Lista 1

MAE-5832: Introducão ao Aprendizado de Máquina

Camila Castro Moreno

Maio 2020

1 Comente sobre o diagrama abaixo. O que o diagrama como um todo ilustra e o que cada componente representa?

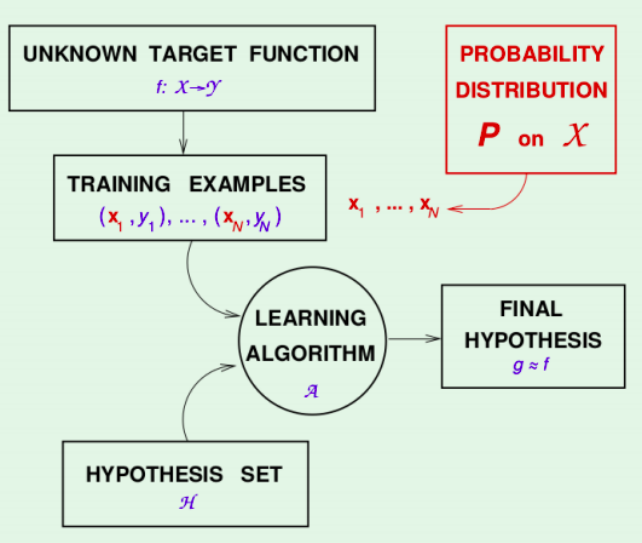

Figure 1: The basic learning setup with added probability distribution.

In Figure 1 we can see that an **unknown probability distribution** P on $\mathcal{X}$ is used to randomly sample n data points of $x \in \mathcal{X}$. These sampled data points along with their respective outputs $y \in \mathcal{Y}$, given by the **unknown target function** $f: \mathcal{X} \mapsto \mathcal{Y}$, make up the **training examples** dataset $\mathcal{D}$. The **learning algorithm** $\mathcal{A}$ uses the training data to explore different hypothesis $h \in \mathcal{H}$, where $\mathcal{H}$ is the **hypothesis set** that has a small error, and tries to find a **final hypotheses** $g \in \mathcal{H}$ that approximates the **unknown target function** f, that is, in which $g(x) \approx f(x)$.

2 O que é E_{in} e E_{out}? Dê um exemplo concreto.

E_{in} is the *in-sample error*, that is, the error rate within a sample, in this case the sampled training dataset $\mathcal{D}$ of size N. It is therefore a measure of training performance. We mathematically define E_{in} of a hypothesis h as:

$$\begin{aligned} E_{in}(h) &= (\text{fraction of } \mathcal{D} \text{ where } f \text{ and } h \text{ disagree}) \\ &= \frac{1}{N}\sum_{n=1}^{N} [\![h(x_n) \neq f(x_n)]\!] \text{ where,} \end{aligned}$$

$$[\![\text{statement}]\!] = \begin{cases} 1 \text{ if statement is true,} \\ 0 \text{ otherwise.} \end{cases}$$

E_{out} is the *out-of-sample error*, that is, the error rate outside of a given sample based on the performance over an entire input space $\mathcal{X}$. E_{out} measures how well the training on $\mathcal{D}$ has generalized to data that wasn't trained. We mathematically define E_{out} of a hypothesis h as:

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

Let's say there's going to be a nationwide vote and we take a poll to evaluate vote intention (in this case our sample or training data). If we model a function or a classifier (our best hypothesis) to this training data, the values incorrectly classified will make up our *in-sample error*. The *out-of-sample error* would be the error after applying our best hypothesis to the entire population, which is technically impossible to obtain. In this way, we can calculate E_{in} but E_{out} can only be estimated.

3 Por que apenas minimizar E_{in} não é suficiente? Em alguma situação pode ser suficiente?

It is not enough to minimize E_{in} because our hypothesis might be overfitting the training data $\mathcal{D}$ and cannot be generalized for data outside of $\mathcal{D}$, that is, we may minimize the E_{in} and still have a large E_{out}. It would only be justifiable to overfit the training data (minimize E_{in}) if our sample size N is large enough, in which case, according to the *Hoeffding Inequality*, as N grows it becomes exponentially unlikely (small probability) that $E_{in}(h)$ will deviate from $E_{out}(h)$ by more than a user-defined tolerance ϵ:

$$P[|E_{in} - E_{out}| > \epsilon] \leq 2e^{-2\epsilon^2 N} \qquad \text{for any } \epsilon > 0. \tag{1}$$

4 Quando consideramos a formulação teórica de aprendizado de máquina, uma das possibilidades é investigar o valor $|E_{in} - E_{out}|$. O que esse valor expressa e porque nos interessa investigar ele?

The absolute difference between E_{in} and E_{out} indicates if the hypothesis is or isn't overfitting the training data in detriment to a whole dataset. If the difference is small, then the hypothesis characterizes the sample and the population in the same manner (this of course does not mean the error is small, just that the hypothesis can extrapolate well outside of the sample data and produce similar results).

5 O prof. Abu-Mostafa menciona recorrentemente o termo hipótese. Ao que ele se refere quando fala em hipótese?

The hypothesis h is a function from the hypothesis set $\mathcal{H}$ that is tested to approximate to the unknown target function f that describes how $\mathcal{X}$ maps to $\mathcal{Y}$. The learning algorithm's final hypothesis g should best approximate to f. In an ideal scenario we'd have $g(x) = f(x)$ for every $x \in \mathcal{X}$. Realistically we'll have $g \approx f$.

6 A desigualdade de Hoeffding, no contexto de aprendizado de máquina, com respeito a uma certa hipótese h, é dada por:

$$P[|E_{in} - E_{out}| > \epsilon] \leq 2e^{-2\epsilon^2 N} \qquad \textbf{for any } \epsilon > 0.$$

Explique o significado dessa desigualdade.

E_{in} is a random variable that depends on the sample and E_{out} is unknown but not random. Although the probability P and the distribution of E_{in} depends on E_{out}, which is unknown, the *Hoeffding Inequality* permits us to bound P by $2e^{-2\epsilon^2 N}$, which does not depend on E_{out}. As mentioned in Exercise 3, the RHS of the inequality shows that the size of N is what affects the bound.

7 A desigualdade de Hoeffding, no contexto de aprendizado de máquina, com respeito a uma situação em que o espaço de hipóteses consiste de M exemplos é dada por:

$$P[|E_{in} - E_{out}| > \epsilon] \leq 2Me^{-2\epsilon^2 N} \qquad \textbf{for any } \epsilon > 0.$$

Comente o significado da diferença entre essa desigualdade e a do item anterior.

For the previous inequality (eq. 1) to be valid, we considered that h was fixed **before** generating the dataset. For a learning setup, this is not the case: the

final hypothesis g **depends** on the data, and therefore can only be selected **after** generating the dataset.

Let's say we have a hypothesis set $\mathcal{H}$ of size M and that a hypothesis $h_m \in \mathcal{H}$ where $m \in M$. We don't need that P be small for all h_m, but that it be small for g. Regardless of the algorithm and sample, g must be one of our h_m's. That being said, it must be true that:

$$|E_{in}(g) - E_{out}(g)| > \epsilon \implies (\exists\ m \in M)\ |E_{in}(h_m) - E_{out}(h_m)| > \epsilon$$

The RHS shows that there must be at least one $m \in M$ in which $E_{in}(h_m) - E_{out}(h_m)| > \epsilon$. All hypotheses h_m are **fixed** (this is important because it will allow us to apply the *Hoeffding Inequality* later).

Let's consider these two rules for events $\mathcal{B}_1$ and $\mathcal{B}_2$ before continuing:

1. If $\mathcal{B}_1 \implies \mathcal{B}_2$ then $P[\mathcal{B}_1] \leq P[\mathcal{B}_2]$,
2. *Boole's Inequality (Union Bound)*: $P[\cup_i \mathcal{B}_i] \leq \sum_i P[\mathcal{B}_i]$.

Applying these two rules to our previous logical implication we have that:

$$\begin{aligned} P[|E_{in}(g) - E_{out}(g)| > \epsilon] &\leq (\exists\ m \in M)\ |E_{in}(h_m) - E_{out}(h_m)| > \epsilon \\ &\leq \sum_{m=1}^{M} P[|E_{in}(h_m) - E_{out}(h_m)| > \epsilon] \end{aligned}$$

Applying the *Hoeffding Inequality* to the M terms and bounding each term in the sum by $2e^{-2\epsilon^2 N}$:

$$P[|E_{in} - E_{out}| > \epsilon] \leq 2Me^{-2\epsilon^2 N} \qquad \text{for any } \epsilon > 0. \tag{2}$$

This second inequality, therefore, is a more uniform version of the previous one, and allows the learning algorithm to to choose any h based on E_{in}. We can expect E_{out} to follow E_{in} uniformly, but for this, it is important that M be finite, since it multiplies the bound that would exist for a single hypothesis.

8 O *bound* $2Me^{-2\epsilon^2 N}$ no item anterior foi obtido aplicando-se o *union-bound*. O que é *union-bound*?

See rule 2 from exercise 7.

Another way to write the union bound in which $\mathcal{B}_m$ is the undesired event $|E_{in} - E_{out}| > \epsilon$ is:

$$P[\mathcal{B}_1 \vee \mathcal{B}_2 \vee \cdots \vee \mathcal{B}_M] \leq P[\mathcal{B}_1] + P[\mathcal{B}_2] + \cdots + P[\mathcal{B}_M].$$

It is important to note that if these events strongly overlap then the inequality produced by the union bound is a gross overstatement.

9 Que elementos são relevantes na definição de dicotomias? O que são dicotomias geradas por um espaço de hipóteses $\mathcal{H}$?

Let's consider binary target functions $f\colon \mathcal{X} \mapsto \{-1, 1\}$. We have different hypotheses $h \in \mathcal{H}$ that we want to approximate to f and we apply them to a finite sample $x_1, \cdots, x_N \in \mathcal{X}$. In doing this we obtain an N-tuple $h(x_1), \cdots, h(x_N)$ of ± 1's which we call a **dichotomy**. A dichotomy essentially splits the sample into two subsets: a subset that maps to -1 and a subset that maps to 1. Mathematically we can define the dichotomies generated by the hypothesis set $\mathcal{H}$ on the sample data as:

$$\mathcal{H}(x_1, \cdots, x_N) = \{(h(x_1), \cdots, h(x_N)) | h \in \mathcal{H}\}. \tag{3}$$

10 O que é o *growth-function*?

First, let's define the **generalization bound**, which is a form of rephrasing the *Hoeffding Inequality* (eq. 2) by considering a tolerance level δ (e.g. $\delta = 0.05$) to bind E_{out} in terms of E_{in}:

$$E_{out}(g) \leq E_{in}(g) + \underbrace{\sqrt{\frac{1}{2N} \ln \frac{2M}{\delta}}}_{\text{error bound/bar}}. \tag{4}$$

Both the generalization bound and the *Hoeffding Inequality* provide a way to characterize the generalization error (the discrepancy between E_{in} and E_{out}). In most interesting learning models $\mathcal{H}$ is an infinite set. Since the error bound depends on the hypothesis set size M, if M is infinite then the error bound becomes infinite, thus rendering the generalization bound meaningless.

Despite the M being infinite in such learning models, hypotheses are generally very similar. If a hypothesis h_1 is very similar to a hypothesis h_2 then the two events "$|E_{in}(h_1) - E_{out}(h_1)| > \epsilon$" and "$|E_{in}(h_2) - E_{out}(h_2)| > \epsilon$" are likely to coincide.

The **growth function** is the quantity that will formalize the effective number of hypotheses to replace M in the generalization bound and it is based on the number of dichotomies (see exercise 9). Mathematically we define the growth function as:

$$m_{\mathcal{H}}(N) = \max_{x_1, \cdots, x_N \in \mathcal{X}} |\mathcal{H}(x_1, \cdots, x_N)|, \tag{5}$$

where $|.|$ is the cardinality (number of elements) in a set.

$m_{\mathcal{H}}(N)$ is therefore the maximum number of dichotomies that can be generated by $\mathcal{H}$ $\forall$ N points. We take all possible subsets of N points $x_1, \cdots, x_N$ from $\mathcal{X}$ and pick the one that results in the most dichotomies.

M and $m_{\mathcal{H}}(N)$ both measure the number of hypotheses in $\mathcal{H}$, but $m_{\mathcal{H}}(N)$ is measured on N points rather than on all of $\mathcal{X}$.

Since $\mathcal{H}(x_1, \cdots, x_N) \subseteq \{-1, +1\}^N$ (the set of all possible dichotomies on any N points) for any $\mathcal{H}$, then $m_{\mathcal{H}}(N)$ is at most $|\{-1, +1\}^N|$ and:

$$m_{\mathcal{H}}(N) \leq 2^N.$$

We say that $\mathcal{H}$ can *shatter* the sample $x_1, \cdots, x_N$ if it can generate all possible dichotomies $\{-1, +1\}^N$ on the sample.

11 No contexto sendo considerado, qual o interesse em dicotomias e *growth-function*?

As mentioned in exercise 10, the growth function is based on dichotomies and used to replace M (size of the hypothesis set) in the generalization bound as to consider an infinite hypothesis set, a characteristic of many interesting learning models.

12 Qual é o interesse em se provar que o *growth-function* é polinomial?

It's not practical to try to compute $m_{\mathcal{H}}(N)$ for every hypothesis set we use, but since $m_{\mathcal{H}}(N)$ is meant to replace M in eq. 2 then we can use an upper bound on $m_{\mathcal{H}}(N)$ instead of the exact value. If no dataset of size k can be shattered by $\mathcal{H}$, then k is said to be a break point for $\mathcal{H}$ and therefore $m_{\mathcal{H}}(N) < 2^k$. It's computationally easier to find break points for $\mathcal{H}$ than to calculate full growth functions. In bounding $m_{\mathcal{H}}(N)$ by a polynomial, the generalization error characterized by the error bound in eq. 4 goes to zero as $N \to \infty$. This means that given a large N we can generalized well to untrained data. The following theorem states that any growth function with a break point is bounded by a polynomial.

Theorem 1 *If $m_{\mathcal{H}}(N) < 2^k$ for some value k, then*

$$m_{\mathcal{H}}(N) \leq \sum_{i=0}^{k-1} \binom{N}{i} \tag{6}$$

$\forall\, N$. The RHS is a polynomial of degree $k - 1$.

This means that as long as our hypothesis set $\mathcal{H}$ has a break point (keeping in mind that the smaller the break point the better the bound), we get a good generalization by means of a polynomial bound on the growth function.

13 Por que não podemos simplesmente trocar o M em $2Me^{-2\epsilon^2 N}$ pelo *growth-function* $m_{\mathcal{H}}(N)$?

First, let's define the VC dimension:

The *Vapnik-Chervonenkis* (VC) dimension, d_{VC}, of a hypothesis set $\mathcal{H}$ is the largest value of N for which $m_{\mathcal{H}}(N) = 2^N$.

By convention we say that if $m_{\mathcal{H}}(N) = 2^N \ \forall \ N$, then $d_{\text{VC}}(\mathcal{H}) = \infty$.

Since $k = d_{\text{VC}} + 1$ is a break point for $m_{\mathcal{H}}$, Theorem 1 can be rewritten in terms of d_{VC}, which will be the order of the polinomial bound on $m_{\mathcal{H}}(N)$:

$$m_{\mathcal{H}}(N) \leq \sum_{i=0}^{d_{\text{VC}}} \binom{N}{i}.$$

Further simplifying the polinomial bound:

$$m_{\mathcal{H}}(N) \leq N^{d_{\text{VC}}} + 1. \tag{7}$$

Let's try to replace M with the growth function in the generalization bound (eq. 4) to see what happens:

$$E_{out}(g) \overset{?}{\leq} E_{in}(g) + \sqrt{\frac{1}{2N} \ln \frac{2m_{\mathcal{H}}(N)}{\delta}}. \tag{8}$$

Since

$$\lim_{x \to \infty} \ln x = \infty,$$

we can see that our bound is maintained as long as $d_{\text{VC}}(\mathcal{H}) \neq \infty$. For any finite value of d_{VC}, the error bar will converge to zero at a speed determined by the VC dimension, since d_{VC} is the order of the polynomial.

We can divide learning models into two classes: "good models" with a finite VC dimension and "bad models" with infinite VC dimension. In the latter it's not possible to make generalization conclusions when replacing M with the growth function, which is why we can't simply substitute one for the other.

14 Relacione dicotomias e *VC dimension.*

When all possible $\{-1, +1\}^N$ dichotomies are generated on a sample $x_1, \cdots, x_N$, then $m_{\mathcal{H}}(N) = 2^N$ (since the growth function measures the maximum number of dichotomies generated by a hypothesis set $\mathcal{H}$). The VC dimension therefore measures the largest value of N for which all dichotomies are generated.

15 Qual é o *VC dimension* do perceptron? Como é feita a demonstração?

There are two steps to computing the VC dimension for the perceptron learning algorithm:

1. First we show that the VC dimension is at least a certain value;
2. then we show that it is at most the same value.

The logical difference between these two steps is given by:

$$d_{\text{VC}} \geq N \iff \exists\ \mathcal{D} \text{ of size } N \mid \mathcal{H} \text{ shatters } \mathcal{D}.$$

Let's analyze different conclusions in the following cases:

1. $\exists\ \mathcal{D}$ of N points that can be shattered by $\mathcal{H}$. In this case, we can conclude that $d_{\text{VC}} \geq N$.
2. $\forall\ \mathcal{D}$ of N points, $\mathcal{H}$ shatters $\mathcal{D}$. In this case, we have more than enough information to conclude that $d_{\text{VC}} \geq N$.
3. $\exists\ \mathcal{D}$ of N points that cannot be shattered by $\mathcal{H}$. Based only on this information, we cannot conclude anything about the value of d_{VC}.
4. $\nexists\ \mathcal{D}$ of N points that can be shattered by $\mathcal{H}$. In this case we can conclude that $d_{\text{VC}} < N$.

For the perceptron algorithm with $d+1$ parameters we have that $d_{\text{VC}} = d+1$. This can be demonstrated as follows:

1. **First, we show that $\mathbf{d_{VC} \geq d+1}$, by proving that $\exists\ \mathbf{d+1}$ points that the perceptron can shatter:**

   Let's consider the input space $\mathcal{X} = \{1\} \times \mathbb{R}^d$ where $\mathbb{R}^d$ is the d-dimensional Euclidean space, and let $\mathcal{Y} = \{-1, +1\}$ be the output space. In this case, the perceptron algorithm has $d+1$ dimensions. Let's represent as a column vector $\mathrm{x} = [x_0, x_1, \cdots, x_d]$, where $x_0 = 1$. We can write x as:

$$\mathrm{x} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 1 & 1 & 0 & \dots & 0 \\ 1 & 0 & 1 & \dots & 0 \\ \vdots & & & \ddots & \\ 1 & 0 & 0 & \dots & 1 \end{bmatrix},$$

   where x is a nonsingular (invertible) $d+1 \times d+1$ square matrix. We need to find one column weight vector

$$\mathrm{w} = [w_0, w_1, \cdots, w_d]^{\mathrm{T}},$$

being that $^{\mathrm{T}}$ denotes the transpose of a vector and w_0 is the bias of the perceptron formula, that satisfies:

$$y = \text{sign}(\mathrm{w}^{\mathrm{T}}\mathrm{x}) = \begin{cases} 1 \text{ if } \mathrm{w}^{\mathrm{T}}\mathrm{x} > 0 \\ -1 \text{ otherwise.} \end{cases}$$

If we make $y = \mathrm{w}^T\mathrm{x}$ then one solution is $\mathrm{w}^T = \mathrm{x}^{-1}y$. Since a set of $d+1$ points can be shattered by the perceptron formula, this implies that $d_{\mathrm{VC}} \geq d+1$.

2. **Second, we show that $\mathbf{d_{VC} \leq d+1}$, by showing that we cannot shatter a set of $\mathbf{d+2}$ points:**

   $\forall\ d+2$ points we have $\mathrm{x} = [x_0, x_1, \cdots, x_d, x_{d+1}]$. In this case we have more points $(d+2)$ than dimensions $(d+1)$, which implies that the vectors of x are linearly dependent and we can describe them mathematically as:

   $$\mathrm{x}_j = \sum_{i \neq j} a_i \mathrm{x}_i,$$

   where not all a_i's are zeros. We only need to show that there is one dichotomy that can't be generated on $d+2$ points to show that the perceptron cannot shatter $d+2$ points. Let's consider the dichotomy in which x_i's with $a_i \neq 0$ will get $y_i = \text{sign}(a_i)$ and x_j gets $y_j = -1$ and prove that it is impossible to implement. We have that:

   $$\mathrm{x}_j = \sum_{i \neq j} a_i \mathrm{x}_i \implies \mathrm{w}^{\mathrm{T}}\mathrm{x}_j = \sum_{i \neq j} a_i \mathrm{w}^{\mathrm{T}}\mathrm{x}_i$$

   If $y_i = \text{sign}(\mathrm{w}^{\mathrm{T}}\mathrm{x}_i) = \text{sign}(a_i)$, then from here we have two cases: 1) Where $\text{sign}(\mathrm{w}^{\mathrm{T}}\mathrm{x}_i) > 0 \implies a_i > 0$ or 2) where $\text{sign}(\mathrm{w}^{\mathrm{T}}\mathrm{x}_i) < 0 \implies a_i < 0$. For both cases we have that $a_i \mathrm{w}^{\mathrm{T}}\mathrm{x}_i > 0$. This implies that:

   $$\mathrm{w}^{\mathrm{T}}\mathrm{x}_j = \sum_{i \neq j} a_i \mathrm{w}^{\mathrm{T}}\mathrm{x}_i > 0$$

   and $y_j = \text{sign}(\mathrm{w}^{\mathrm{T}}\mathrm{x}_i) = 1$, rather than -1 as proposed by the dichotomy.

Since we proved that $d_{\mathrm{VC}} \geq d+1$ and $d_{\mathrm{VC}} \leq d+1$ this implies that $d_{\mathrm{VC}} = d+1$.

16 Dissemos que o *VC dimension* relaciona-se com a expressividade do espaço de hipóteses. Comente sobre isso.

We proved in Exercise 15 that the VC dimension of the perceptron learning algorithm is equal to the number of parameters in the model, $d+1$. This means that one way to look at the VC dimension is that it measures the 'effective' number of parameters or the 'degrees of freedom' in the learning model. The more parameters the model has, the more diverse its hypothesis set $\mathcal{H}$ will be, resulting in a larger value for the growth function $m_{\mathcal{H}}(N)$ (remembering that the growth function formalizes the effective number of hypotheses).

17 Ao final, o *bound* inicial $2Me^{-2\epsilon^2 N}$ acaba sendo substituído por um *bound* que depende do *VC dimension*. Qual é esse novo *bound*? E o que podemos dizer sobre esse novo *bound*, em termos do *VC dimension*?

For the generalization bound in Equation 8 to be true when replacing M with the growth function $m_{\mathcal{H}}(N)$, a few modifications need to be made for the inequality to hold true. The VC generalization bound $\forall$ binary target functions f, $\forall$ $\mathcal{H}$, $\forall$ learning algorithm $\mathcal{A}$ and $\forall$ input probability distribution P is given by the following theorem, where δ is a user defined tolerance:

Theorem 2 *(VC Generalization Bound)*

$$E_{out}(g) \leq E_{in}(g) + \sqrt{\frac{8}{N} \ln \frac{4m_{\mathcal{H}}(2N)}{\delta}}, \quad \forall\, \delta > 0, \tag{9}$$

with probability $\geq 1 - \delta$.

Through Theorem 2 we can say that, as long as the VC dimension is finite, the error bar converges to zero, but at a slower rate than in Equation 8 since $m_{\mathcal{H}}(2N)$ is a polinomial of order d_{VC} in N (see Equation 7). All this means that with a large N of data, all hypothesis in $\mathcal{H}$ with finite d_{VC} will generalize well from E_{in} to E_{out}.

This makes the VC generalization bound the most important mathematical result in the theory of learning since it establishes the feasibility of learning with infinite hypothesis sets.

18 De acordo com a desigualdade *VC*, como podemos calcular o número de amostras necessárias para se garantir uma certa precisão, com probabilidade 0.9, por exemplo?

The sample complexity denotes how many training examples N are needed to achience a certain generalization performance, which is specified by two parameters: ϵ and δ. The tolerance error ϵ determines the allowed generalization error and the confidence paramter δ determines how often the error tolerance ϵ is violated. Observing how fast N grows as ϵ and δ become smaller shows us how much data is needed to get good generalization.

We can estimate the sample complexity using the VC bound:

$$N \geq \frac{8}{\epsilon^2} \ln \left(\frac{4m_{\mathcal{H}}(2N)}{\delta} \right) \tag{10}$$

Replacing the growth function by its polynomial upper bound (based on the VC dimension) in Equation 10 we get:

$$N \geq \frac{8}{\epsilon^2} \ln\left(\frac{4((2N)^{d_{\text{VC}}}+1)}{\delta}\right). \tag{11}$$

Both bounds 10 and 11 are implicit, because N appears on both sides of the inequality. For this reason, iterative methods are used to obtain numerical values for N.

With a probability of 0.9, we have the confidence parameter $\delta = 0.1$. We would also need to consider the VC dimension of the learning model and a certain tolerance error ϵ. Using Equation 11, we calculate the RHS with these values, starting with an initial guess for N. Using the value obtained on the RHS as the new N, we recalculate the RHS iteratively until it converges to the final value.

Let's try out this iterative algorithm in Python. In Listing 1 we calculate N using the values seen in the script and plot the values of N at each iteration, seen in Figure 2. N converges to approximately 40,000.

```
import numpy as np
import matplotlib.pyplot as plt


delta = 0.1 #confidence parameter considering P = 0.9
epsilon = 0.1 #tolerance error
d_vc = 4 #VC dimension
iterations = 10 #number of iterations
Ns = np.empty(iterations+1) #Array to recieve values of N
    throughout all iterations
Ns[0] = 1000 #this will be our initial guess for N

#Iteratively calculate N
for i in range(1,iterations+1):
    Ns[i] = (8/(epsilon**2))*np.log((4*(((2*Ns[i-1])**d_vc)+1))/
    delta)

print("Final N:", Ns[10])
x = np.arange(11)
plt.scatter(x, Ns)
plt.xlabel("Iteration")
plt.ylabel("N (sample complexity)")
plt.show()
```

Listing 1: Python script for Exercise 18

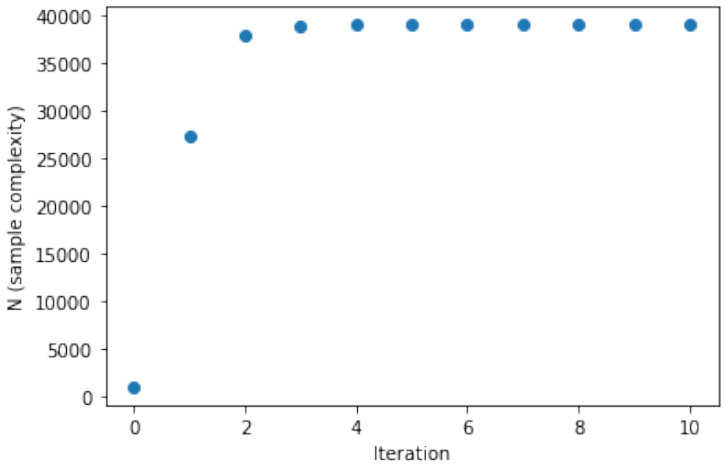

Figure 2: Plot of iteration number versus the calculated RHS. See how N converges to approximately 40,000.

19 De acordo com a desigualdade *VC*, dado uma certa quantidade N de exemplos de treinamento, o que podemos dizer sobre $|E_{in}(h) - E_{out}(h)|$?

Given a certain value of N training examples, we can say that the probability of $|E_{in}(h) - E_{out}(h)| > \epsilon$ will be:

$$P[|E_{in} - E_{out}| > \epsilon] \leq 4m_{\mathcal{H}}(2N)e^{-\frac{1}{8}\epsilon^2 N} \qquad \text{for any } \epsilon > 0, \tag{12}$$

giving us the VC inequality, analogous to the Hoeffding Inequality for M terms (Equation 2). The more N training examples we use, the better we can approximate E_{in} to E_{out} by means of a respectable (small) bound.

20 Por que garantir apenas $|E_{in}(h) - E_{out}(h)| < \epsilon$ não é suficiente?

To show that is not enough to garantee that $|E_{in}(h) - E_{out}(h)| < \epsilon$ let's take the VC generalization bound (Equation 14) and replace the growth function with by its polynomial upper bound (based on the VC dimension):

$$E_{out}(g) \leq E_{in}(g) + \sqrt{\frac{8}{N} \ln \left(\frac{4((2N)^{d_{\text{VC}}} + 1)}{\delta} \right)}, \quad \forall\, \delta > 0, \tag{13}$$

If $N = 100$, $\delta = 0.1$ and $d_{\text{VC}} = 1$ we have:

$$E_{out}(g) \leq E_{in}(g) + \sqrt{\frac{8}{100} \ln\left(\frac{4(201)}{0.1}\right)} \approx E_{in}(g) + 0.848, \tag{14}$$

with a confidence of $\geq 90\%$. In a case where $E_{in} = 0$, E_{out} will still be close to 1, which means the bound ϵ is a poor one. This means that even if $|E_{in}(h) - E_{out}(h)| < \epsilon$, ϵ may still be a loose bound.

21 Na sua opinião, o *VC bound* poderia ser melhorado? O que poderia ser explorado para se obter um bound teórico menor do que o *VC bound*?

Since the VC bound is generalized to work for all learning algorithms, it it more likely to be "pessimistic", considering worst case scenarios with looser bounds. In theory, if a bound were to be deduced for a specific learning algorithm, it could be optimized for that specific algorithm. For example, we bound the growth function instead of actually computing it for the hypothesis set. Despite it not being practical, it would produce a better bound for a learning algorithm of interest.

22 As Lectures 02, 05, 06 e 07 cobrem o *VC theory* (embora em contexto restrito à classificação binária). Escreva aqui, com suas palavras e de forma sucinta, como você explicaria o *VC theory* para um colega interessado em aspectos teóricos de aprendizado de máquina.

The whole purpose of the VC theory is to approximate the in-sample error to the out-of-sample error and prove that learning is indeed feasible by means of generalizing from the training examples to data outside of our sample. The VC dimension, by means of the VC bound, allows us to bound the in-sample error to the out-of-sample error. It replaces the hypothesis set size in the bound, because many interesting learning models have an infinite hypothesis set. We use the concept of dichotomies generated by hypothesis set to define the growth function that formalizes the effective number of hypothesis to replace the set size (the growth function is finite whereas the hypothesis set if infinite in this case). We can then describe the growth function in terms of the VC dimension, which measures the largest value of N for which all dichotomies are generated. We get the VC generalization bound which proves the feasibility of learning when working with infinite hypothesis sets.