7.1 Perceptron vs. WINNOW

In the previous lecture we were introduced to the Perceptron algorithm, which learns halfspaces in $n$ dimensions with a mistake bound of $O(1/\sigma^2)$. Here $\sigma$ is the geometric margin of the sample set, defined as

$$\sigma = \min_{x \in X} \frac{|(x, h)|}{|x|}$$

where $h$ is the normal to the hyperplane. It is clear that $\sigma$ could be very smalls for certain distributions of points $x_1, \ldots, x_m \in S^{n-1}$ on the unit sphere.

Let’s compare the WINNOW and Perceptron algorithms:

- **WINNOW** - learns disjunctions on $k \leq n$ variables with a mistake bound of $O(k \log n)$.
- **Perceptron** - learns halfspaces in $n$ dimensions with the mistake bound described above.

Note that a disjunction over $x \in \{0, 1\}^n$ is a type of halfspace:

$$\text{OR}(x_1, \ldots, x_m) \equiv \sum_{i=1}^{m} x_i \geq 1 \equiv \text{SIGN}(\sum_{i=1}^{m} x_i - 1)$$

Take an example: the “unknown literal” concept class $\{x_i\}$ with $1 \leq i \leq n$. If we run WINNOW on this concept class we obtain a mistake bound of $\log n$. What if we run the Perceptron algorithm? We can map $\{0, 1\}^n$ onto $S^{n-1}$ through normalization, where $(1, \ldots, 1)$ becomes $(1/\sqrt{n}, \ldots, 1/\sqrt{n})$. What is the margin for this set of points? One separating halfspace is $x_i > 0$, but the margin could be as small as $1/n$:

$$(0, 1/\sqrt{n} - 1, \ldots, 1/\sqrt{n} - 1) \equiv (0, 1, \ldots, 1) \rightarrow \text{FALSE}$$

$$(1/\sqrt{n}, \ldots, 1/\sqrt{n}) \equiv (1, \ldots, 1) \rightarrow \text{TRUE}$$

So $\sigma^2 \approx 1/n$, and the mistake bound of Perceptron in this case is $O(n)$. 

7-1
7.2 WINNOW: Beyond Disjunctions

Let’s state WINNOW in its full generality. First, define function $f$ as

$$f(x_1, \ldots, x_n) = \begin{cases} 1 & \text{if } \sum_{i=1}^{n} u_i x_i \geq 1 - \delta \text{ for } u_1, \ldots, u_n \geq 0 \\ 0 & \text{if } \sum_{i=1}^{n} u_i x_i < 1 - \delta \text{ for } u_1, \ldots, u_n \geq 0 \end{cases}$$

where $x \in \{0, 1\}^n$. Note that $f$ has a mistake bound approximately

$$O\left( \frac{n}{\delta^2} + \frac{\ln \Theta}{\delta^2} \sum_{i=1}^{n} u_i \right)$$ (7.1)

for all $\Theta > 0$. Now let’s check that this expression agrees with our earlier analysis of the WINNOW mistake bound.

$$\sum_{i=1}^{k} x_i > 1 \quad \forall x, \text{OR}(x_i) = \text{TRUE} \Rightarrow \sum_{i} x_i \geq 1$$

Choose $\delta = 1/2$ to obtain a mistake bound, from Equation 7.1, of:

$$O\left( \frac{4n}{\Theta} + 4k \ln \Theta \right)$$

Then choose $\Theta = n$ to obtain a mistake bound of $O(k \ln n)$. The learning performance of WINNOW is thus consistent across our analyses.

Now we can examine the performance of WINNOW on the concept class of halfspaces. Let $h$ be a halfspace such that

$$h = \text{SIGN}(\sum_{i=1}^{n} w_i x_i - \gamma)$$

where $\gamma, w_1, \ldots, w_n$ are positive integers. Define $W$ to be the sum of the weights $w_1, \ldots, w_n$ and $\gamma$. The mistake bound of WINNOW in this case is $O(W^2 \log n)$; the performance of WINNOW depends on the weights of the halfspace. Restating our definition of $h$, we have:

$$\sum_{i=1}^{n} w_i x_i \geq \gamma \quad \text{if } x \text{ labeled TRUE}$$

$$\sum_{i=1}^{n} w_i x_i < \gamma \quad \text{if } x \text{ labeled FALSE}$$

Dividing by $\gamma$:

$$\sum_{i=1}^{n} \frac{w_i}{\gamma} x_i \geq 1 \quad \text{if } x \text{ labeled TRUE}$$

$$\sum_{i=1}^{n} \frac{w_i}{\gamma} x_i < 1 \quad \text{if } x \text{ labeled FALSE}$$

What can we choose $\gamma$ to be? It must always be $\leq 1 - \min(w_i/\gamma)$. Applying Equation 7.1, we obtain:

$$\frac{n}{(1 - \frac{\gamma}{w_i})^2 \Theta} + \frac{\ln \Theta}{(1 - \frac{\gamma}{w_i})^2 W}$$
If we choose \( \Theta = n \), this expression becomes:

\[
\approx W(\frac{w_i}{\gamma}) \ln n \leq W^2 \log n
\]

So for WINNOW, a weight \( W \) halfspace on \( n \) variables can be learned with mistake bound \( O(W^2 \log n) \).

### 7.3 Decision Lists on \( k \) Variables

Remember that we gave a PAC algorithm for decision lists on \( k \) variables, then showed that you need \( \approx O(n \log n/\epsilon) \) examples.

How do we realize a decision list as a linear threshold function? Recall that a decision list of length \( k \) is a set of variables of the form \( x_1, x_i, x_j, \ldots \), equivalent to:

\[
\text{SIGN}(2^k x_1 - 2^{k-1} x_i + 2^{k-2} x_j - \ldots)
\]

Applying our analysis from Section 7.2, we know that the mistake bound of WINNOW on this problem will be \( O(2^{2k} \log n) \). This bound is much worse than our performance on disjunctions, \( O(n \log n) \).

### 7.4 Exploring WINNOW

Now we will explore the WINNOW in greater detail. Recall that the halfspace we want to learn can be defined as:

\[
\sum_{i=1}^{n} u_i x_i \geq 1 \quad \text{if } x \text{ labeled TRUE} \quad (7.2)
\]

\[
\sum_{i=1}^{n} u_i x_i < 1 \quad \text{if } x \text{ labeled TRUE} \quad (7.3)
\]

WINNOW learns this halfspace with the mistake bound defined in Equation 7.1.

Our initial halfspace is \( \sum w_i x_i \geq \Theta \), where \( \forall i, w_i = 1 \). Let \( \alpha = 1 + \delta/2 \). Every time we predict negative where the real label is positive, we promote each \( w_i \) such that \( x_i = 1 \): so \( w'_i = w_i \alpha \). Every time we generate a false positive, we demote all \( w_i \) such that \( x_i = 1 \): so \( w'_i = w_i / \alpha \).

Let \( u \) be the number of promotions and \( v \) be the number of demotions performed by WINNOW. We will prove two assertions:

\[
v \leq \frac{\alpha}{\alpha - 1} * \frac{n}{\Theta} + \alpha u \quad (7.4)
\]

\[
u + v \leq \frac{\alpha}{\alpha + 1} * \frac{n}{\Theta} + (1 + \alpha)u \quad (7.5)
\]
Let \( w_i^{\text{bef}} \) and \( w_i^{\text{aft}} \) denote the values of weight \( i \) before and after, respectively, a promotion or demotion. More precisely, for a promotion, \( w_i^{\text{aft}} = w_i^{\text{bef}} + x_i(\alpha - 1)w_i^{\text{bef}} \). For a promotion to occur, we must have incorrectly predicted false, so it must be the case that

\[
\sum_{i=1}^{n} w_i^{\text{bef}} x_i \leq \Theta
\]

\( x_i = 1 - \sum_{i=1}^{n} w_i^{\text{aft}} \leq \sum_{i=1}^{n} w_i^{\text{bef}} + (\alpha - 1)\Theta \quad \text{← after promotion} \)

\[
1 - \sum_{i=1}^{n} w_i^{\text{aft}} \leq \sum_{i=1}^{n} w_i^{\text{bef}} + (1 - \frac{1}{\alpha})\Theta \quad \text{← after demotion} \]

Initially, what is the sum of the \( w_i \)'s?

\[
\sum_{i=1}^{n} w_i \leq n + u(\alpha - 1)\Theta - v(1 - \frac{1}{\alpha})\Theta \geq 0
\]

We know that the sum of the weights is always \( \geq 0 \), so

\[
v \leq \frac{\alpha}{\alpha - 1} \frac{n}{\Theta} + \alpha u
\]

This inequality is equivalent to Equation 7.4. Now, after \( u \) promotions and \( v \) demotions,

\[
\exists i. \log w_i \geq \frac{u - (1 - \delta)u}{\sum_{i=1}^{n} u_i \log \alpha}
\]

And this inequality is equivalent to Equation 7.5. Observe that each \( w_i \leq \alpha \Theta \). If we make a promotion, noting that \( \sum u_i x_i \geq 1 \) since the example is actually positive, then:

\[
w_i^{\text{aft}} = w_i^{\text{bef}} x_i
\]

We can write this equation as

\[
\sum_{i=1}^{n} u_i \log w_i^{\text{aft}} \geq \sum_{i=1}^{n} u_i \log w_i^{\text{bef}} + u_i x_i \log \alpha
\]

where \( u_i x_i \log \alpha \) is at most \( u_i x_i \).

Our exploration of WINNOW continues in the next lecture.