Winnow: Mistake Bound

Theorem 1

Let the target hypothesis be a monotone disjunction on $X = \{0,1\}^n$ *and* s *be the number of atoms in it. Winnow makes at most* $2 + 2s \lg n$ *mistakes, i.e.,*

$$\sum_{k=1}^{\infty} |r_k| \leq 2 + 2s \lg n \tag{1}$$

As $s \leq n$, we have $\sum_{k=1}^{\infty} |r_k| \leq poly(n)$, and thus Winnow learns monotone disjunctions online. It also learns them efficiently (easy to check). The $\lg n$ term makes Winnow a fast learner (compared e.g. to the perceptron) when the number of atoms s in the target disjuction is small ($s \ll n$).

To prove the theorem, we first show two lemmas.

FACULTY OF ELECTRICAL ENGINEERING CTU IN PRAGUE

Winnow: Mistake Bound (cont'd)

Lemma 1

If x_k $(k \in \mathbb{N})$ *is a* false negative *then* $\sum_{i=1}^{n} h_{k+1}^i - \sum_{i=1}^{n} h_k^i \leq \frac{n}{2}$

Proof of Lemma 1: $h_k(x_k) = 0$ since x_k is a false negative, and by (??),

$$\sum_{i=1}^{n} h_k^i x_k^i \leq \frac{n}{2} \tag{2}$$

Furthermore,

$$\sum_{i=1}^{n} h_{k+1}^i - \sum_{i=1}^{n} h_k^i = \sum_{i=1}^{n} (h_{k+1}^i - h_k^i) = \sum_{i=1}^{n} (h_{k+1}^i - h_k^i) x_k^i \tag{3}$$

where the last equality is because for any i such that $x_k^i = 0$, $h_{k+1}^i = h_k^i$ due to the Winnow learning rule.

FACULTY OF ELECTRICAL ENGINEERING CTU IN PRAGUE

Winnow: Mistake Bound (cont'd)

Due to (??),

$$\sum_{i=1}^{n}(h_{k+1}^i - h_k^i)x_k^i = \sum_{i=1}^{n}(2h_k^i - h_k^i)x_k^i = \sum_{i=1}^{n} h_k^i x_k^i$$

Therefore

$$\sum_{i=1}^{n} h_{k+1}^i - \sum_{i=1}^{n} h_k^i = \sum_{i=1}^{n} h_k^i x_k^i \leq \frac{n}{2}$$

where the inequality is given by (2). This proves the lemma.

FACULTY OF ELECTRICAL ENGINEERING CTU IN PRAGUE

Winnow: Mistake Bound (cont'd)

Lemma 2

If x_k $(k \in \mathbb{N})$ *is a* false positive *then* $\sum_{i=1}^{n} h_k^i - \sum_{i=1}^{n} h_{k+1}^i > \frac{n}{2}$

Proof of Lemma 2: $h_k(x_k) = 1$ since x_k is a false positive, and by (??),

$$\sum_{i=1}^{n} h_k^i x_k^i > \frac{n}{2} \tag{4}$$

The lemma is proven by the equation below, where the first equality is due to the Winnow learning rule as in (3), the second equality due to the same rule prescribing $h_{k+1}^i = 0$ for each i such that $x_k^i = 1$, and the last inequality is from (4):

$$\sum_{i=1}^{n} h_k^i - \sum_{i=1}^{n} h_{k+1}^i = \sum_{i=1}^{n} (h_k^i - h_{k+1}^i) x_k^i = \sum_{i=1}^{n} h_k^i x_k^i > \frac{n}{2}$$

FACULTY OF ELECTRICAL ENGINEERING CTU IN PRAGUE

Winnow: Mistake Bound (cont'd)

Lemma 3

For $\forall k \in \mathbb{N}, i \in [1; n] : h_k^i \leq n$

Proof of Lemma 3: From (??) $h_1^i = 1$. For contradiction, assume the lemma is not true and $k+1$ ($k \in \mathbb{N}$) is the smallest index for which $h_{k+1}^i > n$. Since $h_k^i \leq n$, h_k was promoted to h_{k+1} by (??), implying $x_k^i = 1$ (otherwise h_k^i would not have been promoted) and $h_k^i > n/2$ (promotion doubles the value). But then $\sum_{i=1}^{n} x^i h_k^i > n/2$ so by (??), $y_{k+1} = h_k(x_k) = 1$ so h_k^i was not promoted. This contradiction proves the lemma.

FACULTY OF ELECTRICAL ENGINEERING CTU IN PRAGUE

Winnow: Mistake Bound (cont'd)

Proof of Theorem 1: Let $\mathrm{FN_k}$ ($\mathrm{FP_k}$, respectively) be the number of false negatives (false positives) up to time k so $\mathrm{FN_k} + \mathrm{FP_k}$ is the total number of mistakes up to k. From (??), we have

$$\sum_{i=1}^{n} h_1^i = n$$

From this and Lemmas (1) and (2) we have

$$\sum_{i=1}^{n} h_k^i \leq n + \frac{n}{2}\mathrm{FN_k} - \frac{n}{2}\mathrm{FP_k} \tag{5}$$

Furthermore, since $h_1^i = 1$ and any decrease is only through elimination, which zeros the component, we have for $\forall k \in \mathbb{N}, i \in [1; n]$

$$h_k^i \geq 0 \tag{6}$$

FACULTY OF ELECTRICAL ENGINEERING CTU IN PRAGUE

Winnow: Mistake Bound (cont'd)

From (5) and (6), it holds for $\forall k \in \mathbb{N}$:

$$0 \leq \sum_{i=1}^{n} h_k^i \leq n + \frac{n}{2}\mathrm{FN_k} - \frac{n}{2}\mathrm{FP_k}$$

implying

$$\frac{n}{2}\mathrm{FP_k} \leq n + \frac{n}{2}\mathrm{FN_k}$$

and since $n > 0$, we can multiply this by $\frac{2}{n}$, obtaining

$$\mathrm{FP_k} \leq 2 + \mathrm{FN_k} \tag{7}$$

FACULTY OF ELECTRICAL ENGINEERING CTU IN PRAGUE

Winnow: Mistake Bound (cont'd)

Each promotion doubles h^i where i is one of the s indexes corresponding to the s atoms in the target disjunction. First assume $s > 0$. So after $\mathrm{FN_k}$ promotions, at least one of them was doubled $\frac{\mathrm{FN_k}}{s}$ times or more, thus for $\forall k \in \mathbb{N}, \exists i \in [1; n]$: $h_k^i \geq 2^{\mathrm{FN_k}/s}$, i.e.,

$$\lg h_k^i \geq \frac{\mathrm{FN_k}}{s}$$

From Lemma (3) we further have for $\forall k \in \mathbb{N}, i \in [1; n]$: $\lg h_k^i \leq \lg n$. Thus for $\forall k \in \mathbb{N}, \exists i \in [1; n]$

$$\frac{\mathrm{FN_k}}{s} \leq \lg h_k^i \leq \lg n \tag{8}$$

FACULTY OF ELECTRICAL ENGINEERING CTU IN PRAGUE

Winnow: Mistake Bound (cont'd)

Since we assumed $s > 0$, (8) can be written as

$$\mathrm{FN}_k \leq s \lg n$$

If on the other hand $s = 0$ then the target disjunction is tautologically false, so there are no false negatives, therefore $\mathrm{FN}_k = 0$ and the inequality is satisfied trivially. Combining this with (7) we get for the total number of mistakes

$$\mathrm{FP}_k + \mathrm{FN}_k \leq 2 + 2s \lg n$$

and since this value holds for any $k \in \mathbb{N}$, we can write

$$\sum_{k=1}^{\infty} |r_k| \leq 2 + 2s \lg n$$

which completes the proof. *(exercise problem)*

FACULTY OF ELECTRICAL ENGINEERING CTU IN PRAGUE

Generalization: Mistake Bound

Theorem 2

Let X be contingent conjunctions made of up to n variables and let the target hypothesis be a conjunction. The generalization algorithm makes at most n mistakes, i.e.,

$$\sum_{k=1}^{\infty} |r_k| \leq n \tag{9}$$

Thus the generalization algorithm learns conjunctions from contingent conjunctions online. It also learns them efficiently (easy to check).

As any Boolean tuple can be represented through a contingent conjunction, the theorem implies that the generalization algorithm learn conjunctions online from $X = \{ 0, 1 \}^n$ as well.

FACULTY OF ELECTRICAL ENGINEERING CTU IN PRAGUE

Generalization: Mistake Bound (cont'd)

Proof of Theorem 2. By (??), $x \in X$ is classified positive iff $h \subseteq x$. Let $\overline{h}$ be the target conjunction. So any $x \in X$ is a positive example iff $\overline{h} \subseteq x$.

1. $\overline{h} \subseteq h_1$ because h_1 is set to be the first *positive* example. (Without loss of generality, we start indexing from the instant immediately after receiving the first positive example, thus skipping the waiting stage.)
2. $\forall k : \overline{h} \subseteq h_k$ *implies* $\overline{h} \subseteq h_{k+1}$ If x_k is classified correctly, then $h_{k+1} = h_k$ and the implication holds trivially. If x_k is a false positive, i.e., $\overline{h} \not\subseteq x_k$, and $h_k \subseteq x_k$ then from (??) and (??), $h_{k+1} = h_k$, and again the implication holds trivially. Lastly, if x_k is a false negative, i.e., $\overline{h} \subseteq x_k$, and $h_k \not\subseteq x_k$ then from (??) $h_{k+1} = \mathrm{lgg}(h_k, x_k)$. If $\overline{h} \subseteq h_k$ then both arguments of the lgg are subsumed by $\overline{h}$ and since lgg is a least general generalization, $\overline{h} \subseteq \mathrm{lgg}(h_k, x_k)$. Thus the implication again holds.
3. $\forall k \in \mathbb{N} : \overline{h} \subseteq h_k$ – by induction using (1) and (2).

FACULTY OF ELECTRICAL ENGINEERING CTU IN PRAGUE

Generalization: Mistake Bound (cont'd)

4. On each mistake at k, $h_k \not\subseteq x_k$, i.e x_k is a false negative. This is because if x_k was a false positive, then $\overline{h} \not\subseteq x_k$ and due to (3) also $h_k \not\subseteq x$, but then x is not classified as positive. So *mistakes are made only on positive examples*.
5. On each mistake at k, h_{k+1} has strictly fewer literals than h_k. This is because due to (??), $h_{k+1} = \text{lgg}(h_k, x_k)$, and since lgg is a least general generalization, it must be that $h_{k+1} \subseteq x_k$. From (4), we have $h_k \not\subseteq x_k$. Thus some literals of h_k are not in h_{k+1}.
6. Since examples are contingent, they have at most n literals (each of the n atoms is included either as a positive or negative literal but not both) and since h_1 is the first positive example, it also has at most n literals. Due to (5), at least one literal is removed on each mistake, so the *maximum number of mistakes is n*, which completes the proof.

(*exercise problem*)

FACULTY OF ELECTRICAL ENGINEERING CTU IN PRAGUE

Online Learnability

A concept class $\mathcal{C}$ on X is (efficiently) **learnable from X online** if *there is an algorithm* that learns $\mathcal{C}$ from X (efficiently) online.

If for some hypothesis class $\mathcal{H}$, $\mathcal{C}(\mathcal{H})$ is (efficiently) learnable from X online, we say that $\mathcal{H}$ is (efficiently) learnable from X online. We have seen that

- monotone disjunctions are efficiently learnable online from truth-value assignments by the Winnow algorithm.
- conjunctions are efficiently learnable online from contingent conjunctions (incomplete observations) by the generalization algorithm.

From each of these two results, learnability of general *disjunctions*, which are also called **clauses**, can be proven. We will look at two techniques to achieve that.

FACULTY OF ELECTRICAL ENGINEERING CTU IN PRAGUE

Attribute Expansion

(Efficient) online learnability of *monotone* disjunctions from $X = \{ 0, 1 \}^n$ by an algorithm implies the same for clauses:

- Use the algorithm with $X' = \{ 0, 1 \}^{2n}$, presenting to it each x as

$$x'(x) = x^1, x^2, \ldots, x^n, 1 - x^1, 1 - x^2, \ldots, 1 - x^n$$

Reminder: superscripts are component indexes, not powers!

- The disjunction h' learned from X' is monotone but corresponds to the (non-monotone) disjunction

$$h = \bigvee_{\substack{i \leq n \\ \mathrm{p}_n \in \mathsf{Lits}(h')}} \mathrm{p}_i \bigvee_{\substack{i > n \\ \mathrm{p}_n \in \mathsf{Lits}(h')}} \neg \mathrm{p}_i$$

on X, i.e., $\forall x \in X : h(x) = h'(x'(x))$.

(*Exercise problem*)

FACULTY OF ELECTRICAL ENGINEERING CTU IN PRAGUE

Concept Inversion

(Efficient) online learnability of *conjunctions* from any X implies the same for clauses.

- Use the algorithm with inverted policy h', i.e.

$$h'(x) = 1 - h(x)$$

- When $h = \mathcal{L}_1 \wedge \mathcal{L}_2 \wedge \ldots \wedge \mathcal{L}_s$ (where $\mathcal{L}_i$ are literals) is a conjunction on X, i.e., then $h'(x)$ is the policy prescribed by

$$\neg h = \neg\mathcal{L}_1 \vee \neg\mathcal{L}_2 \vee \ldots \vee \neg\mathcal{L}_s$$

  which is a clause.

The reverse implication can be shown with the same reasoning.

FACULTY OF ELECTRICAL ENGINEERING CTU IN PRAGUE

Online Learnability of Conjunctions and Clauses

With the two techniques, we can prove additional learnability results:

1. *clauses* are efficiently learnable online from $X = \{ 0, 1 \}^n$. (Proof: use Winnow + attribute expansion.)
2. *Conjunctions* are efficiently learnable online from $X = \{ 0, 1 \}^n$. (Proof: use Winnow + attribute expansion + concept inversion).
3. *clauses* are efficiently learnable online from $X =$ *contingent conjunctions*. (Proof: use generalization + concept inversion.)

Since a truth-value tuple can be represented by a contingent conjunction, assertion 2 already follows from the mistake bound of the generalization algorithm and assertion 3 implies assertion 1. However, Winnow used in 1 and 2 gives a better bound ($\mathcal{O}(\lg n)$) than generalization ($\mathcal{O}(n)$).

FACULTY OF ELECTRICAL ENGINEERING CTU IN PRAGUE

s-DNF

An **s-conjunction** is a conjunction with at most s literals. An **s-DNF** is a disjunction of s-conjunctions.

For example, the "accepted form of authentication" concept description

$$\text{password} \vee (\text{fingerprint} \wedge \text{pin}) \vee (\text{facescan} \wedge \text{pin})$$

or the "accepted form of payment" concept description

$$\text{cash} \vee (\text{creditcard} \wedge \neg\text{expired})$$

are both 2-DNF but not a 1-DNF.

We will use the name s-DNF also to denote the *hypothesis class* of s-DNF's.

FACULTY OF ELECTRICAL ENGINEERING CTU IN PRAGUE

Online Learnability of s-DNF

Let $c_1, c_2, \ldots c_{n'}$ be all s-conjunctions on n variables. Using a variant of the attribute expansion technique, we can reduce learning s-DNF from $X = \{ 0,1 \}^n$ to learning monotone disjunctions from $X' = \{ 0,1 \}^{n'}$.

- Use an algorithm for learning a monotone disjunction on n' variables, presenting to it each $x \in X$ as $x'(x) = x'^1, x'^2, \ldots x'^{n'}$ where

$$x'^i(x) = 1 \text{ iff } x \models c_i \tag{10}$$

- The hypothesis h' learned from x' is a monotone disjunction but corresponds to the s-DNF

$$\bigvee_{\{ i | \mathrm{p}_i \in \mathsf{Lits}(h') \}} c_i$$

on X, i.e., $\forall x \in X : h(x) = h'(x'(x))$.

FACULTY OF ELECTRICAL ENGINEERING CTU IN PRAGUE

Online Learnability of s-DNF (cont'd)

Assume we used an algorithm with mistake bound $poly(n')$. So s-DNF is learnable online from $X = \{0,1\}^n$ if $n' \leq poly(n)$.

The number n' of all s-conjunctions on n variables is the sum of the number of conjunctions with exactly $0, 1, \ldots s$ literals and the number of literals is twice the number of atoms, i.e. $2n$:

$$1 + \binom{2n}{1} + \binom{2n}{2} + \ldots + \binom{2n}{s} = \mathcal{O}(n^s) \leq poly(n)$$

Here we used an exponential upper bound for binomial coefficients.

So the class s-DNF (for any constant $s \in \mathbb{N}$) is efficiently learnable online from $X = \{0,1\}^n$. Determining (10) for each of the n' conjunctions takes linear time (verify) so s-DNF is also learnable *efficiently*.

FACULTY OF ELECTRICAL ENGINEERING CTU IN PRAGUE