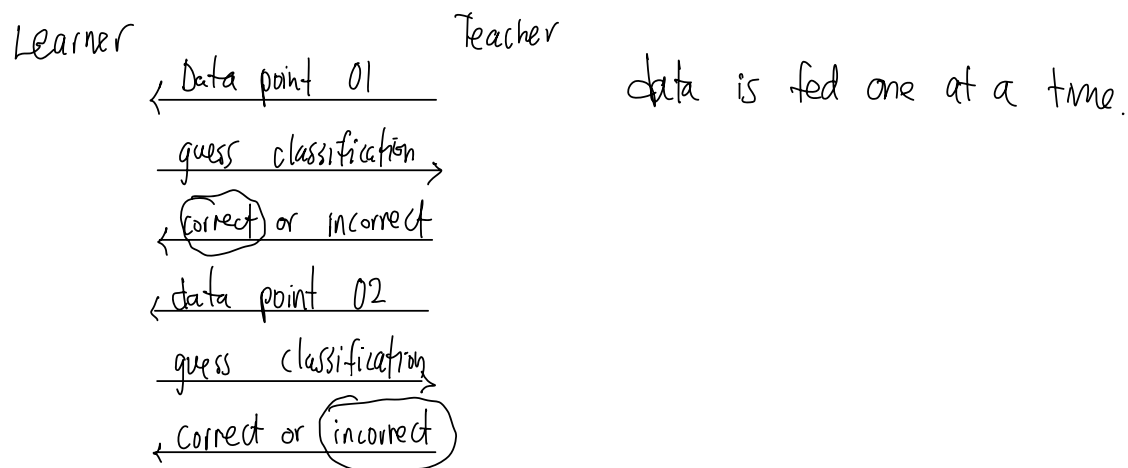


MB Learning : Guarantees a # of mistakes at most.



Counter : # of mistakes increases by 1.

Learner will update internal state.

Mistake-Bounded Learner :

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

Example :

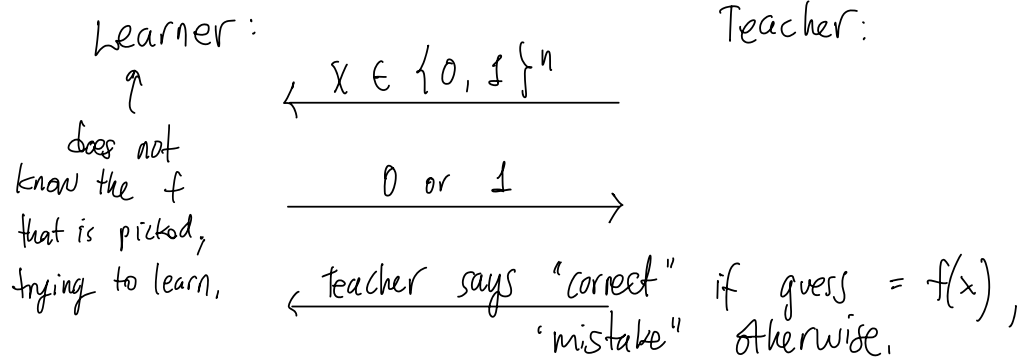
$C = \left\{ \begin{array}{l} \text{monotone disjunctions on } n \text{ variables} \\ \uparrow \text{ class} \quad \uparrow \text{ does not have negations in the literals.} \end{array} \right\}$ "or"

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.

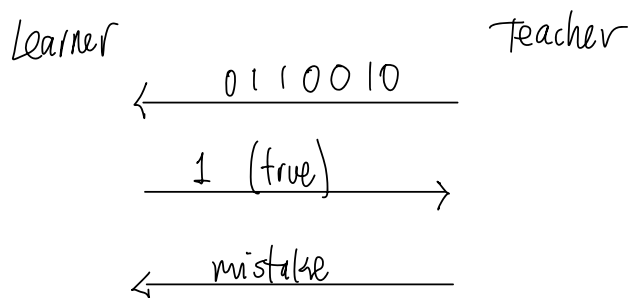


Question: can you come up with a learner/algorithm with mistake bound at most n ?
 \uparrow # of literals we have.

learner will start with the monotone disjunction.

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

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



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

$\left\{ \begin{array}{ll} X_2 & \text{cannot be in the unknown disjunction,} \\ X_3 & " \\ X_6 & " \end{array} \right.$

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

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).

$$f = x_1 \vee \bar{x}_2 \vee x_7 \vee \bar{x}_9$$

\uparrow negations \rightarrow

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

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

use feature expansion.

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

Each $y_i = \bar{x}_i$ (negation of x_i)

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

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

$$\equiv f(x_1, \dots, x_n, y_1, \dots, y_n) = x_2 \vee y_4 \vee x_7 \quad (\text{no negated variables}).$$

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