Last Time:
- LTFs
- OLMC model
- Elimination alg. for monotone disj. variants
- 1-Decision lists

Today:
- OLMC learning alg. for 1-decision lists
- Winnow 1 alg. for sparse monotone disj. (hopefully)
- Winnow 2 alg. for (certain) other LTFs

Back to 1-OLs of length r:
- seq. of r "standard" rules,
- one "default rule"

if T output 0 \implies 2 poss.

4n rules total.

\# of length-r 1-OL over \{0,1\}:
\approx (4n)^r \cdot 2

OLMC Alg. to learn length-r 1-OLs over \{0,1\} with \leq O(nr) mistake bound.

[Always r \leq n, so \leq O(n^2).]
Alg. uses hypotheses which are slight variant of 1-DL:

- hyp. always contains all $4n+2$ rules;
- rules are grouped into levels; each level can have multiple rules. Think of rules within a level as being arranged in lexicographic order.

Alg:

Initially $h$ has one level $\forall$ all $4n+2$ rules

\[
\begin{array}{cccc}
  x_1 & \rightarrow & x_2 & \rightarrow \\
  \boxed{0} & \uparrow & \boxed{0} & \uparrow \\
  \boxed{0} & \uparrow & \boxed{1} & \uparrow \\
  \boxed{0} & \downarrow & \boxed{1} & \\
\end{array}
\]

level 1

* Given $x$: look at level 1 rules, then """""", etc.

Within a level, look thru rules in lex order, looking for first rule whose "if" cond. is satisfied by $x$.

Use that rule to predict.

(If no rule in current level applies, look at next level).

or example $x \Rightarrow y$?

- Update rule: You use a partic. rule in a partic. level for prediction.

  \[
  \begin{array}{c}
    \downarrow \\
    b
  \end{array}
  \]

- if pred. was right: no change.

  \[
  \begin{array}{c}
    \downarrow \\
    b
  \end{array}
  \]

- if pred. wrong move the
Rule that was just used down to next level.

Ex: initially hyp. is \( x_1 \rightarrow x_1 \rightarrow \overline{x}_1 \rightarrow \overline{x}_1 \rightarrow x_2 \ldots \rightarrow 0 \)

Get ex. \( z = 010\ldots \); \( c(z) = 1 \)

\( h(z) = 0 \) move rule \( \overline{x}_i \)

To level 2.

Next hyp:

\( \overline{x}_i \rightarrow x_1 \rightarrow \overline{x}_1 \rightarrow x_2 \ldots \rightarrow \overline{x}_i \rightarrow 0 \)

Thm: The above alg. makes \( O(n^r) \) mistakes when target \( c \) is any 1-DC of length \( r \) over \( \{0,1\} \).

PF: Let \( c = \text{target concept} \)

\( \cdots \rightarrow \ell_r \rightarrow b_{r+1} \)

Claim: first rule in \( c \) never moved below level 1 rule by our alg. if \( \ell \) holds, output is \( b \) so never applies.
Claim 2: 2nd rule in c never moved below level 2: to get pushed to level 3, rule would have to be in level 2 & be 1st rule in hyp whose "if" case applies. Says 1st rule is in level 2; it didn't apply & did apply, so true output value of c on that ex. is b2. So 2nd rule wouldn't have caused a mistake.

Claim i: ith rule in c never moved below level i: same reasoning (if it's in level i & it fires, its pred. is correct.)

So... no rule in c (including final \( \rightarrow b_{r+1} \) rule) will ever be moved below level r+1.

Hence no rule at all will ever be moved below level r+2: for every i, same rule in c (could be \( \rightarrow \)) will apply no deeper than level r+1.

Each mistake moves a rule down a level.

So... each of \( \leq (4n+2) \) rules moves \( \leq (r+1) \) levels, so tot. # mistakes \( \leq (4n+2) \cdot (r+1) = O(nr) \).
Note: comp. efficient: poly(n) time per trial.

Learning sparse disjunctions: Winnow 1.

Recall elim. alg: mist. bound \( \leq n \) for \( C = \{ \text{mon. disj.} \} \).

Suppose target \( c \) is sparse: only has \( k \leq k \) many vars. E.g., \( n \geq 10^5 \)
\[ k = 4 \] need \( n \) mistakes...?

\( C = \{ \text{all } k\text{-sparse mon. disj.} \} \)
\[ c \in C : \quad c(x) = x_{2416} \lor x_{9998} \lor x_{1144} \lor x_{76543} \]

"Very relevant for real-world learning:
most features are irrelevant, we don’t want to "pay" much for them."

Winnow alg: learns with \( O(\sqrt{k} \log n) \) m.b.

Much better than elim. alg. for small \( k \).

\( \Rightarrow \) Winnow uses a hyp. which is an LTF over \( \{0,1\}^n \):

\[ h(x) = \begin{cases} 1 & \text{if } w_1 x_1 + \cdots + w_n x_n \geq \theta \\ 0 & \text{if } w_1 x_1 + \cdots + w_n x_n < \theta \end{cases} \]
Q: can LTFs express disjunctions?

Yes: \( x_1, \ldots, x_n \in \{0, 1\} \).

Here are some LTFs:

- \( x_1 + \ldots + x_k \geq \frac{1}{2} \): holds iff \( x_1, x_2, \ldots, x_k \) holds
- \( x_1 + \ldots + x_k \geq k - \frac{1}{2} \): holds iff \( x_1, x_2, \ldots, x_k \) holds

\[ x_1 + \ldots + x_k \geq r \]: "r-out-of-k" threshold fn.

\[ r = \frac{k}{2} \]: MAJORITY function.

- Any 1-DL can be expressed as an LTF (hint: use different wts for diff vars in DL...)

Winnow: here's Winnow 1 (good for mon. disj.)

Winnow 1 abg: \( \{0, 1\}^n \)

- Initial hyp \( h(x) \) is \( w \cdot x \geq \Theta \) where
\[ \theta = n, \quad \mathbf{w} = (1, \ldots, 1) \quad \forall i = 1, \ldots, n. \]

- Predict using \( h(x) \)
- On false pos \( (h(x) = 1, c(x) = 0) \): for all \( i \) s.t. \( z_i = 1 \), set \( w_i = 0 \). (demotion step)
- On false neg \( (h(x) = 0, c(x) = 1) \): for all \( i \) s.t. \( z_i = 1 \), set \( w_i = 2 \cdot w_i \). (promotion step)
- If \( h(x) = c(x) \), keep \( h \) same.

Makes sense: we know those \( x_i \)'s not in \( c \).

\( \vdash \) wish we had \( w \cdot z \geq \theta \), but had \( w \cdot c < \theta \); makes sense to increase \( w \).

**Lemma:**
1. No \( w \cdot i \) is ever \( < 0 \) \( \checkmark \)
2. In each promotion step, at least one variable in \( c \) is promoted. \( \checkmark \)
3. For every \( i = 1, \ldots, n \), have \( w \cdot c \leq 2n. \)

**Pf:** A \( w \cdot i \) is only promoted if \( i \) is \( < n \). (if \( n \cdot z + z = 1 \), \( w \cdot z \) would be \( 2n \) already, so \( h(x) = 1 \) \( \neq n \) wouldn't promote.)
Lemma 2: Total \# prom. steps, for any K-sparse mon. disj., is \( \leq k \cdot \log(2n) \).

Pf: if demotion happens, no \( x_i \) in \( c \) had \( x_i = 1 \).
So vars in \( c \) never demoted.
Each var. in \( c \) doubles \( \leq \log(2n) \) times before \( \hat{w} \) exceeds \( 2n \), so there are \( \leq k \) vars in \( c \).
Each prom. doubles \( \hat{w} \) of at least one var.
So \( \leq k \cdot \log(2n) \) prom. in total.

Lemma: Let \( d = \# \) demotion steps.
Have \( d \leq p + 1 \).

Pf: Let \( \hat{w} = \sum_{i=1}^{n} w_i \). Initially \( \hat{w} = n \).

• At each demotion step, have
  \[ \hat{w} \cdot 2 = \sum_{i=1}^{n} w_i \cdot 2; = \sum_{i: \hat{w}_i \geq n} w_i \geq n, \] and set all
  these to 0;
  \( \hat{w} \) decreases by at least \( n \).

• At each promot. step, have
  b/c bad prom.
\[ w \cdot z = \sum_{i=1}^{n} w_i z_i = \sum_{i:z_i=1} w_i < \underbrace{n}, \text{ and we double those } w_i \text{'s} \]

So \( W \) increases by \(<n\).

\[ W \geq 0 \text{ always (all } w_i \geq 0), \text{ so } 0 \leq W \leq n - d + p \]

Rearrange: \( d \leq p \cdot l \).

Proves Thm:

Thm: Winnow 1 makes \( \leq 2K \log(2n) + 1 \)

\[ = O(k \log n) \]

mist. when target is \( k \)-sparse mon. disj.

Next time:  
- discuss  
- Winnow 2 (LTFs)  
- Perceptron alg for LTFs

\( \rightarrow \) dual perception, kernelization.