4. Learnability and VC Dimension

4.1. PAC framework

When considering the learning of algorithms, naturally the question arises, what kind of problems can be learned efficiently. Which problems are easier to learn than others? How many training examples are necessary for learning a problem? How precisely can we fit a hypothesis to the function we are looking for? In this chapter we are going to establish and explain a mathematical theory to answer these questions. The PAC framework defines a notion of learnable functions. In PAC, the learnability of a functions is defined the size of the training examples that is needed for finding a well-fitting hypothesis, as well as the complexity of the learning algorithm.

4.1.1. Assumptions and Notation

In the following sections, we will always assume that training and test data are drawn from the same distribution $\mathcal{D}$. That makes sense, because otherwise there would be no connection between training and test, and therefore no way of learning. For this chapter we will limit ourselves to the case of binary classification. Extensions of our results to more general settings can be found in Mohri, Rostamizadeh, and Talwalkar 2012 and Vapnik 1998. Let's quickly fix the notations and notions that will be used.

- The set $\mathcal{X}$ denotes the *input space*, i.e. the set of all possible examples, just as in part one of the thesis
- $\mathcal{Y}$ denotes the output space. For convenience, we will here only treat binary classification, i.e. $\mathcal{Y} = \{-1, 1\}$
- A *target* function or target concept $f : \mathcal{X} \to \mathcal{Y}$ is a mapping from $\mathcal{X}$ to $\mathcal{Y}$. Again, f is unknown and we'll try to approximate them as well as possible.
- A set of concepts can be combined to a concept class C
- The hypothesis set H is a fixed set of possible concepts, i.e. mappings from $\mathcal{X}$ to $\mathcal{Y}$. The learning algorithm tries to find the one concept out of H that approximates the target concept best.

- The *training examples* consist of a sample $S = (x_1, .., x_m)$, as well as the respective labels $(f(x_1), ..., f(x_m))$. This time we assume that the examples are independently and identically distributed (i.i.d.) according to some fixed but unknown distribution $\mathcal{D}$.
- $g_S \in \mathcal{H}$ is the "best" hypothesis our algorithm chooses out of the set of hypotheses, based on the training examples.

We can summarize our learning model with the following illustration:

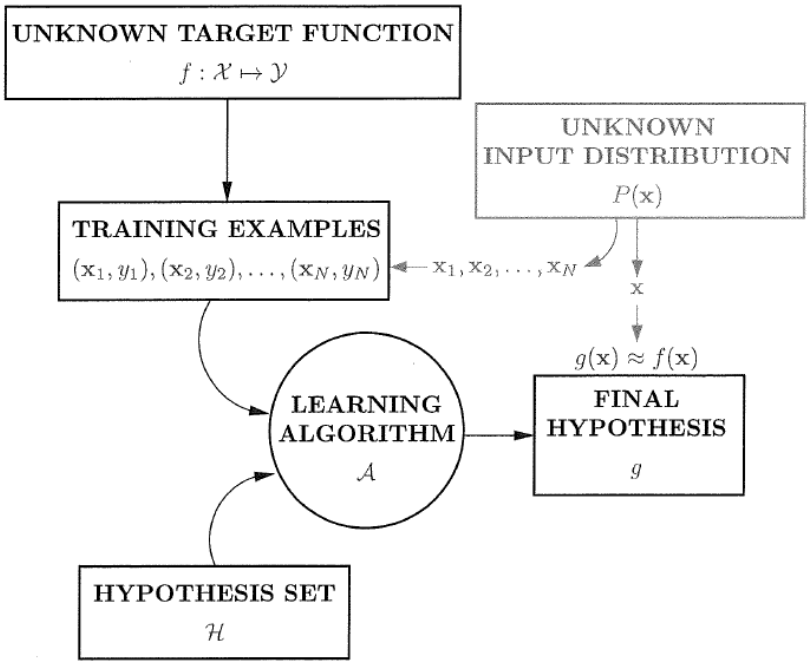


Figure 4.1.: Learning Model after (Abu-Mostafa, Magdon-Ismail, and Lin 2012)[1]

The learning algorithm $\mathcal{A}$ considers a certain hypothesis set H from which he tries to find a certain hypothesis $g \in H$ to approximate an unknown target function $f : \mathcal{X}$ as well as possible. Therefore he has a number of *training examples* $x_1, ..., x_m$ at his disposal, which are drawn from a probability distribution $\mathcal{D}$ and that are labeled according to f. In order to see, how well a function matches the target function, we need to define some kind of performance measure for the hypotheses.

4.1.2. Generalization Error and Empirical Error

Thus we see, that learning is concerned with finding a function g that is as similar as possible to a certain target function or concept f. The problem, however, is that we only have a limited amount of data points at our disposal, whereas the entire target function may be defined on a much larger set. What we don't know yet: Can a data set of limited

size yield enough information to approximate the target function well? And how should our algorithm choose the "best" hypothesis $g \in \mathcal{H}$? It might give us a hypothesis that performs well on the training data, but does not generalize well to hitherto unseen data. So does the data set S reveal something about the classes of the points outside of S? In order to find that out, we need a measure of error between the a target function f and a certain hypothesis $h \in H$:

The first kind of error is the one we can actually measure: The in-sample error (or empirical error) tells us the fraction of S where the target f and a hypothesis $h \in H$ disagree:

Definition 4.1.3 (Empirical Error/In-Sample Error)**.** *Given a hypothesis $h \in H$, a target function f, and an underlying distribution $\mathcal{D}$ of the training examples, the empirical error or in-sample-error of h is defined by*

$$E_{in}(h) = \frac{1}{N} \sum_{n=1}^{N} \mathbb{1}(h(x_n) \neq f(x_n))$$

The empirical error tells us how well a hypothesis performs on the training data set. What we don't know is how well or h performs on the whole input space $\mathcal{X}$, i.e. how large the "true" error of our hypothesis is. This is quantified by the generalization error:

Definition 4.1.4 (Generalization Error/Out-of-Sample-Error)**.** *Given a hypothesis $h \in \mathcal{H}$, a target function f and an underlying distribution $\mathcal{D}$, the generalization error of h is defined by*

$$E_{out}(h) = P(h(x) \neq f(x))$$

Now we can come up with a more sophisticated definition of learning than the one we gave above.

> The learner considers a fixed set of possible concepts H, called a hypothesis set. He receives a sample $S = (x_1, ..., x_m)$ drawn i.i.d. according to a distribution $\mathcal{D}$ as well as the labels $(f(x_1), ..., f(x_m))$, which are based on a specific target function f to learn. His task is to use the labeled sample S to select a hypothesis $g \in \mathcal{H}$ that has a small generalization error.[1]

Let now us shortly make clear what this implies. Our learning algorithm will choose a hypothesis $g \in H$ in order to approximate the unknown target function f. If learning is possible, E_{out} should be approximately zero. This is, however, impossible for us to verify directly, since f and E_{out} remain unknown. We can, however, examine under which conditions $E_{out} - E_{in} \approx 0$. If those conditions are met, then we want to pick a hypothesis

[1] Abu-Mostafa, Magdon-Ismail, and Lin 2012.

$g \in H$ with low empirical error E_{in} and we will know it generalizes well. The in-sample-error we can check directly[1]. So it is our aim to find a hypothesis, whose in-sample error is low and which generalizes well, i.e. a hypothesis

$$E_{in} \approx 0$$

and equally to ensure that

$$E_{in} - E_{out} \approx 0$$

so we can use E_{in} as a proxy for E_{out}.

4.1.5. PAC-learning

We have seen that, for learning to be possible, that the in-sample error of the learned hypothesis should be small and approximate the out-of-sample error: $E_{in} \approx E_{out} \approx 0$. We now formalize this idea by introducing the *Probably Approximately Correct* (PAC)-learning framework:

Definition 4.1.6 (PAC-learning)**.** *A concept class C is said to be PAC-learnable, if there exists an algorithm $\mathcal{A}$ and a polynomial function $poly(\cdot, \cdot)$ such that for any $\epsilon > 0$ and $\delta > 0$, for all distributions $\mathcal{D}$ on input$\mathcal{X}$, the following holds for any training sample size $m \geq poly(1/\epsilon, 1/\delta)$:*

$$P(E_{out}(g) \leq \epsilon) \geq 1 - \delta$$

When such an algorithm $\mathcal{A}$ exists, it is called a PAC-learning algorithm for C.

In some cases, for example for intervals, one can show that the problem is PAC-learnable and derive a specific polynomial bound.

In a more general case, when we want to show that a problem is PAC-learnable, it can be hard to find a concrete polynomial bound. In the following section, we will derive conditions for the learnability for a larger set of functions, that are easier to prove.

[1] Abu-Mostafa, Magdon-Ismail, and Lin 2012, p.100.

4.2. Learning bounds for finite hypothesis sets

4.2.1. Hoeffding's inequality

To qualify the relationship between generalization error and empirical error, we use a bound for the difference of both items. Therefore we use Hoeffding's inequality, one of the most powerful tools in learning theory:

Theorem 4.2.2. *Let $X_1, ..., X_n$ be random independent, identically distributed random variables, such that $0 \leq X \leq 1$. Then,*

$$P(|\frac{1}{n}\sum_{i=1}^{n} -E[X]| \geq \epsilon) \leq e^{-2n\epsilon^2}$$

The reader can find a proof of a more general version of the theorem can be found in the appendix.

Hoeffding's inequality belongs to the concentration inequalities, that give probabilistic bounds for a random variable to be concentrated around its mean. The intuition is as follows: Suppose, we have some random variables. We know, that if we take the average, we usually should get something close to the expectation.

How can we apply this to our learning problem? The Hoeffding Inequality provides a way to characterize the discrepancy between E_{in} and E_{out}. Suppose we only have one possible hypothesis h. Immediately from substituting E_{in} and E_{out}, we obtain:

Theorem 4.2.3. *Let $h \in H$ be a fixed hypothesis, f an arbitrary target function.*

$$P(|E_{in}(h) - E_{out}(h)| \geq \epsilon) \leq \delta = e^{-2n\epsilon^2},$$

for any $\epsilon > 0$, where n denotes the training sample's size.

So now we can give an upper bound for the discrepancy of E_{in} and E_{out}.

Corollary 4.2.4 (Generalization bound - single hypothesis)**.** *Fix a hypothesis $h : \mathcal{X} \to \{0, 1\}$. Then, for any $\delta > 0$, the following inequality holds with probability at least $1 - \delta$:*

$$E_{out}(h) \leq E_{in}(h) + \sqrt{\frac{log\frac{2}{\delta}}{2m}}.$$

Proof. Set the right-hand side of to be equal to δ and solve for ϵ. □

4.2.5. The case of finite hypothesis sets

Consider now a situation in which we have a finite hypothesis class $H = \{h_1, ..., h_k\}$ consisting of k hypotheses. Our training algorithm will pick the hypothesis g with the smallest empirical error. We now want to give some guarantees on the generalization error of g. We already have derived a bound for a fixed hypothesis $h \in H$.

The problem is: g is not fixed, but a random variable depending on the sample S.e S. Therefore we have to find *a uniform convergence bound* $P(|E_{in}(g) - E_{out}(g))$ that holds for the set of all hypotheses, which in particular include g. We need to consider all hypotheses simultaneously. We consider all possible "bad events" with $|E_{in} - E_{out}| > \epsilon$ and get[1]:

$$
\begin{aligned}
\text{"}|E_{in}(g) - E_{out}(g)| \geq \epsilon\text{"} \Rightarrow \text{"}&|E_{in}(h_1) - E_{out}(h_1)| \geq \epsilon\text{"} \\
&\textbf{or } \text{"}|E_{in}(h_2) - E_{out}(h_2)| \geq \epsilon\text{"} \\
&... \\
&\textbf{or } \text{"}|E_{in}(h_k) - E_{out}(h_k)| \geq \epsilon\text{"}
\end{aligned}
$$

Using the union bound and applying 4.2.3 to each hypothesis yields:

$$
\begin{aligned}
P(E_{in}(g) - E_{out}(g)|) \geq \epsilon) &\leq P(\exists h \in H | E_{in}(h) - E_{out}(h)| > \epsilon) \\
&= P\left((|E_{in}(h_1) - E_{out}(h_1)| > \epsilon) \vee ... \vee (|E_{in}(h_k) - E_{out}(h_k)| > \epsilon)\right) \\
&\leq \sum_{h \in H} P(|E_{in}(h) - E_{out}(h) > \epsilon) \\
&\leq 2|H| \exp(-2m\epsilon^2)
\end{aligned}
$$

Setting the right hand side equal to δ and solving for ϵ gives the following learning bound:

Theorem 4.2.6 (Learning bound for finite H)**.** *Let H be a finite hypothesis set. Then, for any $\delta > 0$, with probability at least $1 - \delta$, the following inequality holds:*

$$
\forall h \in H, E_{out}(h) \leq Ein(h) + \sqrt{\frac{\log |H| + \log \frac{2}{\delta}}{2m}}
$$

This inequality can be applied to g, but yields a very crude estimation, because applying the union bound we assume the "bad events" with poor generalization are disjoint for

[1] Mohri, Rostamizadeh, and Talwalkar 2012, p.16.

all $h \in H$. We will later see that this is not the case. Notice however that this theorem already shows that when the hypothesis set H is finite, the learning algorithm $\mathcal{A}$ is a PAC-algorithm, since the sample complexity is bounded by a polynomial in $1/\epsilon$ and $1/\delta$.

4.3. Learning bounds for infinite hypothesis sets

We have derivated bounds for finite hypothesis sets. But in machine learning, the hypothesis sets are usually infinite. Consider for example all hypotheses that are parameterized with real numbers, for example the Perceptron or the SVM. There are infinitely many hyperplanes in spaces of whatever dimension. In example **??** we saw, that we can learn from a finite set of training examples, even when the hypothesis set is infinite. In this case, however the bound in formula 4.2.6 will go to infinity. In the following chapter, we will derive learning bounds that can deal with infinite hypothesis sets. To do this, we need a less generous estimation for $|E_{in} - E_{out}|$ than the union bound. We will show that many of the hypotheses in H are similar, meaning that the "bad" regions with $|E_{in}(h_i) - E_{out}(h_i)| \geq \epsilon$ are in fact often overlapping and so the generalization error can be bound by a smaller term.

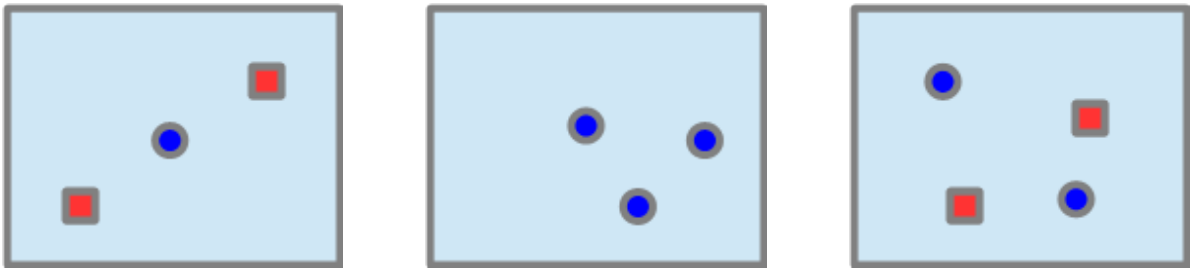
4.3.1. Growth Function

The idea will be the following: We show that the hypotheses in a hypothesis set H can be "similar" to each other and therefore their "bad events" with poor generalization can overlap. Therefore, we define the growth function, that formalizes the number of "effective" hypotheses in a hypothesis set. It tells us how many different hypotheses that H can yield, if one only considers a finite sample of points $x_1, ..., x_n$. Through the growth function, we finally want to replace the factor $|H|$ in the generalization bound 4.2.6. The smaller the growth function, the more similar the hypotheses in H are and the more overlap we have. We will again focus on binary classification here, that is on hypotheses that map $\mathcal{X}$ to $\mathcal{Y} = \{-1, 1\}$.

Definition 4.3.2 (Dichotomy)**.** *Let $x_1, ..., x_m \in \mathcal{X}$. The dichotomies generated by H on these points are defined by*

$$H(x_1, ...x_m) = \{(h(x_1), ..., h(x_m))|h \in H\}.$$

One can think of the dichotomies as the set of hypotheses of H, but seen through the eyes of m points $x_1, ..., x_m$ only. Imagine the input space as a canvas painted with two



(a) (b) (c)

Figure 4.2.: Illustration of the growth function for the 2-dim. perceptron. The dichotomy on (a) cannot be generated by the perceptron, but the choice of the three points in (b) yield a dichotomy that H can shatter. The dichotomy (c) cannot be generated by a perceptron. For 4 points we can get at most 14 dichotomies out of 2^4.

colors for the different labels. It is covered with a veil, which has holes at $x_1, ..., x_m$, so that the observer can not see the whole painting but only single points. Dichotomies are the different patterns of colors one can see through the holes. Some hypotheses may look different on the whole canvas may look the same if you only can see m points, that is, they yield the same dichotomy[1].

The growth function is a map based on the number of dichotomies:

Definition 4.3.3 (Growth Function)**.** *The growth function is defined for a hypothesis set H by*

$$\Pi_H(m) = \max_{x_1,...,x_m \in \mathcal{X}} \{|H(x_1, ..., x_m)| : h \in H\}$$

In other words: $\Pi_H(m)$ is the maximum number of dichotomies that can be generated by H on any m points. If we want to compute the growth function, we consider all possible choices of $x_1, ..., x_m \in \mathcal{X}$ and pick the one that yields the most dichotomies. For any m points, since there are two labels per point available, the maximum number of dichotomies possible is

$$\Pi_H(m) \leq 2^m$$

If H is capable of creating all possible dichotomies on a set of m points $S = \{x_1, ..., x_m\}$, then $H(x_1, ..., x_m) = \{-1, +1\}$ and we say that H *shatters* S. This means that H is as diverse as possible on this sample.

Example 4.3.4 (Perceptron)**.** Let $\mathcal{X}$ be the Euclidean plane and H is the two-dimensional perceptron. We calculate the growth function for $m = 1, 2, 3, 4$. For $m = 1$ we have a single point that has either positive or negative label, therefore $\Pi_H(m) = 2 = 2^1$. For $m = 2$, either both points lie on the same part of the hyperplane or on different parts. In both cases we can switch the labels, and we get $\Pi_H(m) = 4 = 2^2$. In case of $m = 3$ we

[1] Abu-Mostafa, Magdon-Ismail, and Lin 2012.

have two possibilities. If the three points are collinear, not all labelings are possible with one hyperplane as separator. But according to the definition of growth functions we can choose the points that maximize the number of possible labelings. So if we consider the points arranged in a triangle, we can get all possible labelings: $\Pi_H(m) = 2^3$. Either the points have all one color (two possibilites) or one point has a different color than the other (6 possibilites). In case of $m = 4$ datapoints, in Figure 4.2a we see a dichotomy that the perceptron can not generate (The XOR-gate). One can proof that there are no four points that the perceptron cannot shatter, that is whatever points we pick, we will only get at most 14 dichotomies.

We will now show how to compute the entire growth function for some simple hypothesis sets:

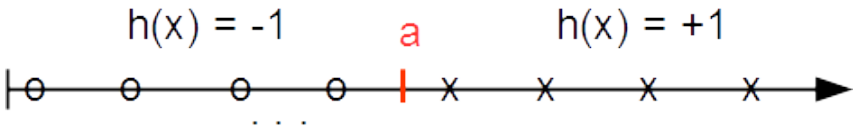


Figure 4.3.: growth function for positive rays

Example 4.3.5 (Positive Rays)**.** H contains all hypotheses $h : \mathbb{R} \mapsto \{-1, +1\}$ with $h(x) = sign(x - a)$, i.e. the hypotheses are defined in one-dimensional input ray and they return -1 to the left of some value a and $+1$ on the right. We notice that given m points, the line is split into $m + 1$ different regions. Which region contains the value a decides the dichotomy we get. As we move a we can get $m + 1$ dichotomies. Since we can't get any more dichotomies on any m points, the growth function is:

$$\Pi_H(m) = m + 1.$$

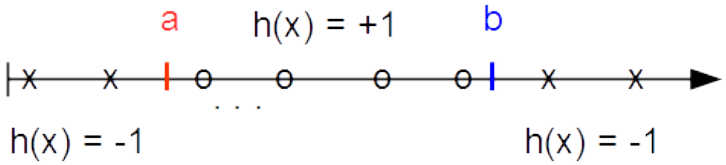


Figure 4.4.: growth function for positive intervals

Example 4.3.6 (Positive Intervals)**.** This time, H consists of all hypotheses in one dimensions that map to 1 within a certain interval and to -1 otherwise. Each hypothesis is defined by the two values at the end of the interval. Again, the line is split into $m + 1 sets$ by the points. Now the dichotomy is specified by which two regions contain the end values of that interval. Therefore we get $\binom{m+1}{2}$ different dichotomies. If both end values fall into

the same region, we get only one dichotomy, that maps all points to -1. We add up the two possibilities and get:

$$\Pi_H(m) = \binom{m+1}{2} + 1 = \frac{1}{2}m^2 + \frac{1}{2}m + 1.$$

In this case $\Pi_H(m)$ grows quadratically, faster than the linear growth function of the simpler example of the positive ray.

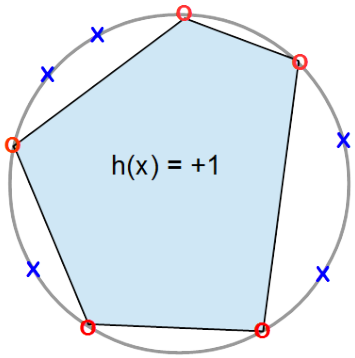


Figure 4.5.: growth function for convex sets

Example 4.3.7 (Convex Sets)**.** H contains all the hypotheses in two dimension $h : \mathbb{R}^2 \mapsto \{-1, 1\}$ that are positive inside some convex set and negative otherwise (a set is convex, if any line connecting two points inside the set lies itself entirely inside the set).

This time, to compute $\Pi_H(m)$we choose the m points wisely. Remember, we may choose the m points in any way that maximizes the number of dichotomies we get. As you can see in Figure ... we choose the m points on the perimeter of a circle. Now we choose an arbitrary pattern of ± 1 on these points and consider the resulting dichotomy. Connecting the points with label $+1$, you get the closed interior of a polygon. The hypothesis made up of this interior is convex, since its vertices are on the perimeter of a circle and it agrees with the dichotomy on all m points. So we get every possible dichotomy on every m points with a convex hypothesis. The growth function has the maximum possible value:

$$\Pi_H(m) = 2^m.$$

Notice: If we had chosen the m points randomly in a plane rather than in a circle, we wouldn't be able to shatter all the points with convex hypothesis. However, the growth function is defined on the maximum of dichotomies, over all possible subsets of m.

In practice, it may not be possible to compute $\Pi_H(m)$ for every hypothesis set and luckily, there is an easier way. In the following part we will show that we can find an upper bound

for $\Pi_H(m)$. As soon as we have found the bound, we can use $\Pi_H(m)$ to replace $|H|$ in 4.2.6.

4.3.8. VC Dimension

We now derive an upper bound for the growth function $\Pi_H(m)$, for all $m \in \mathbb{N}$.For proving the polynomial bound, we define a new combinatorial quantity, the VC dimension. The VC (Vapnik-Chervonenkis) dimension is a single parameter that characterizes the growth function:

Definition 4.3.9. *The VC dimension of a hypothesis set H, denoted by $d_{VC}(H)$, is the largest value of m, for which $\Pi_H(m) = 2^m$. If $\Pi_H(m) = 2^m$, then $d_{VC}(H) = \infty$.*

To illustrate this definition, we will now take a second look at the examples for the growth function to learn their VC-dimension. To find a lower bound we have to simply find a set S that can be shattered by H. To give an upper bound, we need to prove that no set S of $d+1$ points exists, that can be shattered by H, which is usually more difficult.[1]

Example 4.3.10 (Positive rays)**.** We have shown that the growth function for positive rays is $\Pi_H(m) = m + 1$. Only for $m = 0, 1$ we have $\Pi_H(m) = 2^m$, therefore $d_{VC}(H) = 1$.

Example 4.3.11 (Positive intervals)**.** The growth function for positive intervals is $\Pi_H(m) = \frac{1}{2}m^2 + \frac{1}{2}m + 1$. We have $\Pi_H(m) = 2^m$ for $m = 0, 1, 2$ which yields $d_{VC}(H) = 2$.

Example 4.3.12 (Convex sets)**.** We have seen that by arrange convex sets in the right way, sets of every size can be shattered. Therefore $\Pi_H(m) = 2^m$ for all m and $d_{VC}(H) = \infty$

Example 4.3.13 (Perceptrons)**.** The next example will be a bit more elaborate. We will show that for the perceptron in $\mathbb{R}^d$, the VC-dimension is always $d + 1$. We won't explicitely calculate the growth function $\Pi_H(m)$ for all m. Therefore, for determining the VC dimension $d_{VC}(H) = d + 1$, we have to show that
a) The VC dimension is at least d+1: To prove this, we have to find $d + 1$ points in the input space $\mathcal{X} = \mathbb{R}^d$ that the perceptron can shatter. We first consider the set of hyperplanes in $\mathbb{R}^2$. We have already seen that any three non-collinear points in $\mathbb{R}^2$ can be shattered by the Perceptron 4.2c. We also have shown that four points cannot be shattered (XOR-gate can not be realised). Therefore: $d_{VC}(\text{hyperplanes in } \mathbb{R}^2) = 3$. If we now show the general case of hyperplanes in $\mathbb{R}^d$. We pick a set of $d+1$ points in $\mathbb{R}^d$, setting x_0 to be the origin and defining x_i, $i \in \{1, ..., x\}$ as the points whose i-th coordinate is 1 and all others are 0. Let $y_0, ..., y_n \in \{-1, +1\}$ be an arbitrary set of labels for $x_0, x_1, ...x_d$. Let w

[1] Mohri, Rostamizadeh, and Talwalkar 2012, p.40.

be the vector whose ith coordinate is y_i. The perceptron defined by the hyperplane of equation $w \cdot x + \frac{y_0}{2} = 0$ is shatters $x_0, ..., x_d$ because for any $i \in \{0, ..., d\}$:

$$sgn(w \cdot x_i + \frac{y_0}{2}) = sgn(y_i + \frac{y_0}{2}).$$

Now we need to show that
b) the VC-dimension is at most $d + 1$. This is a bit trickier, because we have to show that there is no subset of more than $d + 1$ points that can be shattered by the perceptron. To prove this, we will show that on any $d + 2$ points, there is a dichotomy that can not be realized by the perceptron classifier.

Choose the points $x_1, ..., x_{d+2}$ at random. There are more points than dimension, therefore we must have

$$x_j = \sum_{i \neq j} a_i x_i,$$

i.e. one point is a linear combination of the rest of the points. This will apply to any set of $d + 2$ points you choose. Also, some of the a_i's must be nonzero, because the first coordinate of the x_i's is always one (see definition of the perceptron, first coordinate is one to include the bias term of the hyperplane into the form $w \cdot x$).

Now we show a dichotomy that can't be implemented: Consider the following dichotomy. Let $y_1, ..., y_{d+2}$ the labels of $x_1, ..., x_{d+2}$. Give x_i with nonzero coefficient a_i get the label $+1$, give any label to the x_i with $a_i = 0$ and set $y_j = -1$ as the label of x_j. Let $w \in \mathbb{R}^d + 1$ be the weight vector to any hyperplane h. Now we have

$$x_j = \sum_{i \neq j} a_i x_i \Rightarrow w^T x_j = \sum_{i \neq j} a_i w^T x_i$$

If $y_i = \text{sgn}(w^T x_i) = \text{sgn}(a_i)$, then $a_i w^T x_i > 0$ for all $0 \leq i \leq d$. Then $\text{sgn}(\sum_{i \neq j} a_i w^T x_i) > 0$. However, we set $y_j = \text{sgn}(w^T x_j) = \text{sgn}(\sum_{i \neq j} w^T a_i x_i) < 0$ with gives us a contradiction. The dichotomy can't be implemented on any set of $d + 2$ points by the perceptron classifier.

Combining both parts of the proof, we get: $d_{VC}(\text{hyperplanes in } \mathbb{R}^d) = d + 1$.

Consider again the perceptron in the two-dimensional space. As shown above, its VC-dimension is three. The fact that no four points can be shattered by H limits the number of the dichotomies that can be realized significantly. We exploit that fact to get a bound

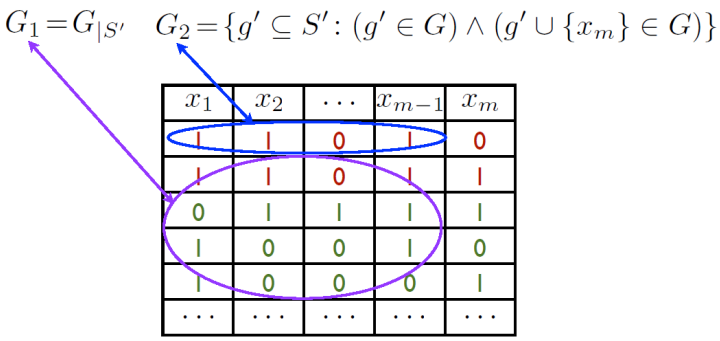


Figure 4.6.: Illustration of construction of G_1 and G_2 in the proof of Sauer's lemma

for $\Pi_H(m)$ for all $m \in \mathbb{N}$. We will prove, that, if the VC-dimension for a set of hypotheses is finite, then there is a polynomial that bounds $\Pi_H(m)$ for all values of m. If such a polynomial exists, and $\Pi_H(m)$ can replace $|H|$ in 4.2.6 then the generalization error will go to zero as $m \to \infty$. The next result uses the VC-dimension to define a bound for the growth function.

Theorem 4.3.14 (Sauer's lemma)**.** *Let H be a hypothesis set with $d_{VC}(H) = d$. Then, for all $m \in \mathbb{N}$, the following inequality holds:*

$$\Pi_H(m) \leq \sum_{i=1}^{d} \binom{m}{i}.$$

Proof. We prove this theorem by induction on $m + d$. Clearly, the statement is true for $m = 1$ and $d = 0$ or $d = 1$. Assume now, that it also holds for $(m-1, d-1)$ and $(m-1, d)$. We fix a set of points $S = \{x_1, ..., x_m\}$ with $\Pi_H(m)$ dichotomies and let $G = H_{|S}$ be the set of dichotomies we get from H on S.

Now we consider the set S without x_m. Let $S' = \{x_1, ..., x_{m-1}\}$. If we "cut off" x_m, there remain dichotomies on m points. Some of them now appear twice, namely those who were labeled identically except for the point x_m. Out of those we take the unique hypotheses and call them G_2.

The other "cut-off" dichotomies(graphic in green) appear only once. We combine them with G_2 and get the set of all unique dichotomies H gives on the first $m-1$ points. Looking at table 4.6 it is easy to see that $|G| = |G_1| + |G_2|$.

We define $G_1' = G_{|S'}$ as the set of unique dichotomies H gives on the first $m-1$ points.

Since $d_{VC}(G_1) \leq d_{VC}(G) \leq d$, we get by using the induction hypothesis and definition of the growth function:

$$|G_1| \leq \Pi_{G_1}(m-1) \leq \sum_{i=0}^{d} \binom{m-1}{i}.$$

Also, by definition of G_2, if a set $Z \in S'$ is shattered by α, then $Z \cup x_m$ is shattered by G

Hence:

$$d_{VC}(G_2) \leq d_{VC}(G) - 1 = d - 1$$

We can now again apply the definition of $\Pi_H(m)$ and the induction hypothesis and get

$$|G_2| \leq \Pi_{G_2}(m-1) \leq \sum_{i=0}^{d-1} \binom{m-1}{i}.$$

Finally, we can conclude,

$$|G| = |G_1| + |G_2| \leq \sum_{i=0}^{d} \binom{m-1}{i} + \sum_{i=0}^{d-1} \binom{m-1}{i} = \sum_{i=0}^{d} \binom{m-1}{i} + \binom{m-1}{i-1} = \sum_{i=0}^{d} \binom{m}{i},$$

which completes the proof. □

Thus we see, there are only two cases: Either the VC dimension $d_{VC} = d$ is finite, and the growth function is bound by a polynomial of degree d, or the VC dimension is infinite and $\Pi_H(m) = 2^m$. In the first case, the growth function can be used as a generalization bound:

Corollary 4.3.15. *Let H be a hypothesis set with $d_{VC}(H) = d$. Then for all $m \geq d$,*

$$\Pi_H(m) \leq \left(\frac{em}{d}\right)^d = O(m^d).$$

This tells us the following: If we can use the growth function in 4.2.6 instead of the factor $|H|$ the generalization error will can be bound, given there have enough training examples.

Proof.

$$\begin{aligned}
\Pi_H(m) &\leq \sum_{i=0}^{d} \binom{m}{i} \\
&\leq \sum_{i=0}^{d} \binom{m}{i} \left(\frac{m}{d}\right)^{d-i} && (m \geq d) \\
&\leq \sum_{i=0}^{m} \binom{m}{i} \left(\frac{m}{d}\right)^{d-i} \\
&= \left(\frac{m}{d}\right)^{d} \sum_{i=0}^{m} \binom{m}{i} \left(\frac{d}{m}\right)^{i} \\
&= \left(\frac{m}{d}\right)^{d} \left(1 + \frac{d}{m}\right)^{m} && \text{(binomial theorem)} \\
&\leq \left(\frac{m}{d}\right)^{d} e^{d}. && ((1-x) < e^{-x})
\end{aligned}$$

□

4.4. VC Generalization Bound

From Sauer's lemma, we can derive a generalization bound that also holds in the case of an hypothesis set of infinite cardinality:

Theorem 4.4.1. *Let H be a family of functions taking values in $-1, +1$ with VC-dimension d. Then, for any $\delta > 0$, with probability at least $1 - \delta$, the following holds for $h \in H$:*

$$E_{out}(h) \leq E_{in}(h) + \sqrt{\frac{8d \log(m/d) + 8 \log \frac{4}{\delta}}{m}}$$

Note that the log factor only plays a minor role in the bound. It is possible to eliminate that factor with a finer analysis.[2] Therefore the discrepancy between in- and out-of-sample-error mostly depends on the VC-dimension of the hypothesis set.

Proof. [3] We will only sketch the idea of the proof here. For a complete proof, see[1]. In the previous chapter we said it was our idea to replace the factor $|H|$ in Hoeffding's theorem. Let's quickly remind us where this factor came from.

For a given hypothesis $h \in H$, the event "$|E_{in} - E_{out}| > \epsilon$" consists of all data points for which the statement is true. We can call this a "bad event", since in this case the

2 Ibid., p.48.
3 Abu-Mostafa, Magdon-Ismail, and Lin 2012, after p. 55 ff.
1 Vapnik 1998.

hypothesis performs well on the training data, but doesn't generalize well to unseen data. The hypothesis g our algorithm chooses, however, is not fixed but a random variable. In the previous section, we used the union bound to estimate the probability that there is a "bad event" in any of the hypotheses. This however, is a very imprecise estimation, since the bad events in fact can strongly overlap.

The greatest part of the proof is concerned with how to account for the overlaps. The argument is the following. Many hypothesis share the same dichotomy on a given training set S, since there are only finitely many dichotomies, even when the hypotheses are infinite. Any statement based on S alone will simultaneously be true or false for all the hypotheses that share the same dichotomy on a particular set S. The growth function lets us account for this kind of hypothesis redundancy in a precise way, i.e. it accounts for the "effective" number of hypotheses.

When H is finite, the redundancy factor will also be infinite, because the infinitely many hypotheses will be distributed amongst finitely many dichotomies. Therefore, the overlap in the "bad events" will be dramatic and the total probability of $|E_{in} - E_{out}|$ will shrink immensely.

So, if we could now immediately replace the number of all hypotheses $|H|$ by the number of effective hypotheses $\Pi_H(m)$ in Hoeffding's inequality, we get the following bound:

$$E_{out}(g) \overset{?}{\leq} E_{in}(g) + \sqrt{\frac{1}{2m} \log \frac{2\Pi_H(m)}{\delta}}$$

This is unfortunately not exactly what the proof gives us. We get a boundary a bit more loose:

$$E_{out} \leq E_{in}(g) + \sqrt{\frac{8}{N} \log \frac{4\Pi_H(2m)}{\delta}}$$

The reason, why $\Pi_H(2m)$ appears in the proof instead of $\Pi_H(m)$ is that the proof uses a sample of $2m$ instead of m points.

To justify, why the condition we get for a sample size $2m$ can replace the original condition for a sample of only m points, we have to shrink the ϵ's by a factor of 4, and we end up with a factor of 2 in the estimate of the overall probability.[2]

If we now use the polynomial bound based on the VC-dimension we got through Sauer's lemma instead $\Pi_H(2m)$, we get the formula we wanted to prove.

[2] Abu-Mostafa, Magdon-Ismail, and Lin 2012, p.55.

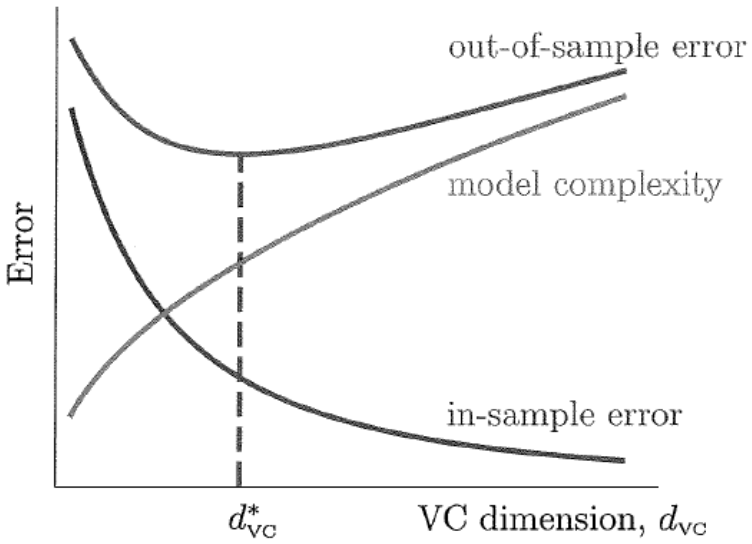


Figure 4.7.: A too simple model, won't fit the training data well and will have a high in-sample error. A more complex learning model, with more features and parameters, has a higher VC dimension d_{VC}. It is likely to fit the training data better, resulting in a lower in-sample error, but the generalization will be worse. Some intermediate d^*_{VC} represents a tradeoff between the two errors.

□

The overall takeaway is the following: The more complex a model we use(i.e. the higher the VC-dimension of the model is) the more likely it is, that we fit the training data well. However,we it gets more likely that the out-of-sample-error will be approximated by the empirical error, which reflects the bias-variance-tradeoff discussed in chapter one. We can attain a minimum for the out-of-sample error at some intermediate VC dimension d^*_{VC}.

Bibliography

Abu-Mostafa, Yaser S., Malik Magdon-Ismail, and Hsuan-Tien Lin (2012). *Learning from data: A short course*. Pasadena, CA: AML Book. ISBN: 1600490069.

Mohri, Mehryar, Afshin Rostamizadeh, and Ameet Talwalkar (2012). *Foundations of machine learning*. Adaptive computation and machine learning series. Cambridge, MA: MIT Press. ISBN: 026201825X.

Vapnik, Vladimir N. (1998). *Statistical learning theory*. Adaptive and learning systems for signal processing, communications, and control. New York and Chichester: Wiley. ISBN: 0471030031.