Last time:
- VC Dimension: "negative results"
- Predicting from Expert Advice
  - Weighted Majority alg \( \in \) (noise-tol. H.A.)

Today:
- Randomized Weighted Majority alg \( \in \) (noise-tol. R.H.A.)
- Start PAC Learning unit
  - motivation
  - basic defs
  - learning \( C = \) intervals of \( R \)

(Admin: PSI due, PS2 out today)

Questions?

Recall:

Orig. WMA: \( \beta = 1 - \epsilon, \epsilon \to 0: \approx \frac{\ln m}{\epsilon} \cdot \ln N \)

If WMA is in 51-49 vote situation: always goes \( \longrightarrow \) predictable; adv. can exploit.
Could have been unpred. (toss coin to choose).

Natural thing to do: if vote is \((p, 1-p)\), toss \( p \)-biased coin to choose which side you use.

Motivates Rand WMA alg:

\( \text{again, } \beta \epsilon 2 \text{ is param.} \)

RWM: \( \text{again, pool of } N \text{ experts.} \)
Again, initialize \( w_i = w \) of expert \( i \) to 1.
- at a given trial, expert \( i \) predicts \( z_i \);
- RWM outputs \( z_i \); w.p. \( \frac{w_i}{W} \), where \( W = \text{sum of all weights} = w_1 + \ldots + w_N \).
- given correct value \( l \), for each \( i \) s.t. \( z_i \neq l \), set \( w_i \leftarrow w_i \cdot \beta \).

(Note: predictions need not be binary!)

**Theorem:** Assume oblivious adv. model (i.e. seq. of true labels is fixed in advance). If best expert in pool makes \( m \) mistakes, then \( \mathbb{E}[\#\text{mist. made by RWM}] \) is

\[
M \leq \frac{m \ln(2) + \ln N}{1 - \beta}.
\]

If \( \beta = 1 - \varepsilon \), \( \Rightarrow m + \frac{1}{\varepsilon} \ln N \)

**Proof:** Fix any seq. of \( T \) trials.

Let \( F_t = \text{frac. of tot wt at } t^{th} \text{ trial on wrong} \).

Have \( \Pr[\text{RWM opt makes mist. on trial } t] = F_t \).

Let \( M = \mathbb{E}[\#\text{mist.}] \).

Have \( M = \sum_{t=1}^{T} F_t \).

Before \( t^{th} \) trial: \( W = (1 - F_t)W + F_t W \).
After $t^{th}$ trial: $W$ becomes $(1-F_t) W + \beta F_t W$

$= W (1 - F_t + \beta F_t)$

$= W (1 - (1-\beta) F_t)$

So final

$W = N \prod_{t=1}^{T} (1 - (1-\beta) F_t)$.

We know $W_{final} \geq w_t^{best}$ of best expert = $\beta^m$.

So, $N \prod_{t=1}^{T} (1 - (1-\beta) F_t) \geq \beta^m$.

$\ln N + \sum_{t=1}^{T} \ln (1 - (1-\beta) F_t) \geq m \ln \beta$

$-\ln N - \sum_{t=1}^{T} \ln (1 - (1-\beta) F_t) \leq m \ln \frac{1}{\beta}$

Recall:

$1 - x \leq e^{-x}$

So

$\ln (1-x) \leq -x$, so $x \leq -\ln (1-x)$

So

$-\ln N + \sum_{t=1}^{T} (-\beta) F_t \leq m \ln \frac{1}{\beta}$.
\[-\ln N + (1-\beta) \sum_{t=1}^{\infty} F_t \leq m \ln \frac{1}{\beta}\]

so

\[M \leq \frac{\ln N + m \ln \frac{1}{\beta}}{1-\beta}.\]

If \( \beta = 0 \), this is RHA.

New Model: 

\[
\begin{array}{c}
\text{Probably} \\
\text{Approximately} \\
\text{Correct} \\
\text{Learning}
\end{array}
\]

(PAC)

OLMB learning:

- very pessimistic/adversarial assumptions.
- penalizing from day 1; \( \tau \)
- MB crit. typically doesn’t give any guar. abst. next example.

PAC learning:

- assumes each ex. \( x \) given to learner is
  - indep. drawn from some fixed (unknown) dist. \( \mathcal{D} \)
  - learn from random examples
  - batch, not online. train alg on set of labeled data pts.
  - output hypothesis \( h : x \rightarrow \{0,1\} \).
  - performance measured by acc. on \( x \sim \mathcal{D} \).
"PAC framework for learning class \( C \) using hypothesis class \( \mathcal{H} \):

- there is a fixed \( c \in C \) \( (\text{learner doesn't know} \ c) \)
- there is a fixed unknown arbitrary dist \( \mathcal{D} \) over \( X \).

- Learner is given training set of \( m \) indep. pairs \((x, c(x))\) where each \( x \sim \mathcal{D} \)

Equivalent P.O.V.:

learner has access to an "example oracle" \( EX(c, \mathcal{D}) \)

\[
EX(c, \mathcal{D}) \rightarrow (x, c(x)) \quad \forall x \sim \mathcal{D}
\]

- Learner does computation on training set, \( \mathcal{D} \) outputs a hyp \( h \in \mathcal{H} \) \( (\mathcal{H} = \text{class of hypothesis}) \)

\[ h : X \rightarrow \{0, 1\} \]

**Def:** Let, \( h, c \) be functions \( X \rightarrow \{0, 1\} \). Let \( \mathcal{D} \) be dist. over \( X \).

The error of \( h \) on \( c \) w.r.t. \( \mathcal{D} \) is

\[
e_{\mathcal{D}}(c, h) = \Pr_{x \sim \mathcal{D}}[h(x) \neq c(x)].
\]
II. Can't expect learning alg's $h$ to achieve 0 error: $\mathcal{D}$ may put very little at on some regions of $\mathbb{X}$; no info about how $c$ classifies such pts.

II. Can't guarantee lower error; possible that all in ex. very atypical.

Crit. For good learning alg: with high prob., hyp. has lower error.

**Def:** (prelim. def. of PAC learning)

"Alg. $A$ PAC learns $C$ using $\mathcal{H}$" means:

- $\forall c \in C$,
- $\forall \text{ dist } \mathcal{D}$ over $\mathbb{X}$,
- $\forall \varepsilon, \delta > 0$, *(\(\varepsilon = \text{error param.,} \\delta = \text{confidence param.})\)*

if $A$ is given $\varepsilon, \delta$ & access to $\text{EX}(c, \mathcal{D})$.

alg $A$ outputs $h \in \mathcal{H}$ s.t.

w. prob. $\geq 1 - \delta$ (over calls alg $A$ makes to $\text{EX}(c, \mathcal{D})$ & any internal rand. of $A$),

have $e_{\mathcal{D}}(h, c) \leq \varepsilon$. 
m = # examples / calls to \( f \); "sample complexity"

Runtime? A computationally efficient algo. runs in time \( \text{poly}(\frac{1}{\varepsilon}, \frac{1}{\delta}) \).
Usually \( X = \mathbb{R}^n \) or \( \text{SO}(n) \); if so, eff. algo. should be \( \text{poly}(n, \frac{1}{\varepsilon}, \frac{1}{\delta}) \).

\( \frac{3}{\varepsilon^2} + \frac{1}{\delta^2} \) 😊

If \( \mathcal{H} = \mathcal{C} \), we say \( A \) is "proper"

Ex: PAC Learning Intervals

\[ X = [0, 1], \quad \mathcal{C} = \text{all closed intervals } [a, b] \]

PAC learning: there's some \( D \) over \( X \), unknown to us.

Target interval \( c = [a, b] \).

Alg \( A \):
- draw \( m \) examples

\[ a' \quad b' \]

—we'll specify this
\( \text{Let } a' = \text{smallest pos ex} \)
\( b' = \text{largest pos ex.} \)
\( A \text{ outputs } h_{xp} \quad h = [a', b'] \).

Error of \( h \) on \( c \) w.r.t. \( \mathcal{Y} \)
\( \text{total} = \text{area of} \)

\( A \) only makes false neg. errors.

To show: w.h.p., \( h \) "doesn't miss much mass".

\( \text{Next time: analysis; show } m = O \left( \frac{1}{\varepsilon} \cdot \ln \frac{1}{\delta} \right) \)
suffices for \( (\varepsilon, \delta) \)-PAC learning.