Chapter 8

Pattern recognition

Let us introduce the pattern recognition problem using computer science notation (its good for you to see that too, as you will probably be reading a bunch of cs papers in the future).

Suppose we have n training data points $T_n := ((X_i, Y_i))_{i=1}^n$ and are interested in a classification rule $h(X)$ that uses T_n to *predict*, i.e., assign labels to previously unseen data X.

Thus, we want our classification rule h, which is typically an algorithm, to perform well on previously unseen data by learning from the training data. This is known as *generalization.*

The space $\mathcal{X}$ where X_i belongs to is called the *instance space* or *feature space* and the space $\mathcal{Y}$ where Y_i belongs to is called the *label space.*

Typically, $\mathcal{X}$ is a subset of $\mathbb{R}^d$ and $\mathcal{Y}$ is binary label space either as $\{0,1\}$ or $\{-1,1\}$. For example, $\mathcal{X}$ can be $\{0,1\}^d$ to indicate the presence or absence of something in the instance space, say a specific set of words in an email if the task is to classify emails with labels 0 and 1 for non-spam or spam.

Remark 8.1. *To connect back to our previous terminology, we see that the data space $\mathbb{X} = (\mathcal{X}, \mathcal{Y})$ (we will later write $\mathbb{X} \times \mathbb{Y}$ to avoid X being used for both feature and label) is split into the featue space and the label space. The random variable we are observing is a pair (X, Y) and a collection of n samples is the training dataset.*

8.1 Linear Classifiers

Let us say that we are trying to device a classification rule based on instance space $\mathcal{X} = \mathbb{R}^d$ and label space $\mathcal{Y} = \{-1, 1\}$.

One of the simplest such rule involves taking weighted sums of the x_i's until it exceeds a threshold to determine if it should be labelled by +1 or not. Such rules involve finding a hyperplane of dimension $d-1$ to separate

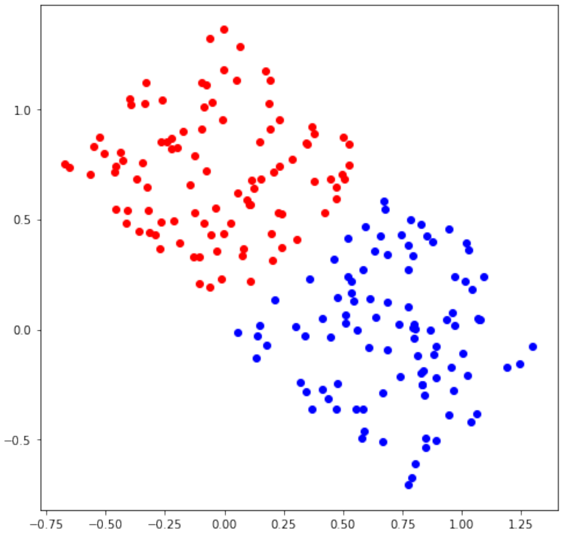

Figure 8.1: Linearly separable data with labels +1 or red and −1 or blue.

out the data with the same labels on each side of the hyperplane. Such a classification rule is called a *linear separator*.

Fig. 8.1 shows an example of data that is *linearly separable* and thus ideal for linear separators.

8.1.1 Linearly Separable Dataset

Consider the following linearly separable dataset, where we can draw a line (hyper-plane in $\mathbb{R}^2$) to separate the data points with different labels on either side of the line.

8.1.2 The perceptron algorithm

In the history of artificial intelligence and neural network research, linear classifiers of this type were called **perceptrons.** (Fisher's linear discrimination analysis (LDA, 1936) is also given by a linear classifier). In this section we present a training algorithm for a perceptron, invented in 1957 at the Cornell Aeronautical Laboratory by Frank Rosenblatt. The algorithm created a great deal of interest when it was first introduced. As we will see, it is guaranteed to converge if there exists a hyperplane that correctly classifies the training data.

The perceptron algorithm tries to find a linear separator, i.e. a hyperplane in $\mathbb{R}^d$ that separates the two classes. The task is thus to find w and t such that for the training data S, the data consists of pairs (x_i, l_i) the x_i represents our features and the l_i our labels or target.

$$\begin{aligned} w \cdot x_i > t & \quad \text{for each } x_i \text{ labeled } +1 \\ w \cdot x_i < t & \quad \text{for each } x_i \text{ labeled } -1 \end{aligned}$$

Adding a new coordinate to our space allows us to consider $\hat{x}_i = (x_i, 1)$ and $\hat{w} = (w, t)$, this allows us to rewrite the inequalities above as

$$(\hat{w} \cdot \hat{x}_i) l_i > 0.$$

The algorithm

1. $w = 0$

2. while there exists x_i with $x_i y_i \cdot w \leq 0$, update $w := w + x_i y_i$

[12]:

```
@interact
def _(n_steps=(0,(0..63))):
        # X = (n_points,3)
        # W = (n_points,3)
        n_points = X.shape[0]
        W = np.array([0,0,0])
        P=points(zip(X1[:,0],X1[:,1]),color='blue')
        P+=points(zip(X2[:,0],X2[:,1]),color='red')

        k = 0
        max_iter=10000
        j = 0
        while ((k < n_steps) and (j < max_iter)):
                i = j % n_points
                j+=1
                if (X[i,:]@W * yall[i] <= 0):
                        W = W + X[i,:]*yall[i]
                        P+=points(X[i,:2],color='yellow')
                        k+=1
        print(W)
```

Theorem 8.2. *If there exists w^* such that $w^* \cdot x_i y_i \geq 1$ for all i. Then the perceptron algorithm finds a w satisfying $w \cdot x_i y_i \geq 0$ for all i in at most $r^2 |w^*|^2$ updates, where $r = \max_i |x_i|$.*

Proof. Lets say we are given a sequence of points $D_n = \{(x_1, y_1), \ldots, (x_n, y_n)\}$. Let w_i, for $i = 0, \ldots$, denote the weight at update i. That is, $w_0 = 0$ and for every update i, there is a corresponding index $I(i) \in \{1, \ldots, n\}$ such that

$$w_i \cdot x_{I(i)} y_{I(i)} \leq 0.$$

Since we then update the weights, we also have

$$w_{i+1} = w_i + x_{I(i)} y_{I(i)}.$$

Assuming the statement is true, you expect that $w_i \to w^*$ in some capacity, at least the direction should be close. Secondly, we also know that the weights most likely grow since we are always adding vectors to it. With this in mind, let us track two quantities and see how they react to an update: First let us compute

$$w_{i+1} \cdot w^* = (w_i + x_{I(i)} y_{I(i)}) \cdot w^* w_i \cdot w^* + w^* \cdot x_{I(i)} y_{I(i)} \geq w_i \cdot w^* + 1. \tag{8.1}$$

Then let us compute

$$\begin{aligned} |w_{i+1}|^2 &= w_{i+1} \cdot w_{i+1} = (w_i + x_{I(i)} y_{I(i)}) \cdot (w_i + x_{I(i)} y_{I(i)}) \\ &= |w_i|^2 + 2 w_i \cdot x_{I(i)} y_{I(i)} + |x_{I(i)}|^2 \\ &\leq |w_i|^2 + r^2. \end{aligned} \tag{8.2}$$

Since $w_0 = 0$ we get from iterating (8.1) and (8.2) for $i = 0, \ldots, m$ that

$$\begin{aligned} w_m \cdot w^* &\geq m \\ |w_m|^2 &\leq m r^2. \end{aligned}$$

If we use Cauchy-Schwartz for the dot-product we get from the above that

$$m \leq w_m \cdot w^* \leq |w_m||w^*| \leq |w^*| \sqrt{m} r.$$

Now, dividing the left hand side and right hand side with $\sqrt{m}$ and skipping the middle we get

$$\sqrt{m} \leq |w^*| r,$$

which when squared gives the result. □

So this theorem guarantees that if the two classes can be separated then the preceptron will also find a separator in finite time. This is interesting since finding the plane that minimizes the error is NP-hard, so this tells us that if there is separation the problem is “easy”.

What about non-linearly separable data. Let $B_r = \{x \in \mathbb{R}^2 : |x| < r\}$, then for instance

$$X = (B_4 \setminus B_3) \cup B_1$$

and let $g^* = \mathbb{1}_{B_1}$. We cannot separate these sets using a linear classifier, see simulation in notebooks.

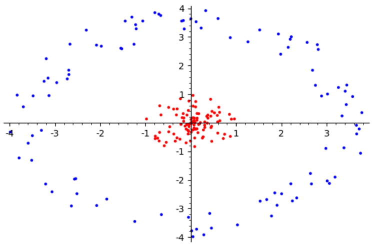

Figure 8.2: Linearly non-separable data in two dimensions.

8.2 Kernelization

What about non-linearly separable data. Take for instance the data formed by the following operations with points at different distances from the origin in two dimensions, as shown in Figure 8.2.

$$X = (B_4 \setminus B_3) \cup B_1$$

and let $c^* = B_1$. We cannot separate these sets using a linear classifier

[25]:
```
A = np.random.normal(size=(100,2))
A_unit = A/(np.linalg.norm(A,axis=1).reshape(-1,1))
radial_A = 3+np.random.uniform(size=(100,1))
P=points(A_unit*radial_A,color=’blue’)

B = np.random.normal(size=(100,2))
B_unit = B/(np.linalg.norm(B,axis=1).reshape(-1,1))
radial_B = np.random.uniform(size=(100,1))
P+=points(B_unit*radial_B,color=’red’)
P.show()
```

[25]: we can however separate the following mapping of X. Namely in $\mathbb{R}^2$ we can do

$$\phi(x) = (x_1, x_2, x_1^2 + x_2^2) \in \mathbb{R}^3$$

This is clearly linearly separable as we can see from the following 3d plot shown in Figure 8.3.

[27]:
```
A_2d = A_unit*radial_A
A_3d = np.concatenate([A_2d,np.linalg.norm(A_2d,axis=1).reshape(-1,1)^2],axis=1)
B_2d = B_unit*radial_B
B_3d = np.concatenate([B_2d,np.linalg.norm(B_2d,axis=1).reshape(-1,1)^2],axis=1)

P=points(A_3d,size=20,color=’blue’)
P+=points(B_3d,size=20,color=’red’)
P.show()
```

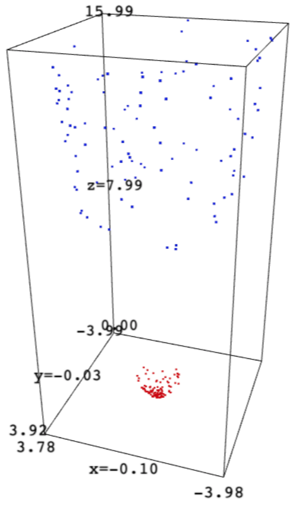

Figure 8.3: Linearly separable in three dimensions after $(x_1, x_2) \mapsto (x_1, x_2, x_1^2 + x_2^2)$.

> Remember the extra dimension that we always add to simplify notation. Therefore the full ϕ in the above examples is $\hat{\phi}(x) = (x_1, x_2, x_1^2 + x_2^2, 1)$.

So if we transform the $x \to \phi(x)$ for some good transformation ϕ then our perceptron will try to solve

$$w \cdot \phi(x_i) l_i > 0$$

furthermore, remember how we constructed w using the perceptron algorithms, i.e. using additions of $x_i l_i$, which transforms into $\phi(x_i) l_i$, and we start with $w = 0$, this gives that the weight has the form

$$w = \sum_{i=1}^{n} c_i \phi(x_i)$$

for numbers c_i. The perceptron algorithm becomes just addition and subtraction of certain c_i's by 1.

Furthermore

$$w \cdot \phi(x_i) = \sum_{j=1}^{n} c_j \phi(x_j) \cdot \phi(x_i) = \sum_{j=1}^{n} c_j k_{ij}$$

where $k_{ij} = \phi(x_i) \cdot \phi(x_j)$.

Is it easy to find such a mapping ϕ? No, it is actually quite difficult. Furthermore, if the mapping ϕ is high dimensional we might need to do alot of computation, which is not so efficient. What if we had a function $k(x, y)$ that could be written as

$$k(x, y) = \phi(x) \cdot \phi(y)$$

for some ϕ and k is easier to compute, then our life would be simpler. Also, what if we are given a function $k(x, y)$ and we would like to know if it is a "kernel function".

Lemma 8.3. *Given a sequence of points $\{x_i\}_{i=1}^n$, $x_i \in \mathbb{R}^d$, and given an $n \times n$ matrix K, which is symmetric and positive semidefinite. Then, there is a mapping $\phi : \mathbb{R}^d \to \mathbb{R}^m$ (for some m) such that $K_{ij} = \phi(x_i) \cdot \phi(x_j)$.*

Exercise 8.4. *Prove the above Lemma using the following outline for it:*

1. *$K = Q\Lambda Q^T$ (eigendecomposition)*

2. *K is positive definite, all eigenvalues ≥ 0, so we can define $B = Q\Lambda^{1/2}$.*

3. *$K = BB^T$*

4. *define $\phi(x_i) = B_{i\cdot}$, i.e. the i:th row of B, then $K_{ij} = \phi(x_i) \cdot \phi(x_j)$.*

5. *What is the size of m?*

We now have a way to identify whenever a matrix K is a kernel matrix. There are some standard choices of kernel functions one could try, that produces positive semi-definite matrices whenever all points x_i are distinct.

Definition 8.5. *We call a function $k(x, y) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ a kernel function if there is a mapping $\phi : \mathbb{R}^d \to \mathbb{R}^m$ (for some m) such that $k(x, y) = \phi(x) \cdot \phi(y)$.*

Theorem 8.6. *Suppose $k_1, k_2 : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ are kernel functions. Then*

1. *For any constant $c \geq 0$, ck_1 is a kernel function.*

2. *For any scalar function f, $k(x, y) = f(x)f(y)k_1(x, y)$ is a kernel function.*

3. *$k_1 + k_2$ is a kernel function. 4. k_1k_2 is a kernel function.*

Exercise 8.7. *Prove Theorem 8.6.*

Corollary 8.8. *The following functions are kernel function*

- *$k(x, y) = (\gamma x \cdot y + r)^k$, ($k \in \mathbb{N}$) polynomial*
- *$k(x, y) = x \cdot y$, linear*

Exercise 8.9. *Prove the above corollary by multiple applications of Theorem 8.6.*

8.2.1 Other types of Kernels

The above concept of a kernel function is a simplification, we can also have the following kernels for which ϕ maps to "infinite dimensions":

1. $k(x, y) = e^{-\gamma|x-y|}$, called Radial Basis Function
2. $k(x, y) = \tanh(\gamma x \cdot y + r)$, sigmoidal

8.3 Theoretical guarantees

Recall that in the pattern recognition model (Section 4.1.3) we assume that the supervisors conditional distribution $F(y|x)$ is discrete, and can take k different values, $y = 0, \ldots, k-1$.

Recall the $0-1$ loss function for $z = (x, y)$

$$L(z, u) = \begin{cases} 0 & \text{if } y = u \\ 1 & \text{if } y \neq u \end{cases}$$

that is, the loss is 1 if u is the wrong value and 0 if it is correct. The pattern recognition problem is the problem of minimizing the functional

$$R(g) = \int L(y, g(x)) dF(x, y) = \mathbb{E}\left[L(Y, g(X))\right]$$

where $(X, Y) \sim F(x, y)$.

Also recall that,

$$\mathbb{E}\left[L(Y, g_\lambda(X))\right] = \mathbb{P}(\{Y \neq g_\lambda(X)\}).$$

As we have alluded to in Section 5.3 is that we want to minimize the empirical version of the risk and hope that we can get reasonable concentration estimates, as in Theorem 5.27.

Definition 8.10. *Given a probability triple $(\Omega, \mathcal{F}, \mathbb{P})$, and assume that $Z = ((X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n)) \overset{\text{IID}}{\sim} F(x, y)$ is a sequence of $\mathbb{R}^{m+1}$ valued random variables taking values in the data space $\mathbb{X} \times \mathbb{Y}$. We define the empirical risk for a function $g : \mathbb{X} \to \mathbb{Y}$ as*

$$\hat{R}_n(g) = \hat{R}_n(Z; g) = \frac{1}{n} \sum_{i=1}^{n} L(Y_i, g(X_i)).$$

Note that the empirical risk is a statistic evaluated on Z.

We would like to minimize the risk, but we only have access to $Z = ((X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n)) \overset{\text{IID}}{\sim} F(x, y)$, so we can in practice only try to minimize the empirical risk. So given a model space $\mathcal{M}$ we consider

$$\hat{g}_n^* := \hat{g}_n^*(Z) := \operatorname*{argmin}_{g \in \mathcal{M}} \hat{R}_n(g).$$

The first realization here is now that $\hat{g}_n^*$ is a random variable that depends on Z, which means that it is not immediate that

$$\hat{R}_n(Z; g_n^*(Z))$$

is unbiased, in fact, since we are minimizing the risk it is quite possibly downward biased. However, even if it is downward biased one could hope that in some cases the bias is small when n is large.

8.3.1 Guarantees with a held out testing set

The problem with using the value of the empirical risk above is that we are evaluating on the "training-data", if we however have access to a testing dataset, then we can give some guarantees in the pattern recognition problem.

Consider a data set $T_{n+m} := \{(X_1, Y_1), \ldots, (X_{n+m}, Y_{n+m})\}$ sampled i.i.d from $F_{X,Y}$. We consider $\{(X_1, Y_1), \ldots, (X_n, Y_n)\}$ (which we dub the training data) and $\{(X_{n+1}, Y_{n+1}), \ldots, (X_{n+m}, Y_{n+m})\}$ (which we dub the testing data). Define $\hat{\phi}$ the **empirical risk minimizer** on the **training dataset**, namely

$$\hat{R}_n(\hat{\phi}) = \min_{\phi \in \mathcal{M}} \hat{R}_n(\phi)$$

then since the **testing dataset** is independent of the training dataset and hence $\hat{\phi}$ is independent of the testing data, we deduce using Theorem 3.6 that if $\hat{R}_m(\phi)$ denotes the empirical risk over the testing dataset we have (Provided R is nice enough)

$$\mathbb{P}(|\hat{R}_m(\hat{\phi}) - R(\hat{\phi})| > \epsilon \mid T_n) < 2e^{-C\epsilon^2 n}. \tag{8.3}$$

Now again using the tower property we can get

$$\mathbb{P}(|\hat{R}_m(\hat{\phi}) - R(\hat{\phi})| > \epsilon) < 2e^{-C\epsilon^2 n}. \tag{8.4}$$

This is a procedure which always works when the loss is bounded, like $0-1$ loss. The risk on the testing data-set can even be exchanged for another loss, i.e. different than the training loss.

Exercise 8.11. *In the above we are mentioning that R needs to be nice enough, why is that? Does $0-1$ loss work? Why?*

Furthermore, we used the tower property to derive (8.4) from (8.3), how does this work?

Remark 8.12. *In many machine learning text-books that are practically oriented, you will see the recommendation that the training/testing split should be 70/30. In the pattern recognition problem this doesn't make much sense,* *it is better to determine with how high probability you want the bound to hold and use that to choose the number of samples to reach a tight enough interval.*

Remark 8.13. *Note that the testing estimate above is only valid if done once, as the probability is over the testing set. Since we only have one, it can only be used once.*

It should be noted that during other courses you will encounter what seems to be a violation to the above, i.e. with the introduction of a test set and a validation set. In this setup the test set is used several times to select the best out of a set of different models, the best model is then chosen and the performance is evaluated on the **validation data**. In this setup one is not too concerned with the fact that the test set is used several times, as it is only used to select the model. The final performance evaluation, done once will satisfy the concentration bound (8.3).

Exercise 8.14. *In kaggle competitions and other online data-science competitions, several teams try to produce a model on a training data-set. When the team submits their proposed model it is validated on a hidden test-set (same for all teams). The teams are then ranked according to the score on the hidden test set. Does this mean that the best team had the best model? Do you think it would look the same if we repeated this with other hidden test-sets?*

8.3.2 Other test metrics

In the practical usage of the pattern recognition problem, one often sees the use of the test-metrics Precision and Recall. For class 1 they are defined as the following conditional probabilities

$$\begin{aligned}\text{Precision:} \quad & \mathbb{P}(Y = 1 \mid g(X) = 1) \\ \text{Recall:} \quad & \mathbb{P}(g(X) = 1 \mid Y = 1).\end{aligned}$$

Recall in particular is often used in medical testing and are then called sensitivity.

Exercise 8.15. *Lets say we have trained an ML model and gotten g as output, and lets say we want to use the test-set to estimate precision and recall. Lets say you wish to give a guarantee using for instance "concentration of measure" in the following scenarios*

1. *The function g is always* 1
2. *The function g is always* 0

3. *$P(Y = 1)$ is close to 0*

4. *$P(Y = 1)$ is close to 1.*

Which of these problems are easier and which are harder? What if we switch to estimating precision and recall for class 0?

8.4 Empirical Risk Minimization for Linear Classifiers

If we restrict the **"complexity"** of the decision functions, complexity will be defined. Then, we can give some guarantees without having to resort to a test-set that can only be used once. A form of a-priori estimate. These estimates are very very strong, but a bit restrictive.

Consider the data space $\mathbb{X} \times \mathbb{Y} = \mathbb{R}^m \times \{0, 1\}$, then a linear decision function is given as

$$\phi_a(x) = \begin{cases} 1 & \text{if } (a')^T x + a_0 > 0 \\ 0 & \text{otherwise} \end{cases}$$

where $a = (a', a_0) \in \mathbb{R}^{m+1}$. The model space corresponding to linear decision rules is

$$\mathcal{M} = \left\{\phi_a : a \in \mathbb{R}^{m+1}\right\} = \{\text{ all linear decision rules }\}.$$

We can index $\mathcal{M}$ using $a \in \mathbb{R}^{m+1}$.

8.4.1 A classifier with finitely many hyperplanes (without testing)

Instead of choosing $\mathcal{M}$ to be indexed by $a \in \mathbb{R}^{m+1}$ with uncountably infinitely many possibilities for ϕ, we will limit ourself to minimizing the empirical risk over finitely many linear decision rules – exactly $2\binom{n}{m}$ that are defined by each of the m choices from the n training points in $D_n = \{X_1, \dots, X_n\}$.

Consider choosing m arbitrary points $(X_{i_1}, X_{i_2}, \dots, X_{i_m})$ from the training data $\{X_1, X_2, \dots, X_n\}$, and let $(a')^T x + a_0 = 0$ be the hyper-plane containing these m points, i.e., $x = (x_{i_1}, x_{i_2}, \dots, x_{i_m})$. If we assume that X_i are continuous random variables, the m points are in *general position*[1] with probability 1 and this hyperplane is unique determining two decision rules:

$$\phi_+(x) = \begin{cases} 1 & \text{if } a^T x + a_0 > 0 \\ 0 & \text{otherwise} \end{cases}$$

[1] https://en.wikipedia.org/wiki/General_position and Exercise 8.17

and

$$\phi_-(x) = \begin{cases} 1 & \text{if } a^T x + a_0 < 0 \\ 0 & \text{otherwise} \end{cases}$$

with empirical risks or misclassification training errors $\hat{R}_n(\phi_+)$ and $\hat{R}_n(\phi_-)$. Thus to each m-tuple of data points we can associate two decision rules to yield a total of $2\binom{n}{m}$ such decision rules. Let us denote these decision rules by $\mathcal{M}_n = \{\phi_1, \dots, \phi_{2\binom{n}{m}}\}$.

Now let $\hat{\phi}$ be a decision rule that minimizes $\hat{R}_n(\phi_i)$ over all $i \in \{1, 2, \dots, 2\binom{n}{m}\}$, i.e.

$$\hat{R}_n(\hat{\phi}) = \min_{\phi \in \mathcal{M}_n} \hat{R}_n(\phi). \tag{8.5}$$

Theorem 8.16. *Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability triple and consider (X_1, Y_1), …, (X_n, Y_n) be an i.i.d. sequence of random variables, X_i being continuous and taking values in $\mathbb{R}^m$ and $Y_i \in \{0, 1\}$ is discrete. If $\hat{\phi}$ is defined as in (8.5), then, if $2m/n \le \epsilon \le 1$, we have*

$$\mathbb{P}\{R(\hat{\phi}) > \inf_{\mathcal{M}} R(\phi) + \epsilon\} \le e^{2m\epsilon} \left(2\binom{n}{m} + 1 \right) e^{-n\epsilon^2/2}.$$

Proof. Denote $\phi^* \in \mathcal{M}$ a decision rule that satisfies

$$R(\phi^*) = \inf_{\mathcal{M}} R(\phi).$$

Furthermore, note that if we take a $\phi \in \mathcal{M}$ then at most it can make m better predictions than $\hat{\phi}$, as it could be that the plane that defines ϕ does not touch any X_i,

$$\hat{R}_n(\hat{\phi}) \le \hat{R}_n(\phi) + \frac{m}{n}.$$

Now elementary considerations give together with the above,

$$\begin{aligned} I_0 := R(\hat{\phi}) - \inf R(\phi) &= R(\hat{\phi}) - \hat{R}_n(\hat{\phi}) + \hat{R}_n(\hat{\phi}) - \inf R(\phi) \\ &\le R(\hat{\phi}) - \hat{R}_n(\hat{\phi}) + \hat{R}_n(\phi^*) - R(\phi^*) + \frac{m}{n} \\ &\le \max_{1 \le i \le n} (R(\phi_i) - \hat{R}_n(\phi_i)) + \hat{R}_n(\phi^*) - R(\phi^*) + \frac{m}{n} \\ &=: \max_{1 \le i \le n} I_{1,i} + I_2 + \frac{m}{n} \end{aligned}$$

The reason we want to bound the difference this way is that we wish to apply concentration inequalities to both of these terms, i.e. we want to use the fact that if n is large, there is a high probability that the empirical risk is close to the true risk.

From the properties of probability (monotonicity and Boole's inequality) we get (denoting $N = 2\binom{m}{n}$)

$$\begin{aligned}\mathbb{P}(I_0 > \epsilon) &\leq \mathbb{P}(\max_{1\leq i\leq N} I_{1,i} + I_2 + \frac{m}{n} > \epsilon) \\ &\leq \mathbb{P}(\max_{1\leq i\leq N} I_{1,i} > \epsilon/2) + \mathbb{P}(I_2 > \epsilon/2 - \frac{m}{n}) \\ &\leq \sum_{i=1}^{N} \mathbb{P}(I_{1,i} > \epsilon/2) + \mathbb{P}(I_2 > \epsilon/2 - \frac{m}{n}) \qquad (8.6)\end{aligned}$$

Now, let us first bound $\mathbb{P}(I_2 > \epsilon/2 - m/n)$. This is easy, because $L(Y_i, \phi^*(X_i))$ is a sequence of independent bounded random variables, i.e. we can apply Hoeffdings inequality Theorem 3.6 to get (if $\epsilon/2 - m/n > 0$)

$$\mathbb{P}(I_2 > \epsilon/2 - m/n) = \mathbb{P}(\hat{R}_n(\phi^*) - R(\phi^*) > \epsilon/2 - \frac{m}{n}) \leq e^{-2n(\epsilon/2 - \frac{m}{n})^2}. \qquad (8.7)$$

To bound $\mathbb{P}(I_{1,i} > \epsilon/2)$ we will make use of the tower property Theorem 2.60 and note that

$$\mathbb{P}(I_{1,i} > \epsilon/2) = \mathbb{E}\left[\mathbb{P}(I_{1,i} > \epsilon/2 \mid X_{i_1}, \ldots, X_{i_m})\right] \qquad (8.8)$$

Let $K_i = \{i_1, \ldots, i_m\}$ be the set of indices of points used to construct ϕ_i, then

$$\begin{aligned}&\mathbb{P}(I_{1,i} > \epsilon/2 \mid \{X_k, k \in K_i\}) \\ &\quad= \mathbb{P}\left(R(\phi_i) - \frac{1}{n}\sum_{j=1}^{n} L(Y_j, \phi_i(X_j)) > \epsilon/2 \;\middle|\; \{X_k, k \in K_i\}\right) \\ &\quad\leq \mathbb{P}\left(R(\phi_i) - \frac{1}{n}\sum_{j=1, j\notin K_i}^{n} L(Y_j, \phi_i(X_j)) > \epsilon/2 \;\middle|\; \{X_k, k \in K_i\}\right).\end{aligned}$$

Now, from this point we can go different paths, but we will use the observation that all $L(Y_j, \phi_i(X_j))$ are Bernoulli$(R(\phi_i))$ for $j \notin K_i$, if we add $Z_1, \ldots, Z_m$ i.i.d. from Bernoulli$(R(\phi_i))$, also independent of $L(Y_j, \phi_i(X_j))$

for $j \notin K_i$, then we can apply Hoeffding to the following

$$\begin{aligned}
&\mathbb{P}\left(R(\phi_i) - \frac{1}{n}\sum_{j=1, j\notin K}^{n} L(Y_j, \phi_i(X_j)) > \epsilon/2 \,\middle|\, \{X_k, k \in K\}\right) \\
&\quad \le \mathbb{P}\left(R(\phi_i) - \frac{1}{n}\sum_{j=1, j\notin J}^{n} L(Y_j, \phi_i(X_j)) - \frac{1}{n}\sum_{j=1}^{m} Z_i > \epsilon/2 - \frac{m}{n} \,\middle|\, \{X_k, k \in K\}\right) \\
&\quad \le e^{-2n(\epsilon/2 - m/n)^2}.
\end{aligned}$$

Recalling (8.6)–(8.8) we obtain

$$\begin{aligned}
\mathbb{P}(I_0 > \epsilon) &\le 2\binom{n}{m} e^{-2n(\epsilon/2 - m/n)^2} + e^{-2n(\epsilon/2 - \frac{m}{n})^2} \\
&= \left(2\binom{n}{m} + 1\right) e^{-2n(\epsilon/2 - m/n)^2}.
\end{aligned}$$

The final step is to realize that

$$2n(\epsilon/2 - m/n)^2 > \frac{n\epsilon^2}{2} - 2m\epsilon.$$

□

Exercise 8.17. *Consider $X_1, \dots, X_{m+1}$ be i.i.d. $\mathbb{R}^m$ valued continuous random variables. Use the tower property to prove that the probability of these points to lie in the same hyperplane is zero. Using this, show the assumption of general position needed to construct the decision rules we worked with above.*

8.5 Preliminaries for VC theory

Definition 8.18 (empirical measure)**.**

$$\mu_n(A) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}_{X_i \in A}$$

We can deduce also that

Lemma 8.19.

$$R(\phi_n^*) - \inf_{\phi\in\mathcal{C}} R(\phi) \le 2 \sup_{\phi\in\mathcal{C}} |\hat{R}_n(\phi) - R(\phi)|$$

8.6 VC theory

We have just studied a specific class of decision rules that where constructed using the observations, and we see that the rule selected by (8.5) is indeed very good. It performs very closely to the best possible!! However our method of proof relies on the way that the rule is constructed and in particular does not allow us to minimize the empirical risk over $\mathcal{M}$ instead of $\mathcal{M}_n$. To cope with this, we need to develop some theory that stems from the works of Vapnik and Chervnonenkis, [SLT]. For this we will transition from viewing the decision rules as functions to viewing them as sets, as a decision rule from $\mathcal{M}$ splits $\mathbb{R}^m \times \{0,1\}$ into two pieces, one which gives zero loss and one part which gives loss 1. Define,

$$\mathcal{A} := \{\{(x,y) \in \mathbb{R}^m \times \{0,1\} : L(y,\phi(x)) = 1\} : \phi \in \mathcal{M}\}, \tag{8.9}$$

and denote by $\nu = dF(x,y)$ and ν_n the empirical measure with respect to the data set $D_n = \{(X_1,Y_1),\dots,(X_n,Y_n)\}$. The above definition allows us to rephrase

$$\mathbb{P}(\sup_{\mathcal{M}} |R_n(\phi) - R(\phi)| > \epsilon) = \mathbb{P}(\sup_{A\in\mathcal{A}} |\nu_n(A) - \nu(A)| > \epsilon). \tag{8.10}$$

Exercise 8.20. *Derive (8.10). Think about the following, given a decision function $\phi \in \mathcal{M}$, then for this function ϕ there is a corresponding set $A \in \mathcal{A}$ as in (8.9). The measure $\nu(A)$ is simply*

$$\nu(A) = \mathbb{P}((X,Y) \in A) = \mathbb{P}(L(Y,\phi(X)) = 1) = \mathbb{P}(Y \neq \phi(X))$$

and the empirical measure is

$$\nu_n(A) = \frac{1}{n}\sum_{i=1}^n \mathbb{1}_{(X_i,Y_i)\in A} = \frac{1}{n}\sum_{i=1}^n \mathbb{1}_{L(Y_i,\phi(X_i))=1} = \frac{1}{n}\sum_{i=1}^n L(Y_i,\phi(X_i)).$$

This puts our problem in the framework of uniform convergence of empirical measures **UCEM**. The uniform is because we have the supremum inside the probability. Recall, we have already seen an example of this!! If we consider the sets $A = (-\infty, a)$ then we are in the setting of Theorem 5.27.

If $|\mathcal{A}| < \infty$, then we could simply use Hoeffdings inequality, Theorem 3.6, to obtain

$$P(\sup_{A\in\mathcal{A}} |\nu_n(A) - \nu(A)| > \epsilon) \leq 2|\mathcal{A}|e^{-2n\epsilon^2}.$$

Exercise 8.21. *Use the Union bound and Theorem 3.6 to prove the above inequality.*

However, even in the simple case of linear decision functions, $\mathcal{M}$, we have an uncountably infinite set.

Lemma 8.22. *Consider a sequence of i.i.d. random variables $Z_1, \dots, Z_{2n} \sim \nu$, split this into two datasets $D_n = \{Z_1, \dots, Z_n\}$ and $D'_n = \{Z_n{+}1, \dots, Z_2 n\}$. Let $n\epsilon^2 \geq 2$, then if we denote ν_n, ν'_n the empirical measure over D_n and D'_n respectively, we have*

$$\mathbb{P}\left[\sup_{A \in \mathcal{A}} |\nu(A) - \nu_n(A)| > \epsilon\right] \leq 2\,\mathbb{P}\left[\sup_{A \in \mathcal{A}} \left|\nu_n(A) - \nu'_n(A)\right| > \frac{\epsilon}{2}\right]$$

Proof. Let $A^* \in \mathcal{A}$ be a set for which $|\nu_n(A^*) - \nu(A^*)| > \epsilon$, if such a set exists, otherwise we can let A^* be a fixed set in $\mathcal{A}$. **NOTE:** A^* is a random set that depends on D_n. Now

$$\begin{aligned}
&\mathbb{P}(\sup_{A \in \mathcal{A}} \left|\nu_n(A) - \nu'_n(A)\right| > \epsilon/2) \\
&\quad \geq \mathbb{P}(\left|\nu_n(A^*) - \nu'_n(A^*)\right| > \epsilon/2) \\
&\quad \geq \mathbb{P}(|\nu(A^*) - \nu_n(A^*)| > \epsilon, \left|\nu(A^*) - \nu'_n(A^*)\right| < \epsilon/2) \\
&\quad = \mathbb{E}\left[\mathbb{1}_{|\nu(A^*) - \nu_n(A^*)| > \epsilon}\,\mathbb{P}(\left|\nu(A^*) - \nu'_n(A^*)\right| < \epsilon/2 \;\middle|\; D_n)\right]
\end{aligned}$$

Now, conditioned on D_n, the set A^* is fixed and $\nu'_n(A^*)$ is the mean of n independent Bernoulli($\nu(A^*)$) r.v.s., hence using Chebyshev's inequality (Proposition 3.2)

$$\mathbb{P}(\left|\nu(A^*) - \nu'_n(A^*)\right| < \epsilon/2 \;\middle|\; D_n) \geq 1 - \frac{\frac{1}{n}\nu(A^*)(1 - \nu(A^*))}{(\epsilon/2)^2} \geq 1/2,$$

in the last step we used the assumption $n\epsilon^2 \geq 2$. Putting it all together we now get

$$\begin{aligned}
\mathbb{P}(\sup_{A \in \mathcal{A}} \left|\nu_n(A) - \nu'_n(A)\right| > \epsilon/2) &\geq \frac{1}{2}\,\mathbb{E}\left[\mathbb{1}_{|\nu(A^*) - \nu_n(A^*)| > \epsilon}\right] \\
&= \frac{1}{2}\,\mathbb{P}(|\nu(A^*) - \nu_n(A^*)| > \epsilon) \\
&= \frac{1}{2}\,\mathbb{P}(\sup_{A \in \mathcal{A}} |\nu(A) - \nu_n(A)| > \epsilon)
\end{aligned}$$

where in the last step we used the definition of A^*. □

Now, this lemma gives us precisely what we want, namely to reduce the size of $\mathcal{A}$ from infinite to finite. This we do as follows

- Given a dataset $D_k = \{Z_1, \dots, Z_k\}$ we say that $A, B \in \mathcal{A}$ are equivalent if $A \cap D_k = B \cap D_k$.
- This equivalence relation defines equivalence classes on $\mathcal{A}$ given D_k, let us denote this set $\mathcal{A}_{D_k}$.

- It is clear that
  1. $\mathcal{A}_{D_k}$ is finite,
  2. it satisfies $|\mathcal{A}_{D_k}| \leq |2^{D_k}|$,
  3. non-decreasing with k. (In most interesting cases it grows with k).

Example 8.23. *Consider $\mathbb{R}^2$ and consider the set $\mathcal{M}$ linear decision rules and construct the corresponding $\mathcal{A}$. Let us now take say two points and labels $(x_0, y_0), (x_1, y_1) \in \mathbb{R}^2 \times \{0, 1\}$. Given two different linear functions $\phi_1, \phi_2 \in \mathcal{M}$ construct $A, B \in \mathcal{A}$, as follows*

$$A = \{(x, y) \in \mathbb{R}^2 \times \{0, 1\}, L(y, \phi_1(x)) = 1\}$$
$$B = \{(x, y) \in \mathbb{R}^2 \times \{0, 1\}, L(y, \phi_2(x)) = 1\}$$

we would then say that A, B are equivalent if $A \cap \{(x_0, y_0), (x_1, y_1)\} = B \cap \{(x_0, y_0), (x_1, y_1)\}$. What does this mean? It means that $L(y_0, \phi_1(x_0)) = L(y_0, \phi_2(x_0))$ and $L(y_1, \phi_1(x_1)) = L(y_1, \phi_2(x_1))$ which is the same as saying that $\phi_1(x_0) = \phi_2(x_0)$ and $\phi_1(x_1) = \phi_2(x_1)$. In short, the two decision functions ϕ_1, ϕ_2 assigns the same values to x_0, x_1 and are thus undistinguishable on these points.

Again, let us repeat. The concept of grouping together several decision functions is to say that on our dataset each labeling has with it an equivalence class of decision functions which produce said labeling.

Definition 8.24. *The largest size of $\mathcal{A}_{D_n}$ for a given n is called the shattering number for $\mathcal{A}$ given n*

$$s(\mathcal{A}, n) = \sup_{x_1, \ldots, x_n} |\mathcal{A}_{\{x_1, \ldots, x_n\}}|$$

Example 8.25. *Consider the decision function as being an inequality i.e. $\phi(x) = \mathbb{1}_{x > x_0}$. Then the corresponding sets $\mathcal{A}$ is as follows*

$$\begin{aligned} &\{(x, y), x \in \mathbb{R}, y \in \{0, 1\} : \mathbb{1}_{x > x_0} \neq y\} \\ = &\{(x, y) : x \in (0, x_0], y = 1\} \cup \{(x, y) : x \in (x_0, \infty), y = 0\}. \end{aligned}$$

Consider now two points $(x_1, 1), (x_2, 0)$ with $x_1 < x_2$, then we are counting the number of sets of the form $\{(x_1, 1), (x_2, 0)\} \cap A$. For any decision function there is a threshold x_0, either $x_0 < x_1, x_2$ and we get the set

$$\{(x_2, 0)\}$$

if $x_1 < x_0 < x_2$ then we get

$$\{(x_1, 1), (x_2, 0)\}$$

and if $x_1 < x_2 < x_0$

$$\{(x_1, 1)\}$$

Thus all in all we have three sets. Note that we did not actually create the empty set.

This is all equivalent to saying that the number of distinct labelings that the decision function can produce for two points is 3.

Definition 8.26. *For a set of points $G_n = \{z_1, \dots, z_n\}$ we say that $\mathcal{A}$ shatters G_n if*

$$|\mathcal{A}_{\{z_1,\dots,z_n\}}| = 2^n$$

What does the above actually mean in terms of the decision function? It means that the decision function can produce all possible labelings of the corresponding set $\{x_1, \dots, x_n\}$, with $z_i = (x_i, y_i)$.

Lemma 8.27. *Let $\sigma_1, \dots, \sigma_n$ be a i.i.d. sequence of Rademacher random variables, (i.e. is equal to 1 or -1 with equal probability), then*

$$\mathbb{P}\left[\sup_{A \in \mathcal{A}_{D_n \cup D'_n}} \left|\nu_n(A) - \nu'_n(A)\right| > \frac{\epsilon}{2}\right] \leq 4s(\mathcal{A}, n) \sup_{A \in \mathcal{A}} \mathbb{P}\left[\frac{1}{n}\left|\sum_{i=1}^{n} \sigma_i \mathbb{1}_A(Z_i)\right| > \frac{\epsilon}{4}\right]$$

Proof. The above definition of equivalence classes suggests that we should be able to use the union bound to prove this lemma. We however need to circumvent the technical hurdle that the equivalence classes depend on $D_n \cup D'_n$. This can however be done by again performing a symmetrization with respect to the sign of $\nu_n(A) - \nu'_n(A)$ (this is the right hand side of the above), for details see [PTPR, Thm 12.4]. □

Lemma 8.28.

$$\mathbb{P}\left[\frac{1}{n}\left|\sum_{i=1}^{n} \sigma_i \mathbb{1}_A(Z_i)\right| > \frac{\epsilon}{4}\right] \leq 2e^{-n\epsilon^2/32}$$

Proof.

$$\mathbb{P}\left[\frac{1}{n}\left|\sum_{i=1}^{n}\sigma_i \mathbb{1}_A(Z_i)\right| > \frac{\epsilon}{4}\right] = \mathbb{E}\left[\mathbb{P}\left[\frac{1}{n}\left|\sum_{i=1}^{n}\sigma_i \mathbb{1}_A(Z_i)\right| > \frac{\epsilon}{4}\,\middle|\, D_n\right]\right]$$

Now the conditional probability is easy to bound, as given D_n $\sum_{i=1}^{n}\sigma_i \mathbb{1}_A(Z_i)$ is the sum of n independent and bounded (in $\{-1,1\}$) random variables, we can use Hoeffdings bound (Theorem 3.6) to get

$$\mathbb{P}\left[\frac{1}{n}\left|\sum_{i=1}^{n}\sigma_i \mathbb{1}_A(Z_i)\right| > \frac{\epsilon}{4}\,\middle|\, D_n\right] \leq 2e^{-n\epsilon^2/32}$$

which proves the lemma. □

We are now ready to prove the VC generalization bound

Theorem 8.29. *Consider a sequence $Z_1,\ldots,Z_n \sim \nu$ of i.i.d. random variables and let $\mathcal{A}$ be a set of ν-measurable sets, then if $n\epsilon^2 \geq 2$ we have*

$$\mathbb{P}(\sup_{A\in\mathcal{A}} |\nu_n(A) - \nu(A)| > \epsilon) \leq 8s(\mathcal{A},n)e^{-n\epsilon^2/32}$$

Exercise 8.30. *Prove the above theorem using Lemmas 8.22, 8.27 and 8.28.*

We also have this immediate corollary

Corollary 8.31. *Let $(\Omega,\mathcal{F},\mathbb{P})$ be a probability triple and consider (X_1,Y_1), ..., (X_n,Y_n) be an i.i.d. sequence of random variables, X_i being continuous and taking values in $\mathbb{R}^m$ and $Y_i \in \{0,1\}$ is discrete. Then if $n(\epsilon)^2 \geq 8$ the following holds*

$$\mathbb{P}(\sup_{\mathcal{M}} |R_n(\phi) - R(\phi)| > \epsilon) \leq 8s(\mathcal{A},n)e^{-n\epsilon^2/64}.$$

In the above, $\mathcal{A}$ is derived from $\mathcal{M}$ as in (8.9).

8.7 Vapnik Chervonenkis dimension

Definition 8.32. *The VC-dimension of $\mathcal{A}$, denoted by $\mathcal{V}_\mathcal{A}$, equals the largest integer $n \geq 1$ such that*

$$s(\mathcal{A},n) = 2^n.$$

If the above equality holds for all n we say that $\mathcal{V}_\mathcal{A} = \infty$.

Example 8.33. *Consider sets of the form*

$$\mathcal{A} = \{(0, x_0] \times \{1\}\} \cup \{(x_0, \infty) \times \{0\} : x_0 \in \mathbb{R}\}$$

which corresponds to a decision rule as in Example 8.25. What is $s(\mathcal{A}, 1)$? Consider a single points (x_1, y_1), $x_1 \in \mathbb{R}$ and $y_1 \in \{0, 1\}$. Then for $y_1 = 0$ we get

$$\mathcal{A}_{(x_1,y_1)} = \{\{(x_1, y_1)\}, \emptyset\}$$

and for $y_1 = 1$ we get also two sets. So $s(\mathcal{A}, 1) = 2$ which is 2^1 so $\mathcal{V}_A \geq 1$. Now consider two points, we already saw this in Example 8.25 where we only got 3 sets which is less than $2^2 = 4$ (check that this is so for any other labeling as well). Conclusion is that $\mathcal{V}_A = 1$.

Example 8.34. *Lets consider the sets*

$$\mathcal{A} = \{\{(a, b) \times (c, d)\} \times \{0\} \cup \{(a, b) \times (c, d)\}^C \times \{1\} \subset \mathbb{R}^2 \times \{0, 1\} : a < b, c < d\}$$

that is, this corresponds to a classifier which classifies everything within an axis parallel rectangle as 1 and outside as 0. Now, the realization should be that it is enough to check how many different labelings we can create using rectangles. Consider a diamond pattern set of points in $\mathbb{R}^2$, i.e. $\{(1, 0), (0, 1), (-1, 0), (0, -1)\}$, using axis-parallel rectangles it should be clear that we can create all possible labelings.

Now consider 5 points, how do we realize that no matter how we do this can we create all labelings. To see this, assume for contradiction we can produce any labeling. Find the minimum enclosing rectangle for the five points. Let us now think that the points which touch the edges will be given as class 1 and the last point as class 0, but this point is inside the set and as such cannot be labeled as that. This gives us that the VC dimension of these axis parallel rectangles is 4.

Example 8.35. *Let us now consider the following class of sets consisting of polygons as in the example above. Then for any number of points placed along a unit circle we can produce any labeling, why? because if we select a subset of the points and then construct a polygon with those points as corners then this will label them as we wish. This means that any number of points can be labeled and we thus get that the VC dimension is infinite.*

Lemma 8.36 (Sauer–Shelah lemma (1972)). *For any positive integer N we have*

$$s(\mathcal{H}, N) \leq \sum_{i=0}^{\mathcal{V}_{\mathcal{H}}-1} \binom{N}{i}.$$

Proof. Out of scope. □

Lemma 8.37. *The Sauer–Shelah lemma (Lemma 8.36) is a polynomial upper bound, i.e.*

$$\sum_{i=0}^{k-1} \binom{N}{i} \leq \left(\frac{Ne}{k}\right)^k.$$

Proof. Let $\lambda \in (0,1)$ then

$$\begin{aligned} 1 &= (\lambda + (1-\lambda))^N \\ &\geq \sum_{i=1}^{\lambda N} \binom{N}{i} \lambda^i (1-\lambda)^{n-1} \\ &\geq \sum_{i=1}^{\lambda N} \binom{N}{i} \left(\frac{\lambda}{1-\lambda}\right)^{\lambda n} (1-\lambda)^n \end{aligned}$$

Thus

$$\begin{aligned} \sum_{i=1}^{\lambda N} \binom{N}{i} &\leq e^{N((\lambda-1)\log(1-\lambda)-\lambda\log(1-\lambda))} \\ &\leq e^{N(\lambda-\lambda\log(1-\lambda))} \\ &= \left(\frac{eN}{\lambda N}\right)^{\lambda N} \end{aligned}$$

Then for $k = \lambda N$ we have our result. □

Let us now turn our focus back towards the problem of the linear classifier.

Lemma 8.38. *(informal) A linear classifier in $\mathbb{R}^m$ has VC-dimension $m+1$.*

Exercise 8.39. *State and prove exactly what the above lemma means.*

Exercise 8.40. *Continuing on from the above exercise, apply Corollary 8.31 and Lemmas 8.36 and 8.37 to obtain a generalization of Theorem 8.16 which you state and prove.*

8.8 What if you don't care about $\inf R(\phi)$?

We begin with an extension of Theorem 3.6 (Hoeffdings theorem) to a sequence of dependent variables, specifically we consider

Definition 8.41. *A sequence of random variables $V_1, \dots$ is a martingale difference sequence if*

$$\mathbb{E}\left[V_{i+1} \mid V_1, \dots, V_i\right] = 0, \quad a.s.$$

for every $i > 0$. A sequence of random variables $V_1, V_2, \dots$ is called a martingale difference sequence with respect to a sequence of random variables $X_1, X_2, \dots$, if for every $u > 0$, V_i is a function of $X_1, X_2, \dots$, and

$$\mathbb{E}\left[V_{i+1} \mid X_1, \dots, X_i\right] = 0, \quad a.s.$$

This extended version of Hoeffdings theorem will be proved in the theoretical foundations course, if you are interested in the proof, see [PTPR, Thm 9.1].

Theorem 8.42. *Let $X_1, \dots$ be a sequence of RV's and assume that $V_1, \dots$ is a martingale difference sequence with respect to $X_1, \dots$. Assume that there exists random variables $Z_1, \dots$ and nonnegative constants $c_1, \dots$ such that for every $i > 0$, Z_i is a function of $X_1, \dots, X_{i-1}$ and*

$$Z_i \leq V_i \leq Z_i + c_i, \quad a.s.$$

Then for any $\epsilon > 0$ and n

$$\mathbb{P}\left(\sum_{i=1}^n V_i \geq \epsilon\right) \leq e^{-\frac{2\epsilon^2}{\sum_{i=1}^n c_i^2}}$$

and

$$\mathbb{P}\left(\sum_{i=1}^n V_i \leq -\epsilon\right) \leq e^{-\frac{2\epsilon^2}{\sum_{i=1}^n c_i^2}}$$

Interestingly, the empirical error probability of the empirically optimal classifier is with high probability close to its expected value. This is very interesting, and in the case that if the empirical minimizer of the risk is a consistent estimator, we get the best one could hope for. However it should be mentioned again that $R_n(\phi_n^*)$ is quite often a downward biased estimator of $R(\phi^*)$, that is, $\mathbb{E}\left[R_n(\phi_n^*)\right] < \inf R(\phi^*)$. Therefore, this whole deal with complexity tells us that $R_n(\phi_n^*)$ is asymptotically consistent if we have bounded VC-dimension.

Theorem 8.43. *Consider a sequence $(X_1, Y_1), \dots, (X_n, Y_n) \sim \nu$ of i.i.d. random variables, let $\mathcal{C}$ be an arbitrary set of classification rules. Let $\phi_n^* \in \mathcal{C}$ be the rule that minimizes the empirical risk given $(X_1, Y_1), \dots, (X_n, Y_n)$, i.e.*

$$R_n(\phi_n^*) = \min_{\phi \in \mathcal{C}} R_n(\phi).$$

Then for every n and $\epsilon > 0$,

$$\mathbb{P}\left(\left|\hat{R}_n(\phi_n^*) - \mathbb{E}\,[\hat{R}_n(\phi_n^*)]\right| > \epsilon\right) < 2e^{-n\epsilon^2/2}$$

Proof. Begin by defining

$$\begin{aligned} Q_i &:= \mathbb{E}\,[R_n(\phi_n^*) \mid X_1, \ldots, X_i], \quad i = 1, \ldots, n \\ Q_0 &:= \mathbb{E}\,[R_n(\phi_n^*)] \end{aligned}$$

furthermore, note that $Q_n = R_n(\phi_n^*)$ and that

$$Q_{i-1} - \frac{1}{n} \leq Q_i \leq Q_{i-1} + \frac{1}{n}$$

since changing the value of one pair (X_i, Y_i) can only change the risk by $1/n$ (1 extra or one less error in classification). If we now denote

$$V_i = Q_i - Q_{i-1}, \quad i = 1, \ldots, n$$

then V_i is a martingale difference sequence w.r.t. (X_i, Y_i) for $i = 1, \ldots, n$. Furthermore, if we define $Z_i = -1/n$ then

$$Z_i \leq V_i \leq Z_i + \frac{2}{n}.$$

Applying Theorem 8.42 we obtain

$$\mathbb{P}\left(\left|\sum_{i=1}^{n} V_i\right| > \epsilon\right) \leq 2e^{-\frac{2\epsilon^2}{\sum_{i=1}^{n}(2/n)^2}} = 2e^{-n\epsilon^2/2}$$

□

8.9 Bibliography

The perceptron and kernel part is mostly from [BlHo]. Section 8.4 is mostly covered in [PTPR] Chapter 4. Section 8.6 and the rest is scattered around in [PTPR] and other sources.