

Ch 21 Online Learning

Friday, March 27, 2020 3:05 PM

In Shalev-Shwartz + Ben-David, ch 21 Online Learning is first ch of "Part 3: Additional Learning Models"
(21: online learning, 22: clustering, 23: dimensionality reduction, 24: generative models (MLE, Bayes), 25: Feature selection)
most similar to PAC... mostly orthogonal ideas, or (eg ch 24) you'd see in a standard Stat. class

Online Learning

Before, we had training data, then (phase 2) could apply classifier to test data

With online learning, no separate phases: train and test on same data

(many things naturally operate this way, e.g., SPAM filters, medical knowledge, ...)

Also, theory can be completely distribution free: instead, allow it to change, or even be adversarial

We'll have to change how we measure "learning" (don't use risk L_0 anymore)

↳ Connections to game theory

Many similarities to PAC learning though (notions similar to VC-dimension, agnostic or not, online-to-batch conversion)

Popular recently ('05-'15 very hot)

Many online methods are cheap to implement, i.e., can deal w/ data streams

History:

Hannan '57, Rosenblatt '58, Novikoff '62

Modern start w/ Littlestone + Warmuth '89

Plan: ① Binary classification → realizable
 ↳ agnostic

② Regression or Surrogate loss (need convexity)

1a) Online Classification, realizable (focus on learning, not computational complexity)

Setup: T rounds ($t \in [T]$), we assume $T \in \mathbb{N}$ but will take $\lim_{T \rightarrow \infty}$ sometimes

Each round, observe data/features X_t

you (or the algo.) predicts p_t ... then true answer y_t is revealed (for now, $y_t \in \{0, 1\}$)

Goal: make as few mistakes as possible (i.e. 0-1 loss)

So... want a good prediction now and to learn so we make a good prediction in the future

(general examples: • SPAM filtering

• Restaurant problem (we'll see again in Reinforcement Learning)

↳ Example of "reward": $y_t =$ whether you liked the food K_t

How are x_t and y_t generated? Deterministic, Stochastic, adversarial ↗ our analysis, since distribution-free, worst-case

Of course, need some assumptions

(otherwise, choose $y_t = \neg p_t$ and it's hopeless)

Realizable Case Assume (x_t) is arbitrary/adversarial, but $\exists h^* \in \mathcal{H}$ st. $(\forall t) y_t = h^*(x_t)$

Def $M_A(\mathcal{H})$ is the maximum # of mistakes ($p_t \neq y_t$) made by algorithm A on a sequence of data $(x_t, y_t)_{t=1}^T$ over all (x_t) and all $h^* \in \mathcal{H}$

Def Given $S = \{(x_1, h^*(x_1)), \dots, (x_T, h^*(x_T))\}$, $h^* \in \mathcal{H}$, then $M_A(S)$ is # mistakes algo A makes on S, and $M_A(\mathcal{H})$ is supremum of $M_A(S)$ over all such S (arbitrary length), and a bound of the form $M_A(\mathcal{H}) \leq B$ ($\forall T$) is a mistake bound, and \mathcal{H} is online learnable if \exists algo A w/ a mistake bound for \mathcal{H}

Warmup: let $|\mathcal{H}| < \infty$ (like we did for PAC learning)

For PAC learning, our main tool was Empirical Risk Minimization (ERM)

We have something similar:

Algorithm: "Consistent" $|\mathcal{H}| < \infty$

Init. : $V_1 = \mathcal{H}$
 for $t=1, 2, \dots, T$ ↗ so this is an ERM for $S = \{(x_1, y_1), \dots, (x_{t-1}, y_{t-1})\}$
 receive x_t
 choose any $h \in V_t$, predict $p_t = h(x_t)$
 receive true label y_t ↗ consistent
 update (prune) $V_{t+1} = \{h \in V_t : h(x_t) = y_t\}$

Corollary 21.2 A = Consistent then $M_A(\mathcal{H}) \leq |\mathcal{H}| - 1$

prof Every time we make a mistake, $|V_{t+1}| \leq |V_t| - 1$

So for M mistakes,

$$1 \leq |V_t| \leq |\mathcal{H}| - M$$

by realizability \square

Is that good? No. Easy fix:

Algorithm: "Hoeffding" $|\mathcal{H}| < \infty$

Init. : $V_1 = \mathcal{H}$
 for $t=1, 2, \dots, T$
 receive x_t ↗ i.e. pick majority vote
 predict $p_t = \operatorname{argmax}_{r \in \{0,1\}} \left| \{h \in V_t : h(x_t) = r\} \right|$
 receive truth $y_t = h^*(x_t)$
 update $V_{t+1} = \{h \in V_t : h(x_t) = y_t\}$ ↗ same consistent update

Theorem 21.3 $M_{\text{Hoeffding}}(H) \leq \log_2(|H|)$

Proof Everytime we make a mistake, we can reduce size of V_t a lot, since at least half got it wrong

$$|V_{t+1}| \leq \frac{1}{2}|V_t| \quad \text{so} \quad 1 \leq |V_{T+1}| \leq |H|/2^M$$

by realizability

General case (H infinite is OK)

We want a measure of complexity of H , analogous to VC dimension

(Recall: $\text{VCdim}(H) \geq m \Rightarrow H$ can shatter m points, i.e., realize all possible dichotomies)

Analogy/example ("A theory that explains everything, explains nothing")

I create a new physics theory. Of course, it involves some unknown parameters, so in fact I actually have a class of theories $H = \{h_1, \dots, h_k\}$.

What's the fewest number of experiments needed for you to falsify my theory? (I am your adversary)
or, you are my adversary

Also, I am unscrupulous (or, "you can make the data say whatever you want")

i.e., you propose an experiment X_t with outcome $\neg P_t$, then I claim $y_t = P_t$

... but I have to be self-consistent (if you ask X_k again, I can't change my answer)

and I've claimed $y_t = h^*(x_t)$ for some $h^* \in \{h_1, \dots, h_k\}$

You need at least $\log_2(k)$ experiments before you could prove that I'm faking it (or that my theory is false)

(more generally)

Get in the mindset of an adversary.

Your annoying sibling has 1 piece of candy, puts it behind their back, and asks you to choose which hand

You choose "left hand". Probability you're correct? ~~50%~~? 0%.

Because they're your siblings, they cheat as much as possible without you being able to prove it

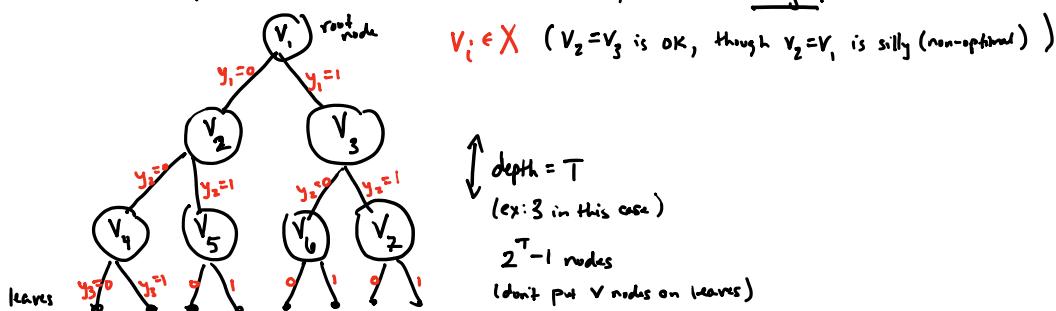
(Notes written during coronavirus quarantine w/ two daughters who just got Easter candy)

Our notion will be Ldim(H) (after Nick Littlestone)

Think of learner vs. environment: environment chooses x_t , learner chooses p_t , then environment chooses y_t

In fact, we only care about rounds where we make mistakes, so let $y_t = \neg p_t$ (i.e. $\neg p_t = 1 - p_t$) but must be realizable by some $h \in H$

To characterize what an adversarial environment can do, make a binary tree



Environment chooses X_t (i.e. V_i), but you choose P_t (so force $y_t = 1 - p_t$ if it's a mistake)
so you choose how to traverse the tree from root to leaf.

$$V_{i_2} = X_{i_2} \dots$$

Def A shattered tree of depth T is a tree w/ nodes $V_1, V_2, \dots, V_{2^{T-1}}$ such that any path from root to leaf (via nodes $(V_{i_1}, V_{i_2}, \dots, V_{i_T})$ w/ labels (y_1, \dots, y_T)) can be realized by some $h \in H$ s.t. $h(X_{i_t}) = y_t \quad \forall t \in [T]$.

↳ i.e. defeats all learners,
or, all learners make at least T mistakes

Def $Ldim(H)$ is the maximal integer T s.t. \exists a shattered tree of depth T
environment's strategy

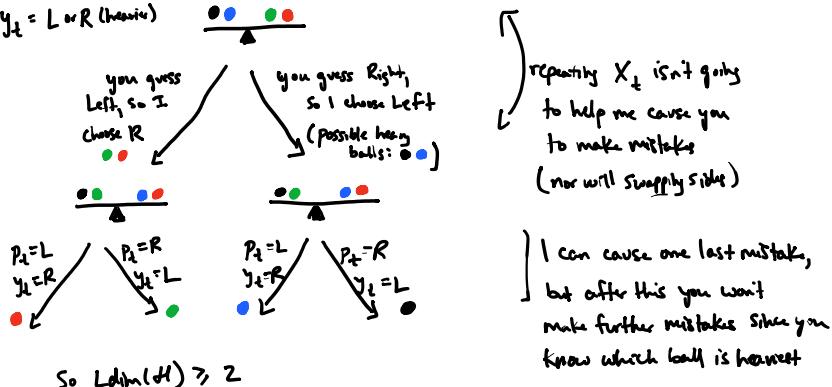
Immediately proves

Lemma 21.6: \forall algo A, $M_A(H) \geq Ldim(H)$

Ex 4 balls  3 identical, 1 is heavier i.e. $|H| = 4$

X_d = Weighing partition

$y_t = L \text{ or } R$ (heavier)



So $Ldim(H) \geq 2$

Ex $Ldim(H) = \log_2(|H|)$ if H finite

(either just see it, or prove via HALVING Algo.)

Ex Unit vectors $X = \{e_1, \dots, e_d\}$, $H = \{h_1, \dots, h_d\}$ $h_j(x) = \begin{cases} 1 & x = j \\ 0 & x \neq j \end{cases}$

then $Ldim(H) = 1$

Why? take $y_1 = 1$, wlog (due to symmetry)

If algo says $p_1 = 1$, then no mistake and its learned h

If algo says $p_1 = 0$, they make a mistake, but they see $y_1 = 1$ and learn h

Either way, the algo won't make any more mistakes

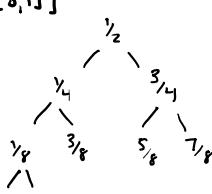
Since d can be arbitrarily large, this shows

$Ldim(H) \ll |H|$ is possible.

Ex $X = [0, 1] \subseteq \mathbb{R}$, $H = \left\{ \frac{1}{[x \leq a]} : a \in [0, 1] \right\}$

Recall $V_Cdim(H) = 1$

b.t $Ldim(H) = \infty$. Choose nodes



$x_1 = \frac{1}{2}$, you guess 1 ($x^* \leq \frac{1}{2}$)

so adversary decides $x^* > \frac{1}{2}$

$x_2 = \frac{3}{4}$, you guess 0 ($x^* > \frac{3}{4}$)

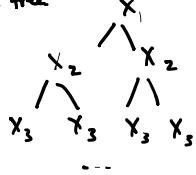
so adversary decides $x^* \leq \frac{3}{4}$

you'll make unbounded
of mistakes.

Ldim vs Vcdim: $\forall H, Vcdim(H) \leq Ldim(H)$, but $<$ possible

Proof: let $d = Vcdim(H)$, and (x_1, \dots, x_d) a shattered set

make the tree



Now, we know $M_A(H) \geq Ldim(H)$. What about $\exists A$ s.t. $M_A(H) = Ldim(H)$?

Yes, constructive (generalize HALVING):

Algo: "Standard Optimal Algorithm" (SOA) for H ($|H| = \infty$ is ok)

Initialize $V_1 = H$

for $t = 1, 2, \dots, T$

receive x_t

predict $p_t = \underset{r \in \{0,1\}}{\operatorname{argmax}} Ldim(\{h \in V_t : h(x_t) = r\})$

receive true label y_t

update $V_{t+1} = \{h \in V_t : h(x_t) = y_t\}$

Lemma 21.7 $M_{SOA}(H) \leq Ldim(H)$

Proof If we make a mistake, $Ldim(V_{t+1}) \leq Ldim(V_t) - 1$. wlog, say we chose $p_t = 0$ (so actually $y_t = 1$)

If not, then $Ldim(V_{t+1}) = Ldim(V_t)$

$\downarrow = V_t^{(1)}$ by update rule

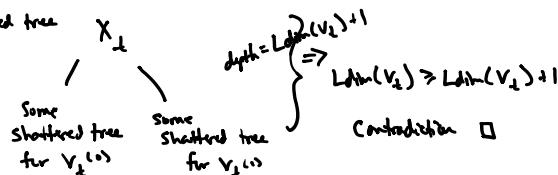
But also $Ldim(V_t) \geq Ldim(V_t^{(0)}) \geq Ldim(V_t^{(1)}) \geq Ldim(V_t)$

$Ldim(A) \geq Ldim(B)$ by how we chose r

if $A \equiv B$

but then if $Ldim(V_t^{(0)}) = Ldim(V_t^{(1)}) = Ldim(V_t)$

can make a new shattered tree



Corollary H has a mistake bound iff $Ldim(H) < \infty$

(16) Online classification, unrealizable case

If we were realizable, it's possible to have a bounded # of mistakes even as $T \rightarrow \infty$

Not true if unrealizable, so need new measure of learning

Want our algo competitive with best fixed predictor, i.e., "regret"

$$\text{Def} \quad \text{Regret}_A(h, T) = \sup_{(x_1, y_1), \dots, (x_T, y_T)} \sum_{t=1}^T |P_t - y_t| - \sum_{t=1}^T |h(x_t) - y_t|$$

this is for ℓ^1 loss,
but can be defined
more generally for
an arbitrary loss

$$\begin{aligned} \text{Def} \quad \text{Regret}_A(H, T) &= \sup_{h \in H} \text{Regret}_A(h, T) = \sup_h \sup_{(x_i, y_i)} \left(\sum |P_t - y_t| - \sum |h(x_t) - y_t| \right) \\ &= \sup_{(x_i, y_i)} \left(\sum |P_t - y_t| + \sup_h -\sum |h(x_t) - y_t| \right) \\ &= \sup_{(x_i, y_i)} \left(\sum |P_t - y_t| - \inf_h \sum |h(x_t) - y_t| \right) \end{aligned}$$

i.e. best expert

We usually have $\inf_{h \in H} \sum_{t=1}^T |h(x_t) - y_t|$ grow sublinearly with T
(if not, then best expert is about as good as always choosing $P_t = 1$)

So if we want our algo to be useful, also need it to be sublinear in T

and want the regret to grow sublinearly in T

(\Rightarrow the difference between our learner and best hypothesis (by hindsight) goes to 0 as $T \rightarrow \infty$)

Unfortunately, this isn't possible (Cover '65)

$$\text{Ex } H = \{h_0, h_1\} \quad h_0(x) = 0 \quad \forall x, \quad h_1(x) = 1 \quad \forall x$$

Not realizable, so adversary has no constraint. You guess (P_1, P_2, \dots, P_T)

and it sets $y_1 = \neg P_1, \dots, y_T = \neg P_T$, so you make T mistakes.

The best-in-class makes $\approx T/2$ mistakes, so regret $\geq T - T/2 = T/2$

which isn't sublinear.

So must limit adversary's power aka mixed strategy (think of rock-paper-scissors...
you need to randomize)

Allow learner to randomize, and look at expectation

Assume randomness is independent of adversary, i.e., you flip a coin, and adversary

knows your general strategy but not the outcome of the coin flip.

In the binary case, output label $\hat{y} = \begin{cases} 1 & \text{w.p. } P_t \\ 0 & \text{w.p. } 1-P_t \end{cases}$

Then expected loss is

$$\mathbb{E} |\hat{y} - y| = P[\hat{y} \neq y] = \left\{ \begin{array}{ll} 1-P_t & y=1 \\ P_t & y=0 \end{array} \right\} = |P_t - y_t|$$

Our results about learnability will be constructive, based on the Weighted majority algorithm

Setup:

think of each $h_i \in \mathcal{H}$ as an **expert**, $i \in [d]$

At each round t , learner chooses advice from these experts,

and we allow a **mixed-strategy**, i.e., choose a prob. distribution $w \in [0,1]^d$, $\sum w_i = 1$,

and choose expert i w.p. w_i This is known as **prediction w/ expert advice**

(ex. choose from several numerical weather models)

Then you see how each expert does,

via a **cost** $v \in [0,1]^d$ ($1 = \text{bad}$, $0 = \text{good}$) so via your strategy,

your **expected loss** is $\sum w_i v_i = \langle w, v \rangle$

Think of this as receiving x_t , choosing $h_i \in \mathcal{H} = \{h_j : j \in [d]\}$, and $v_{t,j} = |h_j(x_t) - y_t|$

so there is an x_t but it's implicit

↑
or more precisely, the loss
associated w/ this projection

Example algorithm: at round $t=1$, guess $w_i = \frac{1}{d}$, receive v_t ,

and for all subsequent rounds $t > 1$, guess $w = v_t / \sum_i v_{t,i}$

... but what if x_t (or v_t)

wasn't representative (or a fluke).

$w = e_i$ where $i \in \operatorname{argmin}_j v_{t,j}$

Better

Algorithm: "Weighted Majority" if this is unknown to you, apply "Doubling Trick"

Input: # experts d , # rounds T

Set $\eta = \sqrt{\frac{2 \log(d)}{T}}$, $\tilde{w}^{(1)} = \mathbf{1} = (1, 1, \dots, 1)^T \in \mathbb{R}^d$

for $t = 1, 2, \dots, T$

normalize $w^{(t)} = \tilde{w}^{(t)} / Z_t = \sum_{i=1}^d \tilde{w}_i^{(t)}$

choose expert i at random w, $P(i) = w_i$

receive costs $v^{(t)}$

incur cost $\langle w^{(t)}, v^{(t)} \rangle$ (i.e., **expected cost**)

update: $\forall i \in [d]$

$$\tilde{w}_i^{(t+1)} = \tilde{w}_i^{(t)} \cdot \exp(-\eta v_i^{(t)}) \quad] \text{ the interesting part}$$

Thm 21.11 (analysis of Weighted-Majority)

If $T > 2 \cdot \log(d)$ (so $\eta < 1$) then $\sum_{t=1}^T \langle w^{(t)}, v^{(t)} \rangle - \min_{i \in [d]} \sum_{t=1}^T v_i^{(t)} \leq \sqrt{2 \cdot \log(d) \cdot T}$

best expert chosen in hindsight

Not T:

$$\sum_{t=1}^d \left(\min_{i \in [d]} v_i^{(t)} \right)$$

$$\begin{aligned} \log\left(\frac{Z_{t+1}}{Z_t}\right) &= \log\left(\sum_i \frac{\tilde{w}_i^{(t)} e^{-\eta v_i^{(t)}}}{Z_t}\right) = \log\left(\sum_i \tilde{w}_i^{(t)} e^{-\eta v_i^{(t)}}\right) \quad \alpha_i = \eta v_i^{(t)} \\ &\leq \log\left(\sum_i \tilde{w}_i^{(t)} \left(1 - \alpha_i + \frac{\alpha_i^2}{2}\right)\right) \quad \text{Since } e^{-\alpha} \leq 1 - \alpha + \alpha^2/2 \quad \alpha \in [0, \infty) \quad (\alpha_i \text{ in book is typo}} \end{aligned}$$

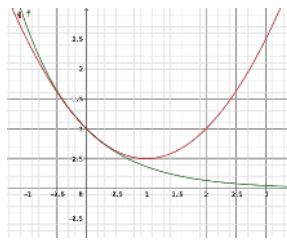
proof $f(x) = e^{-x}$ is 1 -Lipschitz on $(0, \infty)$ so we can use Jensen's inequality

$$e^{-b} = 1 - b + \frac{b^2}{2}, \quad b \in [0, 1]$$

$\geq 1 - b$

$$\text{so } -b = \log(e^{-b}) \geq \log(1-b)$$

$$\begin{aligned} &= \log\left(1 - \sum_i w_i^{(t)} (a_i - \frac{a_i^2}{2})\right) \\ &\leq -\sum_i w_i^{(t)} (\eta v_i^{(t)} - \frac{\eta^2 v_i^{(t)2}}{2}) \\ &= -\eta \langle \omega^{(t)}, v^{(t)} \rangle + \eta^2 \sum_i w_i^{(t)} v_i^{(t)2} \\ &\leq -\eta \langle \omega^{(t)}, v^{(t)} \rangle + \eta^2 \frac{1}{2} \end{aligned}$$



Now sum over $i=1\dots T$ + use telescoping series

$$\begin{aligned} \log(Z_{T+1}) - \log(Z_t) &= \sum_{t=1}^T \log(\frac{Z_{t+1}}{Z_t}) = \sum_{t=1}^T \log\left(\frac{Z_{t+1}}{Z_t}\right) \leq \sum_{t=1}^T \left(-\eta \langle \omega^{(t)}, v^{(t)} \rangle + \eta^2 \frac{1}{2}\right) \\ &\Downarrow \\ &\geq -\log(d) \end{aligned}$$

$$\begin{aligned} \log(Z_{T+1}) &= \log\left(\sum_i \tilde{w}_i^{(T+1)}\right) \quad \leftarrow \tilde{w}_i^{(t+1)} = \tilde{w}_i^{(t)} \cdot \exp(-\eta v_i^{(t)}) \\ &= \log\left(\sum_i e^{-\eta \sum_t v_i^{(t)}}\right) \quad \text{so } \tilde{w}_i^{(T+1)} = \exp(-\eta \sum_t v_i^{(t)}) \\ &\geq \log\left(\max_i e^{-\eta \sum_t v_i^{(t)}}\right) \quad \text{since } \sum_i (\dots) \geq \max_i (\dots) \\ &= -\eta \min_i \sum_t v_i^{(t)} \end{aligned}$$

$$\dots \text{so, re-arrange, } \eta \sum_t \langle \omega^{(t)}, v^{(t)} \rangle - \eta \min_i \sum_t v_i^{(t)} \leq \log(d) + \eta^2 T/2 \quad \frac{\sqrt{d}}{\sqrt{2}} = \frac{\sqrt{T}}{2}$$

$$\text{now } \div \eta \text{ and set } \eta = \sqrt{2 \log(d)/T} \quad \text{i.e. } \frac{\log(d)}{\eta} + \frac{\eta T}{2} = \frac{\sqrt{2 \log(d)/T}}{\sqrt{2 \log(d)/T}} + \frac{\sqrt{2 \log(d)/T} \cdot T}{2} = \sqrt{2 \log(d)/T} \quad \square$$

Wed 4/15/20

Now, use this prediction-with-expert-advice to prove our main learning result for agnostic online classification

(Allowing mixed-strategy, so $E(\# \text{mistakes in } T \text{ rounds}) = \sum_t |p_t - y_t|$)

Thm 21.10 Unrealizable binary online classification, regret bound

For every H , \exists algo. A w, mixed-strategy predictions $p_t \in [0, 1]$ s.t.

FT Sublinear in T

$$\textcircled{(1)} \quad (\forall h \in H) \quad \sum_{t=1}^T |p_t - y_t| - \sum_{t=1}^T |h(x_t) - y_t| \leq \sqrt{2 \min(\log(|H|), \text{Ldim}(H) \cdot \log(T))} \cdot T$$

i.e. best in hindsight

A B

$$\textcircled{(2)} \quad \text{and no algorithm can achieve an expected risk bound smaller than } \Omega(\sqrt{\text{Ldim}(H) \cdot T})$$

\rightarrow we won't prove this part (see Ben-David et al.'09)

Proof $\textcircled{(1)}$ let $|H| < \infty$

This follows from our "weighted-majority" result for expert prediction

$$H = \{h_1, \dots, h_d\}$$

$$V_i^{(t)} = |h_i(x_t) - y_t|, \quad p_t = \sum_i w_i^{(t)} h_i(x_t) \in [0, 1]$$

$$\therefore |p_t - y_t| = |\sum_i w_i^{(t)} h_i(x_t) - y_t - \sum_i w_i^{(t)}| = |\sum_i w_i^{(t)} (h_i(x_t) - y_t)|$$

all i terms have same sign

$$\text{since } y_t = 1 \Rightarrow (\forall i) h_i(x_t) \leq y_t$$

$$y_t = 0 \Rightarrow (\forall i) h_i(x_t) \geq y_t$$

$$= \sum_i w_i^{(t)} |h_i(x_t) - y_t|$$

$$= \sum_i w_i^{(t)} \cdot V_i^{(t)} = \langle \omega^{(t)}, v^{(t)} \rangle$$

$\textcircled{(2)}$ $|H| = \infty$

Strategy: Standard Optimal Algo (SOA) to pick a finite set of experts
... then apply Weighted-Majority.

1st idea: we're not realizable, but let $h \in H$ be the best we can do

$(x_1, y_1), \dots, (x_T, y_T)$ not realizable...

...but $\underbrace{(x_1, h(x_1)), \dots, (x_T, h(x_T))}_{\text{Run SOA on this dataset, and we make } L = \text{Ldim}(H) \text{ mistakes.}}$ is realizable.

Run SOA on this dataset, and we make $L = \text{Ldim}(H)$ mistakes.

Let the mistakes be made at rounds i_1, \dots, i_L

Idea: our "expert" to later use in Weighted Majority will really be an algorithm

Algo: "Expert(i_1, \dots, i_L)" $i_1 < i_2 < \dots < i_L$, class H

Init.: $V_t = H$

For $t=1, \dots, T$

receive x_t

$$V_t^{(r)} = \{h \in V_t : h(x_t) = r\}, r \in \{0, 1\} \quad (\text{as in SOA})$$

$$\tilde{y}_t = \underset{r}{\operatorname{argmax}} \text{Ldim}(V_t^{(r)}) \quad (\text{as in SOA})$$

predict $\hat{y}_t = \tilde{y}_t$ unless $t \in \{i_1, \dots, i_L\}$ SOA made a mistake...
 in which case $\hat{y}_t = 1 - \tilde{y}_t$... so don't make the same mistake twice.

update $V_{t+1} = V_t^{(\hat{y}_t)}$ Since we know (w/ hindsight) that \hat{y}_t is true label

So Expert is like SOA (same V_t) but fixing any mistakes

i.e. We had 2 sources of error: ① best h still isn't perfect (not realizable)

→ ② SOA can't even do as well as h
 "Expert" fixes that

Of course, we don't know where SOA makes the mistakes

Some (i_1, \dots, i_L) , $L \leq \text{Ldim}(H)$

But there are a finite # of such indices, $d := \sum_{L=0}^{\text{Ldim}(H)} \binom{T}{L} \leq \left(\frac{eT}{\text{Ldim}(H)}\right)^{\text{Ldim}(H)}$
 So in principle we could try them all
if $T > \text{Ldim}(H) + 2$
cf. Lemma A.2
(not in practice... $d \approx 3^T$ or more)

and the best of these d experts is at least as good as the best $h \in H$ (w/ hindsight),

$$\text{i.e., } \min_{i \in [d]} \sum_{t=1}^T V_i^{(t)} = \min_{h \in H} \sum_{t=1}^T |h(x_t) - y_t|$$

Now, we can't algorithmically find the best expert in an online fashion, but we

can come close using "Weighted Majority": if $T > 2/\log(d)$ ($\log(d) = \text{Ldim}(H) \log\left(\frac{eT}{\text{Ldim}(H)}\right)$)

$$\text{then } \sum_{i \in [d]} \langle w^{(i)}, V^{(t)} \rangle - \min_{i \in [d]} \sum_{t=1}^T V_i^{(t)} \leq \sqrt{2 \log(d) \cdot T} \quad \text{via Thm 21.11}$$

Expected loss, so also $\sum_i p_i |P_i - y_t|$

□