CSL 758: Advanced Algorithms
Machine Learning
(March 27, 2008 & March 31, 2008)

Professors:
Naveen Garg
Kavitha Telikepalli

Scribe By:
Kunal Malhan (2007MCS2929)
Gaurav Khurana (2007MCS2928)

Machine Learning

As the name suggest, the topic is all about learning from mistakes.

Lets us start with an example:

Say I want to Predict 'Bid on Horse Racing'. For simplicity, assume that there are only two Horses ('+', '– ') and **'n' experts**.

- Every day I take the advice of all the experts.
- Based on their advice, I bid on **One of Two Horses**.
- If the 'Horse on which I make a bid' loses the race, is considered a **mistake** made by me.
- I do sequence of bids.

Day	E1	E2	E3	….	En	I Bid On	Out Come
D1	+	+	+		-	+	+
D2	+	-	-		+	+	-
….							

Aim: To minimize the mistakes I make.

Learning from Expert's Advice (Existence of a Perfect Expert):

A perfect expert is that expert that never makes any mistake (i.e. his/her advice always matches the outcome).

I've a no of experts and don't know who is the perfect expert among them.

Assumption: There exists a perfect expert.

```
ALGO FINDING_PERFECT_EXPERT
  • Initially, On Day 1 all experts are in Set
    MY_FAVORITE_EXPERTS. MY_FAVOURITE_EXPERTS is the set
    in which I assume every expert is a perfect expert and
    would consider their advice.
BEGIN
```

```
• Every day, I take the advice of all the experts in the
  set MY_FAVORITE_EXPERTS and go with the majority. If
  more than half the experts say '+', I would bid on '+'
  horse.
• Whether I bid correctly or in-correctly, I will remove
  those experts from set MY_FAVORITE_EXPERTS, whose
  prediction went wrong on current day. Thus,
  MY_FAVORITE_SETS reflects a more accurate picture of
  perfect experts found till that day.
End ALGO
```

At the end of the each day, the above algorithm would remove those experts who are not perfect from the set MY_FAVORITE_EXPERTS, finally leading to the perfect expert.

Theorem: Total Mistakes (m) done by me are 0 <= m <= log n.

Proof: Here, Perfect Expert never makes a Mistake (i.e. M = 0).

Bounding Mistakes Done By Me (m):

Lower Bound:

Assuming best case, 'Perfect Expert' always remains in the majority set among MY_FAVORITE_EXPERTS and majority always predicts correctly on each day and there is at-least one expert whose prediction goes wrong on that day. Since, I always go with the majority and which always predicts correctly so, I would land up with 0 mistakes before finding the 'Perfect Expert'.

Upper Bound:

As I bid on Horse, which is advised by majority of the experts from the set MY_FAVORITE_EXPERTS, and every time I make a mistake, I reduce the set MY_FAVORITE_EXPERTS at-least by half (I took the advice of the majority and if majority makes a mistake, I would kick out the majority which means more than half of the set). Taking the worst case: half the MY_FAVORITE_EXPERTS say '+' would win and half say '-' would win.

So, After 1 mistake, size of set MY_FAVORITE_EXPERTS = n/2
After 2 mistake, size of set MY_FAVORITE_EXPERTS = n/4
…….
After log n mistake, size of set MY_FAVORITE_EXPERTS = 1

i.e. In the worst case, I will make maximum log (n) mistakes to 'Find the Perfect Expert' and after that I will bid always on horse as advised by 'Perfect Expert'.

Till now we had made an assumption, that there exists a perfect expert which is not true. So, now we ignore that assumption and take real world case.

So we now take all the experts and assign a weight to each one's advice. The weight assigned would depend on the number of mistakes that they make. There is no favorite experts set MY_FAVORITE_EXPERTS.

Next, an algorithm is presented which acts on these weights:

Weighted Majority Algorithm (There is No Perfect Expert):

```
ALGO WEIGHTED_MAJORITY
  • Initially, On Day 1, I assign weight '1' to each
    expert.
BEGIN
  • Every day I compute the 'Sum of Weights' of all
    experts who predict the win of '+' horse and and those
    who say '-' separately.
  • I bid on the horse, which has greater Sum.
  • Reduce the weights of all the experts (by halving each
    one's weight), who predict Wrong on the day.
End ALGO
```

Day		**E1**	**E2**	**E3**	**….**	**En**	**I Bid On**	**Out Come**
D1	Parities Horse	1 +	1 +	1 +		1 -	+	-
D2	Parities Horse	½ +	½ -	½ -		1 +	+	+
….								

Bounding Total Mistakes Done By Me:

Let us say I do **'m'** and best expert makes **'M'** mistakes in T rounds.

Let W_t = Sum of Weights of all the experts at end of round t.
Initially, W_0 is 'n'.

If I do a mistake:

It means experts with more than half of Total Weight predict wrongly and as a result I would reduce weights of all such experts by ½.

So, on every mistake in any round t reduces the total weight by at least $\frac{1}{4} * W_t$.

i.e. $W_{t+1} <= \frac{3}{4} W_t$.

So, finally total weight after T rounds: $W_T <= n (3/4)^m$ (i)

Again, Weight of best expert after T rounds $= 1/(2)^M$

Now, the best expert is one of the experts, the total weight of all experts >= weight of the best expert.

Or $W_T >= 1/(2)^M$ (ii)

from (i) and (ii):

$$1/(2)^M <= W_T <= n(3/4)^m$$
$$M * \log(1/2) <= \log n + m * \log(3/4)$$
$$\frac{M * \log(1/2)}{\log(3/4)} >= \frac{\log n}{\log(3/4)} + m \quad \text{[as log (3/4) is a –ve, so <= becomes >=]}$$

put log (½) = - log (2) and log (¾) = - log (4/3)

$$m <= \log_{(4/3)}(2) * M + \log_{(4/3)} n$$
$$\mathbf{m <= 2.4 * M + k}$$

where k is a constant independent of m and M, but depends on Total number of Experts.

So, above Algorithm is bounding **'mistakes done by me' to 2.4 times 'mistakes done by best Expert'**. So, if best expert makes 25% mistakes, we would mistakes 60% of the time which is not a good bound.

In the next section, we will try to reduce this bound by modifying our strategy of reducing the weights by half.

Reduces Weights differently:

Instead of halving the weights, reduce weight by a factor of 1/(1 + e), where e is a very small number.

If I do a mistake:

That means experts with more than half of total weight predicted wrongly and I would reduce weights of all such experts By 1/(1 + e).

Therefore, sum of weights of all experts in next Round $W_{t+1} = W_t/2 + W_t/(2*(1+e))$

Reduction in Weight $= W_t – (W_t/2 + W_t/(2*(1 + e)))$

Since, e is a small number therefore, 1/(1 + e) can be simplified to (1 – e).

Therefore, reduction in total weight $= W_t – (W_t/2 + W_t(1 - e/2))$

$= W_t * (1 – e/2)$

Total Weight after T rounds would be at most $= n * (1 – e/2)^m$

and at Least it will be Weight of best Expert $= (1 – e)^M$

Using the similar argument the weight of the best expert would be less than the weight of all experts.

Therefore, $(1 – e)^M <= W_T <= n * (1 – e/2)^m$

$$M * \log(1 - e) <= \log n + m * \log(1 - e/2)$$

$$\frac{M * \log(1 - e)}{\log(1 - e/2)} >= \frac{\log n}{\log(1 - e/2)} + m \quad \text{[as log(1 – e/2) is –ve]}$$

put $\log(1 - e/2) = - e/2 +$ [very small Quantity, can be ignored]

and $\dfrac{\log(1 - e)}{\log(1 - e/2)} \approx 2$

$$\mathbf{m <= 2 * M + 2/e * \log n}$$

So, we able to reduce the multiplying factor from 2.4 to 2, which is not again good. Considering the best expert makes mistakes 25% of the time, we would make 50% mistakes (almost equal to tossing a coin).

Randomized Weighted Majority Algorithm (Bid with Probabilities):

Now the Question arise: Can we do better?.. Yes.

Instead of taking going strictly with the majority, we use weights as probabilities. (e.g., if 70% of '+' and 30% of '– ', then pick 70:30). The idea is to smooth out the worst case.

```
ALGO RANDOMIZED_WEIGHTED_MAJORITY
  • Initially, On Day 1, I assign Weight '1' to each
    expert.
BEGIN
  • Every day, I compute the Sum of Weights of all experts
    who bid for '+' horse. Lets us assign this quantity by
    W+. I also compute the weights of all the experts who
    bid for '-' horse, let us assign this quantity by W-.
  • I bid on '+' Horse with probability W+/W, where W =
    total weight of all experts, i.e. W = W+ + W-.
  • Reduce the weights of each expert by (1 – ε) of his/her
    existing weight, whose prediction was wrong on that
    day.
End ALGO
```

Bounding Total Mistakes Done By Me:

On day t, let W_t denotes the total weight of all experts on that day.
Let F_t denotes the total weight of experts, whose prediction was wrong.
and S_t denotes the total weight of experts, whose prediction was correct.

Therefore, $W_{t+1} = S_t + F_t$

Now, F_t made mistake, so their weight reduces to $(1 - e) F_t$.
$F_{t+1} = (1 - e) F_t$.

$W_{t+1} = S_{t+1} + F_{t+1}$
$W_{t+1} = W_t - F_t + (1 - e) * F_t$
$W_{t+1} = W_t - F_t * e.$
$= W_t (1 - e * F_t/W_t)$

$W_{initial} = n.$

Final Weight (after T rounds),
$W_T = W_{initial} *(1 - e * F_1/W_1)*(1 - e * F_2/W_2)*..*(1 - e * F_T/W_T)$
$= n * (1 - e * F_1/W_1)*(1 - e * F_2/W_2)*..*(1 - e * F_T/W_T)$

$$\log W_T = \log n + \sum_i (\log (1 - e * F_i/W_i))$$

$$<\approx \sum_i \log n - \sum_i e * F_i/W_i \qquad \text{(iii)}$$

Best Expert makes M mistakes, therefore

$$W_T >= (1 - e)^M$$
$$\log W_T >= M * \log (1 - e) \qquad \text{(iv)}$$

Expected no. of Mistakes $= \sum_i F_i/W_i$

$<= e^{-1} \log n - e^{-1} \log W_T$ [by (iii)]
$<= e^{-1} \log n - e^{-1} M * \log (1 - \square)$ [by (iv)]
$<= e^{-1} \log n - e^{-1} M * [-e - e^2/2 - \ldots]$
$<= e^{-1} \log n + M * (1 + e/2)$
$<= M * (1 + e/2) + e^{-1} \log n$

$$\mathbf{E[m] <= (1 + e/2) * M + e^{-1} \log n}$$

Assuming, e to be small, therefore we have bounded the number of mistakes made by me to be very closer to the number of mistakes made by the best expert.