Advanced Machine Learning - Hilary Term 2017
2 : Consistent Learners, Occam's Razor

Lecturer: Varun Kanade

1 Occam's Razor

In the first part of this lecture, we'll study an *explanatory framework* for learning. In the PAC learning framework, what is important is a guarantee that, with high probability, the output hypothesis performs well on unseen data, *i.e.,* data drawn from the distribution D. Here we consider the following question: Given $(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)$, where $x_i \in X_n$ and $y_i \in \{0, 1\}$, can we find some hypothesis, $h : X_n \to \{0, 1\}$ that is consistent with the observed data, *i.e.,* for all i, $h(x_i) = y_i$.

If there is no restriction on the output hypothesis, then this can be simply achieved by "memorizing" the data. In particular, one could output a program of the form, *"if $x = x_1$, output y_1, else if $x = x_2$, output y_2, …, else if $x = x_m$, output y_m, else output 0"*. This output hypothesis is correct on all of the observed data and predicts 0 on all other instances. Clearly, we would not consider this as a form of learning. The basic problem here is that the "explanation" of the data is as long as the data itself.

Exercise: Show that there is a DNF formula with $O(m)$ terms that is consistent with the data.

The condition that we want to impose is that the explanation of the data be *succinct*, at the very least, asymptotically shorter than the length of the data itself. In computational learning theory, this is referred to as the *Occam Principle* or *Occam's Razor*, named after the medieval philosopher and theologian, William of Ockham, who expounded the principle that "explanations should be not made unnecessarily complex".[1]

Philosophical Implications*

The notion of *succinct explanations* can be formalised in several ways and has deep connections to various areas of mathematics and philosophy. There are connections to Kolmogorov complexity which leads to the "minimum description length" (MDL) principle. The MDL principle itself can be given a Bayesian interpretation of assigning a larger prior probability to shorter hypotheses. The existence of a short description also implies existence of compression schemes. We will not discuss these issues in detail in this course; the interested student is referred to the following sources as a starting point (de Wolf, 1997; Jaynes, 2003; Grünwald, 2007).

Typically, finding the "shortest hypothesis" consistent with the data may be intractable or even uncomputable. In order to get useful results out of this principle, we do not need to find the shortest description or achieve optimal compression. It turns out that it is enough for the description of the output hypothesis to be slightly shorter than the amount of data observed. We'll formalise this notion to derive PAC-learning algorithms from explanatory hypotheses.

2 Consistent Learning

We'll define the notion of a consistent learning algorithm, or consistent learner, for a concept class C.

[1] This is by no means a wholly accurate depiction of the writings of William of Ockham. Those interested in the history are encouraged to look up the original work.

Definition 1 (Consistent Learner). *We say that an algorithm L is a consistent learner for a concept class C using hypothesis class H, if for all n, for all $c \in C_n$ and for all m, given $(x_1, c(x_1)), (x_2, c(x_2)), \ldots, (x_m, c(x_m))$ as input, where $x_i \in X_n$, L outputs $h \in H_n$ such that for $i = 1, \ldots, m$, $h(x_i) = c(x_i)$. We say that L is an* efficient consistent learner *if the running time of L is polynomial in n,* size(c) *and m.*

A consistent learning algorithm is simply required to output a hypothesis that is consistent with all the training data provided to it. So far, we have not imposed any requirement on the hypothesis class H. This notion of *consistency* is closely related to the empirical risk minimisation principle in the machine learning literature, where the risk is defined using the *zero-one* loss.

The main result we will prove that if H is "small enough", something that is made precise in the theorem below, then a consistent learner can be used to derive a PAC-learning algorithm. This theorem shows that short explanatory hypotheses do in fact also possess predictive power.

Theorem 2 (Occam's Razor, Cardinality Version). *Let C be a concept class and H a hypothesis class. Let L be a consistent learner for C using H. Then for all $n \geq 1$, for all $c \in C_n$, for all D over X_n, for all $0 < \epsilon < 1/2$ and all $0 < \delta < 1/2$, if L is given a sample of size m drawn from $\mathsf{EX}(c, D)$, such that,*

$$m \geq \frac{1}{\epsilon}\left(\log |H_n| + \log \frac{1}{\delta}\right),$$

then L is guaranteed to output a hypothesis $h \in H_n$ that with probability at least $1 - \delta$, satisfies $\mathrm{err}(h) \leq \epsilon$.

If further more, L is an efficient consistent learner, $\log |H_n|$ is polynomial in n and size(c), *and H is polynomially evaluatable, then C is efficiently PAC-learnable using H.*

Proof. Fix a target concept $c \in C_n$ and the target distribution D over X_n. Call a hypothesis, $h \in H_n$ "bad" if $\mathrm{err}(h) \geq \epsilon$. Let A_h be the event that m independent samples drawn from $\mathsf{EX}(c, D)$ are all consistent with h. Then, if h is bad, $\mathbb{P}[A_h] \leq (1 - \epsilon)^m \leq e^{-\epsilon m}$.

Consider the event,

$$\mathcal{E} = \bigcup_{h \in H_n : h \text{ bad}} A_h$$

Then, by a simple application of the union bound we have,

$$\mathbb{P}[\mathcal{E}] \leq \sum_{h \in H_n : h \text{ bad}} \mathbb{P}(A_h) \leq |H_n| \cdot e^{-\epsilon m}$$

Thus, whenever m is larger than the bound given in the statement of the theorem, except with probability δ, no "bad" hypothesis is consistent with m random examples drawn from $\mathsf{EX}(c, D)$. However, any hypothesis that is not "bad", satisfies $\mathrm{err}(h) \leq \epsilon$ as required. □

Remark 3. *The version of the theorem described above only allows H_n to depend on C_n and n. It is possible to have a much more general version, where H_n may depend also on m, ϵ, and δ. As long as $\log |H_n|$ is a small enough function of these parameters, a PAC-learning algorithm can still be derived from a consistent learner. These more general versions appear in the book by Kearns and Vazirani (1994, Chap. 2).*

3 Improved Sample Complexity

Learning CONJUNCTIONS

Let us revisit some of the learning algorithms we've seen so far. We derived an algorithm for learning conjunctions. At the heart of the algorithm was in fact a consistent learner, obtained only using positive examples. Thus, for the conjunction learning algorithm $C_n = H_n$. Note that the number of conjunctions on n literals is 3^n (each variable may appear as a positive literal, negative literal, or not at all).

Our analysis of the conjunction learning algorithm showed that if the number of examples drawn from $\mathsf{EX}(c, D)$ was at least $\frac{2n}{\epsilon}\left(\log(2n) + \log\frac{1}{\delta}\right)$, the output hypothesis with high probability has error at most ϵ. Theorem 2 shows that in fact even a sample of size $\frac{1}{\epsilon}\left(n\log 3 + \log\frac{1}{\delta}\right)$ would suffice.

Learning 3-TERM-DNF

Let us now consider the question of learning 3-TERM-DNF. We have shown that finding a 3-term DNF formula φ that is consistent with a given sample is NP-complete. On the other hand, we saw that it is indeed possible to find a 3-CNF formula that is consistent with a given sample. Let us compare the sample complexity bounds given by Theorem 2 in both of these cases. In order to do that we need good bounds on $|\text{3-TERM-DNF}|$ and $|\text{3-CNF}|$. Any 3-term DNF formula can be encoded using at most $6n$ bits, each term (or a conjunction) can be represented by a a bit string of length $2n$ to indicate whether a variable appears as a positive literal, negative literal, or not at all. Thus, $|\text{3-TERM-DNF}| \leq 2^{6n}$.

Similarly, there are $(2n)^3$ possible clauses with three literals. Thus, each 3-CNF formula can be represented by a bit string of length $(2n)^3$, indicating for each of the possible clauses whether they are present in the formula or not. Thus, $|\text{3-CNF}| \leq 2^{8n^3}$. It is also not hard to show that $|\text{3-CNF}| \geq 2^{\kappa n^3}$ for some universal constant $\kappa > 0$. Thus, it is the case that $\log|\text{3-CNF}| = \Omega(n^3)$. Thus, in order to use a consistent learner that outputs a 3-CNF formula, we need a sample that has size $\Omega\left(\frac{n^3}{\epsilon}\right)$;[2] on the other hand if we had unbounded computational resources and could solve the NP-complete problem of finding a 3-term DNF consistent with a sample, then a sample of size $O\left(\frac{n}{\epsilon}\right)$ is sufficient to guarantee a hypothesis with error at most ϵ (assuming δ is constant). This suggests that there may be tradeoff between running time and sample complexity. However, it does not rule out that there may be another computationally efficient algorithm for learning 3-TERM-DNF that has a better bound in terms of sample complexity. This question is currently open.

References

Ronald de Wolf. Philosophical applications of computational learning theory : Chomskyan innateness and occam's razor. Master's thesis, 1997.

Peter D Grünwald. *The minimum description length principle.* MIT press, 2007.

Edwin T Jaynes. *Probability theory: The logic of science.* Cambridge university press, 2003.

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

[2] At the very least, this is the lower bound we get if we apply Theorem 2. We will see shortly that in fact this is a lower bound on sample complexity for learning 3-CNF, no matter what algorithm is used.