MB Learning : Gaurantees a # of mistakes at most.

Learner   Teacher

← Data point 01
guess classification →
← correct or incorrect
← data point 02
guess classification →
← correct or incorrect

data is fed one at a time.

counter : # of mistakes increases by 1.

Learner will update internal state.

<u>Mistake-Bounded Learner</u> :

☆ we say a learner has mistaken bound t if for every sequence of challenges, learner makes at most t mistakes, guaranteed.

<u>Example</u> :

C = { monotone disjunctions ("or") on n variables }

↑ class — monotone: does not have negations in the literals.

Domain = $\{0, 1\}^n$ = bit string of length n.

$x_1 \vee x_3$ (disjunction = "$\vee$" = another way of saying "or")

$f(x) = x_1 \vee x_7 \vee x_9$. 1 if $x_1, x_7, x_9 = 1$.

$f \in C$: f is a monotonic disjunction.

Learner: Teacher:

does not know the f that is picked, trying to learn.

$\xleftarrow{x \in \{0,1\}^n}$

$\xrightarrow{0 \text{ or } 1}$

$\xleftarrow{\text{teacher says}}$ "correct" if guess $= f(x)$, "mistake" otherwise.

Question: can you come up with a learner/algorithm with mistake bound at most n?

n = # of literals we have.

Learner will start with the monotone disjunction.

$X_1 \vee X_2 \vee X_3 \vee \cdots \vee X_n$.

Guess: Initial state $= X_1 \vee \cdots \vee X_n$ True as long as not all are 0.

Learner   Teacher

$\xleftarrow{0110010}$

$\xrightarrow{1 \text{ (true)}}$

$\xleftarrow{\text{mistake}}$

Every time a mistake is made, we update our monotone disjunction:

X_2 cannot be in the unknown disjunction.

X_3 "

X_6 "

update state: $X_1 \vee \cancel{X_2} \vee \cancel{X_3} \cdots \vee \cancel{X_6} \cdots \vee X_n$.

Key point:

Every time we make a mistake, at least 1 literal is eliminated. At most n literals. Therefore:

$\Rightarrow \boxed{\# \text{ of mistakes } \leq n.}$

Next case: disjunctions.

$C = \{\text{disjunctions}\}$ (as opposed to monotone disjunctions).

$f = x_1 \vee \overline{x_2} \vee x_7 \vee \overline{x_9}$

negations

Question: How can we use the algorithm for monotone disjunction to learn disjunctions?

$x_1, \cdots, x_n \in \{0, 1\}^n$.

use feature expansion.

$\underbrace{x_1, \cdots, x_n}_{n} \xrightarrow{\text{map}} \underbrace{x_1, \cdots x_n, y_1, \cdots, y_n}_{2n}$.

Each $y_i = \overline{x_i}$ (negation of x_i)

Example: $\underbrace{0\ 1\ 1\ 0}_{n=4} \longrightarrow \underbrace{0\ 1\ 1\ 0 \,\vdots\, 1\ 0\ 0\ 1}_{n=8.}$

$f(x_1, \cdots, x_n) = x_2 \vee \overline{x_4} \vee x_7$ (has negated variables)

$\Rightarrow f(x_1 \cdots, x_n, y_1, \cdots, y_n) =$

$x_2 \vee y_4 \vee x_7$ (no negated variables).

using algorithm for monotone disjunction, we have a new algorithm with mistake bound $\leq 2n$, for arbitrary disjunctions.