1 PAC-Learning and VC-Dimension

In this section we will have a look to some statistical learning method. Previously we focused our attention on estimating the generalization error of a given model in order to measure the true performance. The training error usually is not a good indicator of how a model is behaving, but is way easier to estimate than other error terms. So we would like to extract as many information as possible from training error. Also there are cases where the training error is somewhat a good estimation of the performance. For example when we have a good number of samples relative to our hypothesis space we are less prone to overfitting. A question arises naturally. Can we estimate how many samples are necessary given an hypothesis space? We can answer this question in a theoretical way. Please remind that from a practical point of view it is rarely used.

1.1 PAC-Learning

To introduce the ingredients of the theoretical setting we use a character recognition task. Given an array of n bits encoding a binary-valued image we have

- X **Instances set**. In the character recognition problem, the instance space is $X = \{0,1\}^n$. The set of all possible input binary images.
- H **Hypothesis space**. The space where lies all possible combination of parameters.
- C **Set of target concept**. A concept is a subset $c \subset X$. One concept is the set of all patterns of bits in $X = \{0,1\}^n$ that encode a picture of the letter "P".
- $\mathcal{P}$ **Probability distribution over X**. Training instances are generated by a fixed, unknown probability distribution over X

The learner observes a sequence $\mathcal{D}$ of training example $< x, c(x) >$ for some target concept $c \in C$ and x is drawn from $\mathcal{P}$. The learner must output a hypothesis h estimating c. h is evaluated by its performance on subsequent instances drawn according to $\mathcal{P}$

$$L_{true} = P_{x \in \mathcal{P}}(c(x) \neq h(x))$$

Our objective is to bound L_{true} given L_{train}. Now we introduce the so called **version space** $VS_{H,\mathcal{D}}$. It is a subset of H where the training error L_{train} is zero. So the hypothesis h are always correct on training instances. For now, we assume that VS is non-empty. Making some consideration we can bound L_{train} inside VS.

Theorem 1.1 (L_{train} bound in VS)**.** *If the hypothesis space H is finite and $\mathcal{D}$ is a sequence of $N \geq 1$ independent random examples of some target concept c, then for any $0 \leq \epsilon \leq 1$, the probability that $VS_{H,\mathcal{D}}$ contains a hypothesis error greater then ϵ is less than $|H|e^{-\epsilon N}$:*

$$P(\exists h \in H : L_{train}(h) = 0 \wedge L_{true}(h) \geq \epsilon) \leq |H|e^{-\epsilon N}$$

Proof.

$$P((L_{train}(h_1) = 0 \wedge L_{train}(h_1) \geq \epsilon) \vee \cdots \vee (L_{train}(h_{|H|}) = 0 \wedge L_{train}(h_{|H|}) \geq \epsilon))$$

$$\begin{aligned}
&\leq \sum_{h \in H} P(L_{train}(h) = 0 \wedge L_{train}(h) \geq \epsilon) && \text{Union bound} \\
&\leq \sum_{h \in H} P(L_{train}(h) = 0 | L_{train}(h) \geq \epsilon) && \text{Bound using Bayes' rule} \\
&\leq \sum_{h \in H_{bad}} (1-\epsilon)^N && \text{Bound on individual } h_i s \\
&\leq |H|(1-\epsilon)^N && |H|_{bad} \leq |H| \\
&\leq |H| e^{-\epsilon N} && (1 - \epsilon \leq e^{-\epsilon}, for \quad 0 \leq \epsilon \leq 1)
\end{aligned}$$

□

We can notice that the dimension of the hypothesis space influences negatively the bound, in fact a larger searching space will give us less guarantees on the value of L_{true}. On the other hand, having more samples is always better, in fact $e^{-\epsilon N}$ is monotonically decreasing with N. Larger ϵ will lead to smaller bound because we are less demanding on the similarity between L_{true} and L_{train}. Now we can bound the probability of $P(\exists h \in H : L_{train}(h) = 0 \wedge L_{true}(h) \geq \epsilon) \leq |H| e^{-\epsilon N}$. We can set a parameter δ

$$|H| e^{-\epsilon N} \leq \delta$$

After choosing δ we can calculate N or ϵ.
Given ϵ and δ

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

Given N and δ

$$\epsilon \geq \frac{1}{N}\left(ln(|H|) + ln\left(\frac{1}{\delta}\right)\right) \tag{2}$$

Note $|H|$ can be very large. If we take as an example a binary decision problem with M binary inputs(features), the size of H will be 2^{2^M}. So N will have a exponential relationship with M. This is related to the curse of dimensionality.

Example Suppose H contains conjunctions of constraints on up to M boolean attributes (i.e., M literals). In this case $|H| = 3^M$. How many examples are sufficient to ensure with probability at least $(1 - \delta)$ that every h in $VS_{H,\mathcal{D}}$ satisfies $L_{true}(h) \leq \epsilon$?

$$N \geq \frac{1}{\epsilon}\left(Mln(3) + ln\left(\frac{1}{\delta}\right)\right)$$

Now we are ready to define formally what PAC[1]is. Consider a class C of possible target concepts defined over a set of instances X of length n, and a learner L using hypothesis space H.

Definition 1.1. *C is **PAC-learnable** if there exists an algorithm L such that for every $f \in C$, for any distribution $\mathcal{P}$, for any ϵ such that $0 \leq \epsilon < \frac{1}{2}$, and δ such that $0 \leq \delta < \frac{1}{2}$, then algorithm L, with probability at least $(1-\delta)$, outputs a concept h such that $L_{true}(h) \leq \epsilon$ using a number of samples that is polynomial of $\frac{1}{\epsilon}$ and $\frac{1}{\delta}$*

Definition 1.2. *C is **efficiently PAC-learnable** by L using H $\iff$ $\forall c \in C$, distributions $\mathcal{P}$ over X, ϵ such that $0 \leq \epsilon < \frac{1}{2}$, and δ such that $0 \leq \delta < \frac{1}{2}$, algorithm L, with probability at least $(1-\delta)$, outputs a concept h such that $L_{true}(h \leq \epsilon)$, in time that is polynomial in $\frac{1}{\epsilon}$, $\frac{1}{\delta}$, M and size(c)*[2]

Now we need to generalize to problems where the VS is empty, because usually the train error is not equal to zero(agnostic learning). We can simply bound the difference between L_{train} and L_{true} in H.

$$L_{true}(h) \leq L_{train} + \epsilon$$

From now on we will consider only binary classification problems for simplicity. As we did before, we need to find an upper bound for the probability of having a "bad event", which in this case consists in having a gap between L_{train} and L_{true} bigger than ϵ. To achieve this we use the **Hoeffding bound**, which states

Definition 1.3. *For N i.i.d. coin flips $X_1, \ldots, X_N$, where $X_i \in \{0,1\}$ and $0 < \epsilon < 1$, we define the empirical mean $\bar{X} = \frac{1}{N}\sum_{i=1}^{N} X_i$, obtaining the following bound*

$$P(E[\bar{X}] - \bar{X} > \epsilon) \leq e^{-2N\epsilon^2}$$

Theorem 1.2. *Given an hypothesis space H, a dataset $\mathcal{D}$ with N i.i.d. samples, $0 < \epsilon < 1$: for any learned hypothesis h:*

$$P(\exists h \in H | L_{true}(h) - L_{train}(h) > \epsilon) \leq |H| e^{-2N\epsilon^2}$$

This is very similar to what we have found in the non-empty case. Like we did before, we can calculate the number of example needed given ϵ and δ

$$N \geq \frac{1}{2\epsilon^2}\left(ln(|H|) + ln\left(\frac{1}{\delta}\right)\right)$$

Or ϵ given N and δ.

$$\epsilon \geq \sqrt{\frac{ln(|H|) + ln(\frac{1}{\delta})}{2N}}$$

[1]Probably Approximately Correct learning. Probably refers to δ and it is the confidence. Approximately refers to ϵ and it is the accuracy

[2]size(c): Number of bit necessary to express c. It comes from information theory

Now we can rewrite the gap between L_{train} and L_{true} as

$$L_{true}(h) \leq \underbrace{L_{train}(h)}_{Bias} + \underbrace{\sqrt{\frac{ln(|H|) + ln(\frac{1}{\delta})}{2N}}}_{Variance} \tag{3}$$

Once more, we can see how $|H|$ influences the loss function. For large $|H|$ we assume a low bias because it's more probable to find a good h and a high variance. For small $|H|$ we have high bias because we have a low probability of including a good h and low variance. In practice what we are saying is that we have to justify a large H with a lot of data. If we do so the training error will be a good estimation of the overall performance(test/true error).

1.2 VC Dimension

So far we have considered only finite hypothesis space. If we use the bound that we have just found in an infinite[3]H, we would have infinite variance. This is not the case, for infinite H the previous bound is too pessimistic. So we need to find a new one. In the finite H case we encoded the complexity of H in the number of possible hypothesis. In the infinite case we can't do this, so we need to find a new metric to measure the complexity of H. We will use the **VC dimension**. To lay the ground for our theoretical discussion, we need to introduce two definitions,

Definition 1.4 (Dichotomy)**.** *A dichotomy of a set S is a partition of S into two disjoint subsets*

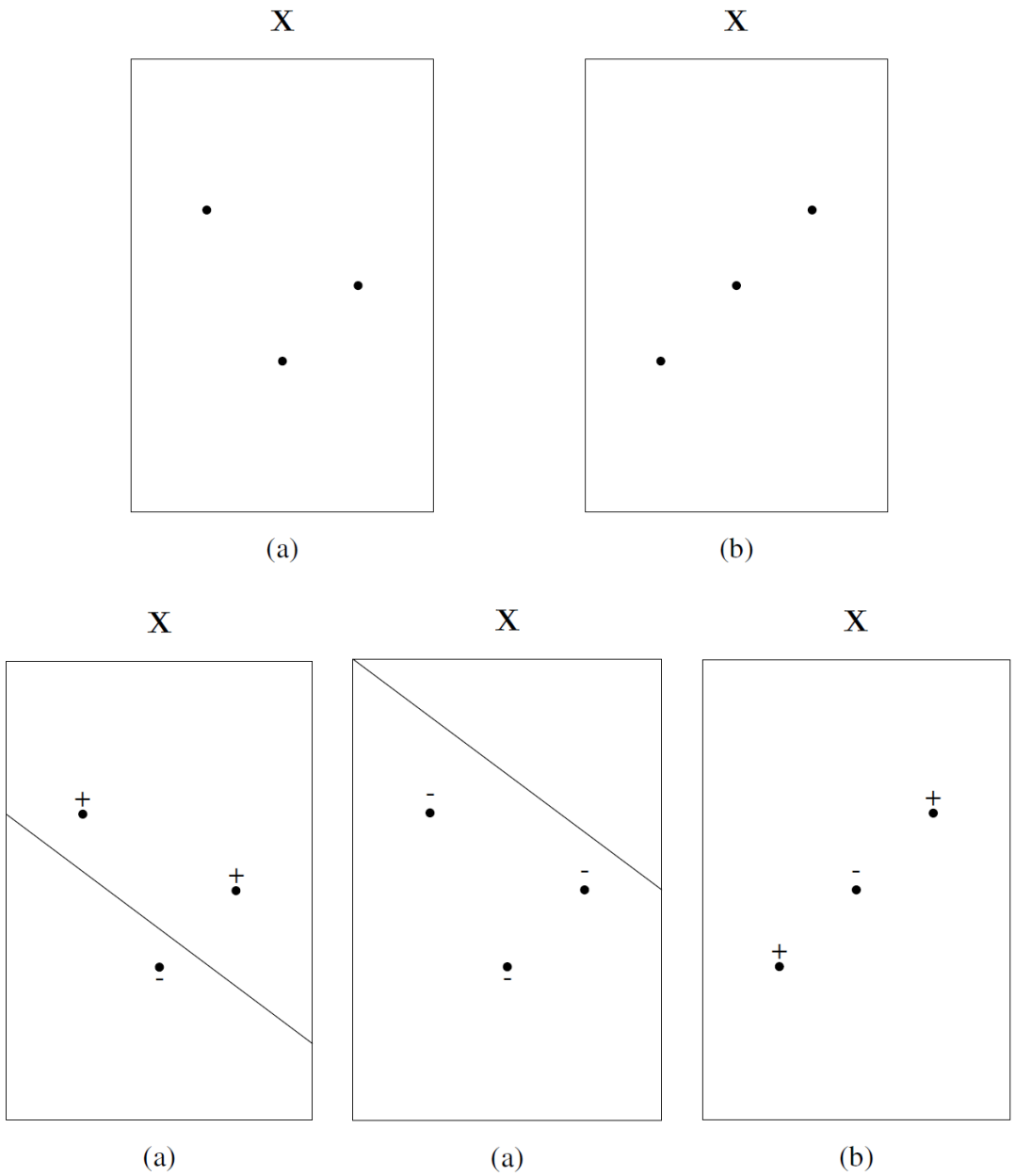
Definition 1.5 (Shattering)**.** *A set of instances S is shattered by hypothesis space H if and only if for every dichotomy of S there exists some hypothesis in H consistent with this dichotomy*

In practice shattering means that if we split our instances set in two not overlapping subset, H shatters S if and only if for every dichotomy we can find an hypothesis which classify correctly every instances. As a reminder we are still considering only binary classification problems.

Example We consider an instances set with three example[4]and an hypothesis space representing a linear classifier. In this case a dichotomy is a specific assignment of class (+) or class (-) to every point. We can have two cases, one where the examples are not aligned and one where they are. H shatters any of the two instances set? Yes, only (a) because we can always find a line that divides every possible dichotomy. In the specific dichotomy shown for (b), we can't find a linear classifier(line), that classifies correctly the three points, so H doesn't shatter (b).

[3]Infinite hypothesis spaces are very common. For example linear regression or classification have infinite hypothesis spaces

[4]This is still infinite because every instance can have a different "position" in the instances set



Note that for four instances, H won't shutter S.[5] Now we can define what is the VC dimension

Definition 1.6 (VC dimension)**.** *The Vapnik-Chervonenkis dimension, $VC(H)$, of hypothesis space H defined over instance space X, is the size of the largest finite subset of X shattered by H. If arbitrarily large finite sets of X can be shattered by H, then $VC(H) \equiv \infty$*

In our previous example, the VC dimension of linear classifiers in two dimension is three. In fact, we can have a configuration of three instances whereby every dichotomy is perfectly separable. This doesn't hold for four instances.

[5]This is the XOR problem. It's not linearly separable

Example Few examples of VC dimensions,

- Linear classifier: $VC(H) = M + 1$, for M features plus the constant term.
- Neural networks: $VC(H) = \#parameters$
- 1-Nearest Neighbor: $VC(H) = \infty$
- SVM with Gaussian Kernel: $VC(H) = \infty$. We will see SVM in future chapters.

Now we can find a new bound for the error between L_{train} and L_{true}, and so we can find how many randomly drawn examples suffice to guarantee an error of at most ϵ with probability at least $(1 - \delta)$

$$N \geq \frac{1}{\epsilon}\left(4log_2\left(\frac{2}{\delta}\right) + 8VC(H)log_2\left(\frac{13}{\epsilon}\right)\right) \tag{4}$$

Equally we can express this as an upper bound for L_{true}

$$L_{true}(h) \leq L_{train}(h) + \sqrt{\frac{VC(H)(ln(\frac{2N}{VC(H)}) + 1) + ln(\frac{4}{\delta})}{N}} \tag{5}$$

Properties

Theorem 1.3. *The VC dimension of a hypothesis space $|H| < \infty$ is bounded from above*

$$VC(H) \leq log_2(|H|)$$

Proof. If $VC(H) = d$ then there exists at least 2^d functions(combination) in H, since there are at least 2^d possible labelings

$$\begin{aligned} |H| &\geq 2^d \\ |H| &\geq 2^{VC(H)} \\ VC(H) &\leq log_2(|H|) \end{aligned}$$

□

Theorem 1.4. *Concept class C with $VC(C) = 1$ is not PAC-learnable.*