SQ and CN

- CN: With prob $1-\eta$ see $c(x)$, with prob η see $1-c(x)$.

  Algorithm A **PAC-learns C with CN** if for any $c \in C$, any $\mathcal{D}$, any $\eta < 1/2$, any $\varepsilon, \delta > 0$, given access to $EX^{\eta}(c, D)$, with prob $1-\delta$, A finds a hypothesis ε-close to c. Allowed time $poly\left(\frac{1}{\varepsilon}, \frac{1}{\delta}, \frac{1}{1-2\eta}, n, size(c)\right)$.

- SQ: Algorithm allowed to ask for statistics over data. Gets answers up to $1/poly$ additive error.

Today: If can learn with an SQ algorithm, then can learn with classification noise.

Converse not known. However, all C known to be learnable with noise are also known to be learnable with SQ.

The Statistical Query Model

- Algorithm asks "what is the probability a labeled example will have property χ? Please tell me up to additive error τ."

- Formally, $\chi : X \times \{0,1\} \to \{0,1\}$. Must be poly-time computable. $\tau \geq 1/poly(\cdots)$.

  If $P_\chi = \mathbf{E}_{c,D}[\chi(x, c(x))] = \Pr_{c,D}(\chi = 1)$ then world responds with $\widehat{P}_\chi \in [P_\chi - \tau, P_\chi + \tau]$.

  (can extend to $\chi : X \times \{0,1\} \to [0,1]$)

- May repeat this $poly(\cdots)$ many times. Algorithm may also ask to see unlabeled examples.

- Algorithm must output a hypothesis with error less than ε. (No δ in this model.)

(no noise in the SQ model)

Example: conjunctions

- Ask for $\Pr[c(x) = 1 \wedge x_i = 0]$ with $\tau = \varepsilon/2n$.

- Produce conjunction of all x_i such that $\widehat{P}_\chi \leq \varepsilon/2n$.

SQ $\Rightarrow$ PAC+CN

Given query χ need to estimate from noisy data. Idea:

- Break χ into part predictably affected by noise and part unaffected by noise.

- Estimate these parts separately.

- More efficient reduction: estimate many queries on same sample. Can even talk of $VCdim$ of possible queries of alg.

Running example: $\chi = 1$ if $c(x) = 1 \wedge x_i = 0$.

How to estimate $\Pr[\chi = 1]$

- CLEAN $= \{x : \chi(x,0) = \chi(x,1)\}$.

- NOISY $= \{x : \chi(x,0) \neq \chi(x,1)\}$.

$$\begin{aligned}\Pr[\chi = 1] \;&=\; \Pr[\chi = 1 \;\wedge\; x \in \text{CLEAN}] \;+ \\ &\;\; \Pr[\chi = 1 \;\wedge\; x \in \text{NOISY}].\end{aligned}$$

- Step 1: First part is easy to estimate from noisy data.

  (easy to test if a given x is in CLEAN)

Proof, contd.

To estimate: $\Pr[\chi = 1 \ \wedge \ x \in \text{NOISY}]$

- First estimate $\Pr[x \in \text{NOISY}]$. If observed ratio $\leq \frac{\tau}{2}$ return 0. Else confident that $\geq \frac{\tau}{4}$.

- Now need to estimate $\Pr[\chi = 1 | x \in \text{NOISY}]$.

  Can write this in terms of $\Pr_\eta[\chi = 1 | x \in \text{NOISY}]$.

- Just need to estimate $\Pr_\eta[\chi = 1 | x \in \text{NOISY}]$ up to additive error $\tau(1 - 2\eta)$.

- If don't know η can guess and verify.

Using SQs: learning Decision Lists

Open Problem: Can noise tolerance be boosted?

- Say for class C, exists alg A that PAC-learns C with noise, for any $\eta < 0.1$.

  Does this imply there must exist an algorithm B that succeeds for all $\eta < 1/2$, with running time $poly(\frac{1}{1-2\eta})$?

- Or, could ask same question for fixed $\mathcal{D}$. E.g., counterexample would be C and D where can achieve the former but not the latter.

- Hard part: given data source with 20% noise, how to run alg A?

How powerful are SQ algs?

- Most algs in practice are (roughly) SQ algorithms.

  E.g., ID3, gradient descent.

- Most are already tolerant to CN.

Can we quantify/characterize the kinds of things doable with SQ algorithms?

SQ-dimension

- Functions f and g are uncorrelated w.r.t $\mathcal{D}$ if

$$\Pr_{x \in \mathcal{D}}[f(x) = g(x)] = 1/2$$

- For a given $C, \mathcal{D}$, let S be the largest set of "pairwise nearly uncorrelated" functions in C.

  For all $f, g \in S$,

$$|\Pr_{x \in \mathcal{D}}[f(x) = g(x)] - \frac{1}{2}| \leq 1/|S|.$$

If $|S| \leq poly(\ldots)$ then you can weak-learn over $\mathcal{D}$.

If $|S| > poly(\ldots)$ then you can't weak-learn over $\mathcal{D}$.

(using an SQ algorithm)

Example: parity

- Let $\mathcal{D}$ be the uniform dist over $\{0,1\}^n$.

- Theorem: Any two parity functions are uncorrelated.

- So, PARITY $= \{$all 2^n parity fns over $\{0,1\}^n\}$ is not SQ-learnable over the uniform dist.

Parity, contd.

- Let $\mathrm{PARITY}_{\log n}$ be the set of parities of $\log n$ variables. Also not SQ-learnable.

- So, DNF and decision trees not SQ-learnable.

  (i.e., no SQ algorithm can guarantee to learn in time poly in n and the size of the target function)

- Can anyone think of a non-SQ alg to learn parity?

  Learnability with noise is big open crypto and coding-theory problem.

Fourier analysis of finite functions