Minimizing Mistakes and Combining Experts

Online Learning Scenario

You are given samples one by one

Your classifier needs to output a label immediately

Afterwards, the classifier is told the true label of the sample and gets to learn based on that

GOAL: Minimize the total number of mistakes made by the classifier

You can make a ton of money in this scenario with the stock market. You try to predict if a stock will go up or down. Later, you can see if the stock went up or down.

Example: OR

Given n-dimensional bit vectors:

(0,1,1,0,0,1,1) → 0

(1,1,0,0,1,1,1) → 1

(1,0,0,1,1,1,0) → 1

In this case, label is $x_4 \vee x_5$

Goal: predict label after each sample is given and make few mistakes

Suppose we know the true label is always calculated as the OR of a few specific bits, but we don't know which bit positions those are

Example: OR

Given n-dimensional bit vectors:

(0,1,1,0,0,1,1) → 0

(1,1,0,0,1,1,1) → 1

(1,0,0,1,1,1,0) → 1

In this case, label is $x_4 \vee x_5$

Goal: predict label after each sample is given and make few mistakes

Algorithm: Start by predicting $x_1 \vee x_2 \vee \ldots \vee x_n$

(Will only make mistakes on samples labeled "0")

If a mistake is made, throw out the variables set to "1" in that sample

What's the maximum number of mistakes this algorithm can make?

For every mistake, at least one variable is discarded, so this can make at most n mistakes

Can there be a deterministic algorithm that always makes less than n mistakes?

No, look at these n samples:

(1,0,0,0,..,0,0)

(0,1,0,0,..,0,0)

(0,0,0,0,..,0,1)

If you run this algorithm forever, how many errors can you make?

Combining Multiple Classifiers

Imagine we have three different classifiers, and they give these outputs on x:

$$C_1(x) = 1 \quad C_2(x) = 0 \quad C_3(x) = 1$$

What should we output?

1 seems like a pretty good number to output, but C2 might be a must better classifier, so a simple majority might not be the best answer

Using "Expert" Advice

Predicting the stock market. Imagine we solicit n "experts" for their advice (will it go up or down?)

Expert 1	Expert 2	Expert 3	Neighbor's Dog	Truth
down	up	up	up	up
down	up	up	down	down
…	…	…	…	…

Imagine there is an expert that is always right. Can we make almost as few mistakes as them?

Here, the always right expert is the neighbors dog. We start off not knowing who is always right. How can we figure out where it is?

The Halving Algorithm

Output the majority vote of all experts who have been always right until now

How many mistakes can this algorithm ever make?

Whenever there is a mistake we throw away at least half of the experts

At most $\log_2(n)$ mistakes

This does not necessarily mean that we can find the always right expert in log(n) iterations

What if there is no expert that is always right?

Proposal #1: Iterated Halving Algorithm. Same as before, but once we've crossed off all experts, restart

How well will this do compared to the best expert?

Makes $\log_2(n)$[OPT +1] mistakes, where OPT is the number of mistakes of the best expert

log(n) mistakes before restarting each time. How many restarts? No more than [OPT+1], where OPT is the number of mistakes the best expert makes.

Proposal #2: Weighted Majority Algorithm

Start with all experts having weight 1

Predict based on weighted majority

Penalize mistakes by cutting weight in half

How many mistakes does this algorithm make compared to the best expert?

M = number of mistakes made so far

m = number of mistakes made so far by best expert

W = total weight (starts at n)

After each mistake W drops by at least 25%

After M mistakes, W is at most: $n(3/4)^M$

Weight of best expert is: $(1/2)^m$

$$(1/2)^m \leq n(3/4)^M$$

$$M \leq 2.4(m + \log_2 n)$$

How do we know halving the weight is best? Maybe some other decrease in weight is better?

Randomized Weighted Majority

Instead of taking majority vote, use weights as probabilities (e.g., if 70% of the weight is on “up”, chose up with 70% chance)

Maybe halving the weight is not optimal? Let’s multiply the weight by $(1-\varepsilon)$ to find the optimal ε

Say at time t we have a fraction F_t of weight on the experts that made a mistake

Probability of making a mistake? F_t

If a mistake is made, how much weight is removed? εF_t

$W_{final} = n(1-\varepsilon F_1)(1-\varepsilon F_2)\ldots$

$\ln(W_{final}) = \ln(n) + \Sigma_t[\ln(1-\varepsilon F_t)] \leq \ln(n) - \varepsilon \Sigma_t F_t$

$= \ln(n) - \varepsilon M$

If best expert has m mistakes, $\ln(W_{final}) > \ln((1-\varepsilon)^m)$

So, $\ln(n) - \varepsilon M > m \ln(1-\varepsilon)$

Which solves to $M \leq (1 + \varepsilon/2)m + (1/\varepsilon)\ln(n)$