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 ① separate phases: train and test on same data
(many things naturally operate this way, eg., SPAM filters, medical knowledge, ...)

Also, theory can be completely ② distribution free: instead, allow it to change, or even be adversarial
↳ connections to game theory

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

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, ie., 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 $\lim_{T\to\infty}$ sometimes
Each round, observe data/features x_t
you (or the alg.) predicts p_t ... then true answer y_t is revealed (for now, $y_t \in \{0,1\}$)
Goal: make as few mistakes as possible (ie. 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 "reveal": y_t = whether you liked the food x_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 $t = 1, 2, \dots, T$
  receive x_t → so this is an ERM for $S = (x_1, y_1), \dots, (x_{t-1}, y_{t-1})$
  Choose any $h \in V_t$, predict $p_t = h(x_t)$
  receive true label y_t
  update (prune) $V_{t+1} = \{h \in V_t : \underbrace{h(x_t) = y_t}_{\text{consistent}}\}$

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

proof Every time we make a mistake, $|V_{t+1}| \leq |V_t| - 1$
So for M mistakes,
$$1 \leq |V_t| \leq |\mathcal{H}| - M$$
(1 ≤ by realizability) □

Is that good? No. Easy fix:

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

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

Thm 21.3 $M_{\text{Halving}}(\mathcal{H}) \le \log_2(|\mathcal{H}|)$

proof Every time we make a mistake, we can reduce size of V_t a lot, since at least half got it wrong

$|V_{t+1}| \le \frac{1}{2}|V_t|$ so $1 \le |V_{T+1}| \le |\mathcal{H}| \, 2^{-M}$ (by realizability) □

General case ($\mathcal{H}$ infinite is OK)

We want a measure of complexity of $\mathcal{H}$, analogous to VC dimension

(Recall: $VCdim(\mathcal{H}) \ge m \Rightarrow \mathcal{H}$ 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 $\mathcal{H} = \{h_1, \dots, h_k\}$.

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 sibling, they cheat as much as possible without you being able to prove it

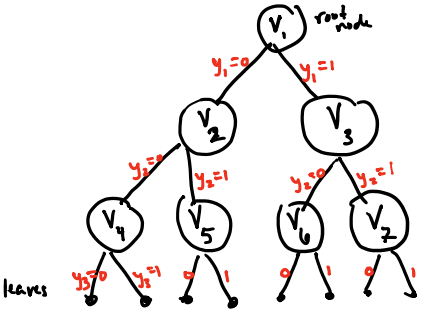
(Notes written during coronavirus quarantine w/ two daughters who just got Easter candy)

Our notion will be $Ldim(\mathcal{H})$ (after Nick Littlestone)

think of learner vs. environment: environment chooses x_t, learner chooses p_t, then environment chooses y_t but must be realizable by some $h \in \mathcal{H}$

In fact, we only care about rounds where we make mistakes, so let $y_t = \neg p_t$ (i.e. $\neg p_t = 1 - p_t$)

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



$V_i \in X$ ($V_2 = V_3$ is OK, though $V_2 = V_1$ is silly (non-optimal))

depth = T (ex: 3 in this case)

$2^T - 1$ nodes

(don't put V nodes on leaves)

Environment chooses X_t (ie. 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.

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 $(V_{i_1}=V_1, V_{i_2}, \dots, V_{i_T})$ w/ labels $(y_1, \dots, y_T)$ can be realized by some $h \in H$ s.t. $h(X_{i_t}) = y_t \quad \forall t \in [T]$. (or V_{i_t})

$V_{i_2} = X_{i_2}$ ...

↳ ie. 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)

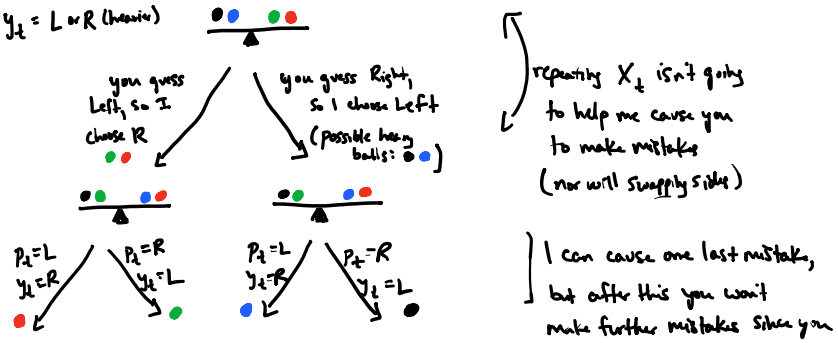
Immediately proves

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

Ex 4 balls ●●●● 3 identical, 1 is heavier ie. $|H| = 4$

X_t = weighing partition

y_t = L or R (heavier)



repeating X_t isn't going to help me cause you to make mistakes (nor will swapping sides)

I can cause one last mistake, but after this you won't make further mistakes since you know which ball is heaviest

So $Ldim(H) \geq 2$

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

(either just see it, or prove via HALVING alg)

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

Then $Ldim(H) = 1$

Why? take $V_1 = 1$, wlog (due to symmetry)

If alg says $P_1 = 1$, then no mistake and its learned h

If alg says $P_1 = 0$, they make a mistake, but they see $y_1 = 1$ and learn h

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

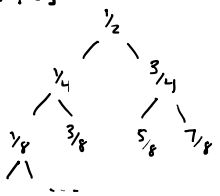
Since d can be arbitrarily large, this shows

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

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

Recall $VCdim(H) = 1$

but $Ldim(H) = \infty$. Choose nodes



$x_1 = 1/2$, you guess 1 ($x^* \le 1/2$)
So adversary decides $x^* > 1/2$

$x_2 = 3/4$, you guess 0 ($x^* > 3/4$)
So adversary decides $x^* \le 3/4$

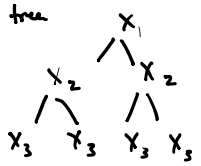
---

you'll make unbounded # of mistakes.

Ldim vs VCdim: $\forall\, H$, $\underline{\text{VCdim}(H)} \le \text{Ldim}(H)$, but $<$ possible (i.e. above example)

proof: let $d = \text{VCdim}(H)$, and $(x_1, \dots, x_d)$ a shattered set

make the tree



Now, we know $m_A(H) \ge \text{Ldim}(H)$. What about $\exists A$ s.t. $m_A(H) = \text{Ldim}(H)$?

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 $p_t = \underset{r \in \{0,1\}}{\text{argmax}}\ \text{Ldim}(\{h \in V_t : h(x_t) = r\})$ (only difference: $V_t^{(r)}$)

receive true label y_t

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

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

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

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

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

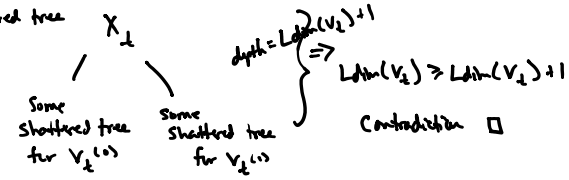
But also $\text{Ldim}(V_t) \ge \text{Ldim}(V_t^{(0)}) \ge \text{Ldim}(V_t^{(1)}) \ge \text{Ldim}(V_t)$

$\text{Ldim}(V_t) \ge \text{Ldim}(V_t^{(0)})$: $\text{Ldim}(A) \ge \text{Ldim}(B)$ if $A \supseteq B$

$\text{Ldim}(V_t^{(0)}) \ge \text{Ldim}(V_t^{(1)})$: by how we chose r

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

can make a new shattered tree



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

(16) Online classification, unrealizable case

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

Not true if unrealizable, so need new measure of learning

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

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

← this is for 1' loss, but can be defined more generally for an arbitrary loss

Def $\text{Regret}_A(\mathcal{H}, T) = \sup_{h \in \mathcal{H}} \text{Regret}_A(h, T) = \sup_h \sup_{(x_i,y_i)} \sum |p_t - y_t| - \sum |h(x_t) - y_t|$

$$= \sup_{(x_i,y_i)} \left(\sum |p_t - y_t| + \sup_h -\sum |h(x_t) - y_t| \right)$$

$$= \sup_{(x_i,y_i)} \left(\sum |p_t - y_t| - \inf_h \sum |h(x_t) - y_t| \right)$$

(i.e. best expert)

We usually have $\inf_{h \in \mathcal{H}} \sum_{t=1}^{T} |h(x_t) - y_t|$ 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

($\Leftrightarrow$ the difference between our learner and best hypothesis (in hindsight) goes to 0 as $T \to \infty$)

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

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

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 $\le T/2$ mistakes, so regret $\ge T - T/2 = T/2$

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

$$\mathbb{E}\,|\hat{y} - y| = \mathbb{P}[\hat{y} \neq y] = \left\{ \begin{matrix} 1-p_t & y=1 \\ p_t & y=0 \end{matrix} \right\} = |p_t - y_t|$$

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 = bad, 0 = 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 $v_{t,j} = h_j(x_t)$

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 → $w = v_1 / \sum_i v_{1,i}$

... but what if x_1 (or v_1) wasn't representative (or a fluke).

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

Better

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

Input: #experts d, #rounds T

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

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

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

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

receive costs $v^{(t)}$

incur cost $\langle w^{(t)}, v^{(t)} \rangle$ (ie, expected cost)

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

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

(Thm 21.11 (analysis of Weighted-Majority))

If $T > 2\log(d)$ (so $\eta < 1$) then

$$\sum_{t=1}^T \underbrace{\langle w^{(t)}, v^{(t)} \rangle}_{\text{expected loss}} - \min_{i \in [d]} \underbrace{\sum_{t=1}^d v_i^{(t)}}_{\text{best expert chosen in hindsight}} \leq \underbrace{\sqrt{2 \cdot \log(d) \cdot T}}_{\text{sublinear in } T}$$

NoT:

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

proof

$$\log\left(\frac{Z_{t+1}}{Z_t}\right) = \log\left(\sum \frac{\tilde{w}_i^{(t)} e^{-\eta v_i^{(t)}}}{Z_t}\right) = \log\left(\sum_i w_i^{(t)} \underbrace{e^{-\eta v_i^{(t)}}}_{e^{-a}}\right) \quad a_i = \eta v_i^{(t)}$$

$$\leq \log\left(\sum_i w_i^{(t)} \left(1 - a_i + \frac{a_i^2}{2}\right)\right)$$

Since $e^{-a} \leq 1 - a + a^2/2 \quad a \in [0, \infty)$ ← $(0,1)$ in book is typo

proof: $f(a) = e^{-a}$, f' is 1-Lipschitz on $(0, \infty)$ so use descent lemma

$\sum w_i = 1$

$e^{-b} = 1 - b + (b')^2/2 \quad b' \in [0,b]$ (Taylor)

$\geq 1 - b$

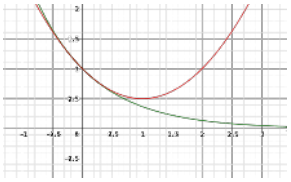
So $-b = \log(e^{-b}) \geq \log(1-b)$

$= \log\left(1 - \underbrace{\sum_i w_i (\eta_i - a_i/2)}_{b}\right)$

$\leq -\overbrace{\sum_i w_i^{(t)} (\eta v_i^{(t)} - \eta^2 v_i^{(t)2}/2)}^{b}$

$= -\eta \langle w^{(t)}, v^{(t)} \rangle + \eta^2/2 \underbrace{\sum_i w_i^{(t)} v_i^{(t)2}}_{\leq 1}$

$\leq -\eta \langle w^{(t)}, v^{(t)} \rangle + \eta^2/2 \qquad \leq \sum w_i = 1$



Now sum over $i = 1 \ldots T$ & use telescoping series

$\log(Z_{T+1}) - \log(Z_1) = \sum_{t=1}^{T} \log(Z_{t+1}) - \log(Z_t) = \sum_{t=1}^{T} \log\left(\frac{Z_{t+1}}{Z_t}\right) \leq \sum_{t=1}^{T} \left(-\eta \langle w^{(t)}, v^{(t)} \rangle + \eta^2/2\right)$

$= -\eta \sum_t \langle w^{(t)}, v^{(t)} \rangle + T \cdot \eta^2/2$

$\log(Z_1) = -\log(d)$

$\log(Z_{T+1}) = \log\left(\sum_i \tilde{w}_i^{(T+1)}\right) \leftarrow \tilde{w}_i^{(t+1)} = \tilde{w}_i^{(t)} \cdot \exp(-\eta v_i^{(t)})$

$= \log\left(\sum_i e^{-\eta \sum_t v_i^{(t)}}\right) \qquad$ so $\tilde{w}_i^{(T+1)} = \exp(-\eta \sum_t v_i^{(t)})$

$\geq \log\left(\max_i e^{-\eta \sum v_i^{(t)}}\right) \qquad$ since $\sum_i (\ldots) \geq \max_i (\ldots)$

$= -\eta \min_i \sum_t v_i^{(t)}$

... so, re-arrange, $\eta \sum_t \langle w^{(t)}, v^{(t)} \rangle - \eta \min_i \sum_t v_i^{(t)} \leq \log(d) + \eta^2 T/2 \qquad 1/\sqrt{2} = \frac{\sqrt{2}}{2}$

now $\div \eta$ and set $\eta = \sqrt{2\log(d)/T}$ i.e. $\frac{\log(d)}{\eta} + \frac{\eta T}{2} = \frac{\sqrt{\log^2(d)}}{\sqrt{2\log(d)/T}} + \frac{\sqrt{2\log(d)/T}\sqrt{T^2}}{2} = \sqrt{2\log(d)/T}$ □

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

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

Thm 21.10 Unrealizable binary online classification, regret bound

for every $\mathcal{H}$, $\exists$ algo. A w/ mixed-strategy predictions $p_t \in [0,1]$ s.t.

(1) $(\forall h \in \mathcal{H})$ $\sum_{t=1}^{T} |p_t - y_t| - \sum_{t=1}^{T} |h(x_t) - y_t| \leq \sqrt{2 \cdot \min(\underbrace{\log(|\mathcal{H}|)}_{A}, \underbrace{Ldim(\mathcal{H}) \cdot \log(eT)}_{B})} \cdot T$

$\sqrt{\text{T}}$ sublinear in T

ie. best in hindsight

Littlestone Dimension

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

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

Proof (A) let $|\mathcal{H}| < \infty$

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

$\mathcal{H} = \{h_1, \ldots, h_d\}$

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

So $|p_t - y_t| = \left|\sum w_i^{(t)} h_i(x_t) - y_t \underbrace{\sum w_i^{(t)}}_{=1}\right| = \left|\sum w_i^{(t)} (h_i(x_t) - y_t)\right|$

all i terms have same sign since $y_t = 1 \Rightarrow (\forall i)\ h_i(x_t) \leq y_t$, $y_t = 0 \Rightarrow (\forall i)\ h_i(x_t) \geq y_t$

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

$= \sum w_i^{(t)} \cdot v_i^{(t)} = \langle w^{(t)}, v^{(t)} \rangle$

(B) $|\mathcal{H}| = \infty$

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

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

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

...but $(x_1, h(x_1)), \dots, (x_T, h(x_T))$ is realizable.

Run SOA on this dataset, and we make $L \le Ldim(\mathcal{H})$ 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 $\mathcal{H}$

Init.: $V_1 = \mathcal{H}$

For $t = 1, \dots, T$

receive x_t

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

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

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

in which case $\hat{y}_t = 1 - \tilde{y}_t$

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