     0.1 0 0 0.9;
     0.2 0.1 0 0];

timestep = 4;

L1 = 0;
L2 = 0;
L3 = 0;

pa = zeros(3,timestep);
pa(:,1) = 1/3;

for t = 1:timestep

    L1 = L1+Z(1,t)
    L2 = L2+Z(2,t)
    L3 = L3+Z(3,t)
    L = [L1 L2 L3]

    if t < 4
        L_min = min(L,[],'all');
        [y,expert] = find(L_min == L);
        pa(expert,t+1) = 1;
    end

end
```

Listing 2: Code exercise 1a strategy B

```
L1 = 0;
L2 = 0;
L3 = 0;

pb = zeros(3,timestep);
```

```
pb(:,1) = 1/3;

for t = 1:timestep-1

    L1 = L1+Z(1,t);
    L2 = L2+Z(2,t);
    L3 = L3+Z(3,t);

    C = exp(-L1)+exp(-L2)+exp(-L3);


    p1 = (exp(-L1))/C;
    p2 = (exp(-L2))/C;
    p3 = (exp(-L3))/C;

    pb(1,t+1) = p1;
    pb(2,t+1) = p2;
    pb(3,t+1) = p3;


end
```

1.2 1b

	t_1	t_2	t_3	t_4
L_A	0.097	0	1	0.9
L_B	0.097	0.029	0.27	0.31

Table 3: Mix Loss of strategy A and strategy B for each time step t

The total mix loss of strategy A is 1.997 and The total mix loss of strategy B is 0.709.

Listing 3: Code exercise 1b strategy A and B

```

mix_loss= zeros(timestep,1);

for t = 1:timestep %t for timesteps

       sum_pz = 0;
       for e = 1:3 %e for experts
           sum_pz = sum_pz + p(e,t) * exp(-Z(e,t))
       end
       mix_loss(t,1) = -log(sum_pz);

end
    total_mix_loss = sum(mix_loss);
```

1.3 1c

Regret strategy A is 1.697, regret strategy B is 0.409. The regret was calculated through the following equation

$$R_4^{strategy} = \sum_{t=1}^{4} I_m(p_t, z_t) - \min_i \sum_{t=1}^{4} z_t^i \tag{1.1}$$

$$R_4^A = 1.997 - 0.3 = 1.697$$

$$R_4^B = 0.709 - 0.3 = 0.409$$

1.4 1d

$$R_t^{strategy} = \sum_{t=1}^{n} I_m(p_t, z_t) - \min_i \sum_{t=1}^{n} z_t^i \tag{1.2}$$

$$\sum_{t=1}^{n} I_m(p_t, z_t) - \min_i \sum_{t=1}^{n} z_t^i \leq log(d) \tag{1.3}$$

$$\sum_{t=1}^{n} I_m(p_t, z_t) \leq log(d) + \min_i \sum_{t=1}^{n} z_t^i \tag{1.4}$$

$$C = log(d) + \min_i \sum_{t=1}^{n} z_t^i \tag{1.5}$$

1.5 1e

The numerical value for C for this example is computed using equation 1.5

$$C = log(3) + 0.3 \tag{1.6}$$

$$C = 1.3986$$

1.6 1f

Strategy B is bounded by C but strategy A is not. This is due to strategy A not being able to perform a crucial step in the proof which sets this bound. This is due to the different definition of p_t.

$$\sum_{t=1}^{n} I_m(p_t, z_t) = \sum_t -ln(\sum_i p_t^i e^{-z_t^i}) \tag{1.7}$$

Transforming p_t^i as follows is key to set the bound:

$$\sum_{t=1}^{n} I_m(p_t, z_t) = \sum_t -ln(\sum_i \frac{e^{-L_{t-1}^i}}{\sum_j e^{-L_{t-1}^j}} e^{-z_t^i}) \tag{1.8}$$

Since strategy A uses $p_t = e_b(t)$ this action cannot be performed.

2 Question 2

2.1 2a

We are looking for the best adversary move, which maximizes the regret.

$$R_1^E = I_m(p_t, z_t) - \min_i z_t^i \tag{2.1}$$

$$R_1^E = -log(p^1 e^{-z^1} + p^2 e^{-z^2}) - \min_i z^i \tag{2.2}$$

To maximize the regret, the term inside the log-term should be as close to zero as possible. Because of the 'min'-term, one of the z^i terms should be kept small. Minimizing the log term can be done by canceling the highest p^i term by setting the accompanying z^i term to infinity (or a sufficiently high number), yielding $p^i \cdot 0$. The remaining z^i term will cancel itself out. Setting z^2 sufficiently high (in case $p^2 \geq p^1$) gives:

$$R_1^E = -log(p^1 e^{-z^1}) - z^1 \tag{2.3}$$

$$R_1^E = -log(p^1) - log(e^{-z^1})) - z^1 \tag{2.4}$$

$$R_1^E = -log(p^1) + z^1 - z^1 \tag{2.5}$$

$$R_1^E = -log(p^1) \tag{2.6}$$

2.2 2b

In this subsection it will be shown that there is always an adversary move such that the regret is larger or equal to log(d). This has in fact partially been shown in question 2a. When the regret is equal to log(d), it means the worst case scenario for the adversary. The worst move the player can make for the adversary is dividing the distribution evenly. In case $d = 2$, this would be $p = (0.5, 0.5)$, indeed log(2) = -log(0.5). When the trust in expert is more distributed, the regret will only increase, since the lower the number inside the log-term, the higher the outcome of equation 2.6 will be.

The regret is bounded (for d =2) by :

$$R_1^E = -log(p^1) \geq log(2) \tag{2.7}$$

Where p^1 is the lowest probability from the p vector.

2.3 2c

The prove from 2a can be used for a more general scenario for which $p_1 = (p_1^1, p_1^2 ... p_1^d)$.

$$R_1^E = -log(p^1 e^{-z^1} + p^2 e^{-z^2} + ... + p^d e^{-z^d}) - \min_i z^i \tag{2.8}$$

Again, all $p^i e^{-z^i}$ terms can be killed, except for the lowest p^i, by setting z^i very large. Again, we end up with equation 2.6. The worst case scenario for the adversary is when $p_1 = (1/d, 1/d...., 1/d)$, when this happens the regret becomes:

$$R_1^E = -log(1/d) = log(d) \tag{2.9}$$

Hence the regret will only grow when the distrubution is less even and makes the regret bounded as follows:

$$R_1^E \geq log(d) \tag{2.10}$$

2.4 2d

We can extend the approach of question 2c to multiple time steps. The regret for every time step is bounded by $R_{En} \geq log(d)$. Hence, adding multiple time steps together will always yield a regret larger or equal to log(d), which is the lowest value for the lowest possible time (t=1).

3 Question 3

3.1 3a

The theoretical guarantee for strategy A (n=2) is:

$$R_2^A \leq 2\frac{\sqrt{4log(d)}}{8} + \frac{log(d)}{\sqrt{4log(d)}} \tag{3.1}$$

$$R_2^A \leq \frac{\sqrt{4}\sqrt{log(d)}}{4} + \frac{log(d)}{2\sqrt{log(d)}} \tag{3.2}$$

$$R_2^A \leq \frac{1}{2}\sqrt{log(d)} + \frac{1}{2}\sqrt{log(d)} \tag{3.3}$$

$$R_2^A \leq \sqrt{log(d)} \tag{3.4}$$

The theoretical guarantee for strategy B (n=2) is:

$$R_2^B \leq \frac{\sqrt{2}\sqrt{log(d)}}{4} + \frac{log(d)}{\sqrt{2}\sqrt{log(d)}} \tag{3.5}$$

$$R_2^B \leq \frac{\sqrt{log(d)}}{2\sqrt{2}} + \frac{2log(d)}{2\sqrt{2}\sqrt{log(d)}} \tag{3.6}$$

$$R_2^B \leq \frac{\sqrt{log(d)}}{2\sqrt{2}} + \frac{2\sqrt{log(d)}}{2\sqrt{2}} \tag{3.7}$$

$$R_2^B \leq \frac{3}{2}\sqrt{\frac{log(d)}{2}} \tag{3.8}$$

3.2 3b

$$\frac{C_2^A}{C_2^B} = \frac{2\sqrt{2}}{3} = 0.9428 \tag{3.9}$$

3.3 3c

Since $\frac{C_2^A}{C_2^B}$ is smaller than zero, C_2^A is smaller and has a tighter bound.

3.4 3d

The theoretical guarantee for strategy A (n=4) is:

$$R_4^A \leq \sqrt{log(d)} + \frac{1}{2}\sqrt{log(d)} \tag{3.10}$$

$$R_4^A \leq \frac{3}{2}\sqrt{log(d)} \tag{3.11}$$

The theoretical guarantee for strategy B(n=4) is:

$$R_4^B \leq \frac{\sqrt{2}\sqrt{log(d)}}{2} + \frac{log(d)}{\sqrt{2log(d)}} \tag{3.12}$$

$$R_4^B \leq \frac{\sqrt{2}}{2}\sqrt{log(d)} + \frac{1}{\sqrt{2}}\sqrt{log(d)} \tag{3.13}$$

$$R_4^B \leq \frac{2\sqrt{log(d)}}{\sqrt{2}} \tag{3.14}$$

3.5 3e

$$\frac{C_4^A}{C_4^B} = \frac{3\sqrt{2}}{4} = 1.0607 \tag{3.15}$$

3.6 3f

Since $\frac{C_2^A}{C_2^B}$ is larger than zero, C_2^B is smaller and has a tighter bound.

3.7 3g

The difference between strategies A and B is the learning rate. A slower learning rate means the algorithm is to a lesser extend adapts to rapid, sudden fluctuations in the data. This could sometimes yield less regret when the data has less correlation. A larger learning rate is more aggressive in compensating for these large fluctuations which would increase the regret. This means that even though the strategy B has a tighter bound than strategy A, its regret might end up being higher.

3.8 3h

Figure 1 shows p_t values for different learning rates.

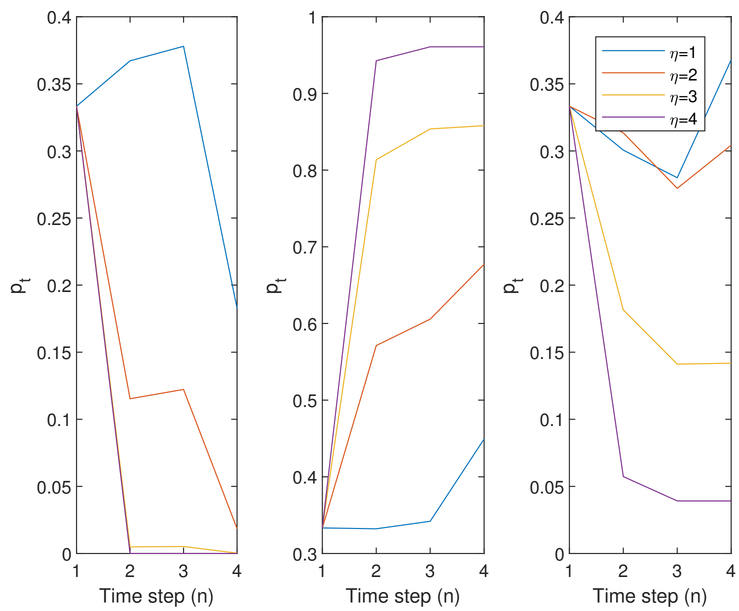


Figure 1: p_t values for different learning rate using the data set given in Question 1a.

From the figure it can be deduced that a higher learning rate weights changes more drastically, resulting in a more aggressive strategy.

3.9 3i

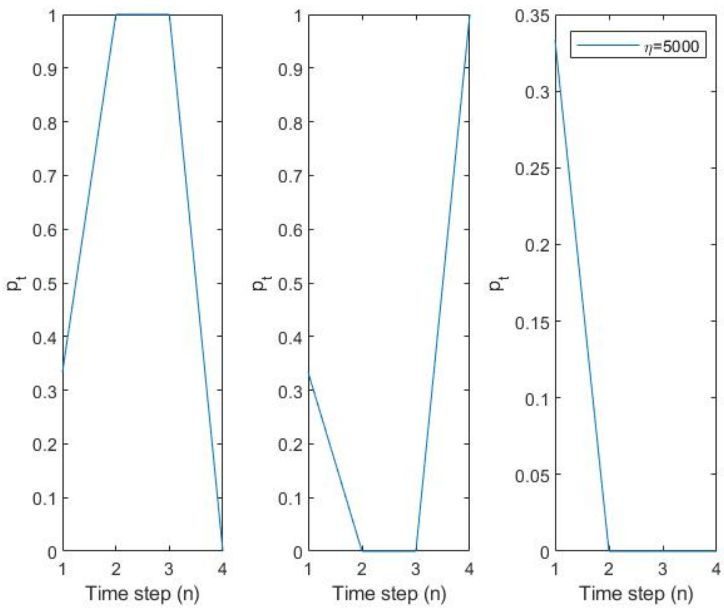


Figure 2: p_t values for an extremely high (mimicking infinity) learning rate rate using the data set given in Question 1a.

An ∞-learning rate will cause an increase of p_t to go to the value one instantly and the remaining $p_t's$ go to zero. In essence we get strategy A from question 1. We still have a guarantee on the regret since the bound C grows linearly when n increases (and η is sufficiently high). When the learning rate is high the formula for C reduces to

$$R_n^E \leq n\frac{\eta}{8}. \tag{3.16}$$

4 Question 4

4.1 4a

$$\sum_{t=1}^{n} I_m(p_t, z_t) = \sum_{t=1}^{n} -log(\frac{W_{t+1}}{r_t W_t} e^{log(r_t)}) \tag{4.1}$$

$$= \sum_{t=1}^{n} -log(\frac{W_{t+1}}{W_t}) \tag{4.2}$$

$$= \sum_{t=1}^{n} -log(W_{t+1}) + log(W_t) \tag{4.3}$$

$$= log(W_1) - log(W_2) + log(W_2) - log(W_3) + log(W_3)... - log(W_{n+1}) \tag{4.4}$$

All terms cancel out except for $-log(W_1)$ and $log(W_{n+1})$. Putting these terms back together in a single log term completes the proof.

$$\sum_{t=1}^{n} I_m(p_t, z_t) = -log(\frac{Wn+1}{W_1}) \tag{4.5}$$

4.2 4b

The difference between the mix loss and the dot loss is that the mix loss yields a non-linear quantity due to the e-power. The mix loss gives lower losses for low confidence with high penalty, but a higher loss when this confidence is increased. In the investing setting the mix loss is appropriate because it takes the trends and stability better into consideration.

4.3 4e

	e_1	e_2	e_3	e_4
Loss	-1.15	-1.36	-1.32	-1.58

Table 4: Loss of each expert

4.4 4f

See Appendix

4.5 4g

The total mix loss of the AA at n=213 is -1.43.

4.6 4h

The regret of AA is 0.22 and was calculated using the equation denoted in the code below.

Listing 4: Code exercise 4h

```
% compute loss of strategy p_t

mix_loss = zeros(time,1);

for t = 1:time

      sum_pz = 0;
```

```
        for e = 1:d
            sum_pz = sum_pz + p(t,e) * exp(-z_t(t,e))
        end
        mix_loss(t,1) = -log(sum_pz);

end
    total_mix_loss_p_t = sum(mix_loss);


% compute losses of experts
loss_per_e = sum(z_t)

L_min_total = min(loss_per_e,[],'all');

% compute regret
 Regret_pt = total_mix_loss_p_t - L_min_total
```

4.7 4i

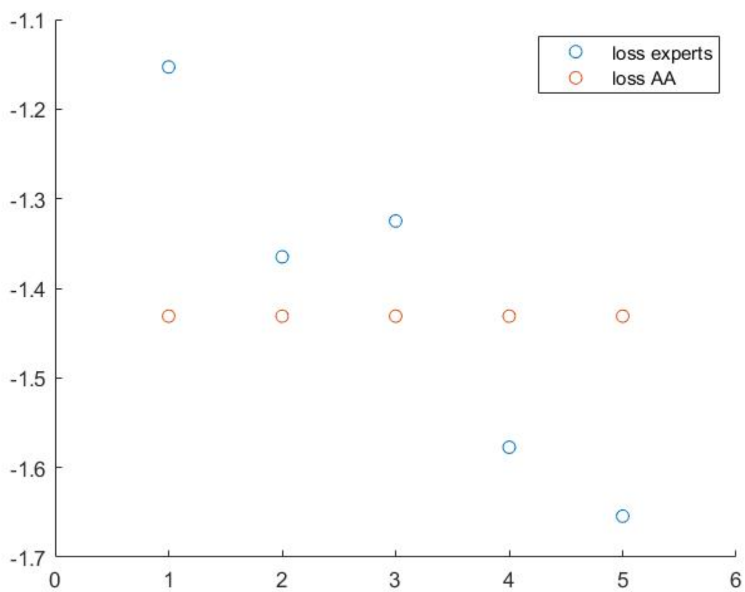


Figure 3: Loss of AA compared with the loss of each expert

It is observed that for the first three experts yield a higher loss than the AA algorithm. The second two experts yield a lower loss than the AA algorithm.

4.8 4j

The theoretical guarantee on the regret sets a bound for log(d) = log(5) = 1.6094. The difference between the AA regret and the bound is 1.3863.

4.9 4k

Currencies data in general is not difficult data since it always follows a trend and the jumps from one time step to the next are never humongous. This is because currencies fluctuations are bounded in some sense.

4.10 4l

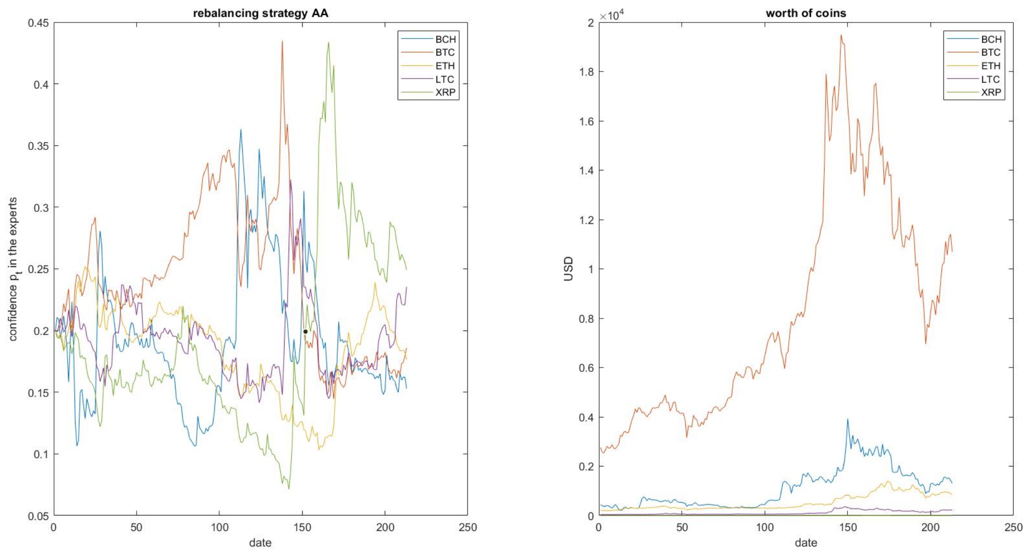


Figure 4: (Left) Visualization of the AA strategy pt vs t. (Right) Visualization of the coin values x_t vs t.

4.11 4m

It is observed that a coin might raise in value, although the AA strategy will invest less in that coin in the next time step. An example of this is illustrated in Figure 5.

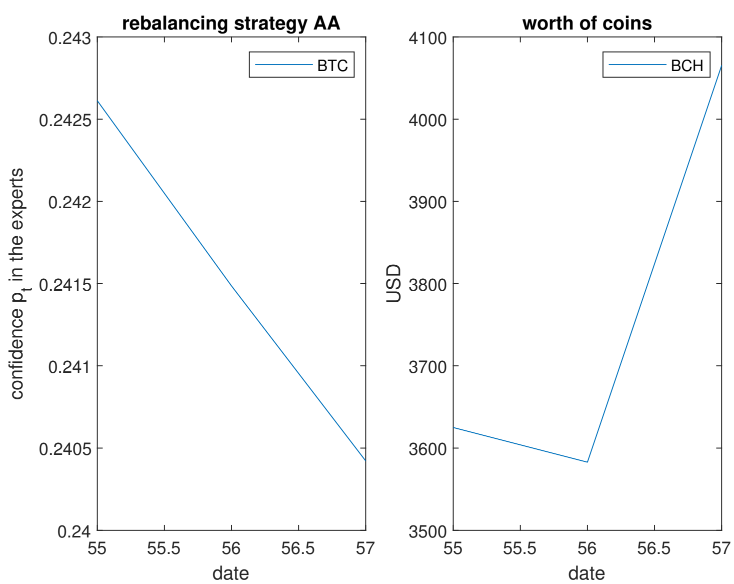


Figure 5: The currency BCT increases in value at time step n = 56 but the investment in the coin decreases.

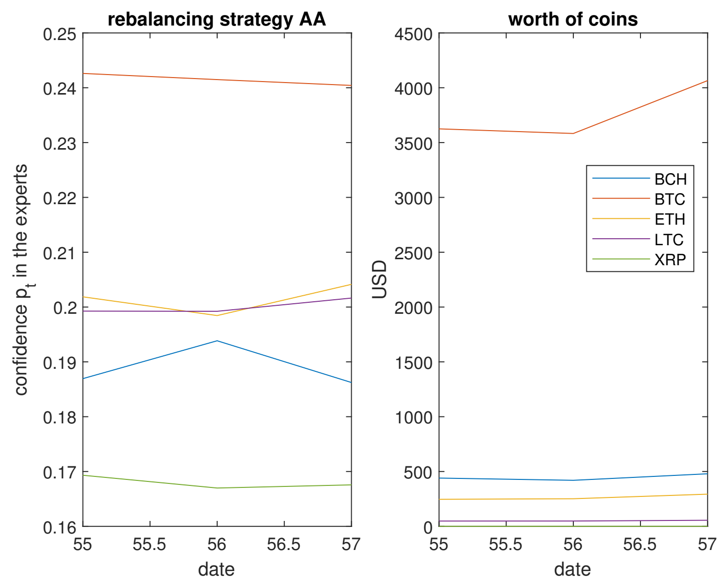


Figure 6: The worth of coins and the confidence in the currencies around time n = 56

This phenomenon can be explained through the fact that the 'BCT' shows an unstable behaviour compared to the other currencies. And although the other currencies do not increase as much, investing in a stable/predictable currency could be more profitable. And since p_t has to sum to one, it has to remove some confidence from the 'BCT' to add it into the other stable currencies.

4.12 4n

If we would have invested ($W_1 = 100$) according to the AA strategy, our wealth would have increased to $W_{214} = 418$.

4.13 4o

In reality we would probably not make a profit this large. This is partly due to the AA-algorithm influencing the currencies in reality, which was not taken into account in this assignment. Also there are additional costs to be able to enter the investing market for example internal expenses, transaction fees etc.

5 Question 5

5.1 5a

The optimal learning rate is given by

$$\frac{df}{d\eta} = -\frac{R^2}{2x^2} + \frac{G^2 n}{2} = 0 \tag{5.1}$$

$$\frac{2x^2}{R^2} = \frac{2}{G^2 n} \tag{5.2}$$

$$\eta = \frac{R}{G\sqrt{n}} \tag{5.3}$$

Next the following inequality will be proven $R_n \leq RG\sqrt{n}$

$$R_n \leq \frac{\frac{R^2}{2R}}{G\sqrt{n}} + \frac{G^2 n R}{2G\sqrt{n}} \tag{5.4}$$

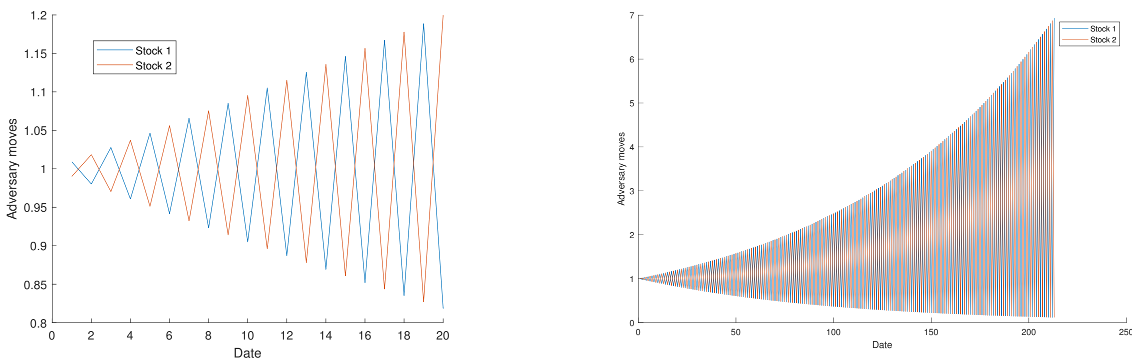
$$R_n \leq \frac{R^2 G\sqrt{n}}{2R} + \frac{G\sqrt{n}R}{2} \tag{5.5}$$

$$R_n \leq \frac{RG\sqrt{n}}{2} + \frac{G\sqrt{n}R}{2} \tag{5.6}$$

$$R_n \leq RG\sqrt{n} \tag{5.7}$$

5.2 5b

The return of two generated stocks are denoted in Figure 7. The wealth created when investing in one or either of the stocks in denoted in Figure 8.



(a) (b)

Figure 7: Two series of numbers ((a) is a zoomed in version of (b)), which are equal but with opposite sign, which represent the returns of two stocks. The decrease of a single stock is higher than its increase, such that when invested in only one stock, your wealth would decrease exponentially.

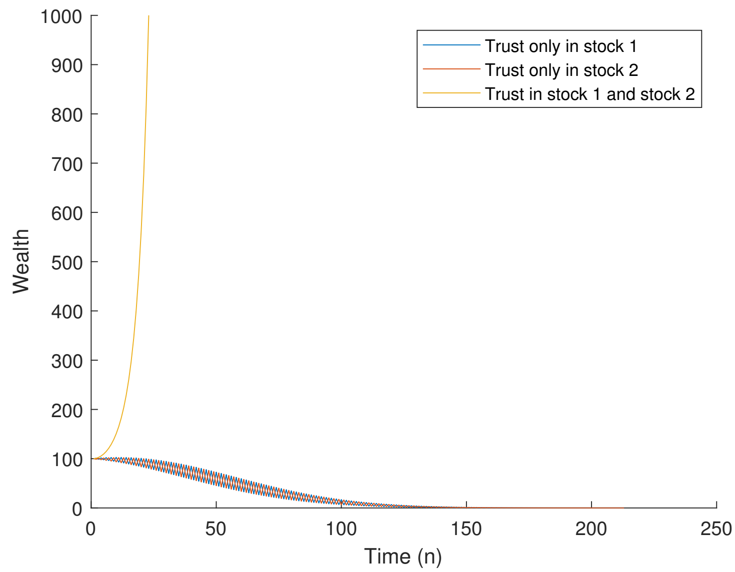


Figure 8: Wealth when investing in a single stock or when investing in both.

It is observed through Figure 8 that investing in a single stock yields an exponential decrease, but when opening the possibility of investing in different stocks, an exponential increase is observed. This is because the decrease of each individual stock is now avoided.

5.3 5c

Algorithm 1 Online Gradient Descent Update Algorithm

Input:
A set of adversary moves $\mathcal{Z}$
a loss function l = $\mathcal{A}$x$\mathcal{Z}$
a closed and convex set $\mathcal{A} \subset R^d$
Initializing:
an initial action $a_0 \in \Delta_d$
for t = 1,2,3...,n **do**
  $W_{t+1} = a_t - \eta \nabla_a l(a_t, z_t)$
  if $W_{t+1} \in \mathcal{A}$ **then**
    $a_{t+1} = W_{t+1}$
  else
    $\prod_{\mathcal{A}}(w_{t+1}) = arg\min_{a \in \mathcal{A}} ||W_{t+1} - a||_2$
  end if
end for

The optimal learning rate is the lowest amount of steps needed to be taken to find the minimum. If the learning rate is taken to large, the steps taken on the level set will be too large and it will be difficult (or impossible) to find the minimum. Taken too small, the algorithm will have to run a lot of times since the step sizes on the level set are small. R and G should be chosen such that the fraction of the learning rate/the step size is appropriate for the size of the data set.

6 Appendix

Listing 5: Code AA algorithm question 4

```
clear all;
load coin_data;

d = 5;
n = 213;

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

% compute strategy p_t (see slides)
time = 213;
L = zeros(time,d);
p = zeros(time,d);
p(1,:) = p(1,:)+0.2


for t = 1:time-1
    C = 0;
    for e = 1:d

        L(t,e) = sum(z_t(1:t,e));

        C = C + exp(-L(t,e));

    end

    for e = 1:d
        p(t+1,e) = exp(-L(t,e))/C;

    end

end

% compute loss of strategy p_t

mix_loss = zeros(time,1);

for t = 1:time

        sum_pz = 0;
        for e = 1:d
            sum_pz = sum_pz + p(t,e) * exp(-z_t(t,e))
        end
        mix_loss(t,1) = -log(sum_pz);

end
    total_mix_loss_p_t = sum(mix_loss);


% compute losses of experts
loss_per_e = sum(z_t)

L_min_total = min(loss_per_e,[],'all');

% compute regret
```

```
 Regret_pt = total_mix_loss_p_t - L_min_total



% compute total gain of investing with strategy p_t
W = zeros(time,1);
W(1) = 100;
for t = 1:time
    W(t+1,:) = p(t,:)*r(t,:)'*W(t);
end
```