Machine Learning Assignment 3

Que Shuhao(4739124)

March 27, 2018

Exercise 1

(a)

1. *Strategy A*
   t = 1:

   choose $P_1 = (1/3, 1/3, 1/3)$
   $Z_1 = (0, 0.1, 0.2)$

   t = 2:

   choose $P_2 = (1, 0, 0)$
   $Z_2 = (0, 0, 0.1)$
   t = 3:

   choose $P_3 = (1, 0, 0)$
   $Z_3 = (1, 0, 0)$
   t = 4:

   choose $P_4 = (0, 1, 0)$
   $Z_4 = (0, 0.9, 0)$

2. *Strategy B*
   According to the following equation:
   $P_t^i = e^{-L_{t-1}^i}/C_{t-1}$
   where
   $L_t^i = \sum_{s=1}^t Z_s^i$
   $C_{t-1} = \sum_{j=1}^d e^{-L_{t-1}^j}$
   I define the actions P_t as follows:

   t = 1:

   choose $P_1 = (1/3, 1/3, 1/3)$
   t = 2:

   choose $P_2 = (0.367, 0.332, 0.301)$
   t = 3:

   choose $P_1 = (0.378, 0.342, 0.280)$
   t = 4:

   choose $P_1 = (0.183, 0.449, 0.368)$

(b)

1. *Strategy A*
   According to the following equation:
   $l_m(P_1, Z_1) = -ln(\sum_{i=1}^{d} P_t^i * e^{-Z_t^i})$
   It can be derived that the total mix loss is:
   $l_m(total) = l_m(P_1, Z_1) + l_m(P_2, Z_2) + l_m(P_3, Z_3) + l_m(P_4, Z_4) = 0.097 + 0 + 1 + 0.9 = 1.997$
   and the corresponding expert regret is:
   $R_n^E = \sum_{t=1}^{4} l_m(P_t, Z_t) - \min_i \sum_{t=1}^{4} Z_t^i = 1.997 - 0.3 = 1.697$

2. *Strategy B*
   According to the following equation:
   $l_m(P_1, Z_1) = -ln(\sum_{i=1}^{d} P_t^i * e^{-Z_t^i})$
   It can be derived that the total mix loss is:
   $l_m(total) = l_m(P_1, Z_1) + l_m(P_2, Z_2) + l_m(P_3, Z_3) + l_m(P_4, Z_4) = 0.097 + 0.029 + 0.273 + 0.310 = 0.709$
   and the corresponding expert regret is:
   $R_n^E = \sum_{t=1}^{4} l_m(P_t, Z_t) - \min_i \sum_{t=1}^{4} Z_t^i = 0.709 - 0.3 = 0.409$

(c)

$R_n^E \leq ln(d) \quad \Rightarrow$
$R_n^E = \sum_{t=1}^{4} l_m(P_t, Z_t) - \min_i \sum_{t=1}^{4} Z_t^i \leq ln(d) \quad \Rightarrow$
$\sum_{t=1}^{4} l_m(P_t, Z_t) \leq ln(d) + \min_i \sum_{t=1}^{4} Z_t^i$
According to this example, d = 3:
$ln(d) + \min_i \sum_{t=1}^{4} Z_t^i = ln(3) + 0.3 = 1.399 \Rightarrow C = 1.399$
for Strategy A:
$\sum_{t=1}^{4} l_m(P_t, Z_t) = 1.997 > ln(3)$
for Strategy B:
$\sum_{t=1}^{4} l_m(P_t, Z_t) = 0.709 < ln(3)$

(d)

Assum for any d, $P_t = (p_1, p_2, ..., p_d)$
$Z_t = (z_1, z_2, ..., z_d)$ and $\min_i Z_t = 0$
$R_n^E = -ln(P_t * Z_t^T)$
$= -ln(\sum_{i=1}^{d} p_i z_i) \geq ln(d)$
$\Rightarrow ln(\frac{1}{\sum_{i=1}^{d} p_i z_i}) \geq ln(d)$
$\Rightarrow \frac{1}{\sum_{i=1}^{d} p_i z_i} \geq d$
$\Rightarrow \sum_{i=1}^{d} p_i z_i \leq \frac{1}{d}$
For any d, we can find Z_t that holds for $p_i z_i \leq \frac{1}{d^2}$ for all $i \in [1, d]$ as long as $z_i \leq \frac{1}{d^2 p_i}$ holds which is totally possible.

Exercise 2

(a)

$R_2^A \leq \frac{2*\eta_A}{8} + \frac{ln(d)}{\eta_A} \Rightarrow R_2^A \leq \sqrt{ln(d)} = C_2^A$
$R_2^B \leq \frac{2*\eta_B}{8} + \frac{ln(d)}{\eta_B} \Rightarrow R_2^B \leq \frac{3\sqrt{2ln(d)}}{4} = C_2^B$
So, C_2^B is tighter for n = 2

(b)

$R_4^A \leq \frac{4*\eta_A}{8} + \frac{ln(d)}{\eta_A} \Rightarrow R_4^A \leq \frac{3}{2}\sqrt{ln(d)} = C_4^A$
$R_4^B \leq \frac{4*\eta_B}{8} + \frac{ln(d)}{\eta_B} \Rightarrow R_4^B \leq \sqrt{2ln(d)} = C_4^B$
This time, C_4^B is still tighter for n = 4

(c)

No. Because $\frac{\partial R_n}{\partial n} = \frac{\eta}{8}$
Therefore, $\frac{\partial R_n^A}{\partial n} \neq \frac{\partial R_n^B}{n}$,
which implies that at different values of n the two regrets reach their own upper bounds. There could exist an "n" at which $R_n^B > R_n^A$.

(d)

$$P_t^i = \frac{e^{-\eta L_{t-1}^i}}{\sum_{j=1}^d e^{-\eta L_{t-1}^i}}$$

$$\lim_{\eta \to \infty} P_t^i = \frac{1}{e^{-\eta(L_{t-1}^{t-2} - L_{t-1}^i)} + e^{-\eta(L_{t-1}^{t-3} - L_{t-1}^i)} + ... + 1}$$

for any $j \neq i$ in d, if there exists:
$L_{t-1}^j - L_{t-1}^i < 0 \Rightarrow L_{t-1}^j < L_{t-1}^i$
then $\lim_{\eta \to \infty} P_t^i = 0$
for all j in d, $j \neq i$ holds:
$L_{t-1}^j - L_{t-1}^i > 0 \Rightarrow L_{t-1}^j > L_{t-1}^i$
which implies that expert i is the best expert, then $\lim_{\eta \to \infty} P_t^i = 1$

Exercise 3

(a)

$$l(a_t, z_t) - l(a, z_t) \leq \nabla_{a_t}^T l(a_t, z_t)(a_t - a)$$

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

$$\| a_{t+1} - a \|^2 = \| \prod_A (a_t - \eta \nabla_{a_t} l(a_t, z_t)) - a \|^2$$

$$\leq \| a_t - \eta \nabla_{a_t} l(a_t, z_t) - a \|^2$$

$$= \| a_t - a \|^2 - 2\eta \nabla_{a_t}^T l(a_t, z_t)(a_t - a) + \| \eta \nabla_{a_t} l(a_t, z_t) \|^2$$

$$\Rightarrow 2\eta \nabla_{a_t}^T l(a_t, z_t)(a_t - a) \leq \| a_t - a \|^2 - \| a_{t+1} - a \|^2 + \| \eta \nabla_{a_t} l(a_t, z_t) \|^2$$

$$\Rightarrow \sum_{t=1}^{n} 2\eta \nabla_{a_t}^T l(a_t, z_t)(a_t - a) \leq \sum_{t=1}^{n} \| a_t - a \|^2 - \| a_{t+1} - a \|^2 + \| \eta \nabla_{a_t} l(a_t, z_t) \|^2$$

$$= \| a_t - a \|^2 - \| a_{t+1} - a \|^2 + \sum_{t=1}^{n} \eta^2 \| \nabla_{a_t} l(a_t, z_t) \|^2$$

$$\leq R^2 + n^2 G^2$$

$$(\| a \| \leq R, \| \nabla_a l(a, z) \| \leq G)$$

$$\Rightarrow \sum_{t=1}^{n} \nabla_{a_t}^T l(a_t, z_t)(a_t - a) \leq \frac{R^2 + n\eta^2 G^2}{2\eta}$$

$$\Rightarrow R_n \leq \frac{R^2}{2\eta} + \frac{nG^2\eta}{2}$$

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

$$\Rightarrow R_n \leq RG\sqrt{n}$$

(b)

As shown in (a), we have:
$\sum_{t=1}^{n} 2\nabla_{a_t}^T \eta_t l(a_t, z_t)(a_t - a) \leq \| a_1 - a \|^2 - \| a_{n+1} - a \|^2 + \sum_{t=1}^{n} \| \nabla_{a_t} \eta_t l(a_t, z_t) \|^2$
Now divide both sides of the inequality by η_t, we have:
$\sum_{t=1}^{n} 2\nabla_{a_t}^T l(a_t, z_t)(a_t - a) \leq \frac{1}{\eta_t} \| a_1 - a \|^2 - \frac{1}{\eta_t} \| a_{n+1} - a \|^2 + \sum_{t=1}^{n} \eta_t \| \nabla_{a_t} l(a_t, z_t) \|^2 \leq \frac{1}{\eta_t} \| a_1 - a \|^2 + \sum_{t=1}^{n} \eta_t \| \nabla_{a_t} l(a_t, z_t) \|^2$
Since we have $\| a \|_2 \leq R$ and $\| nablal(a, z) \|_2 \, leqG$
We can get:
$\sum_{t=1}^{n} 2\nabla_{a_t}^T l(a_t, z_t)(a_t - a) \leq \frac{1}{\eta_t} R^2 + nG^2 \sum_{t=1}^{n} \frac{1}{\eta_t}$
Given that $\eta_t = \frac{R}{G\sqrt{t}}$, we have $\frac{1}{\eta_t} = \frac{G\sqrt{t}}{R} \leq \frac{G\sqrt{n}}{R}$ and $\sum_{t=1}^{n} \frac{1}{\sqrt{t}} = \frac{1}{2\sqrt{n}}$
Therefore, we have:
$\sum_{t=1}^{n} 2\nabla_{a_t}^T l(a_t, z_t)(a_t - a) \leq RG\sqrt{n} + RG\sqrt{n}\frac{1}{2} = \frac{3}{2} RG\sqrt{n}$

Exercise 4

(a)

$W_{t+1} = W_t \sum_{i=1}^{d} P_t^i r_t^i$

(b)

$\sum_{t=1}^{n} -ln(\sum_{i=1}^{d} P_t^i e^{-Z_t^i}) = \sum_{t=1}^{n} -ln(\sum_{i=1}^{d} P_t^i r_t^i) = \sum_{t=1}^{n} -ln\frac{W_{t+1}}{W_t}$
$= -ln(\prod_{t=1}^{n} \frac{W_{t+1}}{W_t}) = -ln(\frac{W_{t+1}}{W_1})$

The mix loss for investment is:
$l_m(P_t, z_t) = -ln(\sum_{i=1}^{d} P_t^i e^{-Z_t^i})$. For day t, the cumulative mix loss is the left side of the equation I just deduced. If the mix loss is small, then $-ln(\frac{W_{t+1}}{W_1})$ will also be small, then $\frac{W_{t+1}}{W_1}$ will be large, which implies that the wealth we get on day t+1 will be larger than that we get on day 1. Therefore, the total mix loss should be as small as possible.

(c)

1. Check the code
2. According to the implemented code, the total loss of each expert is (-1.1530, -1.3649, -1.3249, -1.5774, -1.6543) respectively.
3. According to the implemented code, the expert regret of AA is 0.2232.
4. The total loss of AA, i.e. the mix loss of AA, is -1.4311, while the total loss of experts are (-1.1530, -1.3649, -1.3249, -1.5774, -1.6543) respectively. Clearly, the total losses of experts e_1, e_2, e_3 are all smaller than the total loss of AA, while those of experts e_4, e_5 are larger. The expert regret of AA is 0.2232, while the guaranteed expert regret is ln(5)=1.6094. Therefore, the expert regret of AA is smaller than the guaranteed one. The adversary is not generating difficult data because the expert regret of AA is way smaller than the guaranteed one.
5. According to the Figure 1, we can observe that the values of all the coins expert for BTC experience only slight increases of values through 213 days, while the coin BTC experiences unstable and tremendous changes of value through the same time interval. Although the confidences pt of all the coins experience remarkable changes, there is no explicit correlation between their confidences and the change of their values. The strategy basically follows the principle of assigning different weights to different coins on each day t according to their cumulative mix loss of previous t-1 days to guarantee a threshold for the total mix loss.
6. If we would have invested according to the AA strategy, our wealth would have increased 4.1833.

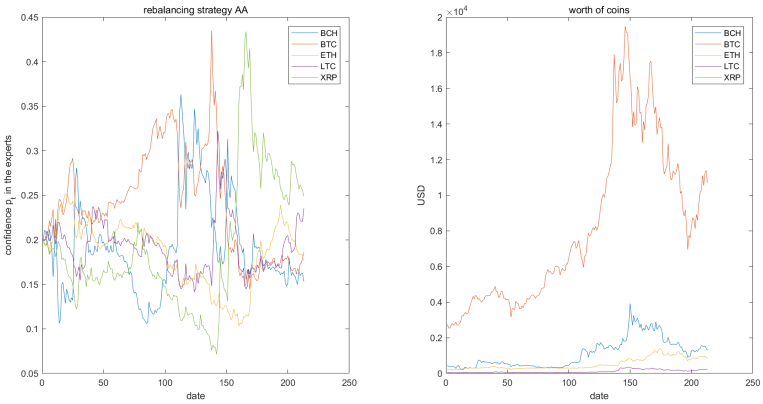

Figure 1: Visualization of strategy AA and values of coins

(d)

1. The second component of the equation to calculate the OCO regret is different from that of the equation to calculate the expert regret considered above. In terms of OCO regret, the minimum total loss is calculated from the optimal fixed action which is not necessarily from the action set $(a_1..a_{213})$ defined in OGD process.

2. We know that $lm(a, z_t) = -ln(a^T r_t)$
   So we have $\frac{\partial lm}{\partial a} = -\frac{\frac{\partial a^T r_t}{a^T}}{a^T r_t}$
   $\frac{\partial a^T r_t}{\partial a^T} = r_t$
   Therefore, we have $\nabla_a lm(a, z_t) = -\frac{r_t}{a^T r_t}$

3. Check the code.

4. Check the code.

5. In the code, R is defined as 1. Because $\| a \|_2 \leq 1$. G is defined as the maximum gradient magnitude, due to which the value of G in the code is assigned by the maximum r magnitude divided by the minimum r magnitude.

6. As shown in Figure 2. If we have used OGD strategy to invest, our wealth would have increased by 6.1428.

7. Check the code.

8. The total loss of OGD is -1.8153, while the loss of best fixed action is -1.4357 which is bigger than the total loss of OGD. The OCO regret of OGD is -0.3796, while the guaranteed OCO regret is 65.7803. Therefore, the adversary is not generating difficult data.

9. When $r_t^i = 0$ happens, the assumption that $\| \nabla l(a, z) \|_2 \leq G$ will be violated since $Z_t^i = -ln(r_t^i)$. In the case of $r_t^i = 0$, the loss Z_t^i will be close to ∞, which implies that it is impossible that an upper boundary G will exist.

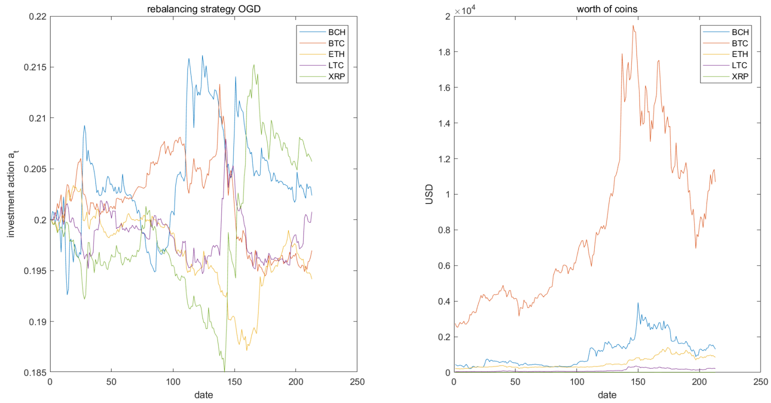

Figure 2: Visualization of strategy OGD and values of coins

Matlab Code

AA.m

```
% Exercise: Aggregating Algorithm (AA)

clear all;
coin_data = load('coin_data.mat');
coin_data = struct2cell(coin_data);

s0 = coin_data{4,1};
s = coin_data{3,1};
r = coin_data{2,1};
symbols_str = coin_data{5,1};

d = 5;
n = 213;

for t=1:n
% compute adversary movez z_t
        z_t(t,:) = -log(r(t,:));
% compute strategy p_t (see slides)
    if t == 1
        % compute cumulative losses of experts --> L_t
        L_t(t,:) = z_t(t,:);
        p_t(t,:) = ones(1,5).*0.2;
    else
        L_t(t,1) = sum(z_t(1:t,1));
        L_t(t,2) = sum(z_t(1:t,2));
        L_t(t,3) = sum(z_t(1:t,3));
        L_t(t,4) = sum(z_t(1:t,4));
        L_t(t,5) = sum(z_t(1:t,5));
```

```
            C_tm1 = sum( exp(-L_t(t-1,:)) );
            p_t(t,1) = exp(-L_t(t-1,1))./C_tm1;
            p_t(t,2) = exp(-L_t(t-1,2))./C_tm1;
            p_t(t,3) = exp(-L_t(t-1,3))./C_tm1;
            p_t(t,4) = exp(-L_t(t-1,4))./C_tm1;
            p_t(t,5) = exp(-L_t(t-1,5))./C_tm1;
        end
end
%total loss of the experts
for i = 1:d
      loss_E(i) = L_t(213,i);
end
% compute loss of strategy p_t --> l_m
      [l_m, g] = mix_loss(p_t', r');

% compute regret --> Rn
  min_z_t = sum(z_t(:,1));
  min_d = 1;

  for j = 1:d
      if min_z_t >= sum(z_t(:,j))
          min_z_t = sum(z_t(:,j));
          min_d = j;
      end
  end

  Rn = sum(l_m) - min_z_t;

% compute total gain of investing with strategy p_t --> w_t/w_1
for t = 1:n
     w_tOverw_t(t,1) = sum(p_t(t,:).*r(t,:));
end
totalGain = prod(w_tOverw_t);
%% 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_t)
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')
```

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 normalized
%       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

    %%% your code here %%%44
    l = -log(dot(a,r));
    g = -r/dot(a,r);
end
```

OGD.m

```
% Exercise: Online Gradient Descent (OGD)
close all;
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
totalLoss = sum(L);

% compute total gain in wealth
totalGain = exp(-totalLoss);
%Alternative method to calcultae totalGain
% Atry = A'
% for t = 1:n
%     w_tOverw_t(t,1) = sum(Atry(t,:).*r(t,:));
% end
% totalGain = prod(w_tOverw_t);

% compute best fixed strategy (you may make use of loss_fixed_action.m and opti
cvx_begin
    variable a_fixed(1,5);
    Vec = [a_fixed; zeros(212,5)];
    minimize( norm((sum(Vec*r'))) ); %the mix loss is defined as lm = -ln(a), t
    subject to
        sum(a_fixed) == 1;
        a_fixed(:) >= 0;
cvx_end

% compute regret
[loss_fixed,g] = loss_fixed_action(a_fixed);
Rn = totalLoss - loss_fixed;

%% 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')
```