The Binomial Weights Algorithm

Vadim Lyubashevsky
Based on the paper “On-line Prediction and Conversion Strategies”
by Nicolo Cesa-Bianchi, Yoav Freund, David P. Helmbold, and Manfred K. Warmuth

9th March 2006
Description of the Problem

The Chip Game

Binomial Weights Algorithm

Extensions
Problem Description

1. An example comes from some domain
2. A set of $N$ experts each label the example 0 or 1
3. Our master algorithm labels the example either 0 or 1
4. Nature reveals the actual label

We know that there is one expert who makes $\leq k$ mistakes

**Goal:** Minimize the number of mistakes made by our master algorithm in the worst case

**Intuition:** Find the best expert as quickly as possible!
The Adversarial Model

1. An example comes from some domain.
2. Adversary picks the labels that each expert outputs.
3. Algorithm makes prediction.
4. Adversary reveals true labeling.
5. If algorithm is wrong, it gets charged with a mistake.
6. All experts who were wrong get charged with a mistake.
Keeping Track of the Experts

Experts Predict
Keeping Track of the Experts
Keeping Track of the Experts

If 1 ball left, and it’s in the last bin – game over
1. An example comes from some domain.
2. Adversary colors the balls pink or orange.
3. Algorithm makes prediction.
4. Adversary reveals true labeling.
5. If algorithm is wrong, it gets charged with a mistake.
6. All balls of the opposite color move up one bin.

Adversary should always reveal labeling different than the algorithm!
Otherwise, balls move for free.
Simplified Adversarial Model

1. An example comes from some domain.
2. Adversary colors balls pink or orange.
3. Algorithm gets charged with a mistake.
4. Algorithm moves balls of one color up one bin.
How to Decide What to Move?

Which is the better move?
How to Decide What to Move?

Need to use math :(  
(But more pictures later :))  
Including a “proof” by picture!
A Simple Special Case

Assume that 1 out of N experts is perfect.

Strategy:

- Label the example as the majority of experts
- Look at nature’s labeling
- Eliminate all experts who were wrong from future consideration

Takes at most $\lfloor \log N \rfloor$ steps to find the perfect expert.
Make at most $\lfloor \log N \rfloor$ mistakes
Every mistake halves the number of possibilities for who the perfect expert could be
Back to the General Case

There is one expert who makes $\leq k$ mistakes

Assume we know that after some $s$ more steps, we will find this expert

Say, some expert has made $j$ mistakes. If he is the expert we’re looking for, then there are

$$\binom{s}{0} + \binom{s}{1} + \cdots + \binom{s}{k-j} = \binom{s}{\leq k-j}$$

ways for him to make at most $k-j$ more mistakes.

We will view this expert as representing $\binom{s}{\leq k-j}$ possibilities. As in the previous example, we want to eliminate possibilities
Eliminating “possibilities”

An expert who made $j$ mistakes represents $\binom{s}{\leq k-j}$ possibilities. If he predicts the next label *correctly*, he will have $\binom{s-1}{\leq k-j}$ possibilities left. If he predicts the next label *incorrectly*, he will have $\binom{s-1}{\leq k-j-1}$ possibilities left.

Notice that

\[
\binom{s-1}{\leq k-j} + \binom{s-1}{\leq k-j-1} = \binom{s}{\leq k-j}
\]
Eliminating ”possibilities”

An expert who made $j$ mistakes votes for label $l_i$. We treat this as

- $(\binom{s-1}{\leq k-j})$ votes for $l_i$
- $(\binom{s-1}{\leq k-j-1})$ votes for $1 - l_i$
The Algorithm

- There are $N$ experts
- Expert $i$ made $e_i$ mistakes and predicts $l_i$ on the next example
- He votes for $l_i$ with weight $\binom{s-1}{\leq k-e_i}$ and votes for $1 - l_i$ with weight $\binom{s-1}{\leq k-e_i-1}$
- We sum up the votes from all experts for each possible label, and output the label with the largest sum.

Things to notice:

- The weight Expert $i$ puts on $l_i$ ($1 - l_i$) is the number of possibilities he will have left for making less than $k$ mistakes if $l_i$ ($1 - l_i$) is the right label.
- If our prediction is wrong, then the number of possibilities is decreased by at least a factor of 2
Calculating $s$

Previous analysis assumed we knew $s$ - the steps needed to find the expert making less than $k$ mistakes

Computing $s$:

- At every step we eliminate at least half the possibilities
- Total number of possibilities is

$$
\sum_{i=1}^{N} \binom{s}{\leq k-e_i}
$$

where $N$ is the number of experts, and $e_i$ is the number of mistakes expert $i$ made.

- So $s = \log \left( \sum_{i=1}^{N} \binom{s}{\leq k-e_i} \right)$.
- Since $s$ has to be an integer,

$$
s = \max \left\{ q \in \mathbb{N} : q \leq \log \left( \sum_{i=1}^{N} \binom{q}{\leq k-e_i} \right) \right\}.
$$
Upper-bound on total number of mistakes

- In the beginning, there are $N$ experts who made 0 mistakes.
- So steps needed is

$$s = \max \left\{ q \in \mathbb{N} : q \leq \log \left( \sum_{i=1}^{N} \binom{q}{\leq k} \right) \right\}$$

$$= \max \left\{ q \in \mathbb{N} : q \leq \log N + \log \binom{q}{\leq k} \right\}$$

A helpful equality:

$$2k = \log \binom{2k+1}{\leq k}$$
Recall Chip Game Dilemma

Which is the better move?
Notes and Observations

- Experts who made fewer mistakes have votes counted more heavily (this is intuitive!)
- The "value" of each expert’s vote is independent of other votes
- The best thing for the adversary is to color half the chips in each bin pink and the other half orange
- The algorithm is essentially optimal when number of experts is large
Natural Extension

What if you know that m experts will make at most k mistakes?
Question: Should the experts who made the fewest mistakes have more votes?
Answer: Not necessarily!
Example

2 experts make no more than 4 mistakes

Ignore this expert
Different Phrasing of the Problem

We know that a $p$ fraction of the experts will make no more than $k$ mistakes.

Simplifying assumptions:

1. Experts can be infinitely split (there are a lot of experts)
2. An optimal adversarial strategy is to split each bin into 2 equal halves (This can actually be proved)

View experts as a "density"
Game Progression
Ending Condition

End when total weight left is $\leq p$
Steps Until $p$ Remains

Proof by picture (as promised):
At time $q$, bin $j$ will have

$$\frac{1}{2^q \binom{q}{j}}$$

We want

$$\sum_{j=0}^{k} \frac{1}{2^q \binom{q}{j}} = \frac{1}{2^q \binom{q}{\leq k}} \leq p$$

Steps $s$ needed until only $p$ percentage remains:

$$s = \min \left\{ q \in \mathbb{N} : \frac{1}{2^q \binom{q}{\leq k}} \leq p \right\}$$
Compatibility With Binomial Weights Analysis

Previous problem: N experts. 1 expert makes no more than k mistakes.

Translation: Do as well as $1/N$ of the experts. So $p = 1/N$. Steps $s$ needed until only $1/N$ percentage remains:

$$s = \min \left\{ q \in \mathbb{N} : \frac{1}{2q} \left( \begin{array}{c} q \\ \leq k \end{array} \right) \leq \frac{1}{N} \right\}$$

$$= \min \left\{ q \in \mathbb{N} : q \geq \log N + \log \left( \begin{array}{c} q \\ \leq k \end{array} \right) \right\}$$

$$= \max \left\{ q \in \mathbb{N} : q \leq \log N + \log \left( \begin{array}{c} q \\ \leq k \end{array} \right) + 1 \right\}$$
The scenario where expert makes no more than $k$ mistakes is similar to Ulam’s game with $k$ lies.

Essentially optimal algorithm for problems has been given.

Some generalization proposed.

Thank You!