

Lecture 21 (11/08/2017): Multiplicative weight update method

Lecturer: Shi Li

Scribe: Minwei Ye

In this lecture we are going to discuss an algorithm that can be served as a general technique solving many other problems including game theory related problems.

21.1 A simple model

There are n experts, indexed by $[n]$. There is an event that happens once every day. The event has two possible outcomes: up and down. Each of the expert predicts outcome of this event in each day. There is an aggregator who wants to predict the event for each day based on the prediction of all these experts had made so far. The aggregator wants to perform as good as the best expert. Specifically, the number of mistakes the aggregator made is within a constant approximation ratio of the number of mistakes the best expert made plus an additive constant.

The following weighted majority algorithm guarantees a $\frac{\ln 2}{\ln 4/3}$ approximation ratio. The ratio can be further improved to $2 + \epsilon$.

Algorithm 1 Weighted majority

```

1:  $w_i^0 \leftarrow 1, \forall i \in [n]$ 
2: for  $t = 1, 2, \dots, T$  do
3:   if  $\sum_{i:\text{predicts up}} w_i^{t-1} \geq \sum_{i:\text{predicts down}} w_i^{t-1}$  then
4:     aggregator predicts "up"
5:   else
6:     aggregator predicts "down"
7:   for every expert  $i$  do
8:     if  $i$  make a mistake then
9:        $w_i^t \leftarrow w_i^{t-1}/2$ 
10:    else
11:       $w_i^t \leftarrow w_i^{t-1}$ 

```

Analysis: Let $\Phi^t = \sum_{i=1}^n w_i^t$.

Let $m_i^t = 1$ if i made a mistake at time t , otherwise $m_i^t = 0$. Let $m^t = 1$ if the aggregator made a mistake at time t , otherwise $m^t = 0$.

There are two ways to estimate Φ^T , one using mistakes of the aggregator for upper bound and the other using mistakes of experts for lower bound. That will lead to an inequality showing the relation between the two:

If $m^t = 1$ then more than a half of Φ^{t-1} will be decreased by a half in the weight updating step.

This means $\Phi^t \leq \Phi^{t-1}/2 + \Phi^{t-1}/4 = \frac{3}{4}\Phi^{t-1}$

If $m^t = 0$ we can only say $\Phi^t \leq \Phi^{t-1}$

In sum,

$$\Phi^T \leq \left(\frac{3}{4}\right)^{\sum_{i=1}^T m^t} \Phi^0 \quad (21.1)$$

On the other hand,

$$\Phi^T = \sum_{i=1}^n w_i^T = \sum_{i=1}^n \left(\frac{1}{2}\right)^{\sum_{i=1}^T m_i^t} \geq \left(\frac{1}{2}\right)^{\sum_{i=1}^T m_i^t} \quad (21.2)$$

for every expert i .

Combining (21.1) and (21.2) we have

$$\begin{aligned} \left(\frac{1}{2}\right)^{\sum_{i=1}^T m_i^t} &\leq \Phi^T \leq \left(\frac{3}{4}\right)^{\sum_{i=1}^T m^t} \Phi^0 = \left(\frac{3}{4}\right)^{\sum_{i=1}^T m^t} n \\ (-\ln 2) \sum_{i=1}^T m_i^t &\leq (-\ln \frac{4}{3}) \sum_{i=1}^T m^t + \ln n && \text{By taking } \ln \\ \sum_{i=1}^T m^t &\leq \frac{\ln 2}{\ln 4/3} \sum_{i=1}^T m_i^t + \frac{\ln n}{\ln 4/3} \end{aligned}$$

The approximation ratio can be further improved to $2(1 + \epsilon)$ by modifying line 9 of the algorithm: instead of halving the weight when the expert make a mistake, we decrease the weight by a factor of $(1 + \epsilon)$. The analysis is left as an exercise.

Furthermore, if we allow randomness for the aggregator in the process of make prediction, the ratio can be improved to be $(1 + \epsilon)$. This result is stated in the next section in the context of a more general expert learning model.

21.2 Prediction with expert's advice game

Below is a description of the game:

for $t = 1 \dots T$: **do**

1. Each expert $i \in [n]$ make some advises.
 2. Aggregator picks some distribution $\vec{p}^t = (p_1^t, p_2^t, \dots, p_n^t)$ over the experts.
 3. Adversary with knowledge of the expert advice and \vec{p}^t , determine a penalty vector $\vec{m}^t \in [-1, 1]^n$
 4. Aggregator observes the penalty and suffers $\vec{p}^t \cdot \vec{m}^t$
-

Notice that here the aggregator doesn't even need to know the advises of the expert. It actually learns from the adversary.

The strategy of the aggregator is the following:

Initially assign each expert i a weight $w_i^0 = 1$. At time t :

- Pick the distribution proportion to the weights, i.e., $\vec{p}_j^t = w_j^{t-1} / \Phi^{t-1}$ where $\Phi^{t-1} = \sum_{i=1}^n w_i^{t-1}$
- After observing the penalty, set $w_i^t = e^{-\epsilon m_i^t} w_i^{t-1}$

Then we have the following theorem:

Theorem 21.1. *For any expert i ,*

$$\frac{1}{T} \sum_{t=1}^T \vec{p}^t \cdot \vec{m}^t \leq \frac{1}{T} \sum_{t=1}^T m_i^t + \frac{\ln n}{T\epsilon} + \epsilon$$

Proof: The proof also makes use of the potential function Φ^T .
On one hand,

$$\begin{aligned} \Phi^t &= \sum_{i=1}^n w_i^t = \sum_{i=1}^n e^{-\epsilon m_i^t} w_i^{t-1} \\ &\leq \sum_{i=1}^n (1 - \epsilon m_i^t + \epsilon^2) w_i^{t-1} \\ &\leq \sum_{i=1}^n (1 + \epsilon^2) w_i^{t-1} + \epsilon \sum_{i=1}^n m_i^t (p_i^t \cdot \Phi^{t-1}) \\ &\leq (1 + \epsilon^2) \Phi^{t-1} - \epsilon \Phi^{t-1} \cdot (\vec{p}^t \cdot \vec{m}^t) \\ &= (1 + \epsilon^2 - \epsilon \cdot (\vec{p}^t \cdot \vec{m}^t)) \Phi^{t-1} \\ &\leq \exp[\epsilon^2 - \epsilon \cdot (\vec{p}^t \cdot \vec{m}^t)] \Phi^{t-1} \end{aligned}$$

Therefore

$$\Phi^T \leq \exp[T\epsilon^2 - \epsilon \sum_{t=1}^T \vec{p}^t \cdot \vec{m}^t] \Phi^0$$

On the other hand, for any expert i

$$w_i^T = \exp[-\epsilon \sum_{t=1}^T m_i^t] \leq \Phi^T \leq \exp[T\epsilon^2 - \epsilon \sum_{t=1}^T \vec{p}^t \cdot \vec{m}^t] \cdot n$$

Taking natural logarithms,

$$-\epsilon \sum_{t=1}^T m_i^t \leq T\epsilon^2 - \epsilon \sum_{t=1}^T \vec{p}^t \cdot \vec{m}^t + \ln n$$

Rearranging the terms and the result follows. ■

If we let $T \geq \frac{\ln n}{\epsilon^2}$, the average number of mistakes of the aggregator is better than any of the experts by an additive error 2ϵ . Furthermore, we can relax the condition on the range of the penalty vector to $[-\rho, \rho]$. This will only add a ρ^2 multiplicative term on T in order to achieve the same additive error. We summarize the result in the following corollary.

Corollary 21.2. *In the above game, if the penalty vector is in $[-\rho, \rho]^n$, $\epsilon \leq \frac{1}{2}$ and $T \geq \frac{4\rho^2 \ln n}{\epsilon^2}$, then for any expert i*

$$\frac{1}{T} \sum_{t=1}^T \vec{p}^t \cdot \vec{m}^t \leq \frac{1}{T} \sum_{t=1}^T m_i^t + \epsilon$$

21.3 Approximate LP feasibility using Multiplicative Weights

21.3.1 Problem formulation

We are given a convex region $K \subseteq \mathbb{R}^m$ as some “simple constraints” where the solution lies. For example, $K = [0, 1]^m$. Also given the “normal” linear constraints $Ax \succeq b$ where $A \in \mathbb{R}^{n \times m}$ and $b \in \mathbb{R}^n$, we want to decide if the LP problem is feasible or not, i.e., decide if $\{x \in K : Ax \succeq b\} = \emptyset$. If the LP problem is feasible, the algorithm is expected to output an $x \in K$ such that $Ax \succeq b - \epsilon \cdot \vec{1}$. Otherwise the algorithm should just declare “infeasible”

21.3.2 Algorithm

We will translate this problem to the expert prediction game problem and using Corollary 21.2 to solve it:

Each expert i corresponds to a linear constraint $A_i x \geq b_i$. The algorithm proceed as follows:

```

for  $t = 1 \dots T$  : do
  choose  $\vec{p}^t$  using multiplicative weight update rule
  Adversary checks if  $\exists x \in K$  s.t  $\vec{p}^t Ax \geq \vec{p}^t b$ 
  if exists such  $x$  then
    adversary assigned the penalty  $A_i x^t - b_i$  to expert  $i$ 
  else
    declare “infeasible”
return  $x^* = \frac{1}{T} \sum_{i=1}^T x^t$ 

```

21.3.3 Analysis

If adversary can't find a feasible solution in K for the constraint $\vec{p}^t Ax \geq \vec{p}^t b$ that means the LP is infeasible. (If the LP is feasible, then $\exists x \in K$ such that the n constraints can be satisfied simultaneously, which means their linear combination can be satisfied, contradiction)

Let $\rho = \sup_{x \in K} \max_i |A_i x - b_i|$. If $T \geq \frac{4\rho^2 \ln n}{\epsilon^2}$, $\epsilon < 1/2$ then $\forall i \in [n]$:

$$0 \leq \frac{1}{T} \sum_{t=1}^T \vec{p}^t \cdot (Ax^t - b) \leq \frac{1}{T} \sum_{t=1}^T (A_i x^t - b_i) + \epsilon = A_i x^* - b_i + \epsilon$$

essentially it means that $Ax^* \succeq b - \epsilon \cdot \vec{1}$. Also since K is convex, $x^* \in K$. So the correctness is proved.