# Computational Learning Theory
# 7 : Statistical Query Learning

Lecturer: Varun Kanade

All the learning frameworks we've studied so far have assumed that the learning algorithms have access to "perfect data". This is clearly not an accurate reflection of what one would encounter when accessing data in the real-world. Over the next few lectures, we will study several models that seek to capture noise or imperfections in the observed data. Some of the learning algorithms we've studied are not robust to noise; we'll study modifications to these algorithms that make them robust to noise. For some concept classes, there is evidence that while these are learnable in the absence of noise, there may not exist any efficient algorithms for learning these classes in the presence of noise.

## 1 Random Classification Noise Model

The model we'll study today is what can be considered as a "white noise" model. The data we receive may have labels flipped indepdently with probability $\eta$. Let us define the example oracle with random classification noise formally.

**Definition 1** (RCN Example oracle)**.** *An example oracle with random classification noise rate $\eta$, for a concept $c$ over instance space $X$ and for distribution $D$ over $X$, when called returns the following: $x \in X$ is drawn according to distribution $D$, with probability $1-\eta$, it returns $(x, c(x))$ and with probability $\eta$ returns $(x, 1-c(x))$.*

Before we study algorithms that can be implemented using a noisy example oracle, let us make a couple of brief remarks. The noise is assumed to be independent in each example; the case when noise could be correlated or a function of the instance can be much harder to deal with. Clearly, no learning algorithm can succeed when the noise rate $\eta = \frac{1}{2}$, as the labels have nothing to do with the target concept in that case. We'll assume that $\eta < \frac{1}{2}$; if $\eta > \frac{1}{2}$, we are just trying to learn the flipped concept. The quantity of interest is $1 - 2\eta$–the smaller this quantity the harder the learning problem. Thus, we will allow the sample complexity and running time of our algorithms to depend polynomially on $\frac{1}{1-2\eta}$. The requirement of the output hypothesis $h$ is exactly the same as in the PAC-learning framework—we want $\mathrm{err}(h; c, D) = \mathbb{P}_{x\sim D}\left[h(x) \neq c(x)\right] \leq \epsilon$, *i.e.*, we are comparing ourselves to the *true* target function (not with a noisy target). Let us formally define the model of PAC-learning in the presence of random classification noise.

**Definition 2** (PAC Learning with Random Classification Noise)**.** *Let $C$ be a concept class and $H$ a hypothesis class. We say that $C$ is PAC-learnable with random classification noise using $H$, if there exists a learning algorithm $L$ that for every $n \geq 1$, target concept $c \in C_n$, distribution $D$ over $X$, accuracy parameter $0 < \epsilon < \frac{1}{2}$, confidence parameter $0 < \delta < \frac{1}{2}$ and noise rate $0 < \eta < \frac{1}{2}$, with access to the noisy example oracle $\mathsf{EX}^\eta(c, D)$ and inputs $\epsilon$, $\delta$, $\mathrm{size}(c)$, and $\eta_0$ such that $\frac{1}{2} > \eta_0 \geq \eta$, outputs a hypothesis $h \in H_n$ such that with probability at least $1-\delta$, $\mathrm{err}(h; c, D) \leq \epsilon$.*

*We say that $C$ is efficiently learnable if $H$ is polynomially evaluatable and the running time of $L$ is polynomial in $n$, $\mathrm{size}(c)$, $\frac{1}{\epsilon}$, $\frac{1}{\delta}$ and $\frac{1}{1-2\eta_0}$.*

In the definition above, instead of the (clean) example oracle $\mathsf{EX}(c, D)$ introduced in the PAC learning framework, the learning algorithm has access to the noisy oracle, $\mathsf{EX}^\eta(c, D)$. The learning algorithm is also given as input a parameter $\eta_0 < \frac{1}{2}$, which is an upper bound on the noise rate. It is straightforward to show that this input, $\eta_0$, is not really required, the learning algorithm can simply try all possible upper bounds (up to some granularity) and then test which of the produced hypotheses is the (almost) best one (see Exercise 5.4 in (Kearns and Vazirani, 1994, Chap. 5)).

## 1.1 Learning Conjunctions

Let us revisit one of the first algorithms we studied, that of learning conjunctions, when there was no noise in the data. The algorithm started with a hypothesis consisting of a conjunction of all $2n$ literals; thus it begins with a hypothesis that always predicts 0. Then for every positive example $(a, 1)$, it deletes the literals present in its hypothesis that cause this example to be classified as negative. As long as sufficiently many examples are used, it is guaranteed that this algorithm outputs a hypothesis that has error at most $\epsilon$.

This algorithm does not work when there is noise in the data. For instance, if an example that was a negative example was observed with label 1 (due to noise), the algorithm may drop several literals from the hypothesis that are actually required. The decisions made by the algorithm are not *robust* as they are based on a single example. We will design a more robust algorithm for learning conjunctions. To begin with, let us continue to assume that the data we recive is noise-free; late, we'll discuss how this more robust algorithm can also be used when the data is noisy.

Let $c$ be the target conjunction and let $\ell$ be a literal that appears in $c$. We will use the notation $\ell(x) = 1$ to indicate that the literal $\ell$ evaluates to 1 (true) on the instance $x \in X$. For any literal $\ell$ that is present in the target conjunction, it holds that $\mathbb{P}_{x\sim D}\left[\ell(x) = 0 \wedge c(x) = 1\right] = 0$. We would like to identify all such literals and put them in the output hypothesis. Of course, it is only important to do this for literals that have a significant probability mass of being false under the distribution. Let us make this idea more concrete.

- A literal $\ell$ is said to be significant if $\mathbb{P}_{x\sim D}\left[\ell(x) = 0\right] \geq \frac{\epsilon}{8n}$
- A literal $\ell$ is harmful if $\mathbb{P}_{x\sim D}\left[\ell(x) = 0 \wedge c(x) = 1\right] \geq \frac{\epsilon}{8n}$

Clearly, all harmful literals are also significant. Let $h$ be a hypothesis that is a conjunction of all literals that are significant, but not harmful. Let $L$ denote the set of all $2n$ literals, $S$ the set of significant literals, and $T$ the set of harmful literals; then we have $T \subseteq S \subseteq L$. Let us analyse the error of $h$.

$$\mathrm{err}(h) = \mathbb{P}_{x\sim D}\left[h(x) = 0 \wedge c(x) = 1\right] + \mathbb{P}_{x\sim D}\left[h(x) = 1 \wedge c(x) = 0\right]$$

If $h(x) = 0$, but $c(x) = 1$, it must be due to a significant literal that is not harmful which was added to $h$. If $h(x) = 1$, but $c(x) = 0$, it must be due to an insignificant literal that was not added to $h$.

$$\begin{aligned}\mathrm{err}(h) &\leq \sum_{\ell\in(S\setminus T)} \mathbb{P}_{x\sim D}\left[\ell(x) = 0 \wedge c(x) = 1\right] + \sum_{\ell\in(L\setminus S)} \mathbb{P}_{x\sim D}\left[\ell(x) = 0\right] \\ &\leq |S \setminus T| \cdot \frac{\epsilon}{8n} + |L \setminus S| \cdot \frac{\epsilon}{8n} \leq \frac{\epsilon}{2}\end{aligned}$$

We haven't said how exactly to find significant and harmful literals; however, it is easy to see that they could be identified correctly with high probability. The size of the sample required to guarantee correctness can be obtained using the Chernoff-Hoefdding bound and is polynomial in $n$, $\frac{1}{\epsilon}$ and $\frac{1}{\delta}$. We've left enough room in our calculations, so that even when using empirical estimates of these probabilities to identify significant and harmful literals, the output hypothesis will have error bounded by $\epsilon$. Compared to the earlier algorithm we've seen, this new algorithm is more robust. A single example is not used to determine whether or not a literal should be present in the target hypothesis. This decision is made using aggregate statistics.

It is not too hard to show that if we know the noise rate $\eta$, these probability estimates can be obtained even when receiving samples from $\mathsf{EX}^{\eta}(c, D)$, rather than $\mathsf{EX}(c, D)$. However, rather than do that we'll see a model that formalises this idea of using 'statistics' to design learning algorithms and show that any algorithm in this framework can always be simulated with access to a noisy example oracle, $\mathsf{EX}^{\eta}(c, D)$.

## 2 Statistical Query Model

In the statistical query model, the learning algorithm is not given any access to examples at all, but instead is given access to a *statistical query* oracle, $\mathsf{STAT}(c, D)$. As was the case in PAC-learning, let $X$ be the instance space, $c : X \to \{0, 1\}$ the target concept, and $D$ the target distribution over $X$. A statistical query is a tuple, $(\chi, \tau)$, where $\chi : X \times \{0, 1\} \to \{0, 1\}$ is a boolean function that takes as input an instance $x \in X$ and a bit $b \in \{0, 1\}$ (one of the two possible labels of the instance), and $\tau$ is the tolerance parameter. The response of the oracle $\mathsf{STAT}(c, D)$ to the query $(\chi, \tau)$, is a value $v \in [0, 1]$, such that,

$$\left| \mathop{\mathbb{E}}_{x \sim D} \left[ \chi(x, c(x)) \right] - v \right| \leq \tau.$$

### Learning Conjunctions using Statistical Queries

Before, we formally define the notion of learning using statistical queries, let us see how the algorithm we described above can be implemented using statistical queries. The first task was to identify significant literals. This can be done easily, for a literal, $\ell$, define the query function, $\chi_\ell : X \times \{0, 1\} \to \{0, 1\}$, as follows:

$$\chi_\ell(x, b) = \begin{cases} 1 & \text{if } \ell(x) = 0 \\ 0 & \text{if } \ell(x) = 1 \end{cases}$$

The above query does not depend on the bit $b \in \{0, 1\}$ at all. Thus, $\chi_\ell(x, b) = \mathbb{1}(\ell(x) = 0)$, and hence $\mathop{\mathbb{E}}_{x \sim D} \left[ \chi_\ell(x, c(x)) \right] = \mathbb{P}_{x \sim D} \left[ \ell(x) = 0 \right]$. We set the tolerance parameter $\tau = \frac{\epsilon}{32n}$. Let $v_\ell$ be the response received from $\mathsf{STAT}(c, D)$ to $(\chi_\ell, \frac{\epsilon}{32n})$, then we treat all literals for which $v_\ell \geq \frac{\epsilon}{16n}$ as significant. This guarantees that any literal that we call as *insignificant* is such that $\mathbb{P}_{x \sim D} \left[ \ell(x) = 1 \right] \leq \frac{3\epsilon}{32n}$.

In order to identify harmful literals, we use the query $\widetilde{\chi}_\ell$ defined as follows:

$$\widetilde{\chi}_\ell(x, b) = \begin{cases} 1 & \text{if } \ell(x) = 0 \text{ and } b = 1 \\ 0 & \text{otherwise} \end{cases}$$

Again it is easy to see that $\widetilde{\chi}_\ell(x, c(x)) = \mathbb{1}(\ell(x) = 0 \wedge c(x) = 1)$, and hence $\mathop{\mathbb{E}}_{x \sim D} \left[ \widetilde{\chi}_\ell(x, c(x)) \right] = \mathbb{P}_{x \sim D} \left[ \ell(x) = 0 \wedge c(x) = 1 \right]$. Thus, the $\mathsf{STAT}(c, D)$ oracle can be used to identify harmful literals. The exact details of choosing the tolerance values and completing the proof are left as an exercise.

### 2.1 Statistical Query Learnability

Let us formally define the notion of learning with statistical queries.

**Definition 3** (Statistical query (SQ) learnability)**.** *Let $C$ be a concept class and $H$ a hypothesis class. We say that $C$ is* efficiently learnable *from* statistical queries *using $H$, if there exists a learning algorithm, $L$, and polynomials $p(\cdot, \cdot, \cdot)$, $q(\cdot, \cdot, \cdot)$, and $r(\cdot, \cdot, \cdot)$ such that, for all $n \geq 1$, for* *every target $c \in C_n$, for every distribution $D$ over $X_n$, for any accuracy parameter $0 < \epsilon < \frac{1}{2}$, $L$ with access to the statistical query oracle, $\mathsf{STAT}(c, D)$, and inputs $\epsilon$ and* $\mathrm{size}(c)$, *satisfies the following:*

- *For any query $(\chi, \tau)$ made by $L$, the predicate $\chi$ can be evaluated in time $q(n, \mathrm{size}(c), \frac{1}{\epsilon})$ and $\frac{1}{\tau}$ is bounded by $r(n, \mathrm{size}(c), \frac{1}{\epsilon})$*
- *$L$ halts in time bounded by $p(n, \mathrm{size}(c), \frac{1}{\epsilon})$*
- *$L$ outputs $h \in H$, such that* $\mathrm{err}(h; c, D) \leq \epsilon$

The confidence parameter $\delta$ no longer appears in the definition; this is because we require the statistical query oracle, $\mathsf{STAT}(c, D)$, to return a value that is within the tolerance with probability 1. We could extend the definition of statistical query learnability to allow randomised algorithms, in which case the confidence parameter would be required to bound the failure of the algorithm itself. Let us first establish the relatively simple result that any concept class that is efficiently SQ-learnable is also efficiently PAC-learnable. We'll only provide a sketch of the proof, but the main idea is that with access to the example oracle, $\mathsf{EX}(c, D)$, the algorithm can simulate $\mathsf{STAT}(c, D)$ with high probability.

**Theorem 4.** *If $C$ is efficiently SQ-learnable using $H$, then $C$ is efficiently PAC-learnable using $H$.*

*Proof.* Let $A$ be the algorithm that learns $C$ using $H$ in the SQ model. Let $k$ be an upper bound on the total number of queries made to the $\mathsf{STAT}(c, D)$ oracle by $A$. We simulate the algorithm $A$; every time a query $(\chi, \tau)$ is made, we draw $m = \Theta(\frac{1}{\tau^2} \log \frac{k}{\delta})$ samples from $\mathsf{EX}(c, D)$, say $(x_1, c(x_1)), \ldots, (x_m, c(x_m))$, and return $\frac{1}{m} \sum_{i=1}^{m} \chi(x_i, c(x_i))$. Using the Chernoff-Hoeffding bound, we know that with probability at least $1 - \frac{\delta}{k}$, the following holds:

$$\left| \frac{1}{m} \sum_{i=1}^{m} \chi(x_i, c(x_i)) - \mathop{\mathbb{E}}_{x \sim D} \left[ \chi(x, c(x)) \right] \right| \leq \tau.$$

When $A$ halts, we simply output the hypothesis $h$ that was produced by $A$. By using the union bound and as $A$ makes at most $k$ queries, we know that with probability at least $1 - \delta$, all simulations of the statistical query oracle used to provide responses to $A$ are valid, and hence the output hypothesis $h$ satisfies, $\mathrm{err}(h; c, D) \leq \epsilon$. □

The more surprising result that we'll prove is that the statistical query oracle can be simulated even with access to noisy examples, *i.e.,* the oracle $\mathsf{EX}^{\eta}(c, D)$. This automatically implies that all concept classes learnable in the SQ framework are also PAC-learnable with random classification noise. This formalises the intuition that "robust" algorithms that make decisions on the basis of statistics rather than individual examples can be adapted to work with noisy data.

## 2.2 Simulating $\mathsf{STAT}(c, D)$ using $\mathsf{EX}^{\eta}(c, D)$

In order to make the mathematical manipulations simpler, we'll assume that the output of boolean functions are in the set $\{-1, 1\}$, rather than the more common $\{0, 1\}$. For reasons that will be clear later, we'll map 0 to 1, and 1 to $-1$. We will also require that the query, $\chi$, is defined as, $\chi : X \times \{-1, 1\} \to \{-1, 1\}$. Note that this still allows us to compute, $\mathbb{P}_{x \sim D} \left[ \chi(x, c(x)) = -1 \right]$, which is what we want as part of the statistical query learning framework, rather easily. To see this, observe that:

$$\mathbb{P}_{x \sim D} \left[ \chi(x, c(x)) = -1 \right] = \mathop{\mathbb{E}}_{x \sim D} \left[ \frac{1 - \chi(x, c(x))}{2} \right] = \frac{1}{2} - \frac{1}{2} \mathop{\mathbb{E}}_{x \sim D} \left[ \chi(x, c(x)) \right]$$

Furthermore, for any query, $\chi : X \times \{-1, 1\} \to \{-1, 1\}$, we can express this as follows:

$$\mathop{\mathbb{E}}_{x \sim D}\left[\chi(x, c(x))\right] = \mathop{\mathbb{E}}_{x \sim D}\left[\chi(x, 1) \cdot \mathbb{1}(c(x) = 1)\right] + \mathop{\mathbb{E}}_{x \sim D}\left[\chi(x, -1) \cdot \mathbb{1}(c(x) = -1)\right]$$

As $c(x) \in \{-1, 1\}$, $\mathbb{1}(c(x) = 1) = \frac{1+c(x)}{2}$; similarly, $\mathbb{1}(c(x) = -1) = \frac{1-c(x)}{2}$. Thus,

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

We observe that $\chi(\cdot, 1)$ and $\chi(\cdot, -1)$ are simply functions from $X \to \{-1, 1\}$. The first two expectations on the RHS above don't depend on the target concept at all; the last two compute the correlation between the some function from $X \to \{-1, 1\}$ and the target. Formally, we allow the learning algorithm to make two kinds of queries—*target independent* and *correlational.* A target-independent query is of the form $(\psi, \tau)$, where $\psi : X \to \{-1, 1\}$ and $\tau \in (0, 1)$, and it receives an answer $v_\psi$ from $\mathsf{STAT}(c, D)$, such that $|\mathop{\mathbb{E}}_{x \sim D}\left[\psi(x)\right] - v_\psi| \le \tau$. A correlational query is also of the form $(\varphi, \tau)$, where $\varphi : X \to \{-1, 1\}$ and $\tau \in (0, 1)$; in this case, however, the response $v_\varphi$ of $\mathsf{STAT}(c, D)$, satisfies, $\left|\mathop{\mathbb{E}}_{x \sim D}\left[\varphi(x)c(x)\right] - v_\varphi\right| \le \tau$. Thus, it suffices to show that we can simulate responses of $\mathsf{STAT}(c, D)$ to target-independent and correlational queries using the noisy oracle $\mathsf{EX}^\eta(c, D)$. For target-independent queries, the label noise is irrelevant, as the response to the query doesn't depend on the target at all. We only need samples drawn from the distribution $D$; these are obtained by simply ignoring the labels of the observed data. We will now show how to simulate the responses of $\mathsf{STAT}(c, D)$ using examples from $\mathsf{EX}^\eta(c, D)$ for correlational queries.

### 2.2.1 Simulating responses to correlational queries

Let $Z \sim B(\eta)$ denote that $Z$ is a random variable taking values in $\{-1, 1\}$, such that $Z = 1$ with probability $1 - \eta$ and $Z = -1$ with probability $\eta$. Observe that $\mathbb{E}[Z] = 1 - 2\eta$. Let $(X, c(X))$ denote a random example drawn from $\mathsf{EX}(c, D)$, then $(X, c(X)Z)$ is distributed as a random example drawn from $\mathsf{EX}^\eta(c, D)$. Consider the following:

$$\begin{aligned}
\mathop{\mathbb{E}}_{(x,y) \sim \mathsf{EX}^\eta(c,D)}\left[\varphi(x) \cdot y\right] &= \mathop{\mathbb{E}}_{x \sim D}\left[\mathop{\mathbb{E}}_{Z \sim B(\eta)}\left[\varphi(x)c(x)Z\right]\right] \\
&= \mathop{\mathbb{E}}_{x \sim D}\left[\varphi(x)c(x)\mathop{\mathbb{E}}_{Z \sim B(\eta)}\left[Z\right]\right] && \text{As } Z \text{ is independent} \\
&= (1 - 2\eta) \cdot \mathop{\mathbb{E}}_{x \sim D}\left[\varphi(x)c(x)\right]
\end{aligned}$$

Suppose we draw $m$ examples from $\mathsf{EX}^\eta(c, D)$, say $(x_1, y_1), \ldots, (x_m, y_m)$ and let $\widehat{v} = \frac{1}{m}\sum_{i=1}^m \varphi(x_i)y_i$. Let $m$ be chosen to be large enough such that $|\widehat{v} - \mathop{\mathbb{E}}_{(x,y) \sim \mathsf{EX}^\eta(c,D)}\left[\varphi(x)y\right]| \le \tau_1$ with probability at least $1 - \delta$. Suppose that we don't have access to the exact value of $\eta$, but only to some $\widehat{\eta} \le \eta_0$ (where $\eta_0$ is an upper-bound on the noise rate) such that $|\widehat{\eta} - \eta| \le \Delta/2$. We have the following:

$$\begin{aligned}\left|\frac{\widehat{v}}{1-2\widehat{\eta}} - \mathop{\mathbb{E}}_{x\sim D}\left[\varphi(x)c(x)\right]\right| &\leq \left|\frac{\widehat{v}}{1-2\widehat{\eta}} - \frac{\widehat{v}}{1-2\eta} + \frac{\widehat{v}}{1-2\eta} - \frac{\mathop{\mathbb{E}}_{(x,y)\sim \mathsf{EX}^{\eta}(c,D)}\left[\varphi(x)y\right]}{1-2\eta}\right| \\ &\leq |\widehat{v}|\cdot\left|\frac{1}{1-2\widehat{\eta}} - \frac{1}{1-2\eta}\right| + \frac{1}{1-2\eta}\cdot\left|\widehat{v} - \mathop{\mathbb{E}}_{(x,y)\sim \mathsf{EX}^{\eta}(c,D)}\left[\varphi(x)y\right]\right| \\ &\leq \frac{2\Delta}{(1-2\eta_0)^2} + \frac{\tau_1}{1-2\eta_0}\end{aligned}$$

If $\Delta$ and $\tau_1$ are chosen so that the RHS above is at most $\tau$, then we have successfuly simulated the response of the statistical query oracle. Choosing such a $\tau_1$ is relatively straightforward, we simply use a sample size $m$ that is polynomially large in $n$, $\frac{1}{1-2\eta_0}$, $\frac{1}{\delta}$ and $\frac{1}{\tau}$. In order to find a suitable $\widehat{\eta}$, we simply run the algorithm with all possible values of $\widehat{\eta} = i\Delta$ for $i = 1, \ldots, \lfloor \frac{\eta_0}{\Delta} \rfloor$. Clearly, one of the values among these satisfies the properties that we require of $\widehat{\eta}$. When we run the procedure for that particular value, we will obtain a $h$ that with high probability satisfies, $\mathrm{err}(h; c, D) \leq \epsilon$. We obtain $h_1, h_2, \ldots, h_{\lfloor \frac{\eta_0}{\Delta} \rfloor}$, and we know that with high probability at least one of theses hypotheses is "good", in that it has error at most $\epsilon$. In order to identify the best (or good enough) hypothesis from this set, we can simply test each of them on a fresh sample drawn from $\mathsf{EX}^{\eta}(c, D)$. Simply using the hypothesis $h$ that has the smallest *empirical* error does the trick.

We've described all the main ingredients of the proof of the following theorem; although we have not provided the formal proof; writing the proof in full detail is left as an exercise.

**Theorem 5.** *If $C$ is efficiently SQ-learnable using $H$, then $C$ is efficiently PAC-learnable with random classification noise using $H$.*

## 3 A hard-to-learn concept class

We have seen an algorithm for learning conjunctions using only statistical queries. In some of the exercises, you are asked to design algorithms for learning rectangles, decision lists, 3-CNF formulae, *etc.* using only statistical queries. Given this one may wonder, if in fact, every concept class that is PAC-learnable is also SQ-learnable. We answer the question in the negative. We show that the class PARITIES is not efficiently learnable using statistical queries. Formally, we will prove the following theorem in this section.

**Theorem 6.** *Any algorithm for learning* PARITIES *using statistical queries with $\epsilon = \frac{1}{10}$, and which makes queries of the form $(\chi, \tau)$, where $\tau \geq \tau_0$ for each query, must make $\Omega(\tau_0^2 \cdot 2^n)$ queries.*

Before we sketch a proof of the result, let us look at the statement of the theorem in greater detail. Efficient SQ learnability requires that the tolerance parameter for any query not be too small. In particular for efficient learning parities to accuracy $\frac{1}{10}$, we require that $\frac{1}{\tau_0}$ be bounded by a polynomial in $n$ (in the case of parities $\mathrm{size}(c) = O(n)$). The statement of the theorem implies that if the inverse tolerance for all queries is bounded by some polynomial in $n$, the algorithm must make $2^{\Omega(n/\log n)}$ queries! In particular, this rules out a polynomial time algorithm for learning PARITIES in the statistical query model. Furthermore, observe that unlike the result where we showed that *proper* learning 3-TERM-DNF is hard unless RP = NP, there are no unproven conjectures required to prove the hardness of learning PARITIES in the statistical query framework. The reason for this is that the proof is purely information-theoretic. In particular, even an algorithm that uses unbounded computation and uses query functions $\chi$

that are not evaluatable in polynomial time (or indeed even uncomputable ones!), requires a superpolynomial number of queries to the $\mathsf{STAT}(c, D)$ oracle.

Let us now sketch a proof of the result. To make our notation simpler, we will assume that the instance space is $\{-1, 1\}^n$, rather than $\{0, 1\}^n$. We will also assume that the output of boolean functions is in the set $\{-1, 1\}$. Then for a subset $S \subseteq [n]$, the parity function on bits in $S$, is defined as:

$$f_S(x) = \prod_{i \in S} x_i$$

If we interpret a bit $x_i = -1$ as being on and $x_i = +1$ as being off, then $f_S(x) = -1$ if and only if an odd number of bits in $S$ are on, *i.e.,* $f_S$ computes the parity over bits in $S$. We will consider $\mathcal{U}$, the uniform distribution over $\{-1, 1\}^n$, as the target distribution. We will assume that the distribution is known to the algorithm, and as a result, without loss of generality, we may assume that the algorithm only makes correlational queries to the oracle $\mathsf{STAT}(c, D)$.

Let us observe some basic facts about $\mathsf{PARITIES}$ with respect to the uniform distribution $\mathcal{U}$. For any non-empty set $S$, $\underset{x\sim\mathcal{U}}{\mathbb{E}}\left[f_S(x)\right] = 0$. This is because the uniform distribution $\mathcal{U}$ over $\{-1, 1\}^n$ can be viewed as a product distribution over the individual bits, *i.e.,* all bits are independent. The distribution over each bit is also uniform, so that $\mathbb{E}\left[x_i\right] = 0$ for each $i$. Then, we also have:

$$\begin{aligned} \underset{x\sim\mathcal{U}}{\mathbb{E}}\left[f_S(x) f_T(x)\right] &= \underset{x\sim\mathcal{U}}{\mathbb{E}}\left[\prod_{i \in S} x_i \prod_{i \in T} x_i\right] \\ &= \underset{x\sim\mathcal{U}}{\mathbb{E}}\left[\prod_{i \in S \Delta T} x_i\right]. \end{aligned}$$

Above, $S \Delta T$ denotes the symmetric difference of the sets $S$ and $T$. This establishes that if $S \neq T$, then $\underset{x\sim\mathcal{U}}{\mathbb{E}}\left[f_S(x) f_T(x)\right] = 0$. Clearly, it is also the case that $\underset{x\sim\mathcal{U}}{\mathbb{E}}\left[(f_S(x))^2\right] = 1$, since $f_S(x) \in \{-1, 1\}$. Since there are $2^n$ such parity functions and there are exactly $2^n$ points in $\{-1, 1\}^n$, the set of parity functions form an orthonormal basis for the vector space of real-valued functions defined over $\{-1, 1\}^n$, with inner product $\langle \varphi, \psi \rangle = \underset{x\sim\mathcal{U}}{\mathbb{E}}\left[\varphi(x)\psi(x)\right]$. Formally, any $\varphi : \{-1, 1\}^n \to \mathbb{R}$ can be expressed as:

$$\varphi(x) = \sum_{S \subseteq [n]} \widehat{\varphi}(S) f_S(x)$$

Furthermore, Parseval's identity establishes that $\underset{x\sim\mathcal{U}}{\mathbb{E}}\left[(\varphi(x))^2\right] = \sum_{S \subseteq [n]} (\widehat{\varphi}(S))^2$. In particular, if $\varphi : \{-1, 1\}^n \to \{-1, 1\}$, then $\sum_{S \subseteq [n]} (\widehat{\varphi}(S))^2 = 1$; the coefficient $\widehat{\varphi}(S) = \underset{x\sim\mathcal{U}}{\mathbb{E}}\left[\varphi(x) f_S(x)\right]$.

Suppose that there is an algorithm, $A$, that learns $\mathsf{PARITIES}$ using at most $q$ queries, each of which has a tolerance parameter larger than $\tau_0$. We show that it must be the case that $q/\tau_0^2 > 2^n - 2$. For the sake of contradiction, suppose that this is not the case. We show that in fact there must be a target parity on which the algorithm fails. The definition of SQ learning requires that a *single* algorithm work for all target functions. We will show that this cannot be the case. We run the algorithm $A$ and every time a query is made, we return the answer 0. Let $h$ be the target hypothesis when $A$ terminates. We will show that when $q/\tau_0^2 \leq 2^n - 2$, there must be at least two parity functions, represented by subsets $S$ and $S'$, such that for all the queries made by the algorithm 0 was a valid answer. Let's first see that this finishes the proof. Clearly, both $f_S$ and $f_{S'}$ could be a valid target; thus it must be the case that $\mathrm{err}(h; f_S, \mathcal{U}) \leq \frac{1}{10}$ and $\mathrm{err}(h; f_{S'}, \mathcal{U}) \leq \frac{1}{10}$. However, $\mathbb{P}_{x\sim\mathcal{U}}\left[f_S(x) \neq f_{S'}(x)\right] = \frac{1}{2}$; thus $h$ cannot be a $\frac{1}{10}$ accurate hypothesis for both $f_S$ and $f_{S'}$.

Let $\varphi_1, \ldots, \varphi_q$ denote the queries made by the algorithm. There is a subtle point to be mentioned here; in principle the queries made by the algorithm could be adaptive, *i.e.,* the query depends on past answers from the oracle $\mathsf{STAT}(c, D)$. However, since we always supply 0 as the answer, we may as well assume that the queries $\varphi_1, \ldots, \varphi_q$ are known ahead of time. We claim that for any $\varphi : \{-1, 1\} \to \{-1, 1\}$, there are at most $\frac{1}{\tau_0^2}$ parity functions, such $\left| \mathop{\mathbb{E}}_{x \sim \mathcal{U}} \left[ \varphi(x) f_S(x) \right] \right| \geq \tau_0$. This follows from that fact that $\sum_{S \subseteq [n]} (\widehat{\varphi}(S))^2 = 1$ and that $\widehat{\varphi}(S) = \mathop{\mathbb{E}}_{x \sim \mathcal{U}} \left[ \varphi(x) f_S(x) \right]$. Thus, a single query rules out at most $\frac{1}{\tau^2}$ parities as the possible target, if we supply a response of 0. Consequently, $q$ queries rule out at most $q/\tau_0^2$ parities as the possible target. Provided, $q/\tau_0^2 \leq 2^n - 2$, there must be at least two different parity functions that are consistent with all the answers supplied to the algorithm, as a simulation of the oracle $\mathsf{STAT}(c, D)$. This completes the proof.

## 4 Bibliographic Notes

The variant of PAC learning with random classification noise was introduced in the work of Angluin and Laird (1988). Our presentation of the statistical query model differs a bit from that in the textbook by Kearns and Vazirani (1994, Chap. 5). Bshouty and Feldman (2002) first used the idea of separating a general statistical query into target-independent and correlational queries. The proof that PARITIES cannot be learnt in the statistical query model appeared in the work of Kearns (1998). The proof provided in the lecture follows along the lines of that introduced by Blum et al. (1994) who also show lower bounds for general concept classes in terms of what is called the *statistical query dimension*.

Of course, the fact that PARITIES is not SQ-learnable does not rule out an efficient algorithm for learning PARITIES is the presence of random classification. Blum et al. (2003) provide an algorithm whose complexity matches that suggested by the lower bound in the SQ model. Improving this dependence is a long standing open problem. It is widely believed that learning parities with noise is hard and cryptographic systems based on the hardness of this and related problems have been desgined. Blum et al. (2003) also demonstrate the existence of a concept class that is not polynomial-time learnable in the SQ model, but can be learnt in polynomial time in the PAC model with random classification noise.

## References

Dana Angluin and Philip Laird. Learning from noisy examples. *Machine Learning*, 2(4):343–370, 1988.

Avrim Blum, Merrick Furst, Jeffrey Jackson, Michael Kearns, Yishay Mansour, and Steven Rudich. Weakly learning dnf and characterizing statistical query learning using fourier analysis. In *Proceedings of the twenty-sixth annual ACM symposium on Theory of computing*, pages 253–262. ACM, 1994.

Avrim Blum, Adam Kalai, and Hal Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *Journal of the ACM (JACM)*, 50(4):506–519, 2003.

Nader H Bshouty and Vitaly Feldman. On using extended statistical queries to avoid membership queries. *Journal of Machine Learning Research*, 2(Feb):359–395, 2002.

Michael Kearns. Efficient noise-tolerant learning from statistical queries. *Journal of the ACM (JACM)*, 45(6):983–1006, 1998.

Michael J. Kearns and Umesh K. Vazirani. *An Introduction to Computational Learning Theory.* The MIT Press, 1994.