Lecturer: Javed Aslam — *Scribe: Jon McAuliffe*

20.1 Outline

- The statistical query model
  - Statistical query learning algorithms
  - Simulators for noise-free statistics
- Choosing a best hypothesis in the statistical query model

20.2 The Statistical Query Model

20.2.1 Motivation

In the previous lecture, we examined the result derived by Angluin and Laird in [1] that PAC learning is possible even in the presence of classification noise, as long as we are able to find a hypothesis $h \in \mathcal{H}$ which minimizes disagreement with a sample of bounded polynomial size. This finding was subject to the information-theoretic barrier of 1/2 on the noise rate η. We also looked at some unfortunate complexity-theoretic results on finding a minimally disagreeing hypothesis, which told us that the problem is NP-hard even for relatively simple concept classes such as monotone conjunctions.

This new difficulty, as well as the obstacles faced by our original PAC algorithms when presented with noisy samples, stems in part from the fact that the PAC approach to learning depends quite sensitively on the classification of every example presented. The algorithms we have examined thus far usually make irretrievable decisions about the desired hypothesis based on individual labels for points $x^{(i)} \in X$.

In light of this realization, it makes sense to consider a formalism for learning which relies on gross statistical properties of a *population* of labelled examples, rather than on the information carried by each individual example. Consider, for example, Figure

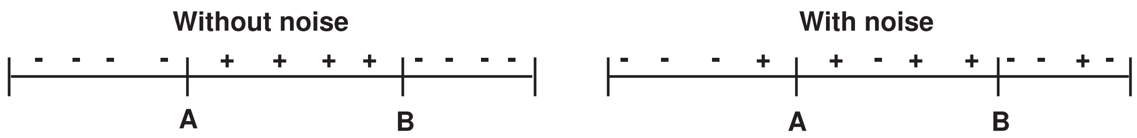

Figure 20.1: Two samples from the target concept $[a, b]$: the left without and the right with noise.

20.1, where we wish to learn subintervals of the unit interval. Although no hypothesis exists which correctly classifies the given noisy sample, the indicated subinterval does have a certain statistical appeal: if the noise rate is $\frac{1}{4}$, then outside of $[a, b]$, roughly 75% of the examples are negative, while within $[a, b]$ exactly the opposite holds.

So if noise affects the statistical characteristics of a sample in a uniform way, it should be possible to design a model of learning which first recovers statistical information about noiseless samples from the statistics of a noisy sample, then uses the recovered information to learn. We will spend the rest of the lecture developing a formalism which makes this statistical intuition precise and general.

20.2.2 The Structure of a Statistical Query Learner

As depicted in Figure 20.2, a learning algorithm in the statistical query model relies on a *statistics oracle* for the information it uses to select an output hypothesis. In particular, the algorithm itself makes no use of information about particular labelled examples, and it expects the answers to its queries to be based on the noise-free characteristics of the target concept. The statistics oracle, in turn, relies on sample data from an examples oracle which may have no noise, classification noise, malicious adversarial noise, or another type of noise altogether. In cases where the data do indeed have noise, the statistics oracle acts as a *simulator*, producing noise-free statistics based on the noisy samples it draws.

The decision to divide the model into an algorithm based on noiseless statistics and an oracle which derives those noiseless statistics from potentially noisy data proves advantageous for two reasons.

Generality Previous efforts to cope with noise involved heuristic techniques incorporated directly into the learning algorithm. Here, a single algorithm automatically succeeds in the presence of several different types of noise, by virtue of

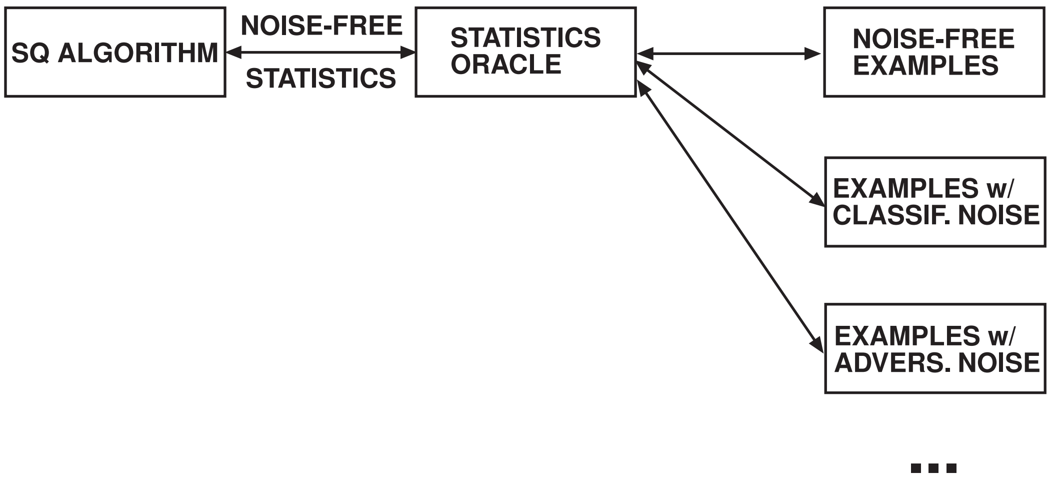

Figure 20.2: A schematic of the interface between algorithm and oracle in the statistical query model.

the noiseless statistics simulator on which it relies.

Efficacy We simply state that nearly all algorithms for PAC learning in the absence of noise can be recast in terms of queries to a noiseless statistics oracle, which immediately suggests the usefulness of the proposed formalism for robust learning.

We now proceed first to formalize the notion of learning by statistical query, setting aside issues of noise, then turn to the definition of simulators which produce noiseless statistics from noisy data.

20.3 Statistical Query Learning Algorithms

20.3.1 Definitions for statistical query learning

Definition 1 *A* **statistical query** *is an ordered pair* (χ, τ)*, where* $\chi : X \times \{0,1\} \to \{0,1\}$ *is an indicator function on labelled examples from the instance space* X*, and* τ *is an error tolerance on the answer returned by the statistics oracle defined below.*

Definition 2 *Let P_χ be the true probability with respect to noiseless examples from a target concept that $\chi = 1$. A* **statistics oracle** *$STAT(c, D)$ for a target concept $c \in \mathcal{C}$ and distribution D on X takes a statistical query as defined above and returns an approximation $\widehat{P}_\chi$ such that $P_\chi - \tau \leq \widehat{P}_\chi \leq P_\chi + \tau$.*

For example, χ may indicate the event $(x \in [a', b']) \wedge (\ell = 1)$. A statistical learning algorithm for the concept class depicted in Figure 20.1 will certainly be able to make use of P_χ in this case. But since the statistics oracle may be using a noisy data source and must draw a bounded number of examples, it can only return the approximation $\widehat{P}_\chi$ which is within $\pm\tau$ of P_χ.

We mention two relevant points about these definitions somewhat prematurely.

- This definition of $STAT(c, D)$ holds for arbitrary $\tau > 0$, though the number of examples will of course depend on how small τ is. We do not address exceptions, but they do exist (*e.g.* when learning from a malicious adversary).
- The learning algorithm expects a correct $\widehat{P}_\chi$ with probability 1. The PAC-learning confidence parameter δ resurfaces in the discussion of simulator design.

Definition 3 *A concept class $\mathcal{C}$ is* **efficiently learnable by statistical queries** *using hypothesis class $\mathcal{H}$ if there exists a learning algorithm L and polynomials $p(\cdot, \cdot)$, $q(\cdot, \cdot)$, and $r(\cdot, \cdot)$ with the following property: for any probability distribution D on X, $(\forall c \in \mathcal{C})(\forall \epsilon, 0 < \epsilon < 1)$ if L is given access to $STAT(c, D)$ and input ϵ, then:*

1. *for every query (χ, τ) submitted,*
   - *χ can be evaluated in time bounded above by $q(\frac{1}{\epsilon}, n)$*
   - *$1/\tau$ is bounded above by $r(\frac{1}{\epsilon}, n)$*
2. *L will halt in time bounded above by $p(\frac{1}{\epsilon}, n)$*
3. *L outputs a hypothesis $h \in \mathcal{H}$ such that error$(h) \leq \epsilon$.*

(Here n denotes the length of any example vector $x \in X$.)

If L is deterministic, then an ϵ-good h will be returned with certainty. We will not consider the case where L is randomized.

20.3.2 An example: SQ-learning monotone conjunctions

The concept class of monotone conjunctions is the set of all boolean formulae of the form $\bigwedge x_{i_k}$. Recall that the PAC-learning algorithm for monotone conjunctions begins with an initial conjunction of all variables $\bigwedge_{i=1}^{n} x_i$. It then draws a sufficiently large sample from $EX(c, D)$ and removes from the initial conjunction all x_i's which are false in any positive example.

The approach in the statistical query model is essentially the same. The statistical query algorithm also begins with a conjunction $h = \bigwedge_{i=1}^{n} x_i$. Then, for each i such that $1 \leq i \leq n$, the algorithm submits a query (χ_i, τ_i), where χ_i is $[(x_i = 0) \wedge (\ell = 1)]$ and $\tau_i = \frac{\epsilon}{2n}$. We know that for any variable present in the target concept c, P_{χ_i} will be 0. Therefore, if $\hat{P}_{\chi_i} > \frac{\epsilon}{2n}$, we have that $P_{\chi_i} > 0$ by choice of τ_i. Hence we remove x_i from h. Furthermore, if $\hat{P}_{\chi_i} \leq \frac{\epsilon}{2n}$, we know $P_{\chi_i} \leq \frac{\epsilon}{n}$, again by choice of τ_i.

Now, no negative examples can be misclassified by the output hypothesis h, because we never remove a variable x_i from the initial conjunction unless it is guaranteed not to be in the target concept. Further, if a positive example is misclassified as negative by h, it is only because an errant x_i is present in h when it should have been removed. But the total probability of such an event is at most $\sum_{x_i \in h} P_{\chi_i} \leq \sum_{x_i \in h} \frac{\epsilon}{n} \leq n\frac{\epsilon}{n} = \epsilon$.

Hence we conclude that our algorithm outputs with certainty a hypothesis h such that error$(h) \leq \epsilon$. The relevant polynomials $p(\cdot,\cdot)$, $q(\cdot,\cdot)$ and $r(\cdot,\cdot)$ are apparent. Thus monotone conjunctions are efficiently SQ-learnable.

We now claim that similar translations into the statistical query model exist for a wide range of efficiently PAC-learnable concept classes. One might speculate about the existence of a "folk theorem" to the effect that every PAC-learning algorithm has a statistical query counterpart. However, the PAC-learnable concept class of parity functions is known *not* to be efficiently SQ-learnable, so we can rule out the existence of such a theorem.

20.3.3 Summary of advantages to SQ-learning algorithms

We now present a brief synopsis of the reasons this modular approach to statistical learning proves especially useful to us.

1. "Nearly" every PAC-learning algorithm finds re-expression in the statistical query formalism.

2. A correct SQ-learning algorithm is also a correct PAC-learning algorithm in the absence of noise.

3. A correct SQ-learning algorithm is also a correct PAC-learning algorithm in the presence of classification noise, malicious adversarial noise, ...

20.4 Simulation of Noise-Free Statistics on Noisy Data

We now turn our attention from learning algorithms which rely on an oracle for information about the statistical properties of noise-free samples to the design of the statistical oracle itself. First we consider how to produce statistical data given examples from a noise-free oracle of the type encountered in the PAC model, then we describe how to simulate the production of noise-free statistics given an oracle with classification noise. Other classes of noise are not considered here.

20.4.1 Producing statistics from a noise-free source of examples

Given that the statistical oracle need not concern itself with the possibility of incorrectly labelled examples, finding a $\widehat{P}_\chi$ within $\pm\tau$ of P_χ for a query (χ, τ) is a straightforward exercise in elementary statistical sampling. The only complication lies in the requirement that the value returned by the statistical oracle is a correct $\widehat{P}_\chi$ with probability at least $1 - \delta$, where δ is the PAC-learning accuracy parameter. In other words, we must handle *δ-budgeting* correctly.

If the number of queries is known beforehand to be fixed, such as in the case of SQ-learning monotone conjunctions, we simply distribute our δ-budget uniformly over the samples drawn for each query. When the number of queries is unknown, two approaches suggest themselves. First, we can avail ourselves of the knowledge that $\sum_{i=1}^{\infty} \frac{1}{i^2} = \frac{\pi^2}{6}$, so that if we let $\delta_i = \frac{6}{\pi^2} \cdot \frac{\delta}{i^2}$, then $\sum_i \delta_i \leq \delta$ as the series converges. Otherwise, we can choose to draw a single large sample after receiving all queries, basing the answer for each query on that sample alone. In this case, uniform convergence tells us that for a sample size only polynomially dependent on the VC-dimension of the query class, we can guarantee with probability at least $1 - \delta$ that all of the estimates $\widehat{P}_{\chi_i}$ will be within $\pm\tau$ of the true P_{χ_i}, as we desire.

20.4.2 Producing noise-free statistics from a sample oracle with classification noise

First, note that the treatment of this topic provided in [3] has some significant errors. We now present an alternate treatment, which is covered in more detail in [2].

Suppose we have a query (χ, τ) to the statistics oracle. We want to return the value $Pr_{EX(c,D)}[\chi = 1]$, that is, a statistic taken with respect to the noise-free oracle $EX(c, D)$, but we have access only to a source of labelled examples with classification noise $EX_{CN}^{\eta}(c, D)$. For the purposes of analysis only, consider these four different oracles:

$EX(c, D)$	The noise-free examples oracle
$EX(\overline{c}, D)$	The noise-free anti-examples oracle, that is, an oracle which outputs $\langle x, \overline{c(x)} \rangle$, where c is the target concept
$EX_{CN}^{\eta}(c, D)$	The examples oracle with classification noise rate $\eta < \frac{1}{2}$
$EX_{CN}^{\eta}(\overline{c}, D)$	The anti-examples oracle (as above) with classification noise rate η

Claim 1

$$EX_{CN}^{\eta}(c, D) = \begin{cases} EX(c, D) & \textit{with probability } 1 - \eta \\ EX(\overline{c}, D) & \textit{with probability } \eta \end{cases}$$

$$EX_{CN}^{\eta}(\overline{c}, D) = \begin{cases} EX(\overline{c}, D) & \textit{with probability } 1 - \eta \\ EX(c, D) & \textit{with probability } \eta \end{cases}$$

Proof: This follows immediately from the definition of the oracles and η. ■

We can now find an expression for $Pr_{EX_{CN}^{\eta}(c,D)}[\chi = 1]$ in terms of $Pr_{EX(c,D)}[\chi = 1]$, $Pr_{EX(\overline{c},D)}[\chi = 1]$, and η, and do so similarly for $Pr_{EX_{CN}^{\eta}(\overline{c},D)}[\chi = 1]$.

Claim 2

$$Pr_{EX_{CN}^{\eta}(c,D)}[\chi = 1] = (1 - \eta) Pr_{EX(c,D)}[\chi = 1] + \eta Pr_{EX(\overline{c},D)}[\chi = 1] \tag{20.1}$$

$$Pr_{EX_{CN}^{\eta}(\overline{c},D)}[\chi = 1] = (1 - \eta) Pr_{EX(\overline{c},D)}[\chi = 1] + \eta Pr_{EX(c,D)}[\chi = 1] \tag{20.2}$$

Proof: These expressions can be derived using Claim 1. ■

Remember that we wish to calculate $Pr_{EX(c,D)}[\chi = 1]$. We can easily compute $Pr_{EX^{\eta}_{CN}(c,D)}[\chi = 1]$ and $Pr_{EX^{\eta}_{CN}(\overline{c},D)}[\chi = 1]$ via normal statistical sampling, and η is an unknown we deal with later. So, multiplying equation 20.1 by $(1-\eta)$, equation 20.2 by η, then subtracting 20.2 from 20.1, we obtain

$$\begin{aligned}
&(1-\eta)Pr_{EX^{\eta}_{CN}(c,D)}[\chi = 1] - \eta Pr_{EX^{\eta}_{CN}(\overline{c},D)}[\chi = 1] \\
&\quad = ((1-\eta)^2 Pr_{EX(c,D)}[\chi = 1] + \eta(1-\eta)Pr_{EX(\overline{c},D)}[\chi = 1]) \\
&\qquad - (\eta(1-\eta)Pr_{EX(\overline{c},D)}[\chi = 1] + \eta^2 Pr_{EX(c,D)}[\chi = 1]) \\
&\quad = ((1-\eta)^2 - \eta^2)Pr_{EX(c,D)}[\chi = 1] \\
&\quad = Pr_{EX(c,D)}[\chi = 1](1 - 2\eta + \eta^2 - \eta^2) \\
&\quad = Pr_{EX(c,D)}[\chi = 1](1 - 2\eta)
\end{aligned}$$

And therefore we have that

$$Pr_{EX(c,D)}[\chi = 1] = \frac{(1-\eta)Pr_{EX^{\eta}_{CN}(c,D)}[\chi = 1] - \eta Pr_{EX^{\eta}_{CN}(\overline{c},D)}[\chi = 1]}{(1-2\eta)} \tag{20.3}$$

Note that we can remove reference to the classification noise anti-oracle in this equation by defining $\chi'(x,\ell) \stackrel{\triangle}{=} \chi(x,\overline{\ell})$, then replacing $Pr_{EX^{\eta}_{CN}(\overline{c},D)}[\chi = 1]$ with $Pr_{EX^{\eta}_{CN}(c,D)}[\chi' = 1]$. Let us forego for a moment concerns about the unknown value η. We now have a method for estimating the value P_χ by calculating statistical estimates on noisy examples, but we need to insure that the returned value $\widehat{P}_\chi$ is within $\pm\tau$ of the true value. In order to accomplish this, we will need to determine what the accuracy bounds should be on the estimates from the noisy data. Finding these bounds is an exercise in *sensitivity analysis.*

20.4.3 Sensitivity analysis

Claim 3 *Suppose we know that for values* $0 \le a, b, c, \tau \le 1$,

$$\begin{aligned}
a &= b - c && (20.4)\\
a &= bc && (20.5)\\
a &= b/c && (20.6)
\end{aligned}$$

and we want to estimate a *within* $\pm\tau$ *in each case. Then for*

20.4 *we must estimate b and c to within $\pm\tau/2$*

20.5 *we must estimate b and c to within $\pm\tau/3$*

20.6 *we must estimate b and c to within $\pm c\tau/3$.*

Proof: 20.4 is obvious. For 20.5, one boundary case is $(b+\frac{\tau}{3})(c+\frac{\tau}{3}) = bc + c\frac{\tau}{3} + b\frac{\tau}{3} + \frac{\tau^2}{9} \leq a + \frac{\tau}{3} + \frac{\tau}{3} + \frac{\tau^2}{9} \leq a + \tau$, which checks. 20.6 is left as an exercise. ∎

Applying these results to our expression for $Pr_{EX(c,D)}[\chi = 1]$, it follows that we must estimate the numerator and denominator of the fraction to within $\pm(1-2\eta)\tau/3$. Repeated application of Claim 3 to the numerator tells us that the tightest estimation interval required for the entire fraction is $\pm(1-2\eta)\tau/18$.

Though we do not know η, we do know an upper bound η_b. Hence we can use an estimation interval of $\pm(1-2\eta_b)\frac{\tau}{18} \leq (1-2\eta)\frac{\tau}{18}$ and, by employing Chernoff bounds, draw a noisy sample(s) large enough to guarantee that $\widehat{P}_\chi$ can be within $\pm\tau$ of P_χ.

The only problem remaining is that equation 20.3 depends on the unknown noise rate η directly, and simply substituting η_b will not guarantee a reasonable $\widehat{P}_\chi$. Further, there is no known method of sampling from $EX_{CN}^{\eta}(c, D)$ so as to compute an estimate of η.

20.5 Choosing the Best $h \in \mathcal{H}$ in the Statistical Query Model

In order to get around the problem that we have insufficient knowledge about η, we simply run the given statistical query algorithm (which relies on our simulated noise-free statistics oracle) many times, with many different guessed values of η, so that we are assured of having at least one run which produces an $h_i \in \mathcal{H}$ such that $\text{error}(h_i) \leq \epsilon$. This is depicted in figure 20.3.

We need to decide on a set of guesses at η small enough to have size bounded by a polynomial, so that it will be feasible to search for the h_i in the set of output hypotheses which minimizes disagreement with the sample drawn from $EX_{CN}^{\eta}(c, D)$. But the set of guesses must also be large enough to guarantee that some ϵ-good hypothesis is produced.

In the previous section, we found that η needs to be known within $\pm(1-2\eta)\tau/18$. However, the value of η is unknown, and the value of τ may be different for each query.

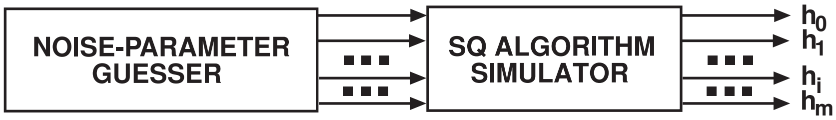

Figure 20.3: Multiple runs of the SQ-learning algorithm, with different guesses at η.

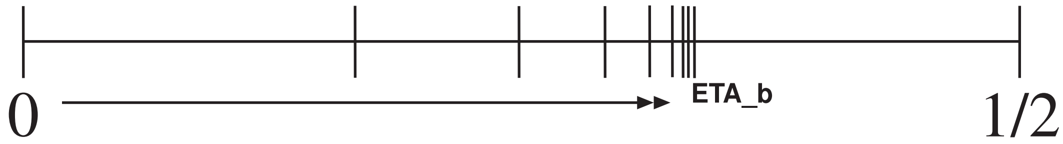

Figure 20.4: A intuitive characterization of guessing at η.

Let τ_{min} be a lower bound on the tolerance required for any query submitted by the SQ learning algorithm. (In practice, τ_{min} is generally simple to calculate from the specification of the SQ learning algorithm.) Since $(1-2\eta_b)\tau_{min}/18 \leq (1-2\eta)\tau/18$, it is sufficient to find a value of η within $\pm(1-2\eta_b)\tau_{min}/18$. One simple way to achieve this is to guess values of η between 0 and η_b uniformly spaced $(1-2\eta_b)\tau_{min}/9$ apart. Note that this will require $\Theta(\frac{1}{\tau_{min}(1-2\eta_b)})$ guesses which is polynomial in the relevant learning parameters as needed.

We can reduce the number of η-guesses by noting that finding a value of η within $\pm(1-2\eta)\tau_{min}/18$ is also sufficient. Thus, for "large" values of η, our guesses need to be "closely" spaced, but for "small" values of η, our guesses can be "further" apart. By employing a guessing strategy similar to that depicted in Figure 20.4, one can show that only $\Theta(\frac{1}{\tau_{min}}\log\frac{1}{1-2\eta_b})$ guesses are required.

References

[1] Angluin, Dana, and Philip Laird. "Learning from noisy examples." *Machine Learning*, 2(4): 343-370, 1988.

[2] Aslam, Javed and Scott Decatur. "Improved Noise-Tolerant Learning and Generalized Statistical Queries." *Harvard University Technical Report TR-17-94*, Center for Research in Computing Technology, Division of Applied Sciences.

[3] Kearns, Michael, and Umesh Vazirani. *An Introduction to Computational Learning Theory*. Cambridge, MA: MIT Press, 1994.