CSCI699: Theory of Machine Learning Fall 2021

Lecture 13: SQ Learning

Instructor: Vatsal Sharan *Scribe: Zhengqi Wu*

Definition 1 (SQ dimension)**.** *The SQ-dimension of a class $\mathcal{C}$ wrt a distribution D over $\mathcal{X}$ is the size of the largest subset $\mathcal{C}' \subseteq \mathcal{C}$ s.t. for all f,g $\in \mathcal{C}'$,*

$$|\Pr_{x\sim D}[f(x) = g(x)] - \frac{1}{2}| < \frac{1}{|C'|}$$

Theorem 2 (Theorem 2)**.** *If $SQ - DIM_D(\mathcal{C}) > poly(d)$ then you cannot efficiently learn $\mathcal{C}$ over D by SQ-algorithms (even "weak-learning" to error $\leq \frac{1}{2} - \frac{1}{poly(d)}$ is impossible).*

Parity function:
$X^d = \{0,1\}^d$
$y = \{0,1\}$
$e = w(x) =< w, x > \ mod\ 2 :\ w \in \{0,1\}^d$

Theorem 3. *PARITIES are efficiently PAC learnable.*

Let U be the uniform distribution over $\{0,1\}^d$.

Claim 4. *Any two parity functions $C_{w1}(x)$ and $C_{w2}(x)$ (where $w_1 \neq w_2$) are uncorrelated:*

$$\Pr_U[C_{w_1}(x) = C_{w_2}(x)] = \frac{1}{2}$$

SQ Learning

It will be convenient to define PARITIES as a function on $\{-1,+1\}^d \mapsto \{-1,+1\}$

$$x^d = \{\pm 1\}^d$$
$$y = \{\pm 1\}$$
$$e = \{C_S(x) = \Pi_{i\in S} x_i : S \subseteq \{1, \dots d\}\}$$

Claim 5. *If $S \neq T$, $Pr_{x\sim U}(C_S(x) = C_T(x)) = \frac{1}{2}$*

Proof

$$\begin{aligned} E_{x\sim U}[C_S(x) \cdot C_T(x)] &= E_{x\sim U}[\prod_{i\in S} x_i \cdot \prod_{i\in T} x_i] \\ &= E_{x\sim U}[\prod_{i\in S\Delta T} x_i] \\ &= 0 \text{ if } S \neq T \text{ (uniform distribution)} \end{aligned}$$

$(S\Delta T = \{S - T\} \cup \{T - S\})$

$$\Pr_{x\sim U}(C_S(x) = C_T(x)) + \Pr_{x\sim U}(C_S(x) \neq C_T(x)) = 1$$
$$\mathop{E}_{x\sim U}[C_S(x) \cdot C_T(x)] = \Pr_{x\sim U}(C_S(x) = C_T(x)) - \Pr_{x\sim U}(C_S(x) \neq C_T(x))$$
$$\mathop{E}_{x\sim U}[C_S(x) \cdot C_T(x)] = 0$$
$$\Pr_{x\sim U}(C_S(x) = C_T(x)) = \frac{1}{2}$$

Corollary 6 (Corollary of Theorem 2)**.** *It is not possible to efficiently learn parities in the SQ model over the uniform distribution.*

Proof
$SQ - DIM_U(\mathcal{C}) = 2^d$
We will now prove theorem 2 for the special case of PARITIES.

Theorem 7 (Hardness of parities in SQ)**.** *Any SQ algorithm for learning SQ over D=u, which makes queries of tolerance $\tau > \tau_{min}$ must make $\Omega(\tau_{min}^2 2^d)$ queries to STAT(c,0).*

Proof We first define a Correlational SQ (CSQ) oracle, which is a modified version of the SQ oracle:
For any query function $\Psi : \mathcal{X} \mapsto \pm$, and tolerance τ, let $P_\Psi = E[\Psi(x) \cdot c(x)]$
Oracle returns $\hat{P_\Psi} \in [P_\Psi - \tau, P_\Psi + \tau]$

Lemma 8. *If learner knows target distribution D, can simulate SQ oracle with CSQ oracle.*

Proof We can decompose any SQ ϕ into:

$$\begin{aligned}
\mathop{E}_{x\sim D}[\phi(x, c(x))] &= \mathop{E}_{x\sim D}[\phi(x,1) \cdot \mathbf{1}(c(x) = 1)] + \mathop{E}_{x\sim D}[\phi(x,1) \cdot \mathbf{1}(c(x) = -1)] \\
&= \mathop{E}_{x\sim D}[\phi(x,1) \cdot (\frac{1 + c(x))}{2})] + \mathop{E}_{x\sim D}[\phi(x,1) \cdot (\frac{1 - c(x))}{2})] \\
&= \frac{1}{2}\left(\mathop{E}_{x\sim D}[\phi(x,1)] + \mathop{E}_{x\sim D}[\phi(x,-1)]\right) \\
&+ \frac{1}{2}\left(\mathop{E}_{x\sim D}[\phi(x,1) \cdot c(x)] + \mathop{E}_{x\sim D}[\phi(x,-1) \cdot c(x)]\right)
\end{aligned}$$

Since we consider $D = U$ (a fixed distribution), suffices to show hardness for CSQ oracle.

Basics of Boolean Function Analysis

Think of any function f: $\{\pm 1\}^d \mapsto \{\pm 1\}$ as a vector $\vec{f}$ of 2^d entries.

$$\left(\frac{1}{2^{d/2}} f(-1,-1,-1), \frac{1}{2^{d/2}} f(-1,-1,1), \ldots \frac{1}{2^{d/2}} f(1,1,1)\right)$$

Note that $< \vec{f}, \vec{g} >= \mathop{E}_{x\sim U}[f(x) \cdot g(x)]$

$$\begin{aligned}
< \vec{f}, \vec{g} > &= \sum_{i=1}^{2^d} \frac{1}{2^d} f(x_i) \cdot \frac{1}{2^d} g(x_i) \\
&= \mathop{E}_{x\sim U}[f(x) \cdot g(x)]
\end{aligned}$$

And $< \vec{f}, \vec{f} >= 1$ Fourier analysis: change "basis" to understand t. Recall that an orthogonal basis for a vector space is a set of orthogonal unit vectors that span the space. If $v_1,\ v_2$ are orthoganal basis for $\mathbb{R}^2$, we can write any vector

$$w =< w, v_1 > v_1 + < w, v_2 > v_2$$

Claim 9. *PARITIES form an orthogonal basis for our vector space.*

Proof Note that for $S \neq T$.

$$\begin{aligned} < \vec{C_S}(x), \vec{C_T}(x) > &= E[C_S(x) \cdot C_T(x)] \\ &= 0 \\ < \vec{C_S}(x), \vec{C_S}(x) > &= 1, \forall s \end{aligned}$$

For any CSQ $\Psi : \{\pm 1\}^d \mapsto \{\pm 1\}$,

$$\vec{\Psi} = \sum_{S:S \subseteq 1 \ldots d}$$

Where $\vec{C_S}$ is the parity function over S.
Note that $\alpha_S = \underset{x \sim U}{E}[\Psi(x) \cdot C_S(x)]$
Exptected response to CSQ Ψ if the target function is $C_S(x)$.
Since $< \vec{\Psi}, \vec{\Psi} >= 1$
$\Sigma \alpha_s^2 = 1$
There can be almost $\dfrac{1}{\tau^2}$ S s.t. $|\alpha_S| \geq \tau$.
Note that if target is S^*, then CSQ oracle can just answer 0 to this query Ψ if $|\alpha_S^*| < \tau$.
We have the target parity function (the set S^r uniformly at random from all possible subsets.

Claim 10. *If algorithm makes less than $\tau^2 2^d$ queries, whp. over choice of S^*, CSQ can answer 0 to all these queries.*

Proof There are 2^d options

Any query Ψ is non-zero on almost $\dfrac{1}{\tau^2}$ options. If S^* is not among these, just answer 0. Since S^r is random, whp cannot "find" if in $O(\tau^2 2^d)$ queries. [Exercise]
This finishes proof.

Theorem 11. *Over the unif dist. D, in the presence of RCN with noise level η, PARITIES are learnable $O(\dfrac{d}{(1-2n)^2})$ samples, whp. (information theoretically)*

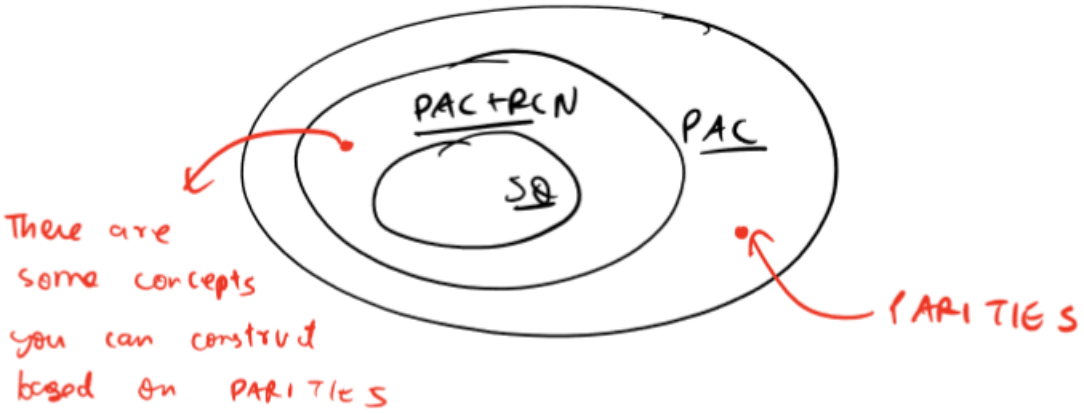
Proof Let $\epsilon = \dfrac{1}{2} - \eta$
We keep label w.p. $\dfrac{1}{2} + \epsilon$
We flip label w.p. $\dfrac{1}{2} - \epsilon$
Let $m = O(\dfrac{d}{\epsilon^2})$ samples from $Ex^\eta(c, D)$.
(Where $C(x) + C_S(x) = \underset{i \in S^*}{\Pi} x^i$)

Claim 12. *With high probability,*

- S^* *is consistent with* $\geq (\frac{1}{2} + \frac{\epsilon}{2})$ *fraction of examples*
- *Any* $S \neq S^*$ *is consistent with*$\leq \frac{1}{2}$ *fraction.*

Proof Exercise. Use Chernoff/Hoeffding inequality and property that parity functions are uncorrelated for $S \neq S^*$

Best know algorithm for learning parity with noise: $2^{\frac{d}{\log d}}$ time (Blum- Wasserman - Kalai 03)
Slightly less than exponential time, non-SQ algorithm.
LPN (Learning Parity with Noise) is believed to be hard.



However, the $2^{\frac{d}{\log d}} alg \Rightarrow$ time algorithm implies that we can learn parities over $O(\log d \log\log d)$ co-coordinates with RCN in poly-time. Therefore

$$\text{SQ} \subset \text{PAC+RCN} \subseteq \text{PAC}$$

Conditioned on hardness of LPN, the final inclusion is proper.

Note that with the exception of Gaussian elimination, almost all known algorithm can be run in SQ model. Therfore, SQ is sort of the frontier of our algorithmic knowledge. To show that some learning problem is hard, many recent papers show hardness in SQ model. Thanks to SQ-dim, we have an information theoretic way to show hardness in SQ.

Boosting

Recall our definition of weak-learning.

Definition 13. *Weak-learning: An algorithm A is a weak learner with edge/advantage* γ *for class* $\mathcal{C}$ *if: for any dist. D and any target* $c \in \mathcal{C}$*, given access to Ex(c,D), w.p.* $(1-\delta)$ *A produces a hypothesis with error(h;c,D)*$\leq \frac{1}{2} - \gamma$.

If A runs in time $\text{poly}(d, \frac{1}{\delta})$ and $\frac{1}{\gamma} > \frac{1}{\text{poly}(d)}$, then $\mathcal{C}$ is efficiently weakly-PAC learnable.

Theorem 14. *If $\mathcal{C}$ is (efficiently) weakly-PAC learnable , then $\mathcal{C}$ is (efficiently) PAC-learnable .*

Proof AdaBoost algorithm of Freund and Schapire (next class).