IE 613: Assignment 1

Manan Doshi
140100015

20 February 2018

Question 1

We see that increasing η in the given range decreases the expected regret. A higher *eta* implies more aggressive exploitation. Note that the error bars for 95% confidence are rawn but are tiny enough to not appear on the plot.

Code

```
import numpy as np
import matplotlib.pyplot as plt

def wma(d,T,eta):
    w_tilde = np.ones([d])
    l       = np.zeros([d,T])
    loss    = 0
    e_loss  = 0
    for t in range(T):
        w          = w_tilde/np.sum(w_tilde)
        adv_choice = np.random.choice(d,p=w)
        l[:-2,t]   = np.random.choice(2, size=8, p=[0.5, 0.5])
        l[-2,t]    = np.random.choice(2, p=[0.6,0.4])
        delta      = 0.1 if t<T/2 else -0.2
        l[-1,t]     = np.random.choice(2, p=[0.5-delta,0.5+delta])
        loss      += l[adv_choice,t]
        e_loss    += w.dot(l[:,t])
        w_tilde    = w_tilde*np.exp(-eta*l[:,t])

    costs    = np.sum(l,axis=1)
    regret   = loss - np.min(costs)
    p_regret = e_loss - np.min(costs)

    return p_regret

d = 10      #Number of advisors
T = 100000  #Number of rounds

c         = np.linspace(0.1,2.1,11)
Eta       = c*np.sqrt(2.0*np.log(d)/T)
n_samples = 30
R         = np.zeros([11,n_samples])
```

```
for i,eta in enumerate(Eta):
    for trial in range(n_samples):
        R[i,trial] = wma(d,T,eta)
        print("Sample: {}, i_c:{}".format(trial,i))

m, s   = np.mean(R, axis=1), np.std(R, axis=1, ddof=1)*1.96/np.sqrt(n_samples)

fig,ax = plt.subplots(figsize=(15,15))
ax.errorbar(c,m,s)
ax.set_xticks(c)
ax.tick_params(axis='both', labelsize=15)
ax.set_xlabel(r"$\frac{\eta}{\sqrt{\frac{2\log(d)}{T}}}$", \\
                fontsize=40, labelpad=20)
ax.set_ylabel(r"Expected regret", fontsize=20, labelpad=30)
plt.savefig(r"./plots/q1.png")
plt.show()
```

Plots

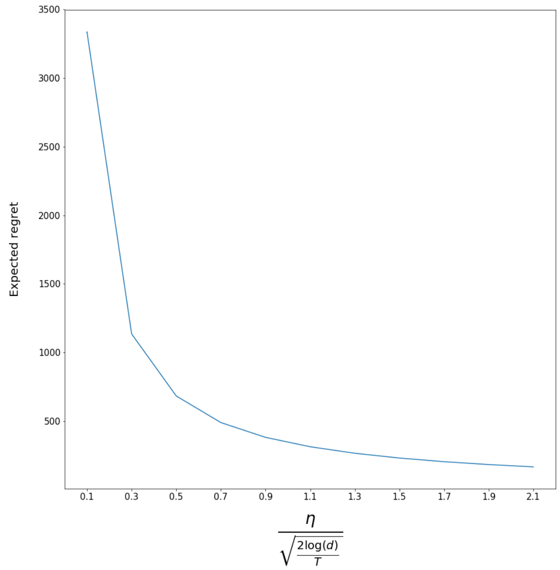

Figure 1: Variation of expected regret with η for the weighted majority algorithm

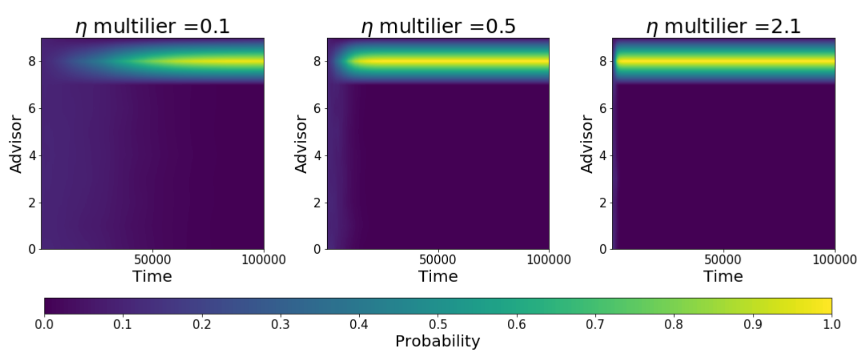

Figure 2: Probability contours with advisors and time. Higher η leads to more aggressive exploitation. The algorithm is not able to switch over to the better advisor after $T/2$ due to lack of exploration

Question 2

Code

EXP3

```
def exp3(d,T,eta):
    e_loss = 0
    elv = 0.5*np.ones([d,2])
    elv[-2,:] = 0.4
    elv[-1,:] = [0.6,0.3]
    w_tilde = np.ones([d])

    for t in range(T):
        w = w_tilde/np.sum(w_tilde)
        adv_choice = np.random.choice(d,p=w)
        e_loss_c    = elv[adv_choice,(2*t)//T]
        l           = np.random.choice(2,\
                        p=[1-e_loss_c, e_loss_c])/w[adv_choice]
        e_loss     += e_loss_c
        w_tilde[adv_choice]     = w_tilde[adv_choice]*np.exp(-eta*l)

    return e_loss - 0.4*T
```

EXP3.P

```
def exp3p(d,T,eta,beta,gamma):
    e_gain = 0
    elv = 0.5*np.ones([d,2])
    elv[-2,:] = 0.4
    elv[-1,:] = [0.6,0.3]
    egv = 1-elv
    G = np.zeros([d])
    w_tilde = np.ones([d])

    for t in range(T):
        w                  = (1-gamma)*(w_tilde/np.sum(w_tilde)) + gamma/d
        adv_choice         = np.random.choice(d,p=w)
        e_gain_c           = egv[adv_choice,(2*t)//T]
        gain               = beta/w
        gain[adv_choice]  += np.random.choice(2,\
                              p=[1-e_gain_c, e_gain_c])/w[adv_choice]
        e_gain            += e_gain_c
        w_tilde            = w_tilde*np.exp(eta*gain)

    return 0.6*T - e_gain
```

EXP3-IX

```python
def exp3ix(d,T,eta,gamma):
    e_loss = 0
    elv = 0.5*np.ones([d,2])
    elv[-2,:] = 0.4
    elv[-1,:] = [0.6,0.3]
    w_tilde = np.ones([d])

    for t in range(T):
        w = w_tilde/np.sum(w_tilde)
        adv_choice = np.random.choice(d,p=w)
        e_loss_c    = elv[adv_choice,(2*t)//T]
        l           = np.random.choice(2,\
                       p=[1-e_loss_c, e_loss_c])/(w[adv_choice]+gamma
    )
        e_loss     += e_loss_c
        w_tilde[adv_choice]     = w_tilde[adv_choice]*np.exp(-eta*l)

    return e_loss - 0.4*T
```

Plot

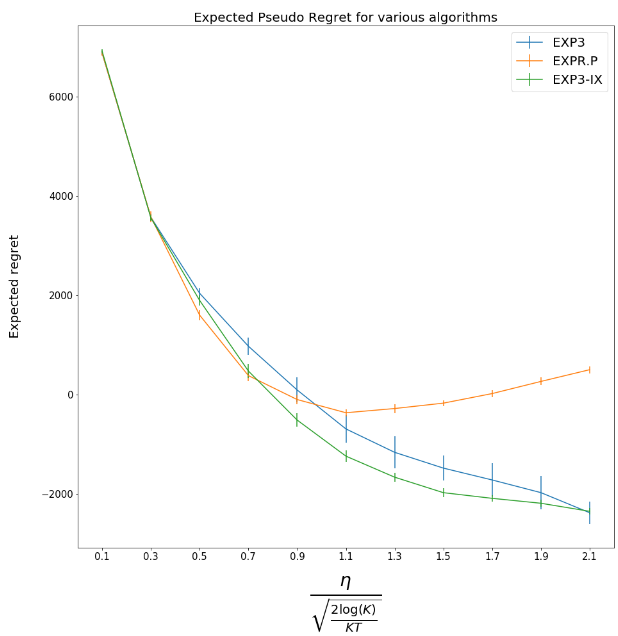

Figure 3: Variation of expected regret with η multiplier

Question 3

Clearly, `EXP3-IX` has the best performance with lower expected regret and lower deviation. The good performance of `EXP3-IX` can be attributed to the fact that it explores and detects the new best advisor after the change in odds at $\frac{T}{2}$. This exploration is not possible in `EXP3`. The bad performance of `EXP3.P` can be attributed to very high exploration rates leading to low exploitation of the current best adviser. This behavious can be clearly seen in the following plot

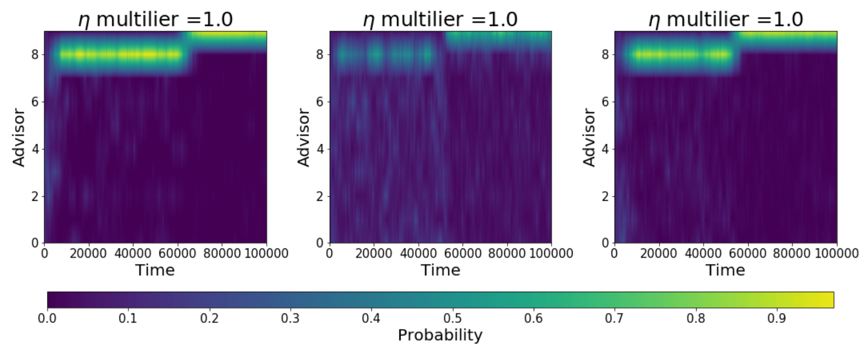

Figure 4: Probability contours with advisors and time for `EXP3`, `EXP3.P` and `EXP3-IX` respectively. The weak exploitation of `EXP3.P` and the slow switching of `EXP3` is apparent here.

Question 4 [1]

The proposed algorithm is the same as the wighted majority algorithm

Algorithm 1: The Weighted Majority Algorithm

Input : Hypothesis class $\mathcal{H}$
Parameter: $\eta \in [0,1]$
Initialize : $\tilde{w}^{(1)} = [1,1,1,...,1]$ in $\mathbb{R}^d$
for $t \leftarrow 1$ **to** T **do**
- Set $w_i^{(t)} = \frac{\tilde{w}_i^{(t)}}{\sum_i \tilde{w}_i^{(t)}}$
- Play i according to the distribution $w^{(t)}$
- Receive loss vector $l_t = \{l_{t,i} : \forall i \in d\}$ where $l_{t,i}$ is the error in prediction of hypthesis h_i
- Update $\forall i, \tilde{w}_i^{(t+1)} = \tilde{w}_i^{(t)} e^{-\eta l_{t,i}}$

We will compute a finite bound for expected number of mistakes of this algorithm on a realizable case with Bernoulli noise. We first make the claim that

$$\mathbb{E}\left[\sum_{s=t+1}^{T} \|\hat{y}_s - f_i^s\|\right]_{w_t} \leq C_\gamma \ln\left(\frac{Z_t}{w_i^t}\right) \tag{1}$$

where i refers to the 'correct' hypothesis. $Z_t = \sum_{i=1}^{d} w_i^t$ and $C_\gamma = \frac{1}{1-2\sqrt{\gamma(1-\gamma)}}$.

We will now prove the above claim using induction. The base case at $t = T$ is trivial since the LHS is 0 and the right side is positive (since $Z_t \geq w_i^t$). We will now split our hypothesis class into two groups based on whether the hypothesis classifies the round at t correctly.

$$u = \sum_{j, f_j^t = f_i^t} w_j^{t-1} \qquad\qquad v = \sum_{j, f_j^t / neq f_i^t} w_j^{t-1}$$

u is thus the total weight of the correct classifiers for the round and v is the total weight of the incorrect classifiers. Probability that the algorithm classifies incorrectly is thus $\frac{v}{Z_{t-1}}$. There is also a chance that the system sends incorrect feedback, say with probability $p \leq \gamma$. If the feedback is incorrect the weight update is $Z_t = e^{-\eta}u + v$ and the update of the weight of the correct hypothesis class is $w_i^t = e^{\eta} w_i^{t-1}$. If the system send correct feedback (with probability $1-p$) and the weight of the correct hypothesis remains unchanged. Expected mistakes from t to T equals the expected number of mistakes at t plus the expected number of mistakes from $t+1$ to T. This leads us to

$$\begin{aligned}
\mathbb{E}\left[\sum_{s=t}^{T} \|\hat{y}_s - f_i^s\|\right]_{w_{t-1}} &= \mathbb{E}\left[\sum_{s=t+1}^{T} \|\hat{y}_s - f_i^s\|\right]_{w_t} + \frac{v}{Z_{t-1}} \\
&\leq \frac{v}{Z_{t-1}} + \mathbb{E}\left[C_\gamma \ln\left(\frac{Z_t}{w_i^t}\right)\right]_{w_t} \\
&= \frac{v}{Z_{t-1}} + p\left[C_\gamma \ln\left(\frac{e^{-\eta}u + v}{e^{-\eta}w_i^{t-1}}\right)\right] + (1-p)\left[C_\gamma \ln\left(\frac{u + e^{-\eta}v}{w_i^{t-1}}\right)\right]
\end{aligned}$$

We will show that the last expression is bounded by the RHS of (1). This involves mathematical manipulations given in the appendix of [1]. Once we have proved (1) we can substitute $t = 0$ to get an upper bound for expectation of number of mistakes.

$$\boxed{\mathbb{E}\left[\sum_{s=1}^{T} \|\hat{y}_s - f_i^s\|\right]_{w_t} \leq C_\gamma \ln(d)}$$

Question 5

Consider an algorithm A whose regret bound for T rounds is $\alpha\sqrt{T}$. For 2^m rounds, the regret bound will be $\alpha\sqrt{2^m}$. Since we do not know the time horizon, we break the time period into pieces of size 2^m where $m = 0, 1, 2, \cdots$. We choose the parameter η in terms of these smaller time periods for every packet.

If the total time horizon is T and the total number of 'packets' is k,

$$\begin{aligned}
\sum_{m=0}^{k-1} 2^m + 1 \quad &\leq T \leq \quad & \sum_{m=0}^{k} 2^m \\
1 + 1 + 2 + 2^2 + 2^3 \cdots + 2^k \quad &\leq T \leq \quad & 1 + 2 + 2^2 + 2^3 \cdots + 2^k \\
2^k \quad &\leq T \leq \quad & (2^{k+1} - 1)
\end{aligned}$$

For a time period of 2^m, regret is $\alpha 2^{\frac{m}{2}}$. Total regret:

$$\begin{aligned}
\mathcal{R} &\leq \sum_{m=0}^{k} \alpha 2^{\frac{m}{2}} \\
&\leq \alpha \left(1 + \sqrt{2} + \sqrt{2}^2 + \cdots + \sqrt{2}^k\right) \\
&\leq \alpha \left(\frac{2^{\frac{k+1}{2}} - 1}{\sqrt{2} - 1}\right) \\
&\leq \alpha \left(\frac{\sqrt{2T} - 1}{\sqrt{2} - 1}\right) \\
&\leq \alpha \left(\frac{\sqrt{2T}}{\sqrt{2} - 1}\right)
\end{aligned}$$

$$\boxed{\mathcal{R} \leq \left(\frac{\sqrt{T}}{\sqrt{2} - 1}\right) \alpha\sqrt{T}}$$

References

[1] Shai Ben-David, Dávid Pál, and Shai Shalev-Shwartz. Agnostic online learning. 2009.