**Weak vs. Strong Learning**

Def. Algorithm $A$ weakly "PAC learns" concept class $C$ if

$$\forall c \in C \quad \forall \delta > 0$$

$$\exists \varepsilon > 0$$

such that $\Pr \left[ h(x) \neq c(x) \right] \leq \frac{1}{2} - \frac{\varepsilon}{2}$

for examples of $C$.

It was conjectured that distribution-free weak learning

was really weaker but surprise!

Can "boost" a weak learner.

Then if $C$ can be weakly learned on any distribution $\mathcal{D}$, then $C$ can be

(strongly) learned. i.e. $\forall \varepsilon, \delta$

Will prove for case of $\mathcal{D}_0 = \mathcal{U}$
Applications

1) "Theoretical"
   - Unit dist Algorithms for poly term DNF weight w-poly threshold fmns
     (Boosting + KM)
   - Ave case vs. worst case complexity

2) practical - Boosting
   Freund-Schapire

Good & Bad Ideas

1) simulate weak learner several times on
   same distribution & take
   majority answer
   or best answer

   gives better confidence
   but doesn't reduce error, what if always get same answer?

2) filter out examples on which current hypothesis
   does well & run weak learner on part where you
do badly.

Problem: given a new
example, how do you
know which section it
is in?
3) Keep some samples on which you are ok in filtering.

Always use majority vote on all previous hypotheses to predict value of new samples.

History: Schapire, Freund-Schapire, Impagliazzo-Servedio-Klivans

Filtering Procedures

- Decide which samples to keep, which to throw out.
- Samples on which so far you guess correctly \( \leq \) need for check future hypotheses incorrectly \( \leq \) need to improve on these.

The Setting

- Given labeled examples

\[(x_1, f(x_1)), (x_2, f(x_2)), \ldots\]

\[x_i \in \mathbb{D}, f \in \mathbb{C}\]

- Given weak learning alg WL which weakly learns (advantage \( \frac{1}{2} \)) on any dist \( \mathbb{D} \)
Boosting Algorithm

Stage 0 (Initialize)

\[ D_0 \leftarrow \emptyset \]

run WL on \( D_0 \) to generate (w.h.p.)

\[ C_i \quad \text{st.} \quad \Pr_{D_0} [ f(x) = C_i(x) ] \geq \frac{1}{2} + \frac{\epsilon}{2} \]

* For \( i = 1 \ldots T = O(\frac{1}{\epsilon^2}) \) stages, stage \( h_{i-1} \) (can stop if \( \text{Majority}(C_1, \ldots, C_i) \) correct on \( \geq 1 - \epsilon \) inp.

1. Construct \( D_i \) via "filtering procedure":
   - Favor pts on which maj of \( C_1, \ldots, C_i \) don't do well
   - But also keep some other points \( \exists \)

   Will specify soon

2. Run WL on examples from \( D_i \) to output

\[ C_{i+1} \quad \text{st.} \quad \Pr_{D_i} [ f(x) = C_{i+1}(x) ] \geq \frac{1}{2} + \frac{\epsilon}{2} \]

* Output \( C = \text{MAJ}(C_1, \ldots, C_T) \)
Filtering procedure

Given new example \( x, f(x) \) from example oracle

1. If majority of \( c_1, \ldots, c_i \) wrong, keep it
   \[ i.e. \quad \frac{i}{2} \leq \frac{1}{\gamma} \]

2. If large majority right, then discard
   \[ i.e. \quad |\text{right} - |\text{wrong}| > \frac{1}{\gamma} \]
   \[ \text{or} \quad |\text{wrong}| \leq \frac{i}{2} - \frac{1}{2\gamma} \]

3. Else, \( |\text{right} - |\text{wrong}| = \frac{\alpha}{\gamma} \) for \( 0 < \alpha < 1 \)
   \[ |\text{wrong} - |\text{right}| = \frac{-\alpha}{\gamma} \]
   So keep with prob = \( 1 - \alpha \)

---

Graph:

- Prob of keeping: \( 1 \)
- Slope = \( 2\gamma \)
Need to show:

1) Output is has nontrivial agreement with $f$

2) # samples needed not too bad

Why could it be bad? If throw out lots of samples, might need to wait a long time before WL can give an output, but if throw out too many samples then you already have a good hypothesis!

Thm: filter outputs new sample in time $\leq \frac{1}{\text{error}[\text{Maj}(C_1 \cdots C_d)]}$

will stop if $\text{Maj}(C_1 \cdots C_d)$ correct on $\geq 1-\varepsilon$ fraction of inputs

so filtering procedure outputs sample with prob $\geq \varepsilon$

($\varepsilon$ in expectation, every $\frac{\varepsilon}{d}$ samples of $f$ at least one makes it thru the filtering system)

$\Rightarrow$ filtering slows down sample collection by $\leq O(d^4 \varepsilon)$

So let's focus on $\mathcal{O}$
**Notation**

- \( R_c(x) = \begin{cases} 1 & \text{if } f(x) = c(x) \\ -1 & \text{if } f(x) \neq c(x) \end{cases} \)
- \( N_i(x) = \sum_{1 \leq j \leq i} R_c(x) \)
- \( M_i(x) = \begin{cases} 1 & \text{if } N_i(x) \leq 0 \\ 0 & \text{if } N_i(x) \geq \frac{1}{\varepsilon} \\ 1 - \frac{1}{\varepsilon} \cdot N_i(x) & \text{otherwise} \end{cases} \)

"is \( c \) correct on \( x \)?"

after iteration \( i \)
how many \( c \)'s correct? (#right - #wrong)

prob of keeping \( x \) in filtering
(after stage \( i \))

note - all "wrong" \( x \) included in \( M \)
also some "right" \( x \) included

Note that new distribution on samples is proportional to \( M_i \):

\[
D_{M_i}(x) = \frac{M_i(x) D_0(x)}{\sum_y M_i(y) D_0(y)}
\]

since \( D_0 \) is assumed to be uniform we can droppp this\( \text{ \underline{not}} \) \( D_{M_i}(x) = D_i \)

\( \sum M_i(y) \) includes all "wrong" \( y \) but also \( y \) for which may not be overwhelming correct

How correct are we w.r.t. \( D_{M_i} \)?

- \( Adv_c(M_i) = \sum_x R_c(x) M_i(x) \)
- \( \Pr_{x \in D_{M_i}}[c(x) = f(x)] = \frac{1}{2} + \frac{Adv_c(M_i)}{\sum_x M_i(x)} \)

"Advantage" of \( c \) on \( M \)

\( \Pr_{c}[\text{correct}] - \Pr_{\text{incorrect}} = \frac{1}{2} \cdot \Pr_{c}[\text{correct}] - 1 \)
Note:

\[ \sum_{i=1}^{n} M_i(x) \geq \varepsilon 2^n \]

\[ Adv_c(M_i) = \gamma \cdot \varepsilon \cdot 2^n \]

Begin Proof

For input \( x \)

let \( A_i(x) \leq \sum_{0 \leq j \leq i-1} R_{c_{j+1}}(x) M_j(x) \)

Claim \( A_i(x) \leq \frac{1}{\varepsilon} + \frac{\varepsilon x}{2} \cdot i \)

bounds advantage per input

only helps after \( \frac{1}{\varepsilon x} \) rounds

Plan for use of claim:

Consider large matrix:

\[ \begin{pmatrix} x_1 & \cdots & x_n \end{pmatrix} \]

for iteration \( i \)

\( (h,j)^{th} \) Entry: \( R_{c_{j+1}}(x_k) M_j(x_k) \)

\( x_k \)'s row sum = \( \sum_{0 \leq j \leq i} R_{c_{j+1}}(x) M_j(x) = A(x) \cdot \frac{1}{\varepsilon} + \frac{\varepsilon x}{2} \cdot i \)

\( j \)'th col sum = \( \sum_{x} R_{c_{j+1}}(x) M_j(x) = Adv_{c_{j+1}}(M_j) \leq \frac{\varepsilon x^2}{2} \)

else algorithm
Goal: lower/upper bound average entry in matrix

**lower bound:**
lower bound average entry in column via
- correctness of WL
- fact that algorithm proceeds
  ⇒ lots of error
  ⇒ $\sum_{x} M_i(x)$ big
  ⇒ lots of progress in WL in absolute terms

**upper bound:**
upper bound rows via claim
- if advantage is too much, lose measure
  this is where majority rule weighting scheme is used
More details:

Assume claim, prove theorem:

Assume haven't terminated at \( i_{0+1} \)th stage

\[ \text{so} \quad \text{error} (C_{i_0}) \leq \varepsilon \]

\[ \forall x \quad M_{i_0}(x) \geq \varepsilon 2^n \]

Claim \( \Rightarrow \)

\[ \sum_x A_{i_{0+1}}(x) = \sum_x \sum_{0 \leq j \leq i_0} R_{C_{j+1}}(x) M_j(x) \quad \text{def of } A_{i_{0+1}} \]

\[ = \sum_{0 \leq j \leq i_0} \text{Adv}_{C_{j+1}}(M_j) \quad \text{def of } \text{Adv}_{C_{j+1}} \]

\[ \leq (\varepsilon 2^n) (i_{0+1}) \]

\[ \leq (\varepsilon 2^n) (i_{0+1}) \]

\[ \text{From "note"} \]

\[ + \sum_x A_{i_{0+1}}(x) \leq \sum_x \left( \frac{1}{\varepsilon \gamma} + \frac{\varepsilon \gamma}{2} (i_{0+1}) \right) \quad \text{claim} \]

\[ \leq 2^n \left( \frac{1}{\varepsilon \gamma} + \frac{\varepsilon \gamma}{2} (i_{0+1}) \right) \]

Putting together:

\[ (\varepsilon \gamma) (i_{0+1}) \leq \frac{1}{\varepsilon \gamma} + \frac{\varepsilon \gamma}{2} (i_{0+1}) \]

so \[ \frac{\varepsilon \gamma}{2} (i_{0+1}) \leq \frac{1}{\varepsilon \gamma} \quad \Rightarrow \quad i_0 \leq \frac{2}{\varepsilon \gamma^2} - 1 \]
Proof of claim:

Question: how can an input $x$ add to cumulative advantage throughout algorithm?

Observations:

- If algorithm's hypotheses $C_i$ are overwhelmingly correct on $x$, then not at all because $x$ gets measure 0

- If algorithm's hypotheses are doing badly (mostly wrong), then not at all because they decrease advantage

Main issue:

Can wander in middle - majority correct but not large majority so have positive measure in increase advantage need to bound this case.
Proof of Claim

First, strange but obvious fact:

Fact "elevator argument"
If one spends any amount of time in an elevator, then no matter what sequence of buttons pushed, one ascends from kth to k+1st floor at most one more time. Then one descends from the k+1st to kth floor.
(Analogous for negative floors −k + −(k−1))

Proof by picture:

For any x, Nj(x)

Match transitions going up with those going down on same level (as in parentheses)
but what is behavior of \( \sum_{j \in k} R_{j}^{(x)} M_{j}^{(x)} \)?

\( \sum_{j \in k} R_{j}^{(x)} M_{j}^{(x)} \)

\( \epsilon \in [0,1] \)

\( \exists \) slope \( \leq 1 \) (in fact \( \leq e^k \))

+ same sign as \( N_{j}^{(x)} \)

Recall: \( A_{x} = \sum_{0 \leq j \leq x-1} R_{j}^{(x)} M_{j}^{(x)} \)

Matching:

For \( k \geq 0 \):

match \( a = j \), s.t. \( N_{j}^{(x)} = k + N_{j+1}^{(x)} = k+1 \)

with \( b = j' \), s.t. \( N_{j'}^{(x)} = k_{n-1} + N_{j+1}^{(x)} = k \)

For \( k < 0 \): analogously match \( -k \) to \( -(k_{n-1}) \)

with \( -(k_{n-1}) \) to \( -k \)

For each matched pair:

Will bound contribution from matched pairs by \( \epsilon \times \delta \) per pair using bound on slope (and total of \( \frac{\epsilon \times \delta \times a}{2} \))
(for each matched pair \((a, b)\) cont.)

\[
R_{Ca+1}(x) M_a(x) + R_{Cb+1}(x) M_b(x) = M_a(x) - M_b(x)
\]

\(N_a(x):=k\)
\(N_b(x):=k+1\)

\(+1\) elevator going up
\(-1\) elevator going down

if \(0 \leq k \leq \frac{1}{\varepsilon_Y}\) or \(0 \leq KH \leq \frac{1}{\varepsilon_Y}\)

then

\[
M_a(x) - M_b(x) = (1 - \varepsilon_Y N_a(x)) - (1 - \varepsilon_Y N_b(x))
\]

\[
= 1 - \varepsilon_Y k - 1 + \varepsilon_Y (KH)
\]

\[
= \varepsilon_Y
\]

else \(M_a(x) - M_b(x) = \begin{cases} 1-1 & \text{on} \\ 0-0 & \end{cases} = 0\)

\(\therefore\) each pair contributes \(\leq \varepsilon_Y\) to sum

\(\leq \frac{1}{2}\) pairs

\[\epsilon \leq \frac{1}{2} \cdot \varepsilon_Y\]

\(\therefore\) total contribution

\[\epsilon \leq \frac{1}{2} \cdot \varepsilon_Y\]
Contribution from unmatched edges:

either all unmatched \( N_i \)'s have negative steps 
or all have positive steps

if all negative:
- \( R_{ej} \)’s all -1
- \( M_j \)'s all \( \in [0,1] \)

:. contribution of \( R_{ej} \) \( M_j(x) \leq 0 \)

if all positive:

if unmatched \( N_i \)'s in \( [0,\frac{1}{\epsilon^2}] \)

- for each \( M_j \in [0,1] \), contribution of \( R_{ej} \) \( M_j(x) \leq 1 \)

- at most \( \frac{1}{\epsilon^2} \) of these

\[ \Rightarrow \text{total contribution} \leq \frac{1}{\epsilon^2} \]

if unmatched \( N_i \) in \( [\frac{1}{\epsilon^2}, \ldots] \)

then \( M_j = 0 \)

\[ \Rightarrow \text{total contribution} = 0 \]

:. Grand total \( \leq \frac{1}{2} \cdot \frac{1}{\epsilon^2} \cdot \lambda + \frac{1}{\epsilon^8} \)