THEOREM: $E[L(T)] \leq (1+\varepsilon) L_*(T) + \frac{\ln d}{\varepsilon}$

$$E[\text{Regret}(T)] \leq \varepsilon L_*(T) + \frac{\ln d}{\varepsilon}$$

PROOF:

Main Idea:

Track $w(t) = \sum_{i=1}^{d} w(t,i)$

Remember:

Expert 1: $w(0,1) = 1$

Day 1: $w(1,1) = (1-\varepsilon)^{L(1,1)}$

Day 2: $w(2,1) = (1-\varepsilon)^{L(1,1) + L(2,1)}$

$$\boxed{w(T,i) = (1-\varepsilon)^{\sum_{t=1}^{T} L(t,i)}}$$

$L(t) = E[\text{loss incurred on day } t]$

$$= \sum_{i=1}^{d} \Pr[\text{we pick expert } i] \cdot L(t,i)$$

$$= \sum_{i=1}^{d} \frac{w(t-1,i)}{\underbrace{\sum_{j=1}^{d} w(t-1,j)}_{\longrightarrow\, w(t-1)}} \cdot L(t,i)$$

$$\boxed{L(t) = \frac{1}{W(t-1)} \cdot \sum_{i=1}^{d} w(t-1,i) \cdot L(t,i)} \quad - (1)$$

$$W(t) = \sum_{i=1}^{d} w(t,i)$$

$$= \sum_{i=1}^{d} w(t-1,i) \cdot \left(1 - \varepsilon \cdot L(t,i)\right)$$

$$= \sum_{i=1}^{d} w(t-1,i) - \varepsilon \sum_{i=1}^{d} w(t-1,i) \cdot L(t,i)$$

$$= W(t-1) - \varepsilon \cdot W(t-1) \cdot L(t) \quad [\text{From } (1)]$$

$$W(t) = W(t-1)\left(1 - \varepsilon L(t)\right)$$

Idea : Compare total weight upper and lower bounds as before.

CLAIM: $1 - x \leq e^{-x} \quad \forall x$

$$W(t) = W(t-1)\left(1 - \varepsilon L(t)\right) \leq W(t-1) \cdot e^{-\varepsilon L(t)}$$

Therefore

$$w(T) \le w(0) \cdot e^{-\varepsilon L(1)} \cdot e^{-\varepsilon L(2)} \cdots\cdots e^{-\varepsilon L(T)}$$

$$= w(0) \cdot e^{-\varepsilon \underbrace{(L(1) + L(2) + \cdots + L(T))}_{\text{Our total loss}}}$$

$$= w(0) \cdot e^{-\varepsilon \underbrace{A(T)}}$$

$$A(T) \equiv \text{Our total expected loss.}$$

Claim: $w(T) \geq (1-\varepsilon)^{L_*(T)} \cdot 1$

Therefore,

$$(1-\varepsilon)^{L_*(T)} \le d \cdot e^{-\varepsilon A(T)}$$

$$L_*(T) \ln(1-\varepsilon) \le \ln d - \varepsilon A(T)$$

$$\varepsilon A(T) \le (-\ln(1-\varepsilon)) \cdot L_*(T) + \ln d$$

$$A(T) \le \left(\frac{-\ln(1-\varepsilon)}{\varepsilon}\right) L_*(T) + \frac{\ln d}{\varepsilon}$$

CLAIM: $$\frac{-\ln(1-x)}{x} \le 1+x \quad \text{if} \quad x < 1/2$$

Therefore, as long as $\varepsilon < 1/2$, we get

$$A(T) \le (1+\varepsilon) L_*(T) + \frac{\ln d}{\varepsilon}.$$

COROLLARY:

Setting $\varepsilon = \sqrt{\frac{\ln d}{T}}$, we get

$$A(T) \le L_*(T) + 2\sqrt{T \ln d}$$

$$\Downarrow$$

We have a "No-Regret" algorithm.

→ This is the best possible regret. (Cannot beat $\Omega(\sqrt{T})$)

→ $L(t, i) \in [0, 1]$

Applications

- Learning
  - Online Learning
  - Portfolio Management
  - Graphical Models
- Solving linear programs
- Solving Semi-definite programs.

BOOSTING:

→ Goal is to get 90% accuracy (Want)

→ We can get 60% accuracy (Have)

Meta-Question: Can we boost "weak-learners" to strong learners.

BOOSTING ON SAMPLES:

Dataset : $(x^1, y^1), (x^2, y^2), \ldots, (x^d, y^d) \in \underbrace{X}_{\text{Domain}} \times \underbrace{L}_{\text{Labels}}$

Hypothesis class H

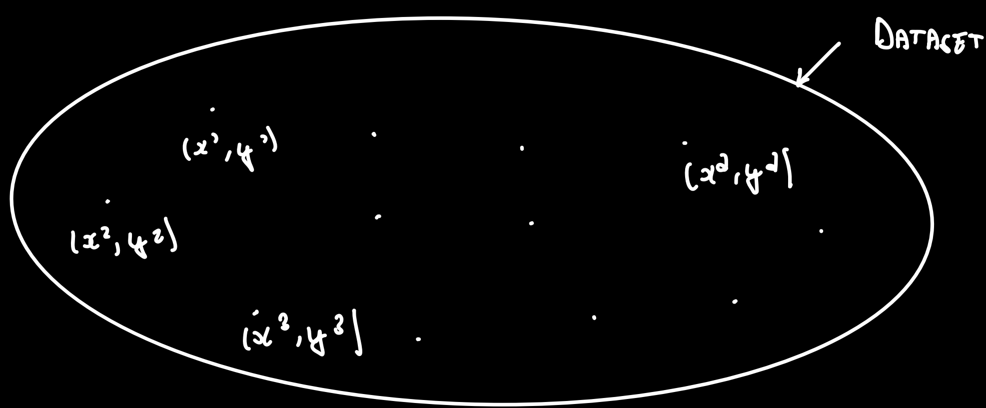
$$\boxed{\begin{array}{l}\text{Weak Learner: For every distribution } D \text{ on the dataset, we can} \\ \text{find a } h \in H, \ \Pr_{(x^i, y^i) \sim D}[h(x^i) \neq y^i] \leq \gamma \\ \qquad\qquad\qquad\qquad\qquad (\text{say } \gamma = 0.4)\end{array}}$$

Goal: Combine a few of these weak-learners to get error $\leq \delta$ (say 0.01).

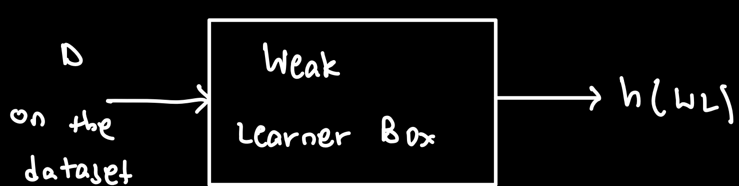
→ Was asked in 1970s by Valiant.

→ Schapire (1989) solved it.

→ Freund and Schapire (1995) gave a practical algorithm (AdaBoost).



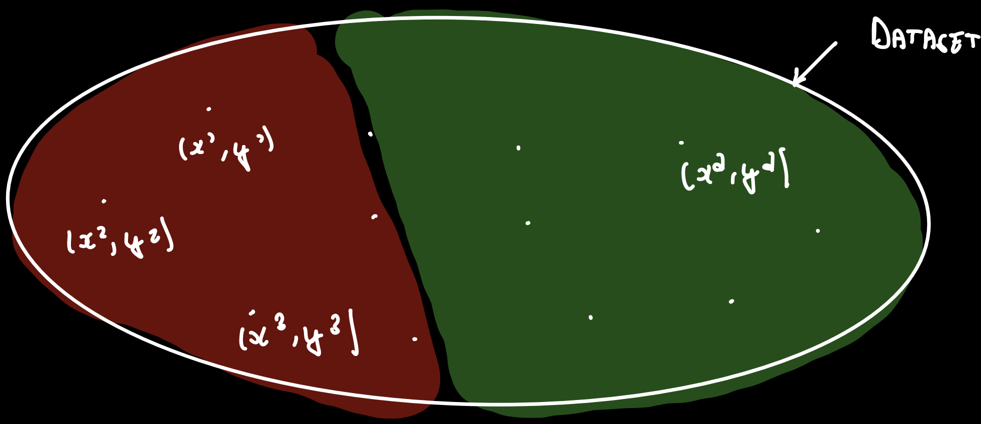
What do we have:



→ Step 1: $D^{(0)} \equiv$ uniform on the whole dataset.

$h^{(0)} = WL(D^{(0)})$

→ Idea: Put more "weight" on the points that you were wrong on before.



- Start with $D^{(0)} \equiv \underbrace{\left(1/d ,\ 1/d ,\ 1/d , \dots ,\ 1/d \right)}$

distribution on the d points in the dataset.

- $h^{(0)} = WL(D^{(0)})$

- For $t = 1, \dots, T$:

  → update $D^{(t-1)}$ to $D^{(t)}$

  → $h^{(t)} = WL(D^{t})$.

- Output $h_{strong} \equiv \text{combine}(h^0, h^1, \dots, h^T)$

Labels $\equiv \{0,1\}$

Idea: Combiner $\equiv$ MAJORITY

Update distributions using MWM.

ADABOOST

$\rightarrow D^{(0)} \equiv (1/d, 1/d, \ldots, 1/d)$

$w(0,i) = 1$ for $i = 1, 2, \ldots, d$

$\rightarrow h^{(0)} = WL(D^{(0)})$

For $t = 1, \ldots, T$:

- Define

$$w(t,i) = \begin{cases} (1-\varepsilon)\, w(t-1,i) & \text{if correct} \\ w(t-1,i) & \text{if wrong} \end{cases}$$

"correct" means $h^{t-1}(x^i) = y^i$

- $D^{(t)}$ = distribution proportional to weights

- $h^t = WL(D^t)$.

- Output $h = MAJ(h^0, h^1, h^2, \ldots, h^{T-1})$

Imaginary "Learning with Experts Game"

	(x^1,y^1)	(x^2,y^2)	$\dots$	(x^d,y^d)
h^0	✓	✓	× × ✓ ×	×
Loss	1	1	0 0 1 0	0
h^1	✓	✓	× × ✓ ×	×
Loss	1	1	0 0 1 0	0
			$\vdots$	
h^{T-1}				

↓

$h(x^1) = y^1$ if there are more than $T/2$ 1's in the column.

Theorem:

Adaboost achieves accuracy $1-\delta$ on the dataset if

$$T \geq \frac{2\ln(1/\delta)}{(1/2 - \gamma)^2}$$

(For example, $\gamma = 0.4$, $\delta = 0.1$

$$\frac{2 \cdot \ln(10)}{(0.1)^2} \approx 600.$$

Why does Adaboost work?

Imagine Adaboost fails to get $1-\delta$ accuracy

$\Rightarrow$ We have at least $d \cdot \delta$ examples where MAJORITY were WRONG!

$\Rightarrow$ "Sum of losses" on that column is $< T/2$.

$\Rightarrow$ if MAJORITY is wrong on example i, then

$$w(T,i) \geq (1-\varepsilon)^{T/2}$$

$$d \cdot \delta \cdot (1-\varepsilon)^{T/2} \leq \sum_{i=1}^{d} w(T,i) \leq d e^{-\gamma T}.$$

$\downarrow$

Assuming Adaboost failed.

ANALYSIS OF BOOSTING:

WL:

$$\Pr_{(x^i, y^i) \sim D}[h(x^i) = y^i] \geq 1 - \gamma$$

THEOREM:

ADABOOST after $T \geq \dfrac{2\ln(1/\delta)}{(1/2 - \gamma)^2}$ rounds achieves

accuracy $(1-\delta)$.

PROOF IDEA: Track $\sum_{i=1}^{d} w(i,t)$ as a function of t.

Remark: ADABOOST is like running a fictional learning with experts game where

$$L(i,t) = \begin{cases} 1 & \text{if } h^t(x^i) = y^i \\ 0 & \text{else.} \end{cases}$$

CLAIM:

$$w(t) \leq w(t-1) \cdot (1-\varepsilon)^{L(t)}$$

expected "loss" of our algorithm.

CLAIM:

$L(t)$: Expected loss of our algorithm.

$$L(t) \geq \text{some value}$$

PROOF:

$$L(t) = E[\text{loss we incur}]$$

$$= \sum_{i=1}^{d} \Pr[i \text{ is picked}] \cdot L(i,t)$$

$$= \sum_{i=1}^{d} \Pr[i \text{ is picked when using distribution } D^{t-1}] \cdot \mathbb{1}(h^t(x^i) = y^i)$$

output using weak learner

$$\geq 1 - \gamma \qquad (\text{because } h^t = WL(D^{t-1}))$$

CLAIM:

$$w(t) \leq w(t-1) \cdot (1-\varepsilon)^{L(t)} \leq w(t-1)(1-\varepsilon)^{(1-\gamma)}$$

CLAIM:

$$w(t) \leq w(t-1) \cdot e^{-\varepsilon(1-\gamma)}$$

So $w(T) \leq w(0) e^{-\varepsilon T(1-\gamma)}$

	(x^1,y^1)	(x^2,y^2)	. . .	(x^d,y^d)
h^0	✓	✓	× × ✓ ×	×
Loss	1	1	0 0 1 0	0
h^1	✓	✓	× × ✓ ×	×
Loss	1	1	0 0 1 0	0
			⋮	
h^{T-1}				

$h \equiv \mathrm{MAJ}(h^0, h^1, \dots h^{T-1})$

Let Bad = All examples where the majority is wrong!

For every i in BAD, we must have the total "loss"

$\left(= \sum_{t=1}^{T} L(i,t)\right)$ is at most $T/2$.

CLAIM:

$$w(i,T) = (1-\varepsilon)^{\sum_{t=1}^{T} L(i,t)}$$

$\Rightarrow$ for every bad index i, $w(i,T) \geq (1-\varepsilon)^{T/2}$

We have:

$$\sum_{i \in BAD} w(i,T) \leq \sum_{i=1}^{d} w(i,T)$$

$$= w(T) \leq w(0) \cdot e^{-\varepsilon T(1-\delta)}$$

$$|BAD| \cdot (1-\varepsilon)^{T/2} \leq d \cdot e^{-\varepsilon T(1-\delta)}$$

$$\left(\frac{|BAD|}{d}\right) \leq e^{-\varepsilon T(1-\delta)} \cdot (1-\varepsilon)^{-T/2}$$

Recall the inequality

$$\frac{-\ln(1-\varepsilon)}{\varepsilon} \leq 1+\varepsilon$$

$$\frac{\ln\left(\frac{1}{1-\varepsilon}\right)}{\varepsilon} \le 1+\varepsilon$$

$$\ln\left(\frac{1}{1-\varepsilon}\right) \le \varepsilon(1+\varepsilon)$$

$$\left(\frac{|BAD|}{d}\right) \le e^{-\varepsilon T(1-\gamma)} \cdot \left(\frac{1}{1-\varepsilon}\right)^{T/2}$$

$$\le e^{-\varepsilon T(1-\gamma)} \cdot e^{\frac{\varepsilon(1+\varepsilon)T}{2}}$$

$$= e^{-\varepsilon T\left((1-\gamma) - 1/2 - \varepsilon/2\right)}$$

$$= e^{-\varepsilon T\left((1/2-\gamma) - \varepsilon/2\right)}$$

Recall:

We get to choose ε. So set $\varepsilon = (1/2 - \gamma)$

$$\frac{|BAD|}{d} \le e^{-(1/2-\gamma)T \cdot 1/2(1/2-\gamma)}$$

$$= e^{\frac{-T(1/2-\gamma)^2}{2}}$$

So each round, proportion of bad examples decreases exponentially.

So, if $T \geq \dfrac{2\ln(1/\delta)}{(1/2-\gamma)^2}$

Then $\dfrac{|BAD|}{d} \leq \delta$

Summary:

Boosting is possible.

→ Is possible with a very practical algorithm: ADABOOST.

→ No regret algorithms are very powerful.

→ We can use learning with experts / MWM for problems that have nothing to do with online learning!

→ "Private" algorithms

→ Graphical Models.

MWM is quite useful when you have to come up with clever distributions.