

CS 49/149: 21st Century Algorithms (Fall 2018): Lecture 2

Date: 18th September, 2018

Topic: Online Decision Making

Scribe: Kevin Tan

Disclaimer: These notes have not gone through scrutiny and in all probability contain errors. Please email errors to kevin.w.tan.19@dartmouth.edu and/or deeparnab@dartmouth.edu.

1 Online Decision Making

In this setting we have m actions available to choose from. At each time step, we want to select a distribution \vec{p} over the set of actions. We define the loss function $\ell_i(t)$ as action i 's loss at a given timestep t . Let $\ell_i(t) \in [-1,1]$. If we sum up the losses across all the actions at any given time-step, we get

$$alg_t = Exp[loss_t] = \sum_{i=1}^m p_i(t) \cdot \ell_i(t)$$

If we sum up alg_t across all time-steps and applying the linearity of expectation, we get

$$ALG = Exp[loss_A] = \sum_{t=1}^T alg_t = \sum_{t=1}^T \sum_{i=1}^m p_i(t) \cdot \ell_i(t)$$

We set our benchmark to be the best possible strategy in hindsight which is not allowed to change over time. That is to say, we sum up the loss of each individual action and pick the action that gives us the lowest loss.

$$OPT = \min_q \sum_{i=1}^m (q_i \cdot \sum_{t=1}^T \ell_i(t))$$

Remark on the optimum strategy: Optimum strategy \vec{q} can be assumed to be a point distribution, i.e. $q_i = 1$ for some i and $q_i = 0$ all other i 's.

Why is this the case? Assume there are only two actions ($m = 2$) and $\sum_{t=1}^T \ell_1(t) = 100$ and $\sum_{t=1}^T \ell_2(t) = 200$. The optimum strategy would be to put all the weights on the first asset since the weights on each action cannot change over time. Putting any weight at all on the second asset would make the strategy non-optimal. We can see the logic of putting all the weight on the action that produce the smallest total loss over time continues to hold as we add more actions. Therefore the optimum strategy must be a point distribution.



Question: *If we change the benchmark to be the best possible strategy in hindsight that is allowed to change over time, is there an upper bound to how much worse the expected loss can be?*

MULTIPLICATIVE WEIGHT UPDATE (MWU)

- Initialize weights $w_i(t)$ for each action with $w_i(1) = 1$ for all i from 1 to m .
- On days $t = 1, \dots, T$:

– We set

$$p_i = \frac{w_i}{\sum_{i=1}^m w_i(t)} = \frac{w_i}{z_t}$$

– Then we receive $\vec{l}(t)$. We then set:

$$w_i(t+1) = w_i(t) \cdot (1 - \eta \cdot \ell_i(t))$$

where η is a parameter set to be less than equal to 1. Note that with this update function, any weight will never be 0.

1.1 Kullback-Leibler divergence

How do we analyze "distance" between two probability distributions? We can use the Kullback-Leibler divergence (KL divergence). A KL divergence can be defined over two probability distributions over the set of all actions. Define x and y as any two arbitrary probability distributions at any time step with the constraint that $x_i \geq 0, i \in m$. We can define the "distance" between the distributions as:

$$D(x||y) = \sum_{i=1}^m x_i \cdot \ln \frac{x_i}{y_i}$$

where $D(x||y)$ is defined only when $y_i > 0, \forall i \in m$ and $0 \cdot \ln 0 = 0$.

We will not go into the proof of the following statements but we know the following identities to be true. We will be using them later in the analysis of the MWU algorithm.

$$D(x||y) \geq 0 \tag{1}$$

$$D(x||y) \leq \ln m \tag{2}$$

1.2 Does $D(q||p(t))$ fall as t increases?

One test we can do to check if MWU is working is to check if $D(q||p(t))$ decreases at each time step. That is to say, we would like to see $p(t)$ converging to q as more time passes. Before we begin

the analysis of the algorithms, the following are useful identities to remember:

$$\begin{aligned}
z(t+1) - z(t) &= \sum_{i=1}^m (w_i(t) - w_i(t)) \\
&= \sum_{i=1}^m ((1 - \eta \ell_i(t) \cdot w_i(t) - w_i(t)) \\
&= -\eta \sum_{i=1}^m \ell_i(t) \cdot w_i(t) \\
&= -\eta \cdot z(t) \cdot \frac{\sum_{i=1}^m w_i(t)}{z(t)} \cdot \ell_i(t) \\
&= -\eta \cdot z(t) \cdot \sum_{i=1}^m (p_i(t) \cdot \ell_i(t)) \\
&= -\eta \cdot z(t) \cdot \text{alg}(t)
\end{aligned}$$

Therefore,

$$z(t+1) = z(t) \cdot (1 - \eta \cdot \text{alg}(t)) \quad (3)$$

Another inequality to keep in mind:

$$\ln(1-x) \geq -x - x^2, \quad (4)$$

if $|x| \leq 1$

Now, let us calculate the change in "distance" between any 2 successive time-steps:

$$\begin{aligned}
&D(q||p(t)) - D(q||p(t+1)) \\
&= \sum_{i=1}^m q_i \cdot \ln \frac{q_i}{p_i(t)} - \sum_{i=1}^m q_i \cdot \ln \frac{q_i}{p_i(t+1)} \\
&= \sum_{i=1}^m q_i \cdot \ln \frac{p_i(t+1)}{p_i(t)} \\
&= \sum_{i=1}^m q_i \cdot \ln \left(\frac{w_i(t+1)}{z_i(t+1)} \cdot \frac{z_i(t)}{w_i(t)} \right) \\
&= \sum_{i=1}^m q_i \cdot \left(\ln \frac{w_i(t+1)}{w_i(t)} + \ln \frac{z(t)}{z(t+1)} \right) \\
&= \ln \frac{z(t)}{z(t+1)} \cdot \sum_{i=1}^m q_i + \sum_{i=1}^m q_i \cdot \ln(1 - \eta \cdot \ell_i(t))
\end{aligned}$$

Using (3) and because $\sum_{i=1}^m q_i = 1$,

$$= \ln \frac{1}{1 - \eta \cdot \text{alg}_t} + \sum_{i=1}^m q_i \cdot \ln(1 - \eta \cdot \ell_i(t))$$

We know that $|\eta \cdot \ell_i(t)| < 1$ because $\ell_i(t) \in [-1, 1]$ and $\eta < 1$. With that and using (4),

$$\begin{aligned} &\geq \eta \cdot alg_t + \sum_{i=1}^m (q_i \cdot (-\eta \cdot \ell_i(t) - \eta^2 \cdot \ell_i(t)^2)) \\ &= \eta \cdot alg_t - \eta \cdot \sum_{i=1}^m q_i \cdot \ell_i(t) - \eta^2 \cdot \sum_{i=1}^m q_i \cdot \ell_i(t)^2 \end{aligned}$$

We note that $\sum_{i=1}^m q_i \cdot \ell_i(t)$ is the loss for our bench mark and we shall denote it as opt_t . $\sum_{i=1}^m q_i \cdot \ell_i(t)^2$ denotes some additional error term which we shall denote as $error(t)$. Therefore we have:

$$D(q||p(t)) - D(q||p(t+1)) \geq \eta(alg_t - opt_t) - \eta^2 error(t)$$

Note the tension here. On one hand we want the LHS of the equation to be large as it means the MWU is rapidly shrinking the "distance" between the distributions at 2 successive time-steps. Yet on the RHS, we do not want the gap between the alg_t and opt_t to be too big.

If we now apply the summation operator on both sides of the equation for all time periods (note: on the LHS if we were to list out all the terms to do the summation over, we would find that all the terms would cancel out except for the following terms):

$$D(q||P_{initial}) - D(q||P_{final}) \geq \eta \cdot (ALG - OPT) - \eta^2 \cdot \sum_{t=1}^T error(t)$$

Observe that the difference in distance between the initial distribution that MWU starts with and the final distribution that MWU ends with is dependent on the difference between ALG and OPT . However intuitively, the difference between ALG and OPT cannot be too big because the distances of the distribution at the end has to be a positive number.

$$ALG - OPT \leq \frac{D(q||P_{initial}) - D(q||P_{final})}{\eta} + \eta \cdot \sum_{t=1}^T \sum_{i=1}^m q_i \cdot \ell_i(t)^2$$

Since $\ell_i(t) \leq 1$, this means $\ell_i(t)^2 \leq 1$ as well. Since $\ell_i(t)^2 \leq 1$, this means that $\sum_{i=1}^m q_i \cdot \ell_i(t)^2 < 1$ as well. Imposing (1) on $D(q||P_{final})$ and (2) on the $D(q||P_{initial})$, we get:

$$ALG - OPT < \frac{\ln m}{\eta} + \eta \cdot T \tag{5}$$

Here again we have a tension. If there was no $\eta \cdot T$ term, then we would choose η to be infinite so we can do as well as our benchmark distribution at the very end. However, the presence of $\eta \cdot T$ also means that if we choose a larger η , there is the potential for our algorithm to perform more poorly.

To more properly see how $ALG - OPT$ must change as T changes, we define $Reg_A(T) = \frac{ALG - OPT}{T}$. Algorithm A is defined to have vanishing regret if $Reg_A(T) \rightarrow 0$ as $T \rightarrow \infty$.

By (5):

$$\text{Reg}_{MWU}(T) \leq \frac{\ln m}{\eta \cdot T} + \eta \tag{6}$$

Setting the RHS terms to equate to each other:

$$\begin{aligned} \frac{\ln m}{\eta \cdot T} &= \eta \\ \frac{\ln m}{T} &= \eta^2 \\ \eta &= \sqrt{\frac{\ln m}{T}} \end{aligned}$$

Substituting $\eta = \sqrt{\frac{\ln m}{T}}$ back into (6), we will find that $\text{Reg}_{MWU} \leq 2 \cdot \sqrt{\frac{\ln m}{T}}$. From this, we can see that MWU has vanishing regret.

Now let us find out about the average performance of the algorithm. Let $\epsilon = 2 \cdot \sqrt{\frac{\ln m}{T}}$, where ϵ is a parameter chosen by us and is the average error we are willing to tolerate.

$$\begin{aligned} \epsilon &= 2 \cdot \sqrt{\frac{\ln m}{T}} \\ \epsilon^2 &= \frac{4 \ln m}{T} \\ T &= \frac{4 \ln m}{\epsilon^2} \end{aligned}$$

Theorem 1. Therefore, if we want the average performance of the algorithm to be within some ϵ of the best hindsight action's loss, we need to run MWU for $T = O(\frac{\ln m}{\epsilon^2})$ rounds with $\eta = \epsilon/2$



Question: Suppose now our distribution over the actions do not sum up to 1 but the weights on each action $\in [-1, 1]$. How does this change our analysis?

2 Application to Linear Classification

In the classic linear classification problem, we have a similar set up to the online decision making problem. We have a set of training data in the form of T tuples $(\vec{a}(t), b(t))$. Let $\vec{a}(t) \in \mathbb{R}^m$ and $b(t) \in \{-1, 1\}$. Assume that all $a_i(t) \in [-1, 1], \forall i, t$. Assume there exists a hyper-plane that separates all the positive and negative $b(t)$'s with a margin. Let this hyper-plane vector be denoted by \vec{q} , where $\forall i, q_i \geq 0$. We can see that in this problem set up, the coordinates of \vec{a} are the actions in the online decision making, and \vec{q} is the same as the optimal strategy.

Now how do we define the loss function? Observe that by the margin assumption, for the all the positive b , $q^T a(t) \geq \gamma$ and for all the negative b , $q^T a(t) < -\gamma$, where γ is the size of the margin. Now, suppose for a distribution vector $\vec{p}(t)$, if $\vec{p}(t)^T \vec{a}(t) \cdot b(t) < 0$, we know that the probability distribution vector has misclassified a point. Hence, we can define $\vec{a}(t) \cdot b(t) = \vec{\ell}_i(t)$ as our loss function.

$$alg_t = E[loss] = \sum_{i=1}^m p_i(t) \cdot \ell_i(t) = \left(\sum_{i=1}^m p_i(t) \cdot a_i(t) \right) \cdot b(t)$$

If we sum up all the losses over all the time steps, we get:

$$ALG = E[loss_A] = \sum_{t=1}^T \vec{p}(t)^T \vec{\ell}(t) = \sum_{t=1}^T \vec{p}(t)^T \vec{a}(t) \cdot b(t) \quad (7)$$

Similarly, we have:

$$opt_t = \sum_{i=1}^m q_i \cdot \ell_i(t) = \sum_{i=1}^m q_i \cdot a_i(t) \cdot b_i(t) \geq \gamma \quad (8)$$

$$OPT = \sum_{t=1}^T opt_t \geq \gamma \cdot T \quad (9)$$

LINEAR CLASSIFICATION (WINNOWER ALGORITHM)

- Initialize weights $w_i(t)$ for each action with $w_i(1) = 1$ for all i from 1 to m .
- On days $t = 1, \dots, T$:

– We set

$$p_i = \frac{w_i}{\sum_{i=1}^m w_i(t)} = \frac{w_i}{z_t}$$

– Consider the hyper-plane $\vec{p}(t)^T \vec{a}(t) \cdot b(t)$:

- * If $\vec{p}(t)^T \vec{a}(t) \cdot b(t) \geq 0$, then we know that everything has been classified correctly. Maintain the probability vector.
- * Else, we know that there was a misclassification. Hence, we need to update all the weights at the timestep with the following algorithm:

$$w_i(t) = w_i(t) \cdot (1 + \eta \cdot \ell_i(t))$$

We then go back to “Consider the hyper-plane $\vec{p}(t)^T \vec{a}(t) \cdot b(t)$ ”

Notice the update step looks different. Instead of multiplying $w_i(t)$ with $(1 - \eta)$ as per MWU, we want to multiply it by $(1 + \eta)$. This is because the larger $\ell_i(t)$ is, the more we want to upweight it (because if it were negative, it meant we were making a mistake).

2.1 Analysis

If we were to follow the analysis in the previous section of the lecture, we would end up with:

$$ALG - OPT \geq -\left(\frac{\ln m}{\eta} + \eta \cdot T\right)$$

We know that $ALG > 0$ because the algorithm is still running after time T . Since we also know (9), we can form the following inequality:

$$\begin{aligned} 0 - \gamma \cdot T > ALG - OPT &\geq -\left(\frac{\ln m}{\eta} + \eta \cdot T\right) \\ \gamma \cdot T &\leq \frac{\ln m}{\eta} + \eta \cdot T \\ T(\gamma - \eta) &\leq \frac{\ln m}{\eta} \\ T &\leq \frac{\ln m}{\eta \cdot (\gamma - \eta)} \end{aligned} \tag{10}$$

Now suppose I want to find the smallest possible T . That means I need $\frac{\ln m}{\eta \cdot (\gamma - \eta)}$ to be as small as possible. However, since m and γ is fixed, that means I can only change the parameter η . I need to maximize $\eta \cdot (\gamma - \eta)$. Taking the first derivative of $\eta \cdot (\gamma - \eta)$ with respect to η :

$$\begin{aligned} \gamma - 2 \cdot \eta &= 0 \\ \eta &= \frac{\gamma}{2} \end{aligned} \tag{11}$$

Substituting (11) into (10), we find that $T \leq \frac{4 \ln m}{\gamma^2}$. The final equation implies that $T \leq \frac{\ln m}{\eta \cdot (\gamma - \eta)}$. We therefore get the following theorem.

Theorem 2.

Assume that all $a_i(t) \in [-1, 1], \forall i, t$.

Assume $b_i \in \{-1, 1\}$

Assume there exists a hyper-plane that separates all the positive and negative $b(t)$'s with a margin γ and that this hyper-plane \vec{q} has $q_i > 0$ and $\sum_{i=1}^m q_i = 1$.

If $\eta = \frac{\gamma}{2}$, the Winnow algorithm needs less than $\frac{4 \ln m}{\gamma^2}$ training examples before it is able to start correctly classifying points.

2.2 Discussion of assumptions

Note that the assumption that $a_i(t) \in [-1, 1]$ and $\|q(t)\| \leq 1$ are weaker conditions than the perceptron algorithm's assumptions of $\|a(t)\|_2 \leq 1$ and $\|q\|_2 \leq 1$. Hence the results of Theorem 2 can also apply to the analysis of the perceptron algorithm.