6.842 Randomness and Computation March 19, 2012

Lecture 12

Lecturer: Ronitt Rubinfeld *Scribe: Cheung Wang Chi*

Today, we are going to discuss the following:

- Models for learning
- Brute force algorithm
- Learning conjunction functions
- Start on Learning theory and Fourier Transform

1 Models for Learning

In the model, the learner aims to learn an unknown function $f : D \to R$, where $|D|$ is finite, but R is a general set, e.g. the set of real numbers. She has access to the *example oracle* $\mathrm{Ex}(f)$, which outputs an example $(x, f(x))$, where $x \in_U D$. Upon obtaining a certain number of random examples, she will call an learning algorithm $\mathcal{A}$ that inputs the observed examples and outputs the hypothesis $h : D \to R$.

In general, one cannot hope to learn a random function. Indeed, when there is no restriction on f, it is too hard to learn f exactly, i.e. to obtain $h \equiv f$, or even to obtain h that is ϵ-close to f. Therefore, we will restrict f to be in a certain class $\mathcal{C}$ of functions, which is also known as the *concept class.* Moreover, the learning algorithm $\mathcal{A}$ is usually specific to the concept class $\mathcal{C}$, as we will see in the case of learning conjunction functions.

Definition 1 *Given a hypothesis h for f, the "error of h with respect to f" is* $\mathrm{err}(h) = \Pr_{x \in_U D}[f(x) \neq h(x)]$

Equivalently, $\mathrm{err}(h)$ is the smallest $\epsilon > 0$ such that h is ϵ-close to f.

Definition 2 *A uniform distribution learning algorithm for the concept class $\mathcal{C}$ is an algorithm $\mathcal{A}$ such that*

1. *Given $\epsilon, \delta > 0$, $\mathcal{A}$ accesses $\mathrm{Ex}(f)$ to obtain random samples $(x_1, f(x_1)), ..., (x_m, f(x_m))$.*
2. *$\mathcal{A}$ outputs hypothesis h such that $\Pr[\mathrm{err}(h) \le \epsilon] \ge 1 - \delta$, where $\mathrm{err}(h)$ is the error of h with respect to f defined earlier.*

Here are the parameters of interest:

- The description of h:
  - Does h have to be in $\mathcal{C}$? We will not require $h \in \mathcal{C}$ in the future, but under the model for *proper learning* we restrict $h \in \mathcal{C}$.
  - We require that h is at least relatively compact, i.e. $\mathbf{size}(h) = O(\log(|\mathcal{C}|))$, and h can be efficiently evaluated.
- The accuracy parameter ϵ.
- The confidence parameter δ.
- The runtime of $\mathcal{A}$.
- The sample complexity m of $\mathcal{A}$, which is the number of samples used by $\mathcal{A}$.

It is desirable to have the runtime and the sample complexity bounded by poly($\log |D|, \frac{1}{\epsilon}, \frac{1}{\delta}$). In fact, we can replace $\frac{1}{\delta}$ with $\log \frac{1}{\delta}$ by repeating $\mathcal{A}$ and picking the best concept learnt.

Comment: Our uniform distribution learning model generalizes naturally to the PAC model. By replacing the assumption of "uniform distribution over D" by "general distribution $\mathcal{D}$ over D", it is the PAC model. To be more precise, in the PAC model, a learner has access to example oracle $\mathrm{Ex}_{\mathcal{D}}(f)$, which generates $(x, f(x))$ with $x \in_{\mathcal{D}} D$. $\mathcal{D}$ is unknown to the learner. The learner aims to output hypothesis h such that $\Pr_{\mathcal{D}}[h \text{ is } \epsilon\text{-close to } f] \geq 1 - \delta$.

2 Brute Force Algorithm for Learning

With reasonable sample complexity but no bound on runtime, for all $\mathcal{C}$ and $\epsilon, \delta > 0$ there is a brute force algorithm for learning $f \in \mathcal{C}$ that achieves the accuracy parameter ϵ and the confidence parameter δ:

Brute Force Algorithm:

1. Draw $M = \frac{1}{\epsilon}\left(\log |\mathcal{C}| + \log \frac{1}{\delta}\right)$ random examples from $\mathrm{Ex}(f)$.
2. Search over all $h \in \mathcal{C}$ until we find one that is consistent with all labelled examples. Choose an arbitrary one if there are > 1 such h.

Behaviour:

- f is always a candidate for h, since f will always be consistent. Thus, the algorithm will definite output a hypothesis.
- We would like to show that for a consistent h, $\Pr[\mathrm{err}(h) \leq \epsilon] \leq 1 - \delta$, i.e. h is not too far from f. Now, say h is *bad* if $\mathrm{err}(h) > \epsilon$, and we see that

  $$\Pr[\text{bad } h \text{ is consistent the } M \text{ examples}] \leq (1-\epsilon)^M$$

  By union bound, we have

  $$\begin{aligned} & \Pr[\text{Any bad } h \text{ consistent with the } M \text{ examples}] \\ \leq \quad & |\mathcal{C}|(1-\epsilon)^M \\ \leq \quad & |\mathcal{C}|\left((1-\epsilon)^{1/\epsilon}\right)^{\log |\mathcal{C}| + \log \frac{1}{\delta}} \\ \leq \quad & \delta. \end{aligned}$$

Therefore, it is unlikely to have any bad h, and the algorithm has the required property. ■

Remarks:

- Since $h \in \mathcal{C}$, this algorithm applies for the proper learning model.
- We did not use the fact that $x \in_U D$, so the brute force algorithm also applies for the PAC model.
- Once we have a good h, we can
  1. predict values of f on new random inputs (with same distribution as in the example oracle).
  2. compress the description of samples. Given samples $(x_1, f(x_1)), ..., (x_m, f(x_m))$, we can rewrite them as $(x_1, ..., x_m) \cup \{\text{Description of } h\}$. Since the description of h requires $O(\log |\mathcal{C}|)$ bits, it is a genuine compression if $m \geq \log |\mathcal{C}|$.

3 Learning Conjunction Functions

Let $\mathcal{C}$ be the family of conjunctions f over $\{0,1\}^n$, for example $f(x) = x_i x_j x_k$. We cannot hope to learn $f \in \mathcal{C}$ with zero error. For example, to distinguish $f_1(x) = \prod_{i=1}^n x_i$ from $f_2(x) = 0$, we need to sample almost all points in general, since f_1, f_2 coincide for all points but $x = (1, ..., 1)$.

If we learn by the brute force algorithm, the sample complexity is $\frac{1}{\epsilon}\left(\log(2^n) + \log\left(\frac{1}{\delta}\right)\right) = \frac{n}{\epsilon} + \frac{1}{\epsilon}\log\left(\frac{1}{\delta}\right)$. But the runtime is an issue. Here is a polynomial time algorithm:

1. Draw poly$(\frac{1}{\epsilon})$ examples to estimate $\Pr[f(x) = 1]$ to the additive error $\pm\frac{\epsilon}{4}$.
2. If the estimate is $< \frac{\epsilon}{2}$, output $h = 0$.
3. Otherwise, draw enough examples such that there are $k = \log(n/\delta)$ positive examples.
4. Let $V = \{i : x_i \text{ is set in the same way in each positive example}\}$.
5. Output $h(x) = \bigwedge_{i \in V} y_i$, where $y_i = x_i$ if $x_i = 1$ in the positive examples, $y_i = \bar{x}_i$ if $x_i = 0$ in the positive examples.

In step (1), it can be shown by the Hoeffding inequality that the estimate has the required additive error bound with high probability.

In step (3), we have $\Pr[f(x) = 1] \geq \frac{\epsilon}{4}$, since the estimate $\geq \frac{\epsilon}{2}$ and the additive error is bounded by $\frac{\epsilon}{4}$. Therefore, in every $O(\frac{1}{\epsilon})$ examples, in expectation we will see a new positive example, i.e. $(x, f(x))$ such that $f(x) = 1$. By sampling $O(k/\epsilon)$ times, we will have k positive examples in expectation, and in fact with high probability, for example by Chernoff inequality.

<u>Behaviour:</u> For i in the conjunction f, i.e. x_i or $\bar{x}_i$ appears in f, then it must be set in the same way in each positive example. Therefore, $i \in V$. For i not in the conjunction f, we know that

$$\begin{aligned} & \Pr[i \in V] \\ = \; & \Pr[x_i \text{ is set in the same way in each of the } K \text{ examples}] \\ = \; & \frac{1}{2^{k-1}}. \end{aligned}$$

This implies that

$$\Pr[\text{Any } i \text{ not in } f \text{ survives}] \leq \frac{n}{2^{k-1}} < \delta,$$

since $k = \log(n/\delta)$. Now, the sample complexity is $O\left(\frac{1}{\epsilon}\log\left(\frac{n}{\delta}\right)\right)$, which is better than the brute force algorithm's. By working backwards, we see that this order bound is tight. ■

4 Learning and Fourier Representation

For approximating the Fourier coefficient of an unknown Boolean function, we have the following lemma.

Lemma 1 *For $f : \{0,1\}^n \to \{0,1\}$, we can approximate any Fourier coefficient $\hat{f}(S)$ to within additive error γ (i.e. $|\text{Output} - \hat{f}(S)| \leq \gamma$) with probability $\geq 1 - \delta$ with $O(\frac{1}{\gamma^2}\log\frac{1}{\delta})$ samples.*

Sketch of Proof Use Hoeffding inequality and the fact that $\hat{f}(S) = 1 - 2\Pr[f = \chi_S]$ ■

It is thought to be unlikely in this model that one can efficiently find the "heavy" Fourier coefficients with non-trivial values. Exhaustive search is intractable, and no other method is immediately clear. If we are dealing with a function class where most of the weight is in the coefficients with small S, we can exhaustively check these values. We will see this approach in work in the next lecture.