MW and Experts

1. **Mistake Bounded Model**
   - difference from CRatio
     - measure the difference, not ratio
     - compare not to the best sequence of adapt decisions the algo could have made
     - but to the best fixed "expert" in hindsight.

   So fixed set of decision profiles could follow (N A team)
   which is best?

   Eg. have several paying algo's can use, which one should we use?

Get: Bounds of the form

\[ \# \text{ mistakes of algo} \leq \# \text{ mistakes of best expert} \times \left( \frac{2}{2 \text{ mult loss}} + \frac{2}{\text{ additive loss}} \right) \]

Ideally: Set mult loss \( = (1 + \delta) \)

and additive loss \( \frac{\log N}{\delta} \)

\( \Rightarrow \) almost as good as best expert "in the long run"
Start slow:

1. Show can get

   \[ A_T \leq O(\log N) \text{ (Best expert + 1) } \]

   (a) What if best expert makes no errors? 
   (b) \[ - - - - - m^* \text{ errors?} \]

   So both additive and multi terms are \( O(\log N) \) ! 😊

2. Then introduce the multiplicative weights (MW) or Weighted Majority (WM) algo.

   \[ A_T \leq 2.4 \cdot m^* + O(\log N) \]

   So much better now!

However: it is a deterministic algorithm, and no deterministic algo can do

\[ A_T \leq (2-\varepsilon) \cdot m^* \]

😢
Randomized Weighted Majority

In the bad example, should have helped beta.

Suppose make random choices of experts.

Then \( E[\text{Agg mistakes}] \leq (1+\epsilon) m^* + O(\sqrt{\log N}) \)

This is the holy grail, cannot do better. 😊

---

BTW, in all these arguments, show that \( \forall \) each expert \( i \)

\[
E[\text{Agg}] \leq \alpha m_i + \beta
\]

\( c \) expert \( i \)'s mistakes

which implies statement for best expert.

---

See the notes for proof [algo].
Slight generalization / restatement of model.

- Each time expert gives a loss value $l_i^t$
  $\Rightarrow$ put them in vector $(l_1^t, l_2^t, \ldots, l_N^t) = \vec{l}(t)$

- We choose probability of picking $1, 2, \ldots, N$ as
  
  $$(p_1^t, p_2^t, \ldots, p_N^t) = \vec{p}(t)$$

Then $\mathbb{E} [\text{loss for no } i] = \langle \vec{p}(t), \vec{l}(t) \rangle$

inner product

$$= \sum_i p_i^t l_i^t$$

So imagine:

Each time the expert panel / Nature / adversary / world

chooses a loss vector $\vec{l}^t$
- can depend on $p_1^t, p_2^t, \ldots, p_N^t$, not on $\vec{p}^t$

We choose a prob. vector $\vec{p}^t$
- which can depend on the past losses $l_1^t, l_2^t, \ldots, l_N^t$
  but not this time
- also can depend on our choices if we want
  (but therefore implied by losses)
And then our actual loss at time $t$ is dot product
\[
\langle \mathbf{p}^t, \mathbf{e}^t \rangle. \quad \text{(can think as "expected" loss)}
\]

We get that for all experts $i$
\[
\sum_t \langle \mathbf{p}^t, \mathbf{e}^t \rangle \leq \sum_t \mathbf{e}^t \cdot (1 + \varepsilon) + \frac{\log N}{\varepsilon}.
\]

This view is useful:

Nature chooses a loss vector $\mathbf{e}^t$ in $\{0, 1\}^N$.

We choose an "action" $\mathbf{p}^t$ in the "probability simplex"
\[
\Delta_N = \{ \mathbf{f} \in [0, 1]^N : \sum f_i = 1 \}\]

Our loss is $\langle \mathbf{p}^t, \mathbf{e}^t \rangle$

Want to compare with the best coordinate
\[
\min_i e_i = \min_{\mathbf{q}^*} \langle \mathbf{e}^t, \mathbf{q}^* \rangle
\]

make sure you believe this (it's simple)
Good: Why is this worth studying?

1. Choose best of multiple optima online
   (really choose the best of N "experts")
   \[\text{\textsuperscript{\textcopyright }maybe self-slyed,}\]

2. Algorithmic idea of multiplicative weights is very powerful

   - e.g. here's a max-flow problem.

   ![Diagram](image)

   All edges have capacity 1 (say).

   Say send flow on \(s \rightarrow a \rightarrow b \rightarrow t\) (unit amount)

   then saturate those edges.

   But not a max flow (since \(F^* = 2 \Rightarrow s \rightarrow a \rightarrow b \rightarrow t\))

   How to proceed. 1. Either use residual graph, etc.
2) Do use MW.

Algo: put lengths 1 on all edges.

push $F^*$ flow on shortest path.

Now set lengths to be $(1 + \varepsilon)$ flow on edge

repeat $\frac{F^* \log n}{\varepsilon}$ times

Take average of all these flows

Etc: each flow under-uses many edges

and over-uses many others.

But averaging them ensures that each edge gets

about the right amount of flow.

Another Example: Boosting in Learning. (HW?)

Suppose you have a weak classifier

Given any weights on data points, correctly classify 51% of data.

Then can use a "majority of $n$ classifiers" to get a

strong classifier

that gets 99% of data right.