Experts and Multiplicative Weights
slides from Avrim Blum

Using “expert” advice
Say we want to predict the stock market.
• We solicit $n$ “experts” for their advice. (Will the market go up or down?)
• We then want to use their advice somehow to make our prediction. E.g.,

<table>
<thead>
<tr>
<th>Expt 1</th>
<th>Expt 2</th>
<th>Expt 3</th>
<th>neighbor’s dog</th>
<th>truth</th>
</tr>
</thead>
<tbody>
<tr>
<td>down</td>
<td>up</td>
<td>up</td>
<td>down</td>
<td>down</td>
</tr>
</tbody>
</table>

Basic question: Is there a strategy that allows us to do nearly as well as best of these in hindsight?
[*“expert” = someone with an opinion. Not necessarily someone who knows anything.*]

Simpler question
• We have $n$ “experts”.
• One of these is perfect (never makes a mistake). We just don’t know which one.
• Can we find a strategy that makes no more than $\lg(n)$ mistakes?

Answer: sure. Just take majority vote over all experts that have been correct so far.

➢ Each mistake cuts # available by factor of 2.
➢ Note: this means ok for $n$ to be very large.

What if no expert is perfect?
Intuition:
Making a mistake doesn’t completely disqualify an expert. So, instead of crossing off, just lower its weight.

Weighted Majority Alg:
– Start with all experts having weight 1.
– Predict based on weighted majority vote.
– Penalize mistakes by cutting weight in half.

Analysis: do nearly as well as best expert in hindsight

• $M =$ # mistakes we’ve made so far.
• $m =$ # mistakes best expert has made so far.
• $W =$ total weight (starts at $n$).
• After each mistake, $W$ drops by at least 25%. So, after $M$ mistakes, $W$ is at most $n(3/4)^M$.
• Weight of best expert is $(1/2)^m$. So,

\[
\begin{align*}
(1/2)^m & \leq n(3/4)^M \\
(4/3)^M & \leq n2^m \\
M & \leq 2.4(m + \lg n)
\end{align*}
\]

So, if $m$ is small, then $M$ is pretty small too.

Randomized Weighted Majority

Instead of taking majority vote, use weights as probabilities. [e.g., if 70% on up, 30% on down, then pick 70:30]
Idea: smooth out the worst case.

Also, generalize $1/2$ to $1-\epsilon$.

Solves to

\[
M \leq \frac{-m \ln (1-\epsilon) + \ln n}{\epsilon} = \frac{1 + \epsilon/2}{} m + \ln n
\]

So,$ M \leq 1.39 m + 2 \ln n \quad (\text{when } \epsilon = 1/2)
M \leq 1.15 m + 4 \ln n \quad (\text{when } \epsilon = 1/4)
M \leq 1.07 m + 8 \ln n \quad (\text{when } \epsilon = 1/8)
Analysis

Say at time $t$ we have fraction $F_t$ of weight on experts that made mistake
So we have probability $F_t$ of making a mistake, and we remove $\epsilon F_t$ fraction of total weight

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

$$\ln W_{\text{final}} = \ln n + \sum_t (1 - \epsilon F_t) \leq \ln n - \epsilon \sum F_t$$

Now solve $\ln n - \epsilon M \geq m \ln (1 - \epsilon)$.

$$M \leq (1 + \epsilon/2)m + \ln n \frac{\epsilon}{\epsilon}$$

An application

Can use this for repeated play of matrix game

Consider cost matrix where all entries are 0 or 1
Rows are different experts. Start each with weight 1.
Notice that RWM is equivalent to “pick expert $i$ with probability $(w_i/\sum_j w_j)$ and go with it”

We assign weights $w_i$, inducing probabilities $p_i = (w_i/\sum_j w_j)$
We choose a random row according to this distribution $p$.
Adversary chooses column. This gives column vector $c$.
We pay expected cost $p.c = \sum_i p_i c_i$.
Update: $w_i = w_i (1 - \epsilon c_i)$.

RWM: matrix view

Since OPT over $T$ steps is at most $T$.

A proof of the Minimax Theorem

RWM gives a clean simple proof of the minimax theorem.

Suppose for contradiction minimax theorem was false.
This means some game $G$ has $V_C > V_R$.
If Column player commits first, there exists a row that gets the Row player at least $V_C$.
But if Row player has to commit first, the Column player can make him get only $V_R$.
Scale matrix so payoffs to row are in $[-1, 0]$.
Observe: payoffs of $-P$ to row = cost of $P$ to row
$\Rightarrow$ can view as costs and hence use RWM
Also, say $V_R = V_C - \delta$

A proof of the Minimax Theorem (contd)

Now consider RWM algorithm against column who plays optimally against row’s distribution (at each time).
In $T$ steps,
1) Alg gets $\geq$ [best row in hindsight] $-T/2 - (\log n)/\epsilon$ [by guarantee of the RWM algorithm]
2) best row in hindsight $\geq T*V_C$ [if row player plays optimally against empirical distr. of column player]
3) But Alg $\leq T*V_R$ [since each time opponent knows your distribution]

By (2)-(3), gap between alg and best row is $\geq \delta*T$.
Contradicts (1) for $\epsilon = \delta/2$ once we have $T \geq (\log n)/\epsilon^2$. 