Lecture 13 Scribe Notes

Instructor: Bobby Kleinberg *Diego Accame (daa67)*

1 Lecture 13 – Monday 20 February 2012 - Weighted Majority Algorithm

Other set of lecture notes at `http://www.cs.cornell.edu/courses/cs683/2007sp/lecnotes/week1.pdf` (skip the first two sections).

1.1 Bit Prediction Problem

A sequence of bits $b_1, b_2, b_3, \ldots, b_T$ is presented to the learner. In each round of the interaction:

1. Experts $1, \ldots, k$ report predictions $a_1(t), a_2(t), \ldots, a_k(t)$ where $a_i(t) \in \{0, 1\}$.
2. Algorithm must predict 0 or 1.
3. True answer b_t is revealed to algorithm.

Goal: Total number of mistakes made is not much more than best expert.

1.2 Online Learning (presented last week)

There is an abstract set of actions $\{1, \ldots, k\}$. You chose one action in each round. Payoff is revealed for every action.
Goal: Get nearly as much payoff as best action.

Relation to Bit prediction problem:

$$\text{Action} \Leftrightarrow \text{Expert}$$
$$\text{Choosing action } i \Leftrightarrow \text{predicting } a_i(t)$$

$$\text{Payoff} = \begin{cases} 0 & \text{if mistake} \\ 1 & \text{if correct} \end{cases}$$

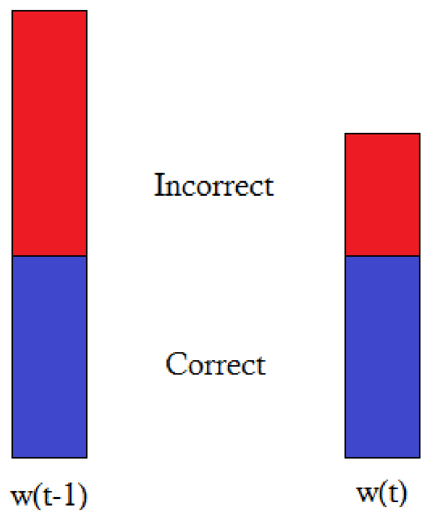
1.3 Weighted Majority Algorithm

(The actual one is actually a family of algorithms. The following is the $\epsilon = 1/2$ version)
Initialize $w(i, 0) = 1$ for $i = 1, \ldots, k$
In round t:

1. Each expert casts a vote with weight $w(i, t-1)$ for bit $a_i(t)$
2. Predict according to weighted majority vote
3. Incorrect experts: $w(i, t) = \frac{1}{2} w(i, t-1)$
4. Correct experts: $w(i, t) = w(i, t-1)$

1.3.1 Analysis

Every time the algorithm makes a mistake, a weighted majority of the experts were wrong and their total weight decreases by 1/2. Let $w(t) :=$ total weight at time t.



If algorithm makes mistake, $w(t) \leq \frac{3}{4}w(t-1)$

If algorithm makes m mistakes in total $\Rightarrow$

$$k(\frac{3}{4})^m \geq w(t) \geq (\frac{1}{2})^{m(1)}$$

If expert i makes only $m(i)$ mistakes:

$$\log(k) + m\log(\frac{3}{4}) \geq m(i)\log(\frac{1}{2})$$
$$-\log(k) + m\log(\frac{4}{3}) \leq m(i)\log(2)$$
$$m \leq \frac{\log 2}{\log(4/3)}m(i) + \frac{\log k}{\log(4/3)}$$

In the general case:

ϵ - learning rate. How fast you perceive change in the reliability of the experts.

Incorrect $\quad w(i,t) = (1-\epsilon)w(i,t-1)$
If $\epsilon \approx 0$ it is very slow but solution is good.
If $\epsilon \approx 1$ it is fast but not very accurate.

Using general $\epsilon > 0$ rather than 1/2 (more detailed proof on other set of notes):

$$\frac{3}{4} \Rightarrow (1 - \frac{\epsilon}{2})$$
$$\frac{1}{4} \Rightarrow (1 - \epsilon)$$
$$k(1 - \frac{\epsilon}{2})^m \geq w(t) \geq (1 - \epsilon)^{m(i)}$$
$$m \leq (\frac{2}{1-\epsilon})m(i) + \frac{2\log k}{\epsilon}$$

It is useful to note that $\frac{2}{1-\epsilon}$ is never less than 2

Fact: Any deterministic bit prediction makes at least twice as many mistakes as best expert in worst case.
Proof: For the case $k = 2$.

expert 1 always predicts 1

expert 2 always predicts 0

Generate b_t by simulating the algorithm, finding its prediction, and flipping that bit.
Algorithm makes t mistakes, best expert makes $\leq t/2$

1.4 Best Expert problem

We have experts $1, \ldots, k$ and costs $0 \leq c(i,t) \leq 1$ which represent the cost of expert i at time t.
In each round:

1. Algorithm chooses $i(t) \in \{1, \ldots, k\}$.
2. Costs $c(1,t), \ldots, c(k,t)$ revealed to algorithm

Step 1 choice can depend on costs in rounds $1, \ldots, t-1$ but not t.
Bit prediction cost:

$$c(i,t) = \begin{cases} 1 & \text{if mistake} \\ 0 & \text{if not} \end{cases}$$

1.5 Hedge algorithm

(a.k.a. Weighted Majority, Randomized Weighted Majority, Multiplicative Weights algorithm)

Choose an ϵ and let $w(i,t) = (1-\epsilon)^{c(i,1)+\ldots+c(i,t-1)}$ and $W(t) = \sum_i w(i,t)$

Sample expert $i(t) = i$ with probability $\frac{w(i,t)}{W(t)}$

1.5.1 Easy half of Analysis

If $i*$ is best expert, $W(T) \geq (1-\epsilon)^{c(i*,1\ldots T)}$

Compare $W(t+1)$ vs. $W(t)$:

Let's note that $(1-\epsilon)^x \leq 1-\epsilon x$ for $0 \leq x \leq 1$

$$\begin{aligned} W(t+1) &= \sum_{i=1}^{k} (1-\epsilon)^{c(i,t)} w(i,t) \\ &\leq \sum_{i} (1-\epsilon c(i,t)) w(i,t) \quad = W(t) - \epsilon W(t) \sum_{i} c(i,t) p(i,t) \qquad p(i,t) = \text{prob of choosing i} \end{aligned}$$

Let w_{*t} be the algorithm state at time t.

$$\mathbb{E}[W(t+1)|w_{*t}] \leq W(t)(1-\epsilon\mathbb{E}[c(i(t),t)|w_{*t}])$$

log is concave so:

$$\begin{aligned} \mathbb{E}[\ln W(t+1)|w_{*t}] &\leq \ln W(t) + \ln(1-\epsilon\mathbb{E}[c(i(t),t)|w_{*t}]) \\ &\leq \ln W(t) - \epsilon\mathbb{E}[c(t)] \quad c(t) \text{ - cost of t} \end{aligned}$$

$$\begin{aligned} \mathbb{E}[\ln W(t+1)] - \mathbb{E}[\ln W(t)] &\leq -\epsilon\mathbb{E}[c(t)] \\ \mathbb{E}[\ln W(t)] - \mathbb{E}[\ln W(t+1)] &\geq \epsilon\mathbb{E}[c(t)] \\ \mathbb{E}[\ln W(0)] - \mathbb{E}[\ln W(T)] &\geq \epsilon\mathbb{E}[\text{algorithm total cost}] \\ \ln k + \ln 1 - \epsilon c(i^*, 1\ldots T) &\geq \epsilon\mathbb{E}[\text{algorithm total cost}] \end{aligned}$$

$$\begin{aligned} \mathbb{E}[\text{algorithm total cost}] &\leq \frac{\ln 1-\epsilon}{\epsilon}\text{cost(best expert)} + \frac{\log k}{\epsilon} \\ &\leq \frac{1}{1-\epsilon}\text{cost(best expert)} + \frac{\log k}{\epsilon} \\ \mathbb{E}[\text{algorithm total cost - best expert cost}] &\leq \frac{\epsilon}{1-\epsilon}\text{cost(best)} + \frac{\log k}{\epsilon} \\ &\leq \frac{\epsilon T}{1-\epsilon} + \frac{\log k}{\epsilon} \quad \epsilon = \sqrt{\frac{\ln k}{T}} \\ &= O(\sqrt{T \ln k}) \end{aligned}$$