(1) Mitchell ch. 2.

## 4.4 Learnability

Languages

- conjunctive hyp
- disj of conj.
~ decision tree
≡ Horn clauses in PC
- ~~Horn clauses~~
logic program lang.
DNF in pred. logic

In this chapter we have seen several hypothesis languages for concept learning, including conjunctions of literals (possibly with internal disjunction), conjunctions of Horn clauses, and clauses in first-order logic. It is intuitively clear that these languages differ in expressivity: for example, a conjunction of literals is also a conjunction of Horn clauses with empty if-part, so Horn theories are strictly more expressive than conjunctive concepts. The downside of a more expressive concept language is that it may be harder to learn. The field of computational learning theory studies exactly this question of *learnability*.

To kick things off we need a *learning model*: a clear statement of what we mean if we say that a concept language is learnable. One of the most common learning models is the model of *probably approximately correct* (*PAC*) learning. PAC-learnability means that there exists a learning algorithm that gets it mostly right, most of the time. The model makes an allowance for mistakes on non-typical examples: hence the 'mostly right' or 'approximately correct'. The model also makes an allowance for sometimes getting it completely wrong, for example when the training data contains lots of non-typical examples: hence the 'most of the time' or 'probably'. We assume that typicality of examples is determined by some unspecified probability distribution $D$, and we evaluate the error rate $err_D$ of a hypothesis with respect to this distribution $D$. More formally, for arbitrary allowable error rate $\epsilon < 1/2$ and failure rate $\delta < 1/2$ we require a PAC-learning algorithm to output with probability at least $1 - \delta$ a hypothesis $h$ such that $err_D < \epsilon$.

Let's assume for the moment that our data is noise-free, and that the target hypothesis is chosen from our hypothesis language. Furthermore, we assume our learner always outputs a hypothesis that is complete and consistent with the training sample. There is a possibility that this zero training error is misleading, and that the hypothesis is actually a 'bad' one, having a true error over the instance space that is larger than $\epsilon$. We just want to make sure that this happens with probability less than $\delta$. I will now show that this can be guaranteed by choosing the training sample large enough. Suppose our hypothesis space $H$ contains a single bad hypothesis, then the probability it is complete and consistent on $m$ independently sampled training examples is at most $(1-\epsilon)^m$. Since $1-\epsilon \leq e^{-\epsilon}$ for any $0 \leq \epsilon \leq 1$, we have that this probability is at most $e^{-m\epsilon}$. We want this to be at most $\delta$, which can be achieved by setting $m \geq \frac{1}{\epsilon} \ln \frac{1}{\delta}$. Now, $H$ may contain several bad hypotheses, say $k \leq |H|$; then the probability that at least one of them is complete and consistent on $m$ independently sampled training examples is at most $k(1-\epsilon)^m \leq |H|(1-\epsilon)^m \leq |H|e^{-m\epsilon}$, which is at most $\delta$ if

$$m \geq \frac{1}{\epsilon}\left(\ln|H| + \ln\frac{1}{\delta}\right) \tag{4.1}$$

This is called the *sample complexity* of a complete and consistent learner. The good news is that it is linear in $1/\epsilon$ and logarithmic in $1/\delta$. Notice that this suggests that it is exponentially cheaper to reduce the failure rate than it is to reduce the error. Any learning algorithm that takes time polynomial in $1/\epsilon$ and $1/\delta$ to process a single training example will therefore also take polynomial training time, another requirement for PAC-learnability. However, finding a complete and consistent hypothesis is not tractable in many hypothesis languages.

Notice that the term $\ln|H|$ arose because in the worst case almost all hypotheses in $H$ are bad. However, in practice this means that the bound in Equation 4.1 is overly pessimistic. Still, it allows us to see that concept languages whose size is exponential in some parameter $n$ are PAC-learnable. For example, the number of conjunctions over $n$ Boolean variables is $3^n$, since each variable can occur unnegated, negated or not at all. Consequently, the sample complexity is $(1/\epsilon)(n \ln 3 + \ln(1/\delta))$. For example, if we set $\delta = 0.05$ and $\epsilon = 0.1$ then the sample complexity is approximately $10(n \cdot 1.1 + 3) = 11n + 30$. For our dolphin example with $n = 4$ this is clearly pessimistic, since there are only $2^4 = 16$ distinct examples! For larger $n$ this is more realistic. Notice also that the PAC model is distribution-free: the learner is not given any information about the instance distribution $D$. This is another source for pessimism in the bound on the sample complexity.

We may not always be able to output a complete and consistent hypothesis: for instance, this may be computationally intractable, the target hypothesis may not be representable in our hypothesis language, or the examples may be noisy. A reasonable strategy would be to choose the hypothesis with lowest training error. A 'bad' hypothesis is then one whose true error exceeds the training error by at least $\epsilon$. Using some results from probability theory, we find that this probability is at most $e^{-2m\epsilon^2}$. As a result, the $1/\epsilon$ factor in Equation 4.1 is replaced by $1/2\epsilon^2$: for $\epsilon = 0.1$ we thus need 5 times as many training examples compared to the previous case.

It has already been mentioned that the $|H|$ term is a weak point in the above analysis. What we need is a measure that doesn't just count the size of the hypothesis space, but rather gives its expressivity or capacity in terms of classification. Such a measure does in fact exist and is called the *VC-dimension* after its inventors Vladimir Vapnik and Alexey Chervonenkis. We will illustrate the main idea by means of an example.

---

**Example 4.7 (Shattering a set of instances).** Consider the following instances:

$$m = \mathsf{ManyTeeth} \land \neg\mathsf{Gills} \land \neg\mathsf{Short} \land \neg\mathsf{Beak}$$

$$g = \neg\mathsf{ManyTeeth} \land \mathsf{Gills} \land \neg\mathsf{Short} \land \neg\mathsf{Beak}$$

$$s = \neg\mathsf{ManyTeeth} \land \neg\mathsf{Gills} \land \mathsf{Short} \land \neg\mathsf{Beak}$$

$b =$ ¬ManyTeeth ∧ ¬Gills ∧ ¬Short ∧ Beak

There are 16 different subsets of the set $\{m, g, s, b\}$. Can each of them be represented by its own conjunctive concept? The answer is yes: for every instance we want to exclude, we add the corresponding negated literal to the conjunction. Thus, $\{m, s\}$ is represented by ¬Gills ∧ ¬Beak, $\{g, s, b\}$ is represented by ¬ManyTeeth, $\{s\}$ is represented by ¬ManyTeeth ∧ ¬Gills ∧ ¬Beak, and so on. We say that this set of four instances is *shattered* by the hypothesis language of conjunctive concepts.

The VC-dimension is the size of the largest set of instances that can be shattered by a particular hypothesis language or model class. The previous example shows that the VC-dimension of conjunctive concepts over $d$ Boolean literals is at least $d$. It is in fact equal to $d$, although this is harder to prove (since it involves showing that no set of $d+1$ instances can be shattered). This measures the capacity of the model class for representing concepts or binary classifiers. As another example, the VC-dimension of a linear classifier in $d$ dimensions is $d+1$: a threshold on the real line can shatter two points but not three (since the middle point cannot be separated from the other two by a single threshold); a straight line in a two-dimensional space can shatter three points but not four; and so on.

The VC-dimension can be used to bound the difference between sample error and true error of a hypothesis (which is the step where $|H|$ appeared in our previous arguments). Consequently, it can also be used to derive a bound on the sample complexity of a complete and consistent learner in terms of the VC-dimension $D$ rather than $|H|$:

$$m \geq \frac{1}{\epsilon} \max\left(8D \log_2 \frac{13}{\epsilon}, 4 \log_2 \frac{2}{\delta}\right) \tag{4.2}$$

We see that the bound is linear in $D$, where previously it was logarithmic in $|H|$. This is natural, since to shatter $D$ points we need at least $2^D$ hypotheses, and so $\log_2 |H| \geq D$. Furthermore, it is still logarithmic in $1/\delta$, but linear times logarithmic in $1/\epsilon$. Plugging in our previous values of $\delta = 0.05$ and $\epsilon = 0.1$, we obtain a sample complexity of $\max(562 \cdot D, 213)$.

We conclude that the VC-dimension allows us to derive the sample complexity of infinite concept classes, as long as they have finite VC-dimension. It is furthermore worth mentioning a classical result from computational learning theory which says that a concept class is PAC-learnable if and only if its VC-dimension is finite.