COS 511: Theoretical Machine Learning

Lecturer: Rob Schapire
Scribe: Jieming Mao

Lecture # 15
April 1, 2013

1 Brief review

1.1 Learning with expert advice

Last time, we started to talk about "learning with expert advice". This model is very different from models we saw earlier in the semester. In this model, the learner can see only one example at a time, and has to make predictions based on the advice of "experts" and the history. Formally, the model can be written as

- $N = \#$ of experts
- for $t = 1, \cdots, T$
  - each expert i predicts $\xi_i \in \{0, 1\}$
  - learner predicts $\hat{y} \in \{0, 1\}$
  - learner observes $y \in \{0, 1\}$
  - (mistake if $y \neq \hat{y}$)

In the previous lecture, we gave a "Halving algorithm" which works when there exists a "perfect" expert. In this case, the number of mistakes that the "Halving algorithm" makes is at most $\lg(N)$.

1.2 Connection with PAC learning

Now we consider an "analogue of the PAC learning model":

- hypothesis space $\mathcal{H} = \{h_1, \cdots, h_N\}$
- target concept $c \in \mathcal{H}$
- on each round
  - get x
  - predict $\hat{y}$
  - observe $c(x)$

This problem is a very natural online learning problem. However, we can relate this problem to "learning with expert advice" by considering each hypothesis h_i as an expert. Since the target concept is chosen inside the hypothesis space, we always have a "perfect" expert. Therefore, we can apply the "Halving algorithm" to solving this problem and the "Halving algorithm" will make at most $\lg(N) = \lg(|\mathcal{H}|)$ mistakes.

2 Bounding the number of mistakes

In the previous section, it is natural to ask whether the "Halving algorithm" is the best possible. In order to answer this problem, we have to define what is "best". Let A be any deterministic algorithm, define

$$M_A(\mathcal{H}) = \max_{c,x}(\#\text{ mistakes made by } A)$$

$M_A(\mathcal{H})$ is the number of mistakes A will make on hypothesis space $\mathcal{H}$ in the worst case. An algorithm is considered good if its $M_A(\mathcal{H})$ is small. Now define

$$opt(\mathcal{H}) = \min_A M_A(\mathcal{H}).$$

We are going to lower bound $opt(\mathcal{H})$ by $VCdim(\mathcal{H})$ as the following theorem:

Theorem 1. $VCdim(\mathcal{H}) \leq opt(\mathcal{H})$

Proof: Let A be any deterministic algorithm. Let $d = VCdim(\mathcal{H})$. Let $x_1, ..., x_d$ be shattered by $\mathcal{H}$. Now we can construct an adversarial strategy that forces A to make d mistakes as following:

- for $t = 1, \cdots, d$
  - present x_t to A
  - $\hat{y}_t = A$'s prediction
  - choose $y_t \neq \hat{y}_t$.

Since $x_1, ..., x_d$ is shattered by $\mathcal{H}$, we know there exists a concept $c \in \mathcal{H}$, such that $c(x_t) = y_t$ for all t. In addition, since A is deterministic, we can simulate it ahead of time. Therefore such adversarial construction is possible. Thus there exists an adversarial strategy to force A to make d mistakes. □

To sum up this section, we have the following result,

$$VCdim(\mathcal{H}) \leq opt(\mathcal{H}) \leq M_{Halving}(\mathcal{H}) \leq \lg(|\mathcal{H}|).$$

3 Weighted majority algorithm

Now we are going back to "learning with expert advice" and we want to devise algorithms when there is no "perfect" expert. In this setting, we will compare the number of mistakes of our algorithm and the number of mistakes of the best expert. Here we modify the "Halving algorithm" to get an algorithm when there is no "perfect" expert. Instead of discarding experts, we will keep a weight for each expert and lower it when this expert makes a mistake. We call this algorithm weighted majority algorithm:

- w_i = weight on expert i
- Parameter $0 \leq \beta < 1$
- $N = \#$ of experts

- Initially $\forall i, w_i = 1$
- for $t = 1, \cdots, T$
  - each expert i predicts $\xi_i \in \{0, 1\}$
  - $q_0 = \sum_{i:\xi_i=0} w_i$, $q_1 = \sum_{i:\xi_i=1} w_i$.
  - learner predicts $\hat{y} \in \{0, 1\}$
  - $\hat{y} = \begin{cases} 1 & \text{if } q_1 > q_0 \\ 0 & \text{else} \end{cases}$ (weighted majority vote)
  - learner observes $y \in \{0, 1\}$
  - (mistake if $y \neq \hat{y}$)
  - for each i, if $\xi_i \neq y$, then $w_i \leftarrow w_i\beta$

Now we are going to analyze weighted majority algorithm(WMA) by proving the following theorem:

Theorem 2. *(# of mistakes of WMA)* $\leq a_\beta$*(# of mistakes of the best expert)* $+c_\beta \lg(N)$*, where* $a_\beta = \frac{\lg(1/\beta)}{\lg(\frac{2}{1+\beta})}$ *and* $c_\beta = \frac{1}{\lg(\frac{2}{1+\beta})}$.

Before proving this theorem, let's first try to understand what this theorem implies. The following table gives a good understanding of the parameter:

β	a_β	c_β
$1/2$	≈ 2.4	≈ 2.4
0	$+\infty$	1
$\to 1$	2	$+\infty$

If we divide both sides of the inequality by T, we get

$$\frac{(\#\text{ of mistakes of WMA})}{T} \leq \frac{a_\beta(\#\text{ of mistakes of the best expert})}{T} + \frac{c_\beta \lg(N)}{T}$$

When $T \to +\infty$, $\frac{c_\beta \lg(N)}{T} \to 0$. Then this theorem means that the rate that WMA makes mistakes is bounded by a constant times the rate that the best expert makes mistakes.
Proof: Define W as the sum of the weights of all the experts: $W = \sum_{i=1}^{N} w_i$. Initially, we have $W = N$. Now we consider how W changes in some round. On some round, without loss of generality, we assume that $y = 0$. Then

$$\begin{aligned} W_{new} &= \sum_{i=1}^{N} w_i^{new} \\ &= \sum_{i:\xi_i=1} w_i\beta + \sum_{i:\xi_i=0} w_i \\ &= q_1\beta + q_0 \\ &= q_1\beta + (W - q_1) \\ &= W - (1-\beta)q_1 \end{aligned}$$

Now suppose WMA makes a mistake. Then

$$\begin{aligned} \hat{y} \neq y &\Rightarrow \hat{y} = 1 \Rightarrow q_1 \geq q_0 \\ &\Rightarrow q_1 \geq \frac{W}{2} \\ &\Rightarrow W_{new} \leq W - (1-\beta)\frac{W}{2} = (\frac{1+\beta}{2})W \end{aligned}$$

So if WMA makes a mistake, the sum of weights W will decrease by multiplying $\frac{1+\beta}{2}$. Therefore, after m mistakes, we get an upper bound of the sum of weights as

$$W \leq N \cdot (\frac{1+\beta}{2})^m.$$

Define L_i to be the number of mistakes that expert i makes. Then we have

$$\forall i, w_i = \beta^{L_i} \leq W \leq N \cdot (\frac{1+\beta}{2})^m.$$

Solving this inequality, and noting that it holds for all experts i, we get

$$m \leq \frac{(\min_i L_i)\lg(1/\beta) + \lg(N)}{\lg(\frac{2}{1+\beta})}.$$

□

4 Randomized weighted majority algorithm

The previous proof has shown that, at best, the number of mistakes of WMA is at most twice the number of mistakes of the best expert. This is a weak result if the best expert actually makes a substantial number of mistakes. In order to get a better algorithm, we have to introduce randomness. The next algorithm we are going to introduce is called randomized weighted majority algorithm. This algorithm is very similar to weighted majority algorithm. The only difference is that rather than making prediction by weighted majority vote, in each round, the algorithm picks an expert with some probability according to its weight, and uses that randomly chosen expert to make its prediction. The algorithm is as following:

- w_i = weight on expert i
- Parameter $0 \leq \beta < 1$
- $N = \#$ of experts
- Initially $\forall i, w_i = 1$
- for $t = 1, \cdots, T$
  - each expert i predicts $\xi_i \in \{0, 1\}$
  - $q_0 = \sum_{i:\xi_i=0} w_i$, $q_1 = \sum_{i:\xi_i=1} w_i$.
  - learner predicts $\hat{y} \in \{0, 1\}$

- $\hat{y} = \begin{cases} 1 & \text{with probability } \dfrac{\sum_{i:\xi_i=1} w_i}{W} = \dfrac{q_1}{W} \\ 0 & \text{with probability } \dfrac{\sum_{i:\xi_i=0} w_i}{W} = \dfrac{q_0}{W} \end{cases}$
- learner observes $y \in \{0, 1\}$
- (mistake if $y \neq \hat{y}$)
- for each i, if $\xi_i \neq y$, then $w_i \leftarrow w_i\beta$

Now let's analyze randomized weight majority algorithm(RWMA) by proving the following theorem:

Theorem 3. *E(# of mistakes of RWMA) $\leq a_\beta$(# of mistakes of the best expert) $+c_\beta \ln(N)$, where $a_\beta = \frac{\ln(1/\beta)}{1-\beta}$ and $c_\beta = \frac{1}{1-\beta}$.*

Notice here we are considering the expectation of number of mistakes RMWA makes because RMWA is a randomized algorithm. When $\beta \to 1$, $a_\beta \to 1$. This means RMWA can do really close to the optimal.

Proof: Similar to the proof of the previous theorem, we prove this theorem by considering the sum of weights W. On some round, define l as the probability that RWMA makes a mistake:

$$l = Pr[\hat{y} \neq y] = \frac{\sum_{i:\xi_i \neq y} w_i}{W}.$$

Then the weight in this round is changed as

$$\begin{aligned} W_{new} &= \sum_{i:\xi_i \neq y} w_i\beta + \sum_{i:\xi_i = y} w_i \\ &= lW\beta + (1-l)W \\ &= W(1 - l(1-\beta)) \end{aligned}$$

Then we can write the sum of weights at the end as

$$\begin{aligned} W_{final} &= N \cdot (1 - l_1(1-\beta)) \cdots (1 - l_t(1-\beta)) \\ &\leq N \prod_{t=1}^{T} \exp(-l_t(1-\beta)) \\ &= N \cdot \exp(-(1-\beta)\sum_{t=1}^{T} l_t) \end{aligned}$$

Then we have

$$\beta^{L_i} \leq w_i \leq W_{final} \leq N \cdot \exp(-(1-\beta)\sum_{t=1}^{T} l_t).$$

Define $L_A = \sum_{t=1}^{T} l_t$ as the expected number of mistakes RWMA makes. We have

$$L_A = \sum_{t=1}^{T} l_t \leq \frac{\ln(1/\beta)}{1-\beta}(\#\text{of mistakes of the best expert}) + \frac{1}{1-\beta}\ln N.$$

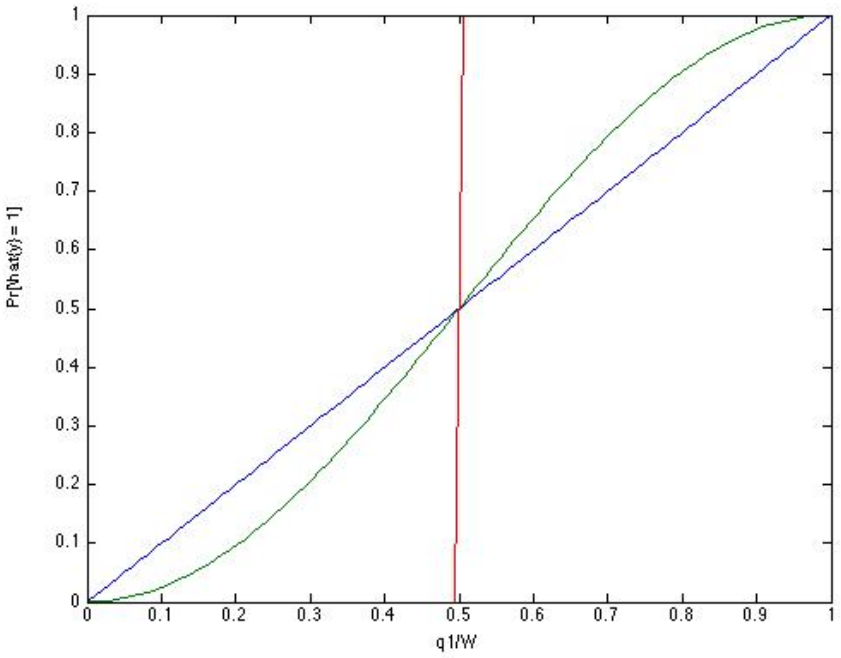
□

5 Variant of RWMA

Let's first discuss how to choose the parameter β for RWMA. Suppose we have an upper on the number of mistakes that the best expert makes as $\min_i L_i \leq K$, then we can choose β as $\frac{1}{1+\sqrt{\frac{2\ln(N)}{K}}}$. Then we have

$$L_A \leq \min_i L_i + \sqrt{2K\ln(N)} + \ln(N).$$

It is natural to ask whether we can do better than RWMA. The answer is yes. The high level idea is to work in the middle of WMA and RWMA. The following figure shows how this algorithm works. The y-axis is the probability of predicting $\hat{y}$ as 1. And the x-axis is the weighted fraction of experts predicting 1. The red line is WMA, the blue line is RWMA, and the green line is the new algorithm.



The new algorithm can reach $a_\beta = \frac{\ln(1/\beta)}{2\ln(\frac{2}{1+\beta})}$ and $c_\beta = \frac{1}{2\ln(\frac{2}{1+\beta})}$. These are exactly half of those of WMA. And if we make the assumption that $\min_i L_i \leq K$ again, the new algorithm has

$$L_A \leq \min_i L_i + \sqrt{K\ln(N)} + \frac{\lg(N)}{2}.$$

If we set $K = 0$ which means there exists a "perfect" expert, we have

$$L_A \leq \frac{\lg(N)}{2}$$

Thus we know this algorithm is better than the "Halving algorithm".

We usually can assume that $K = \frac{T}{2}$ by adding two experts: one always predicts 1 and the other always predicts 0. Then we have

$$L_A \leq \min_i L_i + \sqrt{\frac{T\ln(N)}{2}} + \frac{\lg(N)}{2}.$$

If we divide both sides by T, we get

$$\frac{L_A}{T} \leq \frac{\min_i L_i}{T} + \sqrt{\frac{\ln(N)}{2T}} + \frac{\lg(N)}{2T}.$$

When T gets large, the right hand side of the inequality is dominated by the term $\sqrt{\frac{\ln(N)}{2T}}$.

6 Lower bound

Finally we are going to give a lower bound for "learning with expert advice". This lower bound matches the upper bound we get from the previous section even for constant factors. So this lower bound is tight and we cannot improve the algorithm in the previous section.

To prove the lower bound, we consider the following scenario:

- On each round, the adversary chooses the prediction of each expert randomly, $\xi_i = \begin{cases} 1 & \text{w.p. } 1/2 \\ 0 & \text{w.p. } 1/2 \end{cases}$
- On each round, the adversary chooses the feedback randomly, $y = \begin{cases} 1 & \text{w.p. } 1/2 \\ 0 & \text{w.p. } 1/2 \end{cases}$

For any learning algorithm A,

$$E_{y,\xi}[L_A] = E_\xi[E_y[L_A|\xi]] = T/2.$$

For any expert i, we have

$$E[L_i] = T/2.$$

However, we can prove that

$$E[\min_i L_i] \approx \frac{T}{2} - \sqrt{\frac{T\ln(N)}{2}}.$$

Thus,

$$E[L_A] - E[\min_i L_i] \geq \sqrt{\frac{T\ln(N)}{2}}.$$

The term $\sqrt{\frac{T\ln(N)}{2}}$ meets the dominating term in the previous section. Also, in the proof of the lower bound, the adversary only uses entirely random expert predictions and outcomes. So we would not gain anything by assuming that the data is random rather than adversarial. The bounds we proved in an adversarial setting are just as good as could possibly be obtained even if we assumed that the data are entirely random.