Mistake-bound model:
- View learning as a sequence of stages.
- In each stage, algorithm is given $x$, asked to predict $f(x)$, and then is told correct value.
- Make no assumptions about order of examples.
- Goal is to bound total number of mistakes.

**Alg A learns class $C$ with mistake bound $M$ if A makes $\leq M$ mistakes on any sequence of examples consistent with some $f \in C$.**

Recap from last time
- Last time: PAC model and Occam's razor.
  - If data set has $m \geq (1/\delta)[s \ln(2) + \ln(1/\delta)]$ examples, then whp any consistent hypothesis with size(h) < $s$ has err(h) < $\epsilon$.
  - Equivalently, suffices to have $s \leq (cm - \ln(1/\delta))/\ln(2)$
  - "compression ⇒ learning"
- [KV] book has esp. good coverage of this and related topics.
- Occam bounds ⇒ any class is learnable if computation time is no object.

Recap: open problems
Can one efficiently PAC-learn...
- $C=$fns with only $O(\log n)$ relevant variables? (or even $\tilde{O}(\log \log n)$ or $\omega(1)$ relevant variables)? This is a special case of DTs, DNFs.
- Monotone DNF over uniform D?
- Weak agnostic learning of monomials.

Online learning
- What if we don't want to make assumption that data is coming from some fixed distribution? Or any assumptions at all?
- Can no longer talk about past performance predicting future results.
- Can we hope to say anything interesting??

Idea: mistake bounds & regret bounds.
Simple example: disjunctions
- Suppose features are boolean: \( X = \{0,1\}^n \).
- Target is an OR function, like \( x_3 \lor x_9 \lor x_{12} \).
- Can we find an on-line strategy that makes at most \( n \) mistakes?
  - Sure.
    - Start with \( h(x) = x_1 \lor x_2 \lor \ldots \lor x_n \)
    - Invariant: \( \{\text{vars in } h\} \supseteq \{\text{vars in } f\} \)
    - Mistake on negative: throw out vars in \( h \) set to 1 in \( x \). Maintains invariant and decreases \( |h| \) by 1.
    - No mistakes on positives. So at most \( n \) mistakes total.

3-CNF formulas?
- Can you think of an algorithm to learn 3-SAT formulas from satisfying/non-satisfying assignments that makes at most \( n^3 \) mistakes?

MB model properties
An alg \( A \) is "conservative" if it only changes its state when it makes a mistake.
Claim: if \( C \) is learnable with mistake-bound \( M \), then it is learnable by a conservative alg.
Why?
- Take generic alg \( A \). Create new conservative \( A' \) by running \( A \), but rewinding state if no mistake is made.
- Still \( \leq M \) mistakes because \( A \) still sees a legal sequence of examples.

MB learnable \( \Rightarrow \) PAC learnable
Say alg \( A \) learns \( C \) with mistake-bound \( M \).
Transformation 1:
- Run (conservative) \( A \) until it produces a hyp \( h \) that survives \( \geq (1/\varepsilon)\ln(M/\delta) \) examples.
- \( \Pr(\text{fooled by any given } h) \leq \delta/M. \)
- \( \Pr(\text{fooled ever}) \leq \delta. \)
  - Uses at most \((M/\varepsilon)\ln(M/\delta)\) examples total.
- Fancier method gets \( O(\varepsilon^{-1}[M + \ln(1/\delta)]) \) examples.

MB learnable \( \Rightarrow \) PAC learnable
Say alg \( A \) learns \( C \) with mistake-bound \( M \).
Transformation 2: \( O(\varepsilon^{-1}[M + \ln(1/\delta)]) \) examples
- Run conservative \( A \) for \( O(\varepsilon^{-1}[M + \ln(1/\delta)]) \) examples. Argue that whp at least one of hyps produced has error \( \leq \varepsilon/2 \).
- Test the \( M \) hyps produced on \( O(\varepsilon^{-1}\ln(M/\delta)) \) new examples and take the best.
- Wait on full analysis until we get to Chernoff bounds...
One more example...
- Say we view each example as an integer between 0 and $2^{n-1}$.
- $C = \{[0,a] : a < 2^n\}$. (device fails if it gets too hot)
- In PAC model we could just pick any consistent hypothesis. Does this work in MB model?
- What would work?

What can we do with unbounded computation time?
- "Halving algorithm": take majority vote over all consistent $h \in C$. Makes at most $\log(|C|)$ mistakes.
- What if $C$ has functions of different sizes?
  - For any (prefix-free) representation, can make at most 1 mistake per bit of target.
    - Give each $h$ a weight of $(\frac{1}{2})^{\text{size}(h)}$
    - Total sum of weights $\leq 1$.
    - Take weighted vote. Each mistake removes at least $\frac{1}{2}$ of total weight left.

What can we do with unbounded computation time?
- "Halving algorithm": take majority vote over all consistent $h \in C$. Makes at most $\log(|C|)$ mistakes.
- What if we had a "prior" $p$ over fns in $C$?
  - Weight the vote according to $p$. Make at most $\log(\frac{1}{p})$ mistakes, where $f$ is target fn.
- What if $f$ was really chosen according to $p$?
  - Expected number of mistakes $\leq \sum_p [p \log(\frac{1}{p})]$ = entropy of distribution $p$.

Is halving alg optimal?
- Not necessarily (see hwk).
- Can think of MB model as 2-player game between alg and adversary.
  - Adversary picks $x$ to split $C$ into $C^-(x)$ and $C^+(x)$. [fns that label $x$ as - or + respectively]
  - Alg gets to pick one to throw out.
  - Game ends when all fns left are equivalent.
  - Adversary wants to make game last as long as possible.
- $\text{OPT}(C) = \text{MB}$ when both play optimally.

Is halving alg optimal?
- Halving algorithm: throw out larger set.
- Optimal algorithm: throw out set with larger mistake bound.
- You'll think about this more on the hwk...

What if there is no perfect function?
Think of as $h \in C$ as "experts" giving advice to you. Want to do nearly as well as best of them in hindsight.
These are called "regret bounds".
- Show that our algorithm does nearly as well as best predictor in some class.

We'll look at a strategy whose running time is $O(|C|)$. So, only computationally efficient when $C$ is small.
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.,

```plaintext
<table>
<thead>
<tr>
<th>Exp 1</th>
<th>Exp 2</th>
<th>Exp 3</th>
<th>neighbor’s dog</th>
<th>truth</th>
</tr>
</thead>
<tbody>
<tr>
<td>down</td>
<td>up</td>
<td>up</td>
<td>up</td>
<td>up</td>
</tr>
</tbody>
</table>
... ... ... ... ... ...
```

Can we do nearly as well as best in hindsight? [*expert* = someone with an opinion. Not necessarily someone who knows anything.]

[note: would be trivial in PAC (i.i.d.) setting]

Using “expert” advice

If one expert is perfect, can get \(\leq \lg(n)\) mistakes with halving alg.

But what if none is perfect? Can we do nearly as well as the best one in hindsight?

**Strategy #1:**
• Iterated halving algorithm. Same as before, but once we’ve crossed off all the experts, restart from the beginning.
• Makes at most \(\lg(n) \times \text{OPT} + 1\) mistakes, where \(\text{OPT}\) is #mistakes of the best expert in hindsight.

Seems wasteful. Constantly forgetting what we’ve “learned”. Can we do better?

**Weighted Majority Algorithm**

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.

<table>
<thead>
<tr>
<th>prediction</th>
<th>correct</th>
</tr>
</thead>
<tbody>
<tr>
<td>weights</td>
<td></td>
</tr>
<tr>
<td>predictions</td>
<td>Y Y Y Y</td>
</tr>
<tr>
<td>weights</td>
<td>1 1 1 .5</td>
</tr>
<tr>
<td>predictions</td>
<td>Y Y Y Y</td>
</tr>
<tr>
<td>weights</td>
<td>.5 .5 .5</td>
</tr>
</tbody>
</table>

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.
• \(W = \text{total weight (starts at } n)\).
• After each mistake, \(W\) drops by at least \(1/2\).

So, after \(M\) mistakes, \(W = \text{at most } n(3/4)^M\).

• Weight of best expert is \((1/2)^m\). So, constant ratio

\[
2.4(m + \lg n)
\]

**Randomized Weighted Majority**

2.4\((m + \lg n)\) not so good if the best expert makes a mistake 20% of the time. Can we do better? **Yes.**

• 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 \(\frac{1}{2}\) to \(1 - \epsilon\).

Solves to: \(M \leq \frac{-m \ln(1 - \epsilon) + \ln(n)}{\epsilon} \approx (1 + \epsilon/2)m + \frac{1}{\epsilon} \ln(n)\)

\(M = \text{expected mistakes}\)

\[
M \leq 3.39m + 2 \ln n \quad \Rightarrow \quad \epsilon = 1/2
\]

\[
M \leq 1.15m + 4 \ln n \quad \Rightarrow \quad \epsilon = 1/4
\]

\[
M \leq 1.07m + 9 \ln n \quad \Rightarrow \quad \epsilon = 1/8
\]

**Analysis**

• Say at time \(t\) we have fraction \(F_i\) of weight on experts that made mistake.
• So, we have probability \(F_i\) of making a mistake, and we remove an \(\epsilon F_i\) fraction of the total weight.

\[
W_{\text{final}} = n(1 - \epsilon F_i)(1 - \epsilon F_j)\ldots
\]

\[
\ln(W_{\text{final}}) \leq \ln(n) + \sum_i \ln(1 - \epsilon F_i) \leq \ln(n) - \epsilon \sum_i F_i
\]

\[
= \ln(n) - \epsilon M
\]

• If best expert makes \(m\) mistakes, then \(\ln(W_{\text{final}}) > \ln((1 - \epsilon)^m)\).
• Now solve: \(\ln(n) - \epsilon M > m \ln(1 - \epsilon)\).

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

unlike most worst-case bounds, numbers are pretty good

constant ratio

**Summarizing**

- \( E[\# \text{ mistakes}] \leq (1+\varepsilon)\text{OPT} + \varepsilon^{-1} \log(n) \).

- If set \( \varepsilon = (\log(n) / \text{OPT})^{1/2} \) to balance the two terms out (or use guess-and-double), get bound of \( E[\text{mistakes}] \leq \text{OPT} + 2(\text{OPT} \cdot \log(n))^{1/2} \leq \text{OPT} + 2T(\log n)^{1/2} \).

- Define average regret in T time steps as:

  \[ \text{(avg per-day cost of alg)} - \text{(avg per-day cost of best fixed expert in hindsight)} \]

  Goes to 0 or better as \( T \to \infty \) [= "no-regret" algorithm].

**What can we use this for?**

- Can use to combine multiple algorithms to do nearly as well as best in hindsight.
- Can apply RWM in situations where experts are making choices that cannot be combined.
  - Choose expert i with probability \( p_i = \frac{w_i}{\sum_i w_i} \).
  - Experts could be different strategies for some task, or rows in a matrix game. (Alg generalizes to case where in each time step, each expert gets a cost in \([0,1]\)).

**Minimax Theorem (von Neumann 1928)**

- Every 2-player zero-sum game has a unique value \( V \).
- Minimax optimal strategy for R guarantees R's expected gain at least \( V \).
- Minimax optimal strategy for C guarantees C's expected loss at most \( V \).

**Counterintuitive**: Means it doesn’t hurt to publish your strategy if both players are optimal. (Borel had proved for symmetric 5x5 but thought was false for larger games)

**Nice proof of minimax thm**

- Suppose for contradiction it 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 \([-1,0]\). Say \( V_R = V_C - \delta \).

**Proof, contd**

- Now, consider playing randomized weighted-majority alg as Row, against Col who plays optimally against Row's distrib.
- In T steps,
  - Alg gets \( \geq (1-\varepsilon/2)[\text{best row in hindsight}] - \log(n)/\varepsilon \)
  - BRiH \( \geq TV_C \) [Best against opponent's empirical distribution]
  - Alg \( \leq TV_R \) [Each time, opponent knows your randomized strategy]
  - Gap is \( \delta T \). Contradicts assumption if use \( \varepsilon = \delta \), once \( T > 2\log(n)/\varepsilon^2 \).

**A natural generalization**

- A natural generalization of this setting: say we have a list of n prediction rules, but not all rules fire on any given example.
  - E.g., document classification. Rule: "if <word-X> appears then predict <Y>". E.g., if has football then classify as sports.
- Natural goal: simultaneously, for each rule i, guarantee to do nearly as well as it on the time steps in which it fires.
  - For all i, want \( E[\text{cost}(\text{alg})] \leq (1+\varepsilon)\text{cost}(i) + O(\varepsilon^{-1} \log n) \).
- So, if 90% of documents with football are about sports, we should have error \( \leq 11\% \) on them.

* "Specialists" or "sleeping experts" problem. Will get to this later..."