Mathematics in Machine Learning - DSE@polito

Vaccarino's part

Giuseppe Concialdi
@concialdi_g

Stefano Gioda
@stegd

Alessia Leclercq

Christian Montecchiani
@Christian_Montecchiani

Gabriele Spina
@SmokdGab

Leonardo Tredese
@leonardotredese

Michele Veronesi
@mveronesi

Spring 2022

https://github.com/users/mveronesi/projects/1/views/1
Buy us some alcohol plz https://paypal.me/veronesimichele

Contents

1 Lesson 1 - March 4

1.1 Intro

The process of analyzing data from a domain $\mathcal{X}$, starts by dividing a sampled dataset in 3 parts: training, validation and test.
If the task is supervised learning, we usually try to approximate a function f as following:

- Regression: $\hat{f} \approx f : \mathcal{X} \mapsto \mathbb{R}$
- Classification: $\hat{f} \approx f : \mathcal{X} \mapsto \{l_1, l_2, ..., l_n\}$

To get it, we use an algorithm $\mathcal{A}$ that looks for the approximation $\hat{f}$ that minimally differs from f in a fixed family of functions $\mathcal{H}$. This minimization should be theoretically done on random variables, but usually is performed on the sampled data.
The definition of $\hat{f}$ is:

$$\hat{f} = \underset{\dot{f} \in \mathcal{H}}{\arg\min} Loss(f, \dot{f}) \tag{1}$$

Notation $[m] \equiv \{1, 2, ..., m-1, m\}$
Example the family of linear functions with m parameters is:

$$\mathcal{H}_\theta = \{\theta_0 + \theta_1 x_1 + ... + \theta_m x_m \mid \theta_j \in \mathbb{R}, j \in [m]\} \tag{2}$$

Example Keeping the linear family example, given a dataset $\mathcal{X}$, split in Train and Test.

$$\begin{aligned} \mathcal{X} &= \{(x_i, y_i) \mid x_i \in \mathbb{R}^m, y_i \in \mathbb{R}, i \in \{1, ..., n\}\} \\ &= Train \cup Test, Train \cap Test = \emptyset \end{aligned} \tag{3}$$

and using the sum of squared error loss the minimization becomes:

$$\hat{f} = \underset{(\theta_0, \theta_1, ..., \theta_m)}{\arg\min} \sum_{(x_i, y_i) \in Train} (y_i - f_{(\theta_0, \theta_1, ..., \theta_m)}(x_i))^2 \tag{4}$$

The minimization is then only applied to training data and then the result is checked on test data, defining the following errors:

- **Training Error:** which is the one that gets minimized.

$$\sum_{(x_i, y_i) \in Train} (y_i - \hat{f}(x_i))^2$$

- **Test Error:** which is a representation of the error on new unknown Data.

$$\sum_{(x_j, y_j) \in Test} (y_j - \hat{f}(x_j))^2$$

But this approach is extremely heuristical, as it provides no guaranty on how good the representation of the error on new unknown data is.

1.2 The Statistical learning framework

Let's now formalize statistical learning for classification task in order to prove the correctness of the previous heuristic.

- **Domain Set**, $\mathcal{X}$: which contains objects we want to label.
  For example $x \in \mathcal{X}, x = (x_1, ..., x_m) \in \mathbb{R}^m$
- **Label Set**, $\mathcal{Y}$: which is the set of all possible labels. In this notes will be restricted to either $\{0, 1\}$ or $\{-1, 1\}$.
- **Training Data**, $\mathcal{S} = ((x_1, y_1), ..., (x_m, y_m))$ is a finite sequence (and not a set, as it may contain duplicates) of pairs from $\mathcal{X} \times \mathcal{Y}$
- **Learners output, hypothesis or classifier**, $h : \mathcal{X} \mapsto \mathcal{Y}$: the prediction rule that we will use to label our instances.

- **Data Generator**: all instances are drawn from the same population, equivalent to a probability distribution $\mathcal{D}$ over $\mathcal{X}$, **this distribution is unknown**.
- **Measure of Success or True error**: This is the value that indicates the goodness of the prediction rule applied on data. Lets define the error of the prediction rule $h : \mathcal{X} \mapsto \mathcal{Y}$ as the probability of sampling an instance on which h is wrong:

$$\mathcal{L}_{\mathcal{D},f}(h) \overset{def}{=} \underset{x\sim\mathcal{D}}{\mathbb{P}}[h(x) \neq f(x)] \overset{def}{=} \mathcal{D}(\{x \in \mathcal{X} \mid h(x) \neq f(x)\})$$

  Which is low if the misclassification happens in a rare region of the distribution. For example if h is different from f only in a part of the domain that has no probability of being sampled, than the error is 0.

Usually our setup is the following:

- There are 2 unknowns: the distribution $\mathcal{D}$ the target function $f : \mathcal{X} \mapsto \mathcal{Y}$
- The only thing we known of the setup, is the sampled training set $\mathcal{S} \sim \mathcal{D}^{|\mathcal{S}|}$

The output of a learning Algorithm over S is $h_{\mathcal{S}} : \mathcal{X} \mapsto \mathcal{Y}$, but since f and D are unknown we cannot compute $\mathcal{L}_{\mathcal{D},f}(h_{\mathcal{S}})$. Instead we can compute the **Training Error** defined as such:

$$\mathcal{L}_{\mathcal{S}}(h_{\mathcal{S}}) \overset{def}{=} \frac{|\{i \in [n] \mid h_{\mathcal{S}}(x_i) \neq y_i\}|}{n}, |\mathcal{S}| = n$$

.

$\mathcal{L}_{\mathcal{S}}$ is also known as Empirical Risk. **Empirical Risk Minimization** is the task of looking for h such that it minimizes the training error. When $\mathcal{L}_{\mathcal{S}}(h_{\mathcal{S}}) \ll \mathcal{L}_{\mathcal{D},f}(h_{\mathcal{S}})$ we have the **overfitting** phenomena. Moreover, we can always define a function $h_{\mathcal{S}}$ such that $\mathcal{L}_{\mathcal{S}}(h_{\mathcal{S}}) = 0$ and which will probably overfit:

$$h_{\mathcal{S}}(x) = \begin{cases} y_i & \text{if } \exists i \in [n] : x_i = x \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

1.3 Inductive Bias

Instead of searching h in the set of all functions $\mathcal{Y}^{\mathcal{X}} = \{g : \mathcal{X} \mapsto \mathcal{Y}\}$ (which contains the probably overfitting function), we restrict our selves to a subset $\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$ called **hypothesis class**. This introduces a bias towards the specific kind of predictors belonging to the class. So an empirical risk minimizator of our class is:

$$\text{ERM}_{\mathcal{H}}(\mathcal{S}) \in \underset{h\in\mathcal{H}}{\arg\min} \mathcal{L}_{\mathcal{S}}(h)$$

Finite hypothesis classes

It's a mild restriction as we can choose a vast class of functions.
Realizability assumption: Exists a function in our hypothesis class that has the true error equal to 0.

$$\exists h^{\star} \in \mathcal{H} : \mathcal{L}_{\mathcal{D},f}(h^{\star}) = \underset{x\sim\mathcal{D}}{\mathbb{P}}[f(x) \neq h^{\star}(x)] = 0$$

$$\Rightarrow \mathcal{L}_{\mathcal{S}}(h^{\star}) = 0, \forall \mathcal{S} \sim \mathcal{D}$$

$$\Rightarrow \mathcal{L}_{\mathcal{S}}(h_{\mathcal{S}}) = 0 \text{ if } h_{\mathcal{S}} \in \underset{h\in\mathcal{H}}{\arg\min} \mathcal{L}_{\mathcal{S}}(h), \forall \mathcal{S} \sim \mathcal{D}$$

The previous statement does not imply that $h^{\star} = f$ where f is the labeling rule, as they may differ on part of the domain that has zero probability of being sampled.

I.I.D assumption

The examples of any training set are independently and identically distributed(i.i.d) according to the distribution $\mathcal{D}$. So, sampling an instance has distribution $x \sim \mathcal{D}$, sampling the whole training set has distribution $\mathcal{S} \sim \mathcal{D}^n, |\mathcal{S}| = n$

2 Lesson 2 - March 11

2.1 PAC Learnability

Definition 2.1 (PAC Learnability) *A hypothesis class $\mathcal{H}$ is PAC learnable if there exists a function:*

$$m_{\mathcal{H}} : (0,1)^2 \to \mathbb{N}$$

and a learning algorithm A for which:

- $\forall\ \epsilon \in (0,1)$ *Accuracy parameter.*
- $\forall\ \delta \in (0,1)$ *Probability of getting a non representative sample*
- $\forall\ \mathcal{D}$ *distribution over $\mathcal{X}$*
- $\forall f : \mathcal{X} \to \{0,1\}$

*If the **realizability assumption** holds with respect to $\mathcal{H}$, $\mathcal{D}$ and f then: running an algorithm **A** on $m \geq m_{\mathcal{H}}(\epsilon,\delta)$ **i.i.d.** samples drawn from an unknown distribution $\mathcal{D}$ and labeled by an unknown function f. **A** returns an hypothesis $h_{\mathcal{S}}$ for which:*

$$\mathbb{P}[L_{\mathcal{D},f}(h_{\mathcal{S}}) \leq \epsilon] \geq 1-\delta$$

Probably and **Approximate** are two important concepts which come directly from the fact that:

- **Probability** of having a successful learning algorithm is not 1 but $1-\delta$, hence failures of the learner can happen.
- **Approximate** because we admit a small threshold of error, which is ϵ.

Notice that the PAC leanability leverages the function:

$$m_{\mathcal{H}} : (0,1)^2 \to \mathbb{N}$$

this function is called **sample complexity** and it describes the minimal number of samples that would guarantee at least with probability $1-\delta$ a ϵ-success of the learner.

Let us now recall the conclusion of the analysis of finite hypothesis classes. It can be rephrased as stating:

Corollary 2.1 *Every finite hypothesis class is PAC learnable with sample complexity (assuming **realizability assumption**):*

$$m_{\mathcal{H}} \leq \left\lceil \frac{log(\frac{|\mathcal{H}|}{\delta})}{\epsilon} \right\rceil$$

The model we have just described can be readily generalized. To do that, we consider generalizations in 2 aspects:

1. *Removing Realizability Assumption.* For practical learning tasks, this assumption may be too strong. So we will study *agnostic* PAC model in which this assumption is relaxed.
2. *Learning Problems beyond binary Classification* The learning task that we have been discussing so far has to do with predicting a binary label. However, many learning tasks take a different form. It turns out that our analysis of learning can be readily extended to such scenarios by allowing a variety of loss functions.

2.2 Releasing the Realizability Assumption- Agnostic PAC Learning

The basic PAC setting is deterministic, because the existence of a function f that maps each instance of the training x_i into $f(x_i)$ has been assumed. However, it is more realistic not to assume that the labels are fully determined by the features we measure on the input elements.

Formally, from now on, let $\mathcal{D}$ be a probability distribution over $\mathcal{X} \times \mathcal{Y}$ (as before $\mathcal{X}$ is our domain set and $\mathcal{Y}$ is a set of labels). That is a *joint distribution* over domain and labels. We can view it is composed by 2 parts:

- A distribution $\mathcal{D}_{\mathcal{X}}$ over **unlabeled** domain points (also called *marginal distribution*)
- A *conditional distribution* over labels for each domain point, $\mathcal{D}((x, y)|x)$

We moved from a scenario where given two domain points x_1 and x_2 such that $x_1 = x_2$ they surely have the same label $y = f(x_1) = f(x_2)$. In the agnostic scenario instead, where f does not exist, x_1 and x_2 can have different labels.

This switch in the nature of the labels also brings to a change in the definition of the *true error* and the *empirical error*, which previously depended on $f(\cdot)$.

True Error For a probability distribution, $\mathcal{D}$, over $\mathcal{X} \times \mathcal{Y}$, one can measure how likely h is to make an error when labeled points are randomly drawn according to $\mathcal{D}$. We redefine the **true error** (or risk) of a prediction rule h to be:

$$L_{\mathcal{D}}(h) = \underset{(x,y)\sim\mathcal{D}}{\mathbb{P}}[h(x) \neq y] = \mathcal{D}(\{(x, y) : h(x) \neq y\})$$

However, the learner does not know the data distribution $\mathcal{D}$. What the learner has access to is the training data, S. **Empirical Risk** The definition of the empirical risk remains the same as before, namely:

$$L_S(h) = \frac{|i \in [1, m] : h(x_i) \neq y_i|}{m}$$

At first consider the best possible classifier, often referred to as the **Bayes Optimal Predictor Classifier**.

If $\mathcal{D}$ is the data distribution over ($\mathcal{X} \times \mathcal{Y} = \{0, 1\}$) one could elaborate the best possible labeling function as:

$$f_{\mathcal{D}}(x) = \begin{cases} 1 & if\ \mathbb{P}[y = 1 \mid x]\ \geq \frac{1}{2} \\ 0 & otherwise \end{cases}$$

There is no other classifier $g : \mathcal{X} \to \{0, 1\}$ that has a lower error, that is $\forall\ g,\ L_{\mathcal{D}}(f_{\mathcal{D}}) \leq L_{\mathcal{D}}(g)$. Since we do not know $\mathcal{D}$, we cannot use the optimal predictor $f_{\mathcal{D}}$.

Definition 2.2 (Agnostic PAC Learnability) *A hypothesis class $\mathcal{H}$ is agnostic PAC learnable if there exists a function $m_{\mathcal{H}} : (0, 1)^2 \to \mathbb{N}$ and a learning algorithm **A** such that:*

- $\forall\ (\epsilon, \delta) \in (0, 1)^2$
- $\forall\ \mathcal{D}:$ *distribution over* $\mathcal{X} \times \mathcal{Y}$

Then:

$$A(S) = h\ :\ \mathbb{P}[\ L_{\mathcal{D}}(h) \leq \min_{h' \in \mathcal{H}} L_{\mathcal{D}}(h')\ +\ \epsilon] \geq 1 - \delta$$

$$for\ S :\ |S| = m \geq m_{\mathcal{H}}(\epsilon, \delta)$$

When the Realizability Assumption holds, so we have that $min_{h' \in \mathcal{H}} L_{\mathcal{D}}(h) = 0$, Agnostic PAC learning provides the same guarantee as PAC learning. In that sense, agnostic PAC learning generalizes the definition of PAC learning. Nevertheless, under the definition of agnostic PAC learning, a learner can still declare a success if its error is not much larger than the error achievable by the best predictor included in the class $\mathcal{H}$. This is in contrast to PAC learning, in which the learner is required to achieve a small error in absolute terms and not with respect to the best error achievable by the hypothesis class.

2.3 Releasing the Binary Labeling

As mentioned before, another generalization is to extend the label set, $\mathcal{Y}$, from a binary class to a variety of learning tasks. To do that we have to change the definition of Loss function.

Generalized Loss Function Given any set $\mathcal{H}$ and some domain Z let l be any function from $\mathcal{H} \times Z$ (where Z is coincident to $\mathcal{X} \times \mathcal{Y}$ for classification and $\mathcal{X}$ for clustering.) to the set of non negative real numbers $\mathbb{R}_+$:

$$l : (\mathcal{H} \times Z) \to \mathbb{R}_+$$

we call such functions *loss functions.*

We now define the *risk function* to be the expected loss of a classifier, $h \in \mathcal{H}$, with respect to a probability distribution $\mathcal{D}$ over Z, namely:

$$L_{\mathcal{D}}(h) \; = \; \mathbb{E}_{z\sim\mathcal{D}}[\; l(h,z) \;]$$

That is, we consider the expectation of the loss of h over objects z picked randomly according to $\mathcal{D}$. Similarly, we define the *empirical risk* to be the expected loss over a given sample $S = (z_1, \dots, z_m) \in Z^m$, namely,

$$L_S(h) = \frac{1}{m}\sum_{i=1}^{m} l\;(h, z_i)$$

Commonly used loss functions are:

Loss 0-1 : Typically used in classification (the random variable z ranges over the set of pairs $\mathcal{X} \times \mathcal{Y}$) and the loss function is:

$$l_{0-1}(h,(x,y)) = \begin{cases} 0 & if\ h(x) = y \\ 1 & if\ h(x) \neq y \end{cases}$$

One should note that, for a random variable, α, taking values $\{0,1\}$, we have that: $\mathbb{E}_{\alpha\sim\mathcal{D}}[\alpha] = \mathbb{P}_{\alpha\sim\mathcal{D}}[\alpha = 1]$, which by definition of the true error (which is the *expected value* of the loss function) we obtain that, for **loss 0-1**: $L_{\mathcal{D}}(h) = \mathbb{E}_{z\sim\mathcal{D}}(l(h,z))$.

Squared Loss : Here, our random variable z ranges over the set of pairs $\mathcal{X} \times \mathcal{Y}$ and the loss function is:

$$l_{sq}(h,(x,y)) = (h(x) - y)^2$$

This loss function is typically used in *regression problems.*

2.4 Exercises

Exercise 2.1 Given a training set $S = \{(x_i, f(x_i))\}_{i=1}^m \subseteq (\mathbb{R}^d \times \{0,1\})^m$ show that there exists a polynomial p_S such that $h_S(x) = 1$ if and only if $p_S(x) \geq 0$, where h_S is defined:

$$h_S(x) = \begin{cases} y_i & if\ \exists\ i \in [m]\ such\ that\ x_i = x \\ 0 & otherwise \end{cases}$$

To show that $h_S(x) = 1$ if and only if $p_S(x) \geq 0$, we define:

$$p_S(x) = -\prod_{i\in[m]:y_i=1} ||x - x_i||^2$$

Notice that this function, p_S, is equal to zero only if x is a training point and its label is equal to 1, while for every other point we have that $p_S < 0$. So we can modeled the behaviour of $h_S(x)$ in the following way:

$$h_S = \mathbb{1}_{[p_S(x)\geq 0]} = \begin{cases} 1 & if\ x_i \in S\ and\ y_i = 1 \\ 0 & otherwise \end{cases}$$

Exercise 2.2 Let $\mathcal{H}$ be a class of binary classifiers over a domain $\mathcal{X}$. Let $\mathcal{D}$ be an unknown distribution over $\mathcal{X}$, and let f be the target hypothesis in $\mathcal{H}$. Fix some $h \in \mathcal{H}$. Show that the expected value of $L_S(h)$ over the choice of $S|_x$ equals $L_{(\mathcal{D},f)}(h)$, namely:

$$\mathop{\mathbb{E}}_{S|_x\sim\mathcal{D}^m}[L_S(h)] = L_{(\mathcal{D},f)}(h)$$

Solution By definition we obtain that:

$$\mathop{\mathbb{E}}_{S|_x\sim\mathcal{D}^m}[L_S(h)] = \mathop{\mathbb{E}}_{S|_x\sim\mathcal{D}^m}\left[\frac{1}{m}\sum_{i=1}^{m}\mathbb{1}_{[h(x_i)\neq f(x_i)]}\right]$$

By linearity of the expected value we can put it inside the summation,

$$\mathop{\mathbb{E}}_{S|_x\sim\mathcal{D}^m}\left[\frac{1}{m}\sum_{i=1}^{m}\mathbb{1}_{[h(x_i)\neq f(x_i)]}\right] = \frac{1}{m}\sum_{i=1}^{m}\mathbb{E}\left[\mathbb{1}_{[h(x_i)\neq f(x_i)]}\right]$$

Notice that the indicator is a function that can assume values that are only zero and ones, in particular in this case it assumes value 1 only when $h(x_i) \neq f(x_i)$. And for the property of the indicator function its expected value is equal to to the probability of being one.

$$\frac{1}{m}\sum_{i=1}^{m} \mathbb{E}\left[\mathbb{1}_{[h(x_i)\neq f(x_i)]}\right] = \frac{1}{m}\sum_{i=1}^{m} \mathbb{P}_{x_i\sim\mathcal{D}}\left[\mathbb{1}_{[h(x_i)\neq f(x_i)]}\right]$$

By definition the probability inside the summation is exactly the true error, so just by doing simple math we obtain that:

$$\frac{1}{m}\sum_{i=1}^{m} \mathbb{P}_{x_i\sim\mathcal{D}}\left[\mathbb{1}_{[h(x_i)\neq f(x_i)]}\right] = \frac{1}{m}\cdot m \cdot L_{(\mathcal{D},f)}(h) = L_{(\mathcal{D},f)}(h)$$

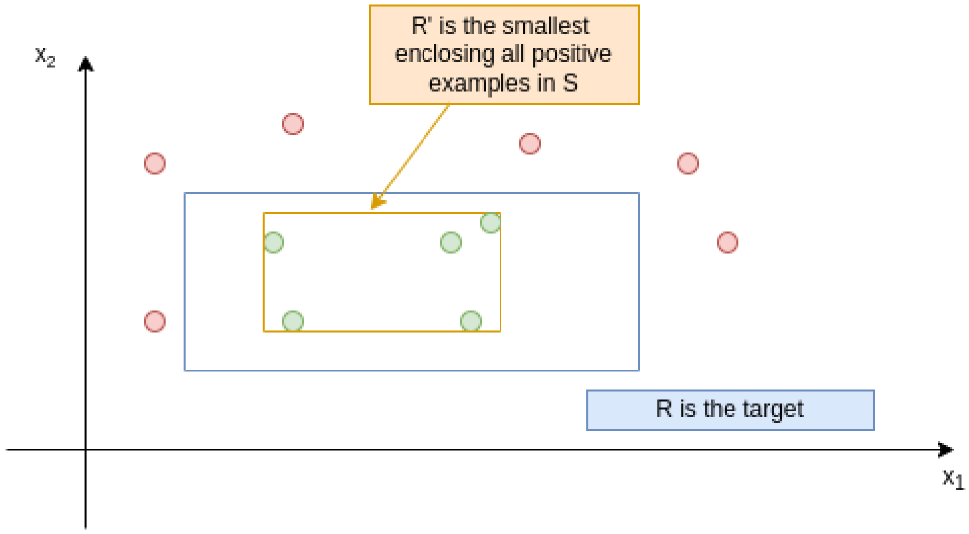
Exercise 2.3 An axis aligned rectangle classifier in the plane is a classifier that assigns the value 1 to a point if and only if it is inside a certain rectangle. Formally, given real numbers $a_1 \leq b_1, a_2 \leq b_2$, define the classifier $h_{(a_1,b_1,a_2,b_2)}$ by

$$h_{(a_1,b_1,a_2,b_2)}(x_1,x_2) = \begin{cases} 1 & if\ a_1 \leq x_1 \leq b_1\ and\ a_2 \leq x_2 \leq b_2 \\ 0 & otherwise \end{cases}$$

The class of all axis aligned rectangles in the plane is defined as

$$\mathcal{H}^2 = \{h_{(a_1,b_1,a_2,b_2)} : a_1 \leq b_1\ and\ a_2 \leq b_2\}$$

Note that this is an infinite size hypothesis class. Throughout this exercise we rely on the realizability assumption. To give a graphical representation of the class:

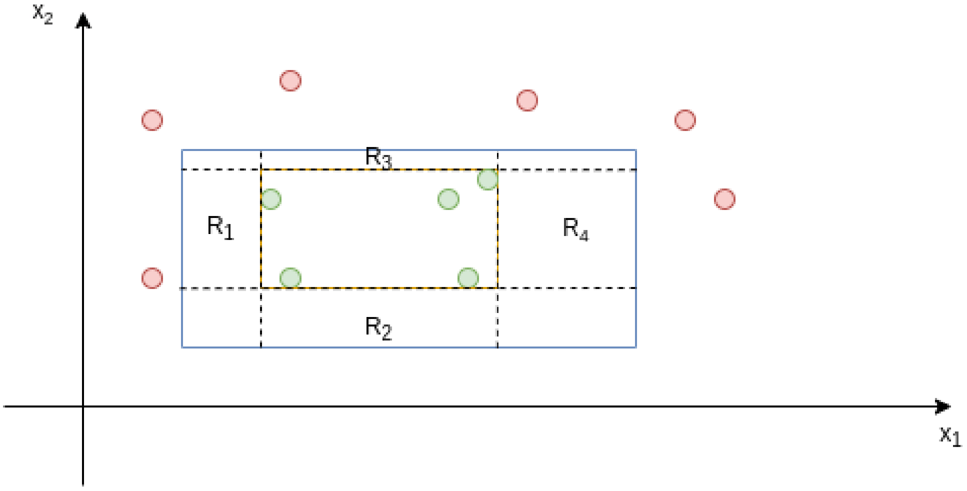


1. The function that satisfies the realizability assumption is defined as $R^\star$. A is the algorithm that returns the smallest rectangle R' enclosing all positive examples in the training set S. By definition $R' \subseteq R^\star$ labels correctly all the positive examples in S. Moreover, since we assumed that the realizability assumption holds, and since the tightest rectangle enclosing all positive examples is returned, all the negative examples are labeled correctly by R' as well. So, we conclude that R' is an ERM.

2. Since $R' \subseteq R^\star$ the true error of R' is going to be:

$$L_{(\mathcal{D},f)}(R') = \mathcal{D}(R^\star \setminus R')$$

   Fix $\epsilon \in (0,1)$. Then define R_1, R_2, R_3, R_4, similarly to the image such that each of their probability mass is exactly $\frac{\epsilon}{4}$.



For each R_i we define the event F_i as drawing a sample $S|_x$ such that it does not contain any point in the area of R_i. Formally:

$$F_i = \{S|_x : S|_x \cap R_i = \emptyset\}$$

Since R' is the minimal rectangle around all positive sampled elements, the probability of misclassifying increases if the positive samples in S are distant from the borders of $R^\star$, so the probability of $L_{(\mathcal{D},f)}(R') > \epsilon$ is upper bounded by the probability of all the F_i happening.

$$\mathcal{D}^m(\{S : L_{(\mathcal{D},f)}(R') > \epsilon\}) \leq \mathcal{D}^m\left(\bigcup_{i=1}^{4} F_i\right) \leq \sum_{i=1}^{4} \mathcal{D}^m(F_i)$$

Let's fix $\mathcal{D}^m(F_i)$ as the probability that all of the instances do not fall in R_i is exactly $(1-\frac{\epsilon}{4})^m$. Therefore,

$$\mathcal{D}^m(F_i) = (1 - \frac{\epsilon}{4})^m \leq e^{-\frac{m\epsilon}{4}}$$

So, taking the expression before and the last one we obtain,

$$\mathcal{D}^m(\{S : L_{\mathcal{D},f}(A(S)) > \epsilon\}) \leq 4 \cdot e^{-\frac{m\epsilon}{4}}$$

3 Lesson 3 - March 18

3.1 PAC vs. APAC

The table below summarizes the main differences between basic PAC and APAC learning.

	basic **PAC**	***agnostic*** **PAC**
Distribution	$\mathcal{D}$ over $\mathcal{X}$	$\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$
Truth	$f \in \mathcal{H}$	not in $\mathcal{H}$ or does not exist
Risk	$\mathcal{D}(\{x : h(x) \neq f(x)\})$	$\mathcal{D}(\{(x, y) : h(x) \neq y\})$
Training	$(x_1, ..., x_m) \sim \mathcal{D}^m, y_i = f(x_i)$	$((x_1, y_1), ..., (x_m, y_m)) \sim \mathcal{D}^m$
Goal	$L_{\mathcal{D},f}(A(S)) \leq \epsilon$	$L_{\mathcal{D}}(A(S)) \leq \min_{h \in \mathcal{H}} L_{\mathcal{D}}(h) + \epsilon$

Basic PAC and the APAC learning are also called respectively **discriminative** and **generative**. In the discriminative setting the objective is to find the function f while in the generative one the aim is to learn the underling distribution $\mathcal{D}$.

3.2 Uniform Convergence is sufficient for Learnability

In the ERM paradigm the hope is that an h that minimizes the empirical risk with respect to S is a risk minimizer with respect to the true data probability distribution as well. For that, it suffices to ensure that the empirical risks of all members of $\mathcal{H}$ are good approximations of their true risk, **hence that uniformly over all hypotheses in the hypothesis class, the empirical risk will be close to the true risk**.

Definition 3.1 (ϵ - representative sample) *A training set S is called ϵ-representative with respect to a domain Z, a hypothesis class $\mathcal{H}$, a loss function l, and a distribution $\mathcal{D}$ if*

$$\forall h \ \in \mathcal{H}, \quad |L_S(h) - L_{\mathcal{D}}(h)| \leq \epsilon$$

Lemma 3.1 (Goodness of ERM result in $\frac{\epsilon}{2}$-representative samples) *Given a $\frac{\epsilon}{2}$-representative sample S, formally:*

$$S : |L_S(h) - L_{\mathcal{D}}(h)| \leq \frac{\epsilon}{2} \quad \forall h \ \in \mathcal{H}$$

Then, any output $h_S \in \mathcal{H}$ of $ERM_{\mathcal{H}}(S)$ satisfies:

$$L_{\mathcal{D}}(h_S) \leq \ min_{h' \in \mathcal{H}} L_{\mathcal{D}}(h') + \epsilon$$

The lemma simply states that whenever a $\frac{\epsilon}{2}$-representative sample S is given, then it is guaranteed that the ERM-rule returns a good hypothesis h_S. Consequently, given the right training set S, the true risk can be controlled by the computation of the empirical risk.

Proof If S is $\frac{\epsilon}{2}$ representative, then:

$$|L_S(h) - L_{\mathcal{D}}(h)| \leq \frac{\epsilon}{2}$$

If this is true $\forall h \ \in \mathcal{H}$ that is true for h_S, which is the output of the $\text{ERM}_{\mathcal{H}}$ rule. So, we have:

$$|L_S(h_S) - L_{\mathcal{D}}(h_S)| \leq \frac{\epsilon}{2}$$

$$L_{\mathcal{D}}(h_S) \leq L_S(h_S) + \frac{\epsilon}{2}$$

By definition of h_S we have that $L_S(h_S) \leq L_S(h) \quad \forall h \ \in \mathcal{H}$, since h_S is the given empirical risk minimizer. This implies:

$$L_{\mathcal{D}}(h_S) \ \leq \ L_S(h_S) + \frac{\epsilon}{2} \ \leq \ L_S(h) + \frac{\epsilon}{2}$$

But if S is $\frac{\epsilon}{2}$ representative and by definition we have that:

$$L_S(h) \leq \ L_{\mathcal{D}}(h) + \frac{\epsilon}{2}$$

Which implies:

$$L_{\mathcal{D}}(h_S) \leq \ L_S(h) + \frac{\epsilon}{2} \ \leq \ L_{\mathcal{D}}(h) + \frac{\epsilon}{2} + \frac{\epsilon}{2} = L_{\mathcal{D}}(h) + \epsilon$$

The previous lemma can be easily translated into a property of the hypothesis class $\mathcal{H}$ by introducing the notion of **Uniform Convergence**.

Definition 3.2 (Uniform Convergence) *A hypothesis class $\mathcal{H}$ has the uniform convergence property (with respect to a domain Z and a loss function l) if there exists a function:*

$$m_{\mathcal{H}}^{UC} : \ (0,1)^2 \to \mathbb{N}$$

such that

- *for every $(\epsilon, \delta) \in (0,1)^2$,*
- *for every $\mathcal{D}$ over Z,*

then following holds: a sample

$$S = (z_1, \ldots, z_m) : \quad |S| = m \geq m_{\mathcal{H}}^{UC}(\epsilon, \delta)$$

of i.i.d observations drawn according to $\mathcal{D}$ is guaranteed to be ϵ-representative with probability $1 - \delta$.

In this case, $m_{\mathcal{H}}^{UC}$ represents the **minimal sample complexity** required to ensure the uniform convergence property, hence to ensure that with probability of at least $1 - \delta$ the sample S is ϵ-representative.

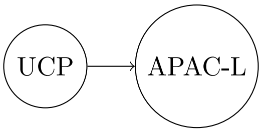
Corollary 3.1 *If a class $\mathcal{H}$ has the uniform convergence property with a function $m_{\mathcal{H}}^{UC}$ then the class is agnostic PAC learnable with sample complexity:*

$$m_{\mathcal{H}}(\epsilon, \delta) \leq m_{\mathcal{H}}^{UC}(\frac{\epsilon}{2}, \delta)$$

Furthermore, in that case, the ERM$_{\mathcal{H}}$ paradigm is a successful agnostic PAC learner for $\mathcal{H}$.

3.3 Equivalent statement

If a class of function $\mathcal{H}$ has the *Uniform Convergence property*, then automatically the ERM on that sample is agnostic PAC learnable and the sample complexity is bounded by $m_{\mathcal{H}}^{UC}(\frac{\epsilon}{2}, \delta)$. In other words, *Uniform Convergence Property* implies *Agnostic PAC.*



We want to show that every finite class $\mathcal{H}$ has the Uniform Convergence property. Basically, we want to show that, for fixed $\epsilon, \delta \in (0,1)$:

$$\mathcal{D}^m(\{S : \forall h \in \mathcal{H}, |L_S(h) - L_{\mathcal{D}}(h)| \leq \epsilon\}) \geq 1 - \delta$$

So we need to find a sample complexity $m > 0$ such that:

- $\forall \mathcal{D}$ over Z
- with probability $\geq 1 - \delta$ over the choice $S = (z_1, \ldots, z_m) \sim \mathcal{D}^m$

it holds that $\forall h \in \mathcal{H} : |L_S(h) - L_{\mathcal{D}}(h)| \leq \epsilon$.

3.4 Concentration inequalities

Concentration inequalities provide bound for stochastic quantities.

Definition 3.3 (Markov's Inequality) *Let Z be a non-negative random variable, then $\forall a > 0$*

$$\mathbb{P}[Z \geq a] \leq \frac{\mathbb{E}[Z]}{a} \tag{6}$$

Definition 3.4 (Chebyshev's Inequality) *Let Z be a non-negative random variable, then $\forall a > 0$*

$$\mathbb{P}[|Z - \mathbb{E}[Z]| \geq a] \leq \frac{\mathbb{V}[Z]}{a^2} \tag{7}$$

Definition 3.5 (Hoeffding's Inequality) *Consider the following definitions:*

- *$Z_1, ..., Z_m$ are i.i.d random variables*

- $\bar{Z} = \frac{1}{m}\sum_{i=1}^{m} Z_i$
- $\forall i : \mathbb{E}[Z_i] = \mu$
- $\forall i : \mathbb{P}[a \leq Z_i \leq b] = 1$

Then $\forall \varepsilon > 0$

$$\mathbb{P}\left[\left|\frac{1}{m}\sum_{i=1}^{m} Z_i - \mu\right| \geq \varepsilon\right] \leq 2e^{\frac{-2m\varepsilon^2}{(b-a)^2}} \tag{8}$$

3.5 Finite Classes are Agnostic PAC Learnable

In view of the preceding Corollary, the claim that every finite hypothesis class is agnostic PAC learnable will follow once we establish that uniform convergence holds for a finite hypothesis class.

Fix some ϵ, δ. We need to find a sample size m that guarantees that for any $\mathcal{D}$, with probability of at least $1-\delta$ of the choice of $S = (x_1, \ldots, x_m)$ samples **i.i.d.** from $\mathcal{D}$ we have that for all $h \in \mathcal{H}$, $|L_S(h) - L_\mathcal{D}(h)| \leq \epsilon$. That is,

$$\mathcal{D}^m(\{S : \forall h \in \mathcal{H}, |L_S(h) - L_\mathcal{D}(h)| \leq \epsilon\}) \geq 1 - \delta$$

Equivalently, we need to show that

$$\mathcal{D}^m(\{S : \exists h \in \mathcal{H}, |L_S(h) - L_\mathcal{D}(h)| > \epsilon\}) < \delta$$

Now, clearly enough,

$$\mathcal{D}^m(\{S : \exists h \in \mathcal{H}, |L_S(h) - L_\mathcal{D}(h)| > \epsilon\}) = \mathcal{D}^m(\bigcup_{h \in \mathcal{H}} \{S : |L_S(h) - L_\mathcal{D}(h)| > \epsilon\})$$

and applying the union bound, we obtain:

$$\mathcal{D}^m(\{S : \exists h \in \mathcal{H}, |L_S(h) - L_\mathcal{D}(h)| > \epsilon\}) \leq \sum_{h \in \mathcal{H}} \mathcal{D}^m(\{S : |L_S(h) - L_\mathcal{D}(h)| > \epsilon\})$$

Now recall that $L_\mathcal{D}(h) = \mathbb{E}_{z \sim \mathcal{D}}\,[l(h, z)]$ and that $L_S(h) = \frac{1}{m}\sum_{i=1}^{m} l(h, z_i)$. Since each z_i is sampled i.i.d. from $\mathcal{D}$, the expected value of the random variable $l(h, z_i)$ is $L_\mathcal{D}$. Which implies:

$$|L_S(h) - L_\mathcal{D}(h)| = |L_S(h) - \mathbb{E}[L_S(h)]|$$

Consequently, $|L_S(h) - L_\mathcal{D}(h)|$ represents the deviation of $L_S(h)$ from its mean. We therefore need to show that the measure of $L_S(h)$ is concentrated around its expectation value.

According to the Law of Large Numbers, when m goes to infinity the empirical average converges to the true expectation. This is true for $L_S(h)$, since it defined as the empirical average of i.i.d. random variables. However, the law of large number is only an asymptotic result and it does not provide any information about the gap between the empirically estimated error and its true value for any given finite sample size. A probability measure for this gap is provided by the Hoeffding's inequality. Lets consider the the random variables $l(h, z_i)$. Since h is fixed and $z_1, \ldots, z_m$ are sampled i.i.d., it follows that $l(h, z_i)$ are also i.i.d. random variables. If we denote $L_\mathcal{D}(h) = \mu$ and further assume that $l \in [0, 1]$, we obtain:

$$\mathcal{D}^m(\{S : |L_S(h) - L_\mathcal{D}(h)| > \epsilon\}) = \mathbb{P}[|\frac{1}{m}\sum_{i=1}^{m} l(h, z_i) - \mu| > \epsilon] \leq 2\exp\{-2\ m\epsilon^2\}$$

But then,

$$\mathcal{D}^m(\{S : \exists h \in \mathcal{H}, |L_S(h) - L_\mathcal{D}(h)| > \epsilon\}) \leq \sum_{h \in \mathcal{H}} 2\exp\{-2\ m\epsilon^2\} =$$
$$= 2|\mathcal{H}|\exp(-2m\epsilon^2)$$

Finally, if we choose

$$m \geq \left\lceil \frac{log(2\ |\mathcal{H}|/\delta)}{2\epsilon^2} \right\rceil$$

(this formula is the minimum number of sample that are sufficient to learn the hypothesis from this class) then

$$\mathcal{D}^m(\{S : \exists h \in \mathcal{H}, |L_S(h) - L_\mathcal{D}(h)| > \epsilon\}) \leq \delta$$

Which is exactly the expression to control the sample complexity in order to ensure as ϵ-representative sample with probability $1 - \delta$. Finally, note that, as previously defined, ϵ-representative implies 2ϵ APAC learnability for any given hypothesis class. So we have that **finite classes** are APAC learnable.

N.B: The class is agnostic PAC learnable using the ERM algorithm with sample complexity:

$$m_{\mathcal{H}}(\epsilon, \delta) \leq m_{\mathcal{H}}^{UC}(\frac{\epsilon}{2}, \delta) \leq \left\lceil \frac{2\log(2|\mathcal{H}|/\delta)}{\epsilon^2} \right\rceil$$

4 Lesson 4 - April 1

4.1 VC-Dimension

Definition 4.1 (Restriction of $\mathcal{H}$ to C) *$\mathcal{H}$ is an hypothesis class of functions $\mathcal{X} \mapsto \{0,1\}$, and C is a subset $\{c_1, c_2, \dots, c_m\} = C \subset \mathcal{X}$ on the domain. The restriction of $\mathcal{H}$ to C is: the set of functions $C \mapsto \{0,1\}$ that can be derived from $\mathcal{H}$. The restriction is represented as:*

$$\mathcal{H}_C = \{(h(c_1), \dots, h(c_m)) : h \in \mathcal{H}\}$$

so we represent each function $C \mapsto \{0,1\}$ as a vector in $\{0,1\}^{|C|}$.

If this restriction corresponds to all the possible sequences of zeros and ones $\mathcal{H}$ shatters the set C.

Definition 4.2 (Shattering) *A hypothesis class $\mathcal{H}$ shatters a finite set $C \subset \mathcal{X}$ if the restriction of $\mathcal{H}$ to C is the set of all function from C to $\{0,1\}$. That is equivalent of saying $|\mathcal{H}_C| = 2^{|C|}$*

The meaning of VC-Dimension is to measure the complexity of an hypothesis class $\mathcal{H}$, which is done by computing what is the restriction C of biggest size that can be shattered by $\mathcal{H}$. Formally

$$\text{VC-Dimension}(\mathcal{H}) = \max_{C \subset \mathcal{X} : |\mathcal{H}_C| = 2^{|C|}} |C|$$

Example: let $\mathcal{H}$ be the class of threshold functions $h_a(x) = \mathbb{1}_{x<a}$ over $\mathbb{R}$. Its VC-dimension($\mathcal{H}$) $= 1$ because given $C = \{c_1\}$ we can impose $a = c_1 + 1$ so $h_a(c_1) = 1$ and imposing $a = c_1 - 1$, so $h_a(c_1) = 0$. But if $C = \{c_1, c_2\}$ with $c_1 < c_2$ the threshold function cannot produce the labeling $h_a(c_1) = 0, h_a(c_2) = 1$. Since it can shatter at most restrictions on subsets of size 1 its VC-dimension is 1.

The **No Free Lunch theorem** (which will be defined in lesson 5) can be briefly summed up as **"if $\mathcal{H}$ shatters some set C with $|C| = 2m$ we cannot expect to learn from a training set $S, |S| \leq m$"**. This theorem is useful to prove the following:

Theorem 1 *: let $\mathcal{H}$ be such that VCdim($\mathcal{H}$) $= \infty$. Then $\mathcal{H}$ is not PAC learnable.*

Proof:

$$\text{VCdim}(\mathcal{H}) = \infty \Rightarrow \forall S : |S| = m, \exists C \subset \mathcal{X} : |C| = 2m \text{ such that } |\mathcal{H}_C| = 2^{|C|}$$

Then by applying No Free Lunch theorem we can state that $\mathcal{H}$ is not learnable for any possible training set size.

Now we will introduce a framework useful to prove VCdim($\mathcal{H}$) $= d$. Its divided in 2 steps:

1. First we need to prove that $\exists C \subset \mathcal{X} : |C| = d$ that can be shattered by $\mathcal{H}$. This implies that VCdim($\mathcal{H}$) $\geq d$.

2. Then we must prove that $\forall C \subset \mathcal{X} : |C| = d+1$, C is not shattered by $\mathcal{H}$. This implies that VCdim($\mathcal{H}$) $< d+1$.

The combination of the previous 2 steps implies that VCdim($\mathcal{H}$) $= d$

Example: let $\mathcal{H}$ the family of indicator functions defined on an interval.

$$\mathcal{H} = \{h_{a,b} \mid a, b \in \mathbb{R}, a < b\}$$

$$h_{a,b}(x) = \mathbf{1}_{x \in [a,b]}(x)$$

Lets take $C_2 = a_1, a_2 : a_1 < a_2$ and $a_1, a_2 \in \mathbb{R}$, $\mathcal{H}$ can produce all possible labelings imposing:

$$\begin{cases} a = a_1 - 1, b = a_2 + 1 & \text{realizes } (1,1) \\ a = a_2 + 1, b = a_2 + 2 & \text{realizes } (0,0) \\ a = a_1 + \frac{(a_2 - a_1)}{2}, b = a_2 + 1 & \text{realizes } (0,1) \\ a = a_1 - 1, b = a_1 + \frac{(a_2 - a_1)}{2} & \text{realizes } (1,0) \end{cases} \tag{9}$$

This implies that VCdim($\mathcal{H}$) ≥ 2. Now, instead lets take $C_3 = a_1, a_2, a_3 : a_1 < a_2 < a_3$ and $a_1, a_2, a_3 \in \mathbb{R}$ The labeling $(1,0,1)$ cannot be realized in C_3 and therefore VCdim($\mathcal{H}$) < 3.
Combining these two results we can conclude that VCdim($\mathcal{H}$) $= 2$.

4.2 Detour on Linear Predictors

Definition 4.3 (Affine functions) *The affine functions class L_d is defined as*

$$L_d = \{h_{w,b} : w \in \mathbb{R}^d, b \in \mathbb{R}\}$$
$$h_{w,b} : \mathbb{R}^d \mapsto \mathbb{R},\ x \mapsto \langle w, x\rangle + b$$

if $h_{w,b}(x) = 0$ it is an equation of an affine hyperplane.

Now lets define the hypothesis class of linear predictors by constructing it as the following composition:

$$\phi \circ h_{w,b} \text{ where } \phi : \mathbb{R} \mapsto \{-1, +1\}$$

Where ϕ is the sign function.

in Machine Learning b is referred to as the **bias**, this leads to the **Homogenization trick**, where we impose:

$$x' = \begin{bmatrix} 1 \\ x_1 \\ \vdots \\ x_d \end{bmatrix}, w' = \begin{bmatrix} b \\ w_1 \\ \vdots \\ w_d \end{bmatrix},$$

in this way it holds that $\langle w, x\rangle + b = \langle w', x'\rangle$, this is a homogenous linear function. Practically we can use this trick in order to treat L_d as a homogenous linear function $h_w(x) = \langle w, x\rangle$

Halfspaces predictors These predictors have a domain $\mathcal{X} = \mathbb{R}^d$ and label set $\mathcal{Y} = \{-1, +1\}$, where the family of functions is:

$$\mathcal{HS}_d = \{x \mapsto sign(h_{w,b}(x)) : h_{w,b} \in L_d\}$$

We define the hyperplane $\pi : \langle w, x\rangle + b = w_1x_1 + \ldots + w_dx_d + b = 0$, and $|h_{w,b}| = d(x, \pi) \cdot ||w||$ where d is the distance of the point x from the plane π, more specifically $d(x, \pi) = \frac{|w_1x_1+\ldots+w_dx_d|}{\sqrt{w_1^2+\ldots+w_2^2}}$.
If the distance is zero, then the point lies on the plane. Moreover, the plane defines two halfspaces, points belonging to the first have positive values $h_{w,b}(x) > 0$ and for the second one negative $h_{w,b}(x) < 0$. This plane is also called **decision boundary**. We will see that this is a valid ERM predictor under realizability assumption, using Linear Programming. Now let's compute the $VCDim(\mathcal{HS}_d)$.

$\forall i \in [d]$, with e_i the i-th entry of the canonical basis. We claim that $\bar{0}, \bar{e_1}, \ldots \bar{e_d}$ can be shattered by $\mathcal{H}$.
Proof: let $y_0, \ldots, y_d : y_i \in -1, +1$ be an arbitrary labeling of $\bar{0}, \bar{e_1}, \ldots, \bar{e_d}$. Now we define $\hat{w} = [y_1 - b, \ldots, y_d - b]$ and $b = y_0$, then:

- $sign(\langle 0, w\rangle + b) = sign(b) = y_0$
- $sign(\langle e_i, w\rangle + b) = sign(y_i - b + b) = sign(y_i) = y_i$

In this way we can produce any arbitrary labeling, and so VCdim($\mathcal{H}$) $\geq d + 1$.
To prove that the VC dimension is $d + 1$ now we will upperbound it using Radon's Lemma:

Lemma 4.1 (Randon's Lemma) *Let $\mathcal{X} \subset \mathbb{R}^d, |\mathcal{X}| = d + 2$ Then there exists two disjoint subsets $\mathcal{X}_1, \mathcal{X}_2 \subset \mathcal{X} : \mathcal{X}_1 \cap \mathcal{X}_2 = \emptyset$ such that the convex hull $convh(\mathcal{X}_1) \cap convh(\mathcal{X}_2) \neq \emptyset$, where:*

$$convh(A) : \{\sum_{P \in A} \alpha_p P \mid \alpha_p \in \mathbb{R}^+, \forall P \in A,\ \underset{p \in A}{\alpha_p} = 1\}$$

For the upperbound lets take $\mathcal{X} = \{x_1, \ldots, x_{d+2}\} \subset \mathbb{R}^d$ and $\mathcal{X}_1, \mathcal{X}_2 \subset \mathcal{X} : \mathcal{X}_1 \cap \mathcal{X}_2 = \emptyset$ and label $\mathcal{X}_1$ with +1, and $\mathcal{X}_2$ with -1. Then, $convh(\mathcal{X}_1) \cap convh(\mathcal{X}_1) \neq \emptyset$ implies that the half space in the intersection can never be correct and, therefore, VCdim($\mathcal{HS}_d = d + 1$).

4.3 VCDim for finite classes

Given a finite $\mathcal{H} :\ |\mathcal{H}| < \infty$, then $\forall C \subset \mathcal{X}$. if $\mathcal{H}$ is restricted to C, than $|\mathcal{H}_C| \leq |\mathcal{H}|$. Consequently, if $|\mathcal{H}| < 2^{|C|}$ then C cannot be shattered. Therefore, it becomes clear that we can shatter C only if:

$$log_2(|\mathcal{H}|) \geq |C| \Rightarrow log_2(|\mathcal{H}|) \geq \text{VCdim}(\mathcal{H}).$$

Also, if $\mathcal{H}$ is finite than VCdim($\mathcal{H}$) is finite, because the VC dimension is built upon shattering, and shattering is based on the cardinality of subsets. So, if the cardinality of the subsets is finite also the VC dimension is going to be finite.
Proposition: *PAC Learnability of finite $\mathcal{H}$ follows from PAC Learnability of classes with finite VC dimension.*

Definition 4.4 (Fundamental Theorem of statistical learning) *$\mathcal{H}$ hypothesis class of functions from $\mathcal{X} \mapsto \{0,1\}$ and the loss l_{0-1}, then the following statements are equivalent:*

1. *$\mathcal{H}$ has the uniform convergence property.*
2. *Any ERM rule is a successful agnostic PAC learner for $\mathcal{H}$*
3. *$\mathcal{H}$ is agnostic PAC learnable*
4. *$\mathcal{H}$ is PAC learnable*
5. *Any ERM rule is a successful PAC learner for $\mathcal{H}$*
6. *$\mathcal{H}$ has finite VC dimension.*

Definition 4.5 (Quantitative Fundamental Theorem of statistical learning) *Let H be an hypothesis class of functions from $\mathcal{X} \mapsto \{0,1\}$ and let the loss function be the 0-1 loss. Assume that $VCdim(\mathcal{H}) < \infty$, than there exists absolute constants C_1, C_2 such that:*

- *$\mathcal{H}$ has the uniform convergence property.*

$$C_1 \frac{d + log(1/\delta)}{\epsilon^2} \leq m_{\mathcal{H}}^{UC}(\epsilon, \delta) \leq C_2 \frac{d + log(1/\delta)}{\epsilon^2}$$

- *$\mathcal{H}$ is agnostic PAC learnable*

$$C_1 \frac{d + log(1/\delta)}{\epsilon^2} \leq m_{\mathcal{H}}(\epsilon, \delta) \leq C_2 \frac{d + log(1/\delta)}{\epsilon^2}$$

- *$\mathcal{H}$ is PAC learnable*

$$C_1 \frac{d + log(1/\delta)}{\epsilon} \leq m_{\mathcal{H}}(\epsilon, \delta) \leq C_2 \frac{d \cdot log(1/\epsilon) + log(1/\delta)}{\epsilon}$$

small note: since $\epsilon < 1$, dividing by epsilon increases the values.

Lets now prove the point 6 of the fundamental theorem of statistical learning implies point 1, and so that a finite VC dimension implies uniform convergence. We will do it by introducing the following function

Definition 4.6 (Growth function) *Let H be a hypothesis class. The growth function $\tau_{\mathcal{H}} : \mathbb{N} \mapsto \mathbb{N}$ of H is defined as the maximum number of functions in the restriction $\mathcal{H}_C$ for subsets C of cardinality m :*

$$\tau_{\mathcal{H}}(m) = \max_{C \in \mathcal{X} : |C| = m} |\mathcal{H}_C|$$

Lemma 4.2 (Sauer's Lemma) *Let $\mathcal{H}$ be a class with $VCdim(\mathcal{H}) = d < \infty$, then:*

$$\forall m \in \mathbb{N} : \tau_{\mathcal{H}}(m) \leq \sum_{i=0}^{d} \binom{m}{i} \text{ with } \frac{m!}{i!(m-i)!} = \binom{m}{i}$$

in particular if m becomes greater than the VCdimension of the class, $\tau_{\mathcal{H}}$ increases polinomially and not exponentially. So:

$$\text{if } m > d + 1 \text{ then } \tau_{\mathcal{H}}(m) \leq (\frac{em}{d})^d$$

5 Lesson 5 - April 22

5.1 No Free Lunch Theorem

Definition 5.1 (No Free Lunch Theorem) *Let A be any learning algorithm for the task of binary classification with respect to the 0-1 loss over a domain $\mathcal{X}$. Let $m < |\mathcal{X}|/2$, representing a training set size. Then, there exists a distribution $\mathcal{D}$ over $\mathcal{X} \times \{0,1\}$ such that:*

- *There exists a function $f : \mathcal{X} \to \{0,1\}$ with $L_{\mathcal{D}}(f) = 0$.*
- *With probability of at least $\frac{1}{7}$ over the choice of $S \sim \mathcal{D}^m$ we have that $L_{\mathcal{D}}(A(S)) \geq \frac{1}{8}$, i.e.*

$$\mathbb{P}_{S\sim\mathcal{D}^m}\left[L_{\mathcal{D}}(A(S)) \geq \frac{1}{8}\right] \geq \frac{1}{7}$$

This theorem states that for every learner, there exists a task on which it fails, even though that task can be successfully learned by another learner. Indeed, a trivial successful learner in this case would be an ERM learner with the hypothesis class $\mathcal{H} = \{f\}$, or more generally, ERM with respect to any finite hypothesis class that contains f and whose size satisfies the equation $m \geq 8\log(\frac{7|\mathcal{H}|}{6})$.

5.2 Exercise 1 - Chapter 5

Text

Prove that $\mathbb{E}_{S\sim\mathcal{D}^m}[L_{\mathcal{D}}(A(S))] \geq \frac{1}{4} \implies \mathbb{P}_{S\sim\mathcal{D}^m}[L_{\mathcal{D}}(A(S)) \geq \frac{1}{8}] \geq \frac{1}{7}$.

Solution

The following corollary of Markov's inequality (3.3) is used in this solution: let Z be a random variable in [0, 1] with $\mathbb{E}[Z] = \mu$, then for any $\alpha \in (0,1)$, $\mathbb{P}[Z > \alpha] \geq \frac{\mu-\alpha}{1-\alpha}$.

Since $L_{\mathcal{D}}(A(S))$ is a random variable in [0, 1] with $\mu = \mathbb{E}_{S\sim\mathcal{D}^m}[L_{\mathcal{D}}(A(S))] \geq \frac{1}{4}$, it possible to directly apply the corollary above by choosing $\alpha = \frac{1}{8}$:

$$\mathbb{P}_{S\sim\mathcal{D}^m}\left[L_{\mathcal{D}}(A(S)) \geq \frac{1}{8}\right] \geq \frac{\mu-\alpha}{1-\alpha} = \frac{1/4-1/8}{1-1/8} = \frac{1}{7}$$

□

5.3 Exercise 1 - Chapter 3

Monotonicity of Sample Complexity: Let $\mathcal{H}$ be a hypothesis class for a binary classification task. Suppose that $\mathcal{H}$ is PAC learnable and its sample complexity is given by $m_{\mathcal{H}}$. Show that $m_{\mathcal{H}}$ is monotonically nonincreasing in each of its parameters. That is, show that given $\delta \in (0,1)$, and given $0 < \varepsilon_1 \leq \varepsilon_2 < 1$, we have that $m_{\mathcal{H}}(\varepsilon_1,\delta) \geq m_{\mathcal{H}}(\varepsilon_2,\delta)$. Similarly, show that given $\varepsilon \in (0,1)$, and given $0 < \delta_1 \leq \delta_2 < 1$, we have $m_{\mathcal{H}}(\varepsilon,\delta_1) \geq m_{\mathcal{H}}(\varepsilon,\delta_2)$.

Solution

Let $\mathcal{D}$ be an unknown distribution over $\mathcal{X}$, and let $f \in \mathcal{H}$ be the target hypothesis. Denote by A an algorithm which learns $\mathcal{H}$ with sample complexity $m_{\mathcal{H}}$. Fix some $\delta \in (0,1)$. Suppose that $0 < \varepsilon_1 \leq \varepsilon_2 < 1$. We need to show that $m_1 \overset{\text{def}}{=} m_{\mathcal{H}}(\varepsilon_1,\delta) \geq m_{\mathcal{H}}(\varepsilon_2,\delta) \overset{\text{def}}{=} m_2$. Given an i.i.d. training sequence of size $m \geq m_1$, we have that with probability at least $1-\delta$, A returns a hypothesis h such that

$$L_{\mathcal{D},f}(h) \leq \varepsilon_1 \leq \varepsilon_2$$

This follows from the definition of PAC learnability and from the fact that $m \geq m_1 \geq m_2$. Since with $m \geq m_1$ the condition on ε_2 is satisfied and we know that $m_{\mathcal{H}}$ is the minimum m satisfying the condition, then it follow that $m_2 \leq m_1$.

This proof showed that the sample complexity is monotonically decreasing in the accuracy parameter ε, the proof for the confidence parameter δ is analogous. □

5.4 Exercise 3 - Chapter 3

Text

Let $\mathcal{X} = \mathbb{R}^2$, $\mathcal{Y} = \{0, 1\}$, and let $\mathcal{H}$ be the class of concentric circles in the plane, that is, $\mathcal{H} = \{h_r : r \in \mathbb{R}_+\}$, where $h_r(x) = \mathbb{1}_{[\|x\| \leq r]}$. Prove that $\mathcal{H}$ is PAC learnable (assume realizability), and its sample complexity is bounded by

$$m_{\mathcal{H}}(\varepsilon, \delta) \leq \left\lceil \frac{\log(1/\delta)}{\varepsilon} \right\rceil$$

Solution

Consider the ERM algorithm A which given a training sequence $S = \{(x_i, y_i), i = 1, \ldots, m\}$, returns the hypothesis $\hat{h}$ corresponding to the "tightest" circle which contains all the positive instances, i.e. the circle with minimum radius ($\hat{r}$) which encircle all the positives. Since the text assumes realizability, then there is a circle h^* with zero generalization error, which is our target, and it has radius r^*.

Choose a scalar $\overline{r} \leq r^*$ such that $\mathcal{D}_{\mathcal{X}}(\{x : \overline{r} \leq \| x \| \leq r^*\}) = \varepsilon$ and define $E = \{x \in \mathbb{R}^2 : \overline{r} \leq \| x \| \leq r^*\}$, i.e. the probability of picking a random point from the distribution $\mathcal{D}$ which is inside the circular crown E is ε. The probability (over drawing S) that $L_{\mathcal{D}}(h_S) \geq \varepsilon$ is bounded above by the probability that no point in S belongs to E, because if a point in S belongs to E then $\hat{r} \geq \overline{r}$, so $L_{\mathcal{D}}(h_S) \leq \varepsilon$. Since the probability of a point being in E is ε, then the probability of no point in S being in E is $(1-\varepsilon)^m$, which is bounded above by $(1-\varepsilon)^m \leq e^{-\varepsilon m}$.

The claimed bound on $m_{\mathcal{H}}$ follow by requiring that $e^{-\varepsilon m} \leq \delta$. □

5.5 Exercise 5 - Chapter 3

Text

Let $\mathcal{X}$ be a domain and let $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_m$ be a sequence of distributions over $\mathcal{X}$. Let $\mathcal{H}$ be a finite class of binary classifiers over $\mathcal{X}$ and let $f \in \mathcal{H}$. Suppose we are getting a sample S of m examples, such that the instances are independent but are not identically distributed; the i-th instance is sampled from $\mathcal{D}_i$ and then y_i is set to be $f(\mathbf{x}_i)$. Let $\bar{\mathcal{D}}_m$ denote the average, that is, $\bar{\mathcal{D}}_m = (\mathcal{D}_1 + \cdots + \mathcal{D}_m)/m$.

Fix an accuracy parameter $\varepsilon \in (0, 1)$. Show that

$$\mathbb{P}[\exists h \in \mathcal{H} \text{ s.t. } L_{(\bar{\mathcal{D}}_m, f)}(h) > \varepsilon \text{ and } L_{(S,f)}(h) = 0] \leq |\mathcal{H}|e^{-\varepsilon m}$$

.

Solution

Fix some $h \in \mathcal{H}$ with $L_{(\bar{\mathcal{D}}_m, f)}(h) = \mathbb{P}_{X \sim \bar{\mathcal{D}}_m}[h(X) \neq f(X)] > \varepsilon$. Then

$$\underset{X \sim \bar{\mathcal{D}}_m}{\mathbb{P}}[h(X) = f(X)] = \frac{\mathbb{P}_{X \sim \mathcal{D}_1}[h(X) = f(X)] + \cdots + \mathbb{P}_{X \sim \mathcal{D}_m}[h(X) = f(X)]}{m} = 1 - \underset{X \sim \bar{\mathcal{D}}_m}{\mathbb{P}}[h(X) \neq f(X)] < 1 - \varepsilon$$

Since we want to compute $\mathbb{P}[\exists h \in \mathcal{H} \text{ s.t. } L_{(\bar{\mathcal{D}}_m, f)}(h) > \varepsilon \text{ and } L_{(S,f)}(h) = 0]$ and h is fixed such that the first condition is true, let's bound $\mathbb{P}[L_S(h) = 0]$:

$$\underset{S \sim \prod_{i=1}^m \mathcal{D}_i}{\mathbb{P}}[L_S(h) = 0] = \prod_{i=1}^m \underset{X \sim \mathcal{D}_i}{\mathbb{P}}[h(X) = f(X)] = \left(\left(\prod_{i=1}^m \underset{X \sim \mathcal{D}_i}{\mathbb{P}}[h(X) = f(X)] \right)^{\frac{1}{m}} \right)^m$$

$$\leq \left(\frac{\sum_{i=1}^m \mathbb{P}_{X \sim \mathcal{D}_i}[h(X) = f(X)]}{m} \right)^m < (1 - \varepsilon)^m \leq e^{-\varepsilon m}$$

The first inequality is given by the geometric-arithmetic mean inequality: $(\prod_{i=1}^m x_i)^{\frac{1}{m}} \leq \frac{\sum_{i=1}^m x_i}{n}$.

Since this result holds for one $h \in \mathcal{H}$, if we take into account all $h \in \mathcal{H}$ with $L_{(\bar{\mathcal{D}}_m, f)}(h) > \varepsilon$ and apply the union bound ($\mathbb{P}\left(\bigcup_i A_i\right) \leq \sum_i \mathbb{P}(A_i)$), we conclude that the probability that there exists some $h \in \mathcal{H}$ with $L_{(\bar{\mathcal{D}}_m, f)}(h) > \varepsilon$, which is consistent with S (i.e. that $\mathbb{P}[L_S(h) = 0]$) is at most $\sum_{h \in \mathcal{H}} e^{-\varepsilon m} = |\mathcal{H}|e^{-\varepsilon m}$. □

5.6 Exercise 7 - Chapter 3

Text

The Bayes optimal predictor: Show that for every probability distribution $\mathcal{D}$, the Bayes optimal predictor $f_{\mathcal{D}}$ is optimal, in the senses that for every classifier g from $\mathcal{X}$ to {0, 1}, $L_{\mathcal{D}}(f_{\mathcal{D}}) \leq L_{\mathcal{D}}(g)$.

Solution

Recall the Bayes optimal predictor: given any probability distribution $\mathcal{D}$ on $\mathcal{X}\times$ {0, 1}, the best label predicting function from $\mathcal{X}$ to {0, 1} is

$$f_{\mathcal{D}}(x) = \begin{cases} 1 \text{ if } \mathbb{P}[y=1|x] \geq \frac{1}{2} \\ 0 \text{ otherwise} \end{cases}$$

Now let $x \in \mathcal{X}$ and $\alpha_x = \mathbb{P}[y=1|x]$ (i.e. α_x is the conditional probability of a positive label given x). We have

$$\begin{aligned} \mathbb{P}[f_{\mathcal{D}}(X) \neq y|X=x] &= \mathbb{1}_{[\alpha_x \geq 1/2]} \cdot \mathbb{P}[Y=0|X=x] + \mathbb{1}_{[\alpha_x < 1/2]} \cdot \mathbb{P}[Y=1|X=x] \\ &= \mathbb{1}_{[\alpha_x \geq 1/2]} \cdot (1-\alpha_x) + \mathbb{1}_{[\alpha_x < 1/2]} \cdot \alpha_x \\ &= \min\{\alpha_x, 1-\alpha_x\} \end{aligned}$$

Let g be a classifier from $\mathcal{X}$ to {0, 1}. We have

$$\begin{aligned} \mathbb{P}[g(X) \neq y|X=x] &= \mathbb{P}[g(X)=0|X=x] \cdot \mathbb{P}[Y=1|X=x] + \mathbb{P}[g(X)=1|X=x] \cdot \mathbb{P}[Y=0|X=x] \\ &= \mathbb{P}[g(X)=0|X=x] \cdot \alpha_x + \mathbb{P}[g(X)=1|X=x] \cdot (1-\alpha_x) \\ &\geq \mathbb{P}[g(X)=0|X=x] \cdot \min\{\alpha_x, 1-\alpha_x\} + \mathbb{P}[g(X)=1|X=x] \cdot \min\{\alpha_x, 1-\alpha_x\} \\ &= \min\{\alpha_x, 1-\alpha_x\} = \mathbb{P}[f_{\mathcal{D}}(X) \neq y|X=x] \end{aligned}$$

Note that for the 0-1 loss

$$L_{\mathcal{D}}(f_{\mathcal{D}}) = \mathbb{E}_{(x,y)\sim\mathcal{D}}[\mathbb{1}_{[f_{\mathcal{D}}(x)\neq y]}] = \mathbb{E}_{x\sim\mathcal{D}_X}\left[\mathbb{E}_{y\sim\mathcal{D}_{Y|x}}\left[\mathbb{1}_{[f_{\mathcal{D}}(x)\neq y}|X=x]\right]\right] = \mathbb{E}_{x\sim\mathcal{D}_X}[\mathbb{P}[f_{\mathcal{D}}(X) \neq y|X=x]]$$

and analogously $L_{\mathcal{D}}(g) = \mathbb{E}_{x\sim\mathcal{D}_X}[\mathbb{P}[g_{\mathcal{D}}(X) \neq y|X=x]]$.

Since we found that $\mathbb{P}[f_{\mathcal{D}}(X) \neq y|X=x] \leq \mathbb{P}[g(X) \neq y|X=x]$, then $L_{\mathcal{D}}(f_{\mathcal{D}}) \leq L_{\mathcal{D}}(g)$. □

5.7 Exercise 1 - Chapter 4

Text

In this exercise, we show that the (ε, δ) requirement on the convergence of errors in our definition of PAC learning, is, in fact, quite close to a simpler looking requirement about averages (or expectations). Prove that the following two statements are equivalent for any learning algorithm A, any probability distribution $\mathcal{D}$, and any loss function whose range is [0, 1]:

1. For every $\varepsilon, \delta > 0$, there exists $m(\varepsilon, \delta)$ such that $\forall m \geq m(\varepsilon, \delta)$
$$\mathop{\mathbb{P}}_{S\sim\mathcal{D}^m}[L_{\mathcal{D}}(A(S)) > \varepsilon] < \delta$$
2. $$\lim_{m\to\infty} \mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}[L_{\mathcal{D}}(A(S))] = 0$$

Solution

- 1 ⇒ 2: Assume that $\forall\varepsilon, \delta \in (0,1)$, and $\forall\mathcal{D}$ over $\mathcal{X} \times \{0,1\}$, there $\exists m(\varepsilon, \delta) \in \mathbb{N}$ such that $\forall m \geq m(\varepsilon, \delta)$, $\mathop{\mathbb{P}}_{S\sim\mathcal{D}^m}[L_{\mathcal{D}}(A(S)) > \varepsilon] < \delta$.

  Then using the definition of expectation
$$\begin{aligned} \mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}[L_{\mathcal{D}}(A(S))] &\leq \mathop{\mathbb{P}}_{S\sim\mathcal{D}^m}[L_{\mathcal{D}}(A(S)) > \varepsilon] \cdot 1 + \mathop{\mathbb{P}}_{S\sim\mathcal{D}^m}[L_{\mathcal{D}}(A(S)) \leq \varepsilon] \cdot \varepsilon \\ &\leq \mathop{\mathbb{P}}_{S\sim\mathcal{D}^m}[L_{\mathcal{D}}(A(S)) > \varepsilon] + \varepsilon < \delta + \varepsilon \end{aligned}$$
  where the last inequality follows from the assumption. Now set $\delta = \varepsilon$, we have for every $\varepsilon > 0$ there exists $m(\varepsilon, \varepsilon)$ such that $\forall m \geq m(\varepsilon, \varepsilon)$, $\mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}[L_{\mathcal{D}}(A(S))] \leq 2\varepsilon$. By taking the limit of ε going to zero, then $m(\varepsilon, \varepsilon)$ goes to infinity, so we can write
$$\lim_{m\to\infty} \mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}[L_{\mathcal{D}}(A(S))] \leq \lim_{\varepsilon\to 0} 2\varepsilon = 0$$
- 2 ⇒ 1: Assume that $\lim_{m\to\infty} \mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}[L_{\mathcal{D}}(A(S))] = 0$. For every $\varepsilon, \delta \in (0,1)$ there exists some $m_0 \in \mathbb{N}$ such that for every $m \geq m_0$, $\mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}[L_{\mathcal{D}}(A(S))] \leq \varepsilon\delta$. By Markov's inequality (3.3),
$$\mathop{\mathbb{P}}_{S\sim\mathcal{D}^m}[L_{\mathcal{D}}(A(S)) > \varepsilon] \leq \frac{\mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}[L_{\mathcal{D}}(A(S))]}{\varepsilon} \leq \frac{\varepsilon\delta}{\varepsilon} = \delta$$

□

5.8 Exercise 2 - Chapter 4

Corollary 5.1 *Let $\mathcal{H}$ be a finite hypothesis class, Z a domain and $\ell : \mathcal{H} \times Z \to [0,1]$ a loss function. Then, $\mathcal{H}$ enjoys the uniform convergence property with sample complexity*

$$m_{\mathcal{H}}^{UC}(\varepsilon, \delta) \leq \left\lceil \frac{\log(2|\mathcal{H}|/\delta)}{2\varepsilon^2} \right\rceil$$

Furthermore, the class is agnostically PAC learnable using the ERM algorithm with sample complexity

$$m_{\mathcal{H}}(\varepsilon, \delta) \leq m_{\mathcal{H}}^{UC}(\varepsilon/2, \delta) \leq \left\lceil \frac{2\log(2|\mathcal{H}|/\delta)}{\varepsilon^2} \right\rceil$$

Text

Bounded loss functions: In the preceding Corollary (5.1) we assumed that the range of the loss function is [0,1]. Prove that if the range of the loss function is $[a, b]$ then the sample complexity satisfies

$$m_{\mathcal{H}}(\varepsilon, \delta) \leq m_{\mathcal{H}}^{UC}(\varepsilon/2, \delta) \leq \left\lceil \frac{2\log(2|\mathcal{H}|/\delta)(b-a)^2}{\varepsilon^2} \right\rceil$$

Solution

Applying the Hoeffding's inequality 3.5 to $L_S(h) = \frac{1}{m}\sum_{i=1}^{m} \ell(h, (x_i, y_i))$ yields:

$$\mathbb{P}_{S\sim\mathcal{D}^m}(|L_S(h) - \mathbb{E}[L_S(h)]| > \varepsilon) = \mathbb{P}_{S\sim\mathcal{D}^m}(|L_S(h) - L_D(h)]| > \varepsilon) \leq 2\exp\left(-\frac{2m\varepsilon^2}{(b-a)^2}\right)$$

We then use the union bound combined with this upper bound to obtain:

$$\mathbb{P}_{S\sim\mathcal{D}^m}(\exists h \in \mathcal{H} : |L_S(h) - L_D(h)| > \varepsilon) \leq \sum_{h\in\mathcal{H}} \mathbb{P}_{S\sim\mathcal{D}^m}(|L_S(h) - L_D(h)]| > \varepsilon) \leq 2|\mathcal{H}|\exp\left(-\frac{2m\varepsilon^2}{(b-a)^2}\right)$$

The desired bound on the sample complexity follow from requiring $2|\mathcal{H}|\exp(-\frac{2m\varepsilon^2}{(b-a)^2}) \leq \delta$ □

6 Lesson 6 - April 29

6.1 Exercise 1 - Chapter 6

Text

Show the following monotonicity property of VC-dimension: for every two hypotesis classes, if $\mathcal{H}' \subseteq \mathcal{H}$ then $\text{VCdim}(\mathcal{H}') \leq \text{VCdim}(\mathcal{H})$.

Solution

If $\mathcal{H}' \subseteq \mathcal{H}$ then $\mathcal{H}'_c \leq \mathcal{H}_c$ $\forall C = \{x_1, \dots x_m\} \subseteq \mathcal{X}$ (i.e., the domain of the functions in $\mathcal{H}$). In particular, if C is shattered by $\mathcal{H}'$ (i.e., $|\mathcal{H}'_C| = 2^{|C|}$) then is shattered also by $\mathcal{H}$ ($|\mathcal{H}_C = 2^{|C|}$ as well). Therefore, $\text{VCdim}(\mathcal{H}') \leq \text{VCdim}(\mathcal{H})$. □

6.2 Exercise 2 - Chapter 6

Text

Given some finite domain set $\mathcal{X}$, and a number $k \leq |\mathcal{X}|$, figure out the VC-dimension of each of the following classes (and prove your claims):

1. $\mathcal{H}^{\mathcal{X}}_{=k} = \{h \in \{0,1\}^{\mathcal{X}} : |\{x : h(x) = 1\}| = k$ (i.e., the set of all functions that assign the value 1 to exactly k elements of $\mathcal{X}$).
2. $\mathcal{H}_{\leq k} = \{h \in \{0,1\}^{\mathcal{X}} : |\{x : h(x) = 1\}| \leq k$ or $|\{x : h(x) = 0\}| \leq k$

Solution part 1

Assume that $\mathcal{H} = \mathcal{H}^{\mathcal{X}}_{=k}$. We have that each $h_i \in \mathcal{H}$ is a list of 0 and 1, for a total of $|\mathcal{X}|$ binary numbers with exactly k labels equal to 1, then $|\mathcal{X}| - k$ labels are equal to zero. We want to prove that $\text{VCdim}(\mathcal{H}^{\mathcal{X}}_{=k}) = \min\{k, |\mathcal{X}| - k\}$. First of all, we show that $\text{VCdim}(\mathcal{H}^{\mathcal{X}}_{=k}) \leq k$. Let $C \subseteq \mathcal{X} : |C| = k + 1$, then there is no $h \in \mathcal{H}$ such that $h(x) = 1$ $\forall x \in C$. In fact, by definition of the class of functions, you have only k elements labelled 1, so having $k + 1$ elements in your set you cannot find a function that assign the label 1 to each element. So, the class does not shatter any set of size $k + 1$. Then $\text{VCdim}(\mathcal{H}) \leq k$ □
Now, let $C \subseteq \mathcal{X} : |C| = |\mathcal{X}| - k + 1$. Since we have a constrain on the number of labels equal to 1 (they must be equal to k), then we have also a constrain on the number of labels equal to 0 (they must be equal to $|\mathcal{X}| - k$). Applying the same reasoning as before, $\mathcal{H}$ does not shatter C because there is no a function $h \in \mathcal{H}$ such that all the labels assigned to elements in C are equal to 0, therefore $\text{VCdim}(\mathcal{H}) \leq |\mathcal{X}| - k$ □
Using the first two claims, we can state that $\text{VCdim}(\mathcal{H}^{\mathcal{X}}_{=k}) \leq \min\{k, |\mathcal{X}| - k\}$ □

We show now that $\text{VCdim}(\mathcal{H}) \geq \min\{k, |\mathcal{X}| - k\}$. Let $C = \{x_1, \dots, x_m\} \subseteq \mathcal{X}$ with $m \leq \min\{k, |\mathcal{X}| - k\}$. Consider $(y_1, \dots, y_m) \in \{0,1\}^m$ the associated vector of labels and $s = \sum_{i=1}^m y_i$ is the number of labels equal to 1. Now let $D = \mathcal{X} \backslash C$ such that $|D| = k - s$, and let $h \in \mathcal{H}$ be such that

$$\begin{cases} h(x_i) = y_i \ \forall x_i \in C \\ h(x) = \mathbb{1}_{[D]} = \begin{cases} 1 \text{ if } x \in D \\ 0 \text{ otherwise} \end{cases} \end{cases}$$

We have that $h \in \mathcal{H}$ because we have assigned exactly k labels equal to 1. Furthermore, $\mathcal{H}$ shatters C because the labels assigned to C (i.e., the binary vector $y_1, \dots y_m$) are arbitrary. Then $\text{VCdim}(\mathcal{H} \geq \min\{k, |\mathcal{X}| - k\}$ □
Using these claims we have proven our statement.

Solution part 2

We impose $\mathcal{H} = \mathcal{H}_{\leq k}$. In this class, the number of labels 1 or the number of labels 0 cannot be greater than k. For example, if $|\mathcal{X}| = 10$ and $k = 3$, the number of labels equal to 1 must be ≤ 3 or ≥ 7. As you can guess, $\text{VCdim}(\mathcal{H}) = k$ because if we take a bigger set you cannot produce all the possible combinations of labels with the functions in $\mathcal{H}$. Now we prove this claim in two steps as before.
First of all we show that the VC-dimension is $\leq k$. Let $C \subseteq \mathcal{X} : |C| = k + 1$, then there is no $h \in \mathcal{H}$ such that $h(x) = 1$ $\forall x \in C$.
Now we show that the VC-dimension is $\geq k$. Let $C = \{x_1, \dots, x_m\} \subseteq \mathcal{X}$ with $m \leq k$. Let $(y_1, \dots, y_m) \in \{0,1\}^m$ then this can be obtained by some $h \in \mathcal{H}$ because you cannot have a number of labels equal to 1 or equal to 0 greater than k (because there are not enough elements in C to violate this constraint).
Then $\text{VCdim}(\mathcal{H}) = k$ □.

6.3 Exercise 3 - Chapter 6

Text

Let $\mathcal{X}$ be the boolean hypercube $\{0,1\}^n$. For a set $I \subseteq \{1,2,\dots,n\}$ we define a parity function h_I as follows. On a binary vector $x = (x_1, x_2, \dots, x_n) \in \{0,1\}^n$,

$$h_I(x) = \left(\sum_{i\in I} x_i\right) \bmod 2$$

(That is, h_I computes parity of bits in I). What is the VC-dimension of the class of all such parity functions

$$\mathcal{H}_{n\text{-parity}} = \{h_I : I \subseteq \{1,2,\dots,n\}\}$$

Solution

Let $\mathcal{H} = \mathcal{H}_{n\text{-parity}}$. As you can guess, $\mathrm{VCdim}(\mathcal{H}) = n$ because if the size of the subset is greater than n you cannot consider those ...
First of all, the cardinality of the class is $|\mathcal{H}| = 2^n$ because, for each entry in the n-dimensional vector, you can consider or not the value in order to compute the parity. Therefore $\mathrm{VCdim}(\mathcal{H}) \le \log_2(|\mathcal{H}|) = n$, because if $\mathcal{H}$ is finite, then $|\mathcal{H}_C| \le |\mathcal{H}|$ (with $C \subseteq \mathcal{X}$). Clearly, C cannot be shattered if $|\mathcal{H}| \le 2^{|C|}$ (because you need at least a function $h \in \mathcal{H}$ for each element in C). $\square$

Now, consider $\mathcal{F} = \{e_1, e_2, \dots, e_n\} \in \{0,1\}^n$ where e_i is the i-th vector of the canonical basis. Then, given $(y_1, \dots, y_n) \in \{0,1\}^n$ the labels of the vectors in $\mathcal{F}$, you can always find a set I to produce the desired labeling of $\mathcal{F}$, just putting $i \in I$ if we want that e_i is labeled with $y_i = 1$, or $i \notin I$ if you want $y_i = 0$. Then the VCdimension of the class is $\ge n$ $\square$ So, we conclude that $\mathrm{VCdim}(\mathcal{H}) = n$.

6.4 Exercise 5 - Chapter 6

Text

VCdimension of axis aligned rectangles in $\mathbb{R}^d$: Let $\mathcal{H}^d_{\text{rec}}$ be the class of axis aligned rectangles in $\mathbb{R}^d$. We have already seen that $\mathrm{VCdim}(\mathcal{H}^2_{\text{rec}}) = 4$. Prove that, in general, $\mathrm{VCdim}(\mathcal{H}^d_{\text{rec}}) = 2 \cdot d$.

Solution

Given $a_i \le b_i$ $\forall i \in [d] = \{1,2,\dots,d\}$, we have that

$$h_{a_1,b_1,\dots,a_d,b_d}(x_1,\dots,x_d) = \prod_{i=1}^{d} \mathbb{1}_{[x_i \in [a_i,b_i]]}$$

i.e., h define a rectangle $\mathbb{R}^d$ and assign 1 to those points which are contained in that rectangle. Consider $S = \{s_1, s_2, \dots, s_{2d}\}$ where $s_i \in \mathbb{R}^d$. Furthermore, we impose

$$s_i = \begin{cases} e_i \text{ if } i \in [d] \\ -e_{i-d} \text{ if } i > d \end{cases}$$

where e_i is the i-th vector of the canonical basis of the space $\mathbb{R}^d$. Now we show that exists a function $h \in \mathcal{H}$ for each possible combination of labeling $(y_1, \dots, y_{2d}) \in \{0,1\}^{2d}$ on S. You can choose

$$a_i = \begin{cases} -2 \text{ if } y_{i+d} = 1 \\ 0 \text{ otherwise} \end{cases} \qquad b_i = \begin{cases} 2 \text{ if } y_i = 1 \\ 0 \text{ otherwise} \end{cases} \qquad \forall i \in [d]$$

i.e., for each dimension of the space you choose a proper length of the rectangle along a specific dimension which guarantees that e_i is contained in it (to better understand, try to visualize this reasoning in $\mathbb{R}^2$). Then $\mathrm{VCdim}(\mathcal{H}) \ge 2d$ $\square$

Now, let $C \subset \mathbb{R}^d : |C| \ge 2d+1$. Then, $\exists x \in C$ such that $\forall j \in [d]$ we have that

- $\exists x' \in C$ with $x'_j \le x_j$
- $\exists x'' \in C$ with $x''_j \ge x_j$

The previous fact come from the Pigeon Hole Principle: if you have m slots and more than m pigeons, than in at least one slot there should be more than one pigeons. In our case, we have d dimensions, and more than d directions, then there should be two vectors with different values along a specific dimension. Therefore the labeling in which x is labeled with 0 and the rest of the points in C are labeled with 1 is not covered by any function $h \in \mathcal{H}$, because either x' or x'' are labeled with 1. Then the VC dimension must be less than $2d+1$ $\square$. We conclude that $\mathrm{VCdim}(\mathcal{H}) = 2d$.

6.5 Exercise 7 - Chapter 6

Text

We have shown that for a finite hypothesis class $\mathcal{H}$, $\text{VCdim}(\mathcal{H}) \leq \lfloor \log(|\mathcal{H}|) \rfloor$. However, this is just an upper bound. The VC-dimension of a class can be much lower than that.

1. Find an example of a class $\mathcal{H}$ of functions over the real interval $\mathcal{X} = [0, 1]$ such that $\mathcal{H}$ is infinite while its VC-dimension is equal to 1.
2. Give an example of a finite hypotesis class $\mathcal{H}$ over the domain $\mathcal{X} = [0, 1]$ where $\text{VCdim}(\mathcal{H}) = \lfloor \log(|\mathcal{H}|) \rfloor$.

Solution part 1

We take

$$\mathcal{H} = \{\mathbb{1}_{[x \geq t]} : t \in \mathbb{R}\}$$

where each class in $\mathcal{H}$ is defined as

$$h_t(x) = \begin{cases} 1 \text{ if } x \geq t \\ 0 \text{ otherwise} \end{cases} \qquad \forall x \in [0, 1]$$

If we take $C \subset \mathcal{X} : |C| = 1$, for example $C = \{0.5\}$ we have that $h_{0.4}$ labels it with 1, while $h_{0.6}$ labels it with 0, therefore all the possible labeling are produced, thus the VC dimension is at least 1.
But if we take $C \subset \mathcal{X} : |C| = 2$, and suppose that $C = \{c_1, c_2\}$ with $c_1 \leq c_2$ without loss of generality, than we cannot find a function $h_t \in \mathcal{H}$ such that $h_t(c_1) = 1$ and $h_t(c_2) = 0$. Then the VC dimension of the class must be less than 2. Thus, it is equal to 1. □

Solution part 2

Let

$$\mathcal{H} = \{h_1 = \mathbb{1}_{[x \leq 1]}, h_2 = \mathbb{1}_{[x \leq 0.5]}\} : |\mathcal{H}| = 2$$

Then we have that $\log_2(|\mathcal{H}|) = \log_2(2) = 1$. If we take $C = \{0.6\}$, we have that $h_1(0.6) = 1$, while $h_2(0.6) = 0$. So, all the possible labeling of C are produced by $\mathcal{H}$, therefore its VC dimension is at least 1 because $|C| = 1$. Now take $C = \{c_1, c_2\}$ with $c_1 \leq c_2$. Then a function $h_i \in \mathcal{H}$ which produces the labeling $h_i(c_1) = 0$ and $h_i(c_2) = 1$ does not exists. Therefore the VC dimension of this class must be less than $|C| = 2$. Thus, it is equal to $\log_2(|\mathcal{H}|) = 1$ □

6.6 Exercise 9 - Chapter 6

Text

Let $\mathcal{H}$ be the class of signed intervals, that is

$$\mathcal{H} = \{h_{a,b,s} : a \leq b, s \in \{-1, 1\}\}$$

where

$$h_{a,b,s}(x) = \begin{cases} s \text{ if } x \in [a, b] \\ -s \text{ if } x \notin [a, b] \end{cases}$$

with $a, b, x \in \mathbb{R}$. Calculate the VC-dimension of this class.

Solution

If we take $C = \{1, 2, 3\}$ we have that

1	2	3	a	b	s
-	-	-	0.5	3.5	1
-	-	+	2.5	3.5	1
-	+	-	1.5	2.5	1
-	+	+	1.5	3.5	1
+	-	-	0.5	1.5	1
+	-	+	1.5	2.5	-1
+	+	-	0.5	2.5	1
+	+	+	0.5	3.5	1

The above table shows that each possible combination of labeling can be produced by $\mathcal{H}$, then its VC dimension it is equal or greater than $|C| = 3$.
Let now $C = \{x_1, x_2, x_3, x_4\}$ with $x_1 \le x_2 \le x_3 \le x_4$ without loss of generality. Then the labeling $(y_1, y_2, y_3, y_4) = (-1, 1, -1, 1)$ cannot be obtained by any function $h \in \mathcal{H}$, i.e., a combination of the parameters a, b, s which produce that specific labeling does not exist. Then, the VC dimension must be less than $|C| = 4$. Thus, the VC dimension of the class is equal to 3.

6.7 Exercise 11 - Chapter 6

Text

VCdim of union: Let $\mathcal{H}_1, \dots, \mathcal{H}_r$ be hypotesis classes over some fixed domain set $\mathcal{X}$. Let $d = \max_i \text{VCdim}(\mathcal{H}_i)$ and assume for simplicity that $d \ge 3$.

1. Prove that
$$\text{VCdim}\left(\bigcup_{i=1}^{r} \mathcal{H}_i\right) \le 4d\log(2d) + 2\log(r)$$
Hint: Take a set of k examples and assume that they are shattered by the union class. Therefore, the union class can produce all the 2^k possible labelings on these examples. Use Sauer's lemma to show that the union class cannot produce more than rk^d labelings. Therefore $2^k < rk^d$.

2. Prove that for $r = 2$ it holds that
$$\text{VCdim}(\mathcal{H}_1 \cup \mathcal{H}_2) \le 2d + 1$$

Solution part 1

We assume without loss of generality that $\forall i \in [r]$, $\text{VCdim}(\mathcal{H}_i) = d \ge 3$. Let $\mathcal{H} = \bigcup_{i=1}^{r} \mathcal{H}_i$. Let $k \in [d]$ then it follows that $\tau_{\mathcal{H}}(k) = 2^k$, recall that $\tau_{\mathcal{H}} : \mathbb{N} \to \mathbb{N}$ is the growth function (4.6) defined as

$$\tau_{\mathcal{H}}(m) = \max_{C \subset \mathcal{X} : |C| = m} |\mathcal{H}_C|$$

Clearly, if $\text{VCdim}(\mathcal{H}) = d$ then $\forall m \le d$ we have $\tau_{\mathcal{H}}(m) = 2^m$.

By definition, since $\mathcal{H}$ is the union of $\mathcal{H}_i$

$$\tau_{\mathcal{H}}(k) \le \sum_{i=1}^{r} \tau_{\mathcal{H}_i}(k)$$

Then

$$\begin{aligned} 2^k = \tau_{\mathcal{H}}(k) \le \sum_{i=1}^{r} \tau_{\mathcal{H}_i}(k) \le r\left(\frac{em}{d}\right)^d = r\left(\frac{e}{d}\right)^d m^d & \quad \text{applying Sauer's lemma 4.2 to each } \tau_{\mathcal{H}_i}(k) \\ < rm^d & \quad \text{since } d \ge 3 > e, \text{ so } \left(\tfrac{e}{d}\right)^d < 1 \end{aligned}$$

We found that $2^k < rm^d$, which can be rewritten as $k < \log_2(rm^d) = \log_2(r) + d\log_2(m)$.

Given the following lemma:

Lemma 6.1 *Let $a \ge 1$ and $b > 0$. Then, $x \ge 4a\log(2a) + 2b \implies x \ge a\log(x) + b$.*

We can revert the implication, which becomes: $x < a\log(x) + b \implies x < 4a\log(2a) + 2b$. Taking $a = d \ge 3$ and $b = \log_2(r) > 0$, then $k < \log(r) + d\log(m) \implies k < 4d\log(2d) + 2\log(r)$ □

Solution part 2

By applying the previous result with $r = 2$, we obtain $\text{VCdim}(\mathcal{H}_1 \cup \mathcal{H}_2) \le 4d\log(2d) + 2$, which is weaker result with respect to what we want to prove. As before, we may assume without loss of generality that $\text{VCdim}(\mathcal{H}_1) = \text{VCdim}(\mathcal{H}_2) = d$. Let $\mathcal{H} = \mathcal{H}_1 \cup \mathcal{H}_2$. Let k be a positive integer such that $k \ge 2d + 2$. We show that $\tau_{\mathcal{H}}(k) < 2^k$ by applying the Sauer's lemma 4.2 to $\tau_{\mathcal{H}_1}(k)$ and $\tau_{\mathcal{H}_2}(k)$ and by using the binomial coefficient property $\binom{n}{j} = \binom{n}{n-j}$,

$$\begin{aligned}
\tau_{\mathcal{H}}(k) \leq \tau_{\mathcal{H}_1}(k) + \tau_{\mathcal{H}_2}(k) &\leq \sum_{i=0}^{d}\binom{k}{i} + \sum_{i=0}^{d}\binom{k}{i} = \sum_{i=0}^{d}\binom{k}{i} + \sum_{i=0}^{d}\binom{k}{k-i} \\
&= \sum_{i=0}^{d}\binom{k}{i} + \sum_{i=k-d}^{d}\binom{k}{i} \leq \sum_{i=0}^{d}\binom{k}{i} + \sum_{i=d+2}^{d}\binom{k}{i} \\
&< \sum_{i=0}^{d}\binom{k}{i} + \sum_{i=d+1}^{d}\binom{k}{i} = \sum_{i=0}^{k}\binom{k}{i} = 2^k
\end{aligned}$$

□

6.8 Exercise 10 - Chapter 6

Text

Let $\mathcal{H}$ be a class of functions from $\mathcal{X}$ to {0,1}.

1. Prove that if VCdim($\mathcal{H}$) $\geq d$, for any d, then for some probability distribution $\mathcal{D}$ over $\mathcal{X}\times$ {0,1}, for every sample size m,

$$\mathbb{E}_{S\sim\mathcal{D}^m}[L_{\mathcal{D}}(A(S))] \geq \min_{h\in\mathcal{H}} L_{\mathcal{D}}(h) + \frac{d-m}{2d}$$

Hint: Use Exercise 3 in Chapter 5.

2. Prove that for every $\mathcal{H}$ that is PAC learnable, VCdim($\mathcal{H}$) $< \infty$. (Note that this is the implication 3 → 6 in the Fundamental Theorem of statistical learning 4.4).

Solution part 1

From Exercise 3 in Chapter 5 we get the following result: Let A be a learning algorithm for the task of binary classification, let the training set size m be $m < \frac{|\mathcal{X}|}{k}$, then there exists a distribution $\mathcal{D}$ over $\mathcal{X} \times \{0,1\}$ such that:

- There exists a function $f : \mathcal{X} \to \{0,1\} with L_{\mathcal{D}}(f) = 0$;
- $\mathbb{E}_{S\sim\mathcal{D}^m}[L_{\mathcal{D}}(A(S))] \geq \frac{1}{2} - \frac{1}{2k} = \frac{k-1}{2k}$.

We may assume that $m < d$, since otherwise the statement we want to prove is meaningless. Let C be a shattered set of size d. We may assume without loss of generality that $\mathcal{X} = C$ (since we can always choose distributions which are concentrated on C). Note that $\mathcal{H}$ contains all the functions from C to {0,1}. According to the results of Exercise 3 in Chapter 5 for every alogrithm there exists a distribution $\mathcal{D}$ for which $\min_{h\in\mathcal{H}} L_{\mathcal{D}}(h) = 0$, but $\mathbb{E}[L_{\mathcal{D}}(A(S))] \geq \frac{k-1}{2k} = \frac{d-m}{2d}$, the last equality is obtained substituting $k = \frac{d}{m}$ □

Solution part 2

We prove that if VCdim($\mathcal{X}$)= ∞, then A, a learning algorithm, fails to PAC learn $\mathcal{H}$. In particular we want to show that $\mathbb{P}[L_{\mathcal{D}}(A(S)) > \varepsilon] < \delta$ doesn't hold. Choose $\varepsilon = \frac{1}{16}$ and $\delta = \frac{1}{14}$. For any $m \in \mathbb{N}$, there exists a shattered set of size $d = 2m$, since VCdim($\mathcal{X}$)= ∞. Then assuming PAC learnability and applying the above, we obtain that there exits a distribution $\mathcal{D}$ for which $\min_{h\in\mathcal{H}} L_{\mathcal{D}}(h) = 0$, but $\mathbb{E}[L_{\mathcal{D}}(A(S))] \geq \frac{d-m}{2d} = \frac{1}{4}$ by applying the result of the first part of the exercise with $d = 2m$. Given the following lemma:

Lemma 6.2 *Let Z be a random variable that takes values in [0,1]. Assume that $\mathbb{E}[Z] = \mu$. Then for any $a \in$ (0,1), $\mathbb{P}[Z > 1-a] \geq \frac{\mu-(1-a)}{a}$.*

This also implies that for every $a \in$ (0,1), $\mathbb{P}[Z > a] \geq \frac{\mu-a}{1-a} \geq \mu - a$.

Finally,

$$\mathbb{P}[L_{\mathcal{D}}(A(S)) > \frac{1}{8} > \varepsilon] \geq \frac{\mathbb{E}[L_{\mathcal{D}}(A(S))] - \frac{1}{8}}{1-\frac{1}{8}} \geq \frac{\frac{1}{4}+\frac{1}{8}}{\frac{7}{8}} = \frac{1}{7} > \delta$$

So we proved that $\mathbb{P}[L_{\mathcal{D}}(A(S)) > \varepsilon] > \delta$, in contradiction to the PAC learnability definition. □

6.9 Exercise 8 - Chapter 6

Text

It is often the case that the VC-dimension of a hypothesis class equals (or can be bounded above by) the number of parameters one needs to set in order to define each hypothesis in the class. For instance, if $\mathcal{H}$ is the class of axis aligned rectangles in $\mathbb{R}^d$, then $\text{VCdim}(\mathcal{H}) = 2d$, which is equal to the number of parameters used to define a rectangle in $\mathbb{R}^d$. Here is an example that shows that this is not always the case. We will see that a hypothesis class might be very complex and even not learnable, although it has a small number of parameters.

Consider the domain $\mathcal{X} = \mathbb{R}$, and the hypothesis class Let $\mathcal{H}$ be a class of functions from $\mathcal{X}$ to {0,1}.

$$\mathcal{H} = \{x \mapsto \lceil \sin(\theta x) \rceil : \theta \in \mathbb{R}\}$$

(here, we take $\lceil -1 \rceil = 0$). Prove that $\text{VCdim}(\mathcal{H}) = \infty$.

Hint: There is more than one way to prove the required result. One option is by applying the following lemma: if $0.x_1x_2x_3\ldots$, is the binary expansion of $x \in (0,1)$, then for any natural number m, $\lceil \sin(2^m \pi x) \rceil = (1 - x_m)$, provided that $\exists k \geq m$ such that $x_k = 1$.

Solution

To prove $\text{VCdim}(\mathcal{H}) = \infty$, we need to pick n points which are shattered by $\mathcal{H}$, for any n. To do so, we construct n points $x_1, \ldots, x_n \in (0,1)$ such that each x_i has 2^n bits after the comma in the binary expansion and the by taking the m-th bit for each x_i we obtain all possible labellings of $x_1, \ldots, x_n$ as m goes from 1 to 2^n. For example, with $n = 3$ we have

$$\begin{array}{rcccccccccccc} x_1 & = & 0 & . & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ x_2 & = & 0 & . & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ x_3 & = & 0 & . & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{array}$$

In general,

$$\begin{array}{rcccccccc} x_1 & = & 0 & . & 0 & 0 & \ldots & 1 & 1 \\ x_2 & = & 0 & . & 0 & 0 & \ldots & 1 & 1 \\ & & & & & & \vdots & & \\ x_{n-1} & = & 0 & . & 0 & 0 & \ldots & 1 & 1 \\ x_n & = & 0 & . & 0 & 1 & \ldots & 0 & 1 \end{array}$$

For example, to give the labeling 1 for all instances, we just pick $m = 1$, i.e. the fist bit column in the binary expansion, so $h(x) = \lceil \sin(2^1 \pi x) \rceil$, because given x_i^1 the first bit of x_i, it returns (see *Hint*) $(1 - x_i^1) = 1 - 0 = 1$, since the first bit is zero for every x_i. If we wish to label 1 $x_1, \ldots, x_{n-1}$ and 0 x_n, it is enough to choose $m = 2$, so $h(x) = \lceil \sin(2^2 \pi x) \rceil$.

We conclude that $x_1, \ldots, x_n$ can be given any labeling by some $h \in \mathcal{H}$, so it is shattered. This can be done for any n, so $\text{VCdim}(\mathcal{H}) = \infty$. □

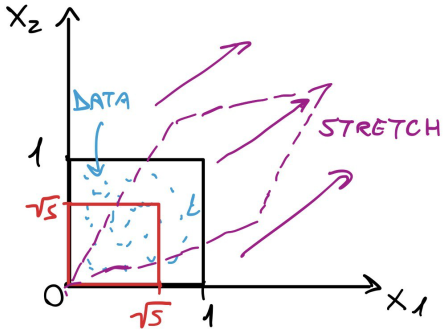


Figure 1: Hypercube in $\mathbb{R}^2$

7 Lesson 7 - May 6

7.1 The curse of dimensionality

In this section high dimensional spaces are presented as well as some consequences they imply.

7.1.1 In high dimensional spaces, nobody can hear you scream

Usually, when an algorithm requires distances between points in a space (for example $\mathbb{R}^d$) to be computed, a distance function must be defined. For example, one could use the Euclidean norm, namely:

$$d(x,y) = ||x-y||_2 = \sqrt{\sum_{i=1}^{d}(x_i - y_i)^2}$$

The distance computations are needed in multiple algorithms, for example in KNN, or in Linear Regression. However, computing distances with highly dimensional data will result in a tricky process since, as we will show, in such a case the concept of neighborhood between data points vanishes.

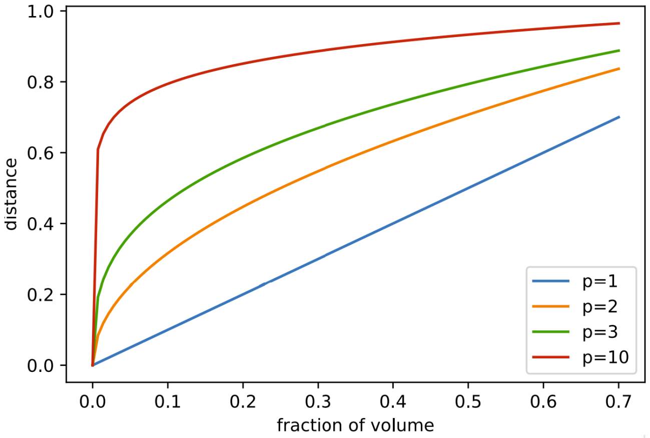
Given, for example, a square in $[0,1]^2 \subset \mathbb{R}^2$, its main diagonal is $\sqrt{2}$ long. If the dimension increases, then the cube in $[0,1]^3 \subset \mathbb{R}^3$ has $\sqrt{3}$ long diagonal. In general, given a d-dimensional hypercube in $[0,1]^d \subset \mathbb{R}^d$, then its main diagonal can be rewritten as:

$$||\underbrace{(1,\dots,1)}_{d\text{ times}}||_2 = \sqrt{d}$$

To show our point, lets assume data is $X \in [0,1]^p \subset \mathbb{R}^{m\times p}$, where $m < \infty$ is the number of samples. To capture a neighborhood which represents a fraction s of the hypercube volume, the edge length should be $s^{1/p}$, so that the volume of the hypercube containing the data is $V_s = (s^{1/p})^p = s$. An example in $\mathbb{R}^2$ is represented in Figure 1. In the $[0,1]^2$ square case, for example, in order to find a fraction s of the hypercube containing the data, the edge must have length $s^{1/2}$, so that the volume becomes $\sqrt{s}\cdot\sqrt{s} = s$.

The edge's length $s^{1/p}$ is a p-root function of a number, which is a continuous and derivable function everywhere except the origin. In fact, if $f(s) = s^{1/p}$ then $\lim_{s\to 0^+} f'(s) = +\infty$.

The following picture shows the behaviour of the $s^{1/p}$ function as the dimensionality p increases and, one may also notice that even with a low dimension the function grows very fast. Consequently, a very big portion of volume is required in order to get a low fraction of data.



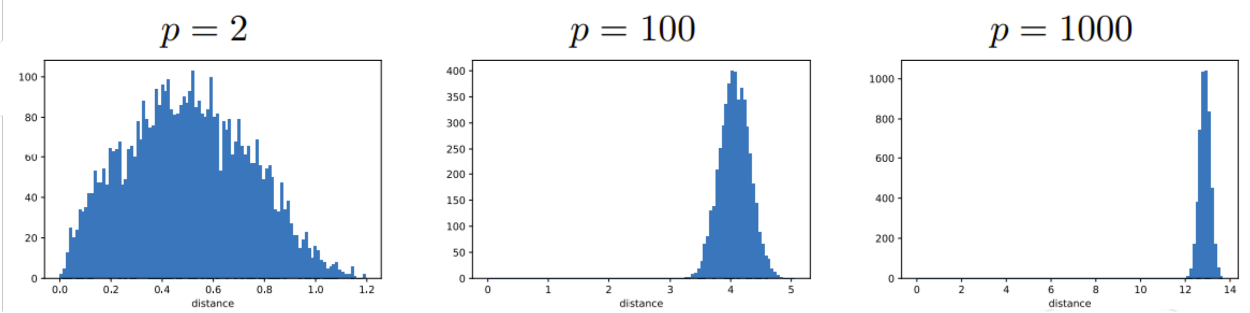
For example, when $p = 10$, 85% of the original hypercube is required to get one third of the data. This shows that all the points (which were located uniformly in the hypercube) are located in a single portion. To conclude, points in high dimensional spaces are isolated.

One could also imagine the hypercube as a mass having unitary volume in $\mathbb{R}^p$ and which is filled with m samples drawn accordingly to an uniform distribution. When the dimensionality p increases, the main diagonal of the hypercube stretches and the data concentrates on the main diagonal (see Figure 1 for a graphical representation in $\mathbb{R}^2$).

Another proof that points in high dimensional spaces are isolated, is that if you take an hypercube with edge length $s = 0.1$ inside the data space (the hypercube with edge length 1), its volume is $0.1^p \to 0$ as $p \to +\infty$, then the probability to capture a data point is very small. To conclude, in order to overcome the problem, the number of data points should increase exponentially with the dimensionality.

7.1.2 Nearest Neighbors

Now lets see how nearest neighbors behave when p increases. Let X, Y be two independent random variables, with uniform random distribution on $[0,1]^p$. The mean square distance $||X-Y||^2$ satisfies $\alpha = \mathbb{E}[||X-Y||^2] = p/6$ and $\beta = Std[||X-Y||^2] \approx 0.2 \cdot \sqrt{p}$. If you take the ratio $\beta/\alpha \approx 1/\sqrt{p}$, one can notice that it goes to zero as p increases.



From the distance distributions in the picture one can observe that, as p increases:

- The average of the distribution grows quickly
- The variance of the distribution increases slowly
- The minimal distance between two points increases
- All points get at the same distance from each other

Since all the distances between points become similar, it gets very difficult to define the Nearest Neighbor of a data point and the concept of ranking distances vanishes.

7.1.3 Concentration phenomena

Until now we have worked with cubes, now let us consider balls. The volume of a ball of radius r in a p-dimensional space is defined as

$$V_p(r) = r^p \cdot \frac{\pi^{p/2}}{\Gamma(p/2+1)}$$

where the Γ function is a generalization of the factorial. In particular, when its argument is an integer, the Γ is exactly the factorial. Then, what happens to $V_p(r)$ as the dimension p increases?

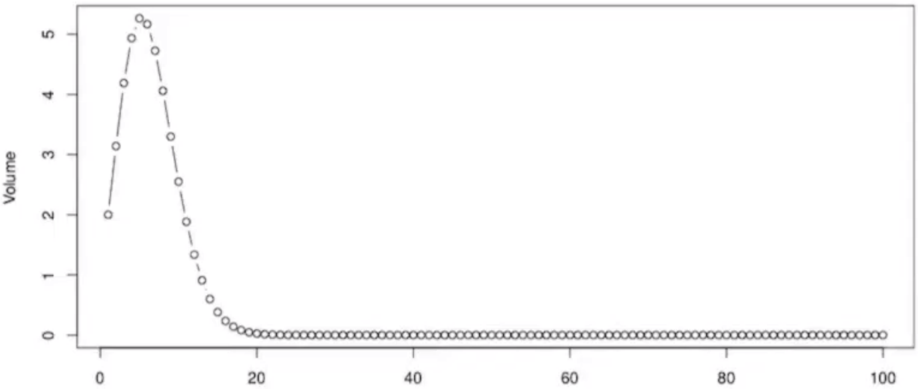


Fig. Volume of a ball of radius 1 regarding to the dimension p.

As one can see from the previous image, the volume of the ball decreases more than exponentially fast, until it gets negligible. Note that the volume corresponds to the probability of finding something inside the sphere. The consequence is that if you want to cover an hypercube of unitary volume $[0,1]^d$ with a union of n hyper-spheres of unitary radius, you need

$$n \geq \frac{1}{V_p} = \frac{\Gamma(p/2+1)}{\pi^{p/2}} \to (\frac{p}{2\pi e})^{p/2}\sqrt{p\pi}$$

So clearly $n \to \infty$ because $V_p \to 0$ as p grows. For example, when $p = 100$ we have $n = 42 \cdot 10^{39}$.

We now define the surface of a d-dimensional hyper-sphere of radius r as

$$S_p(r) = \frac{2\pi^{p/2}}{\Gamma(p/2)} \cdot r^{p-1}$$

Two interesting facts arise when looking at the ratio between the volume and the surface:

1. $\lim_{p\to+\infty} V_p(r) = \lim_{p\to+\infty} S_p(r) = 0 \; \forall r > 0$
2. $\lim_{p\to+\infty} \frac{V_p(r)}{S_p(r)} = \lim_{p\to+\infty} \frac{\Gamma(p/2)}{2\Gamma(p/2+1)} r = 0$

Moral: **the volume goes to zero faster than the surface**. This implies that, when the dimension increases, all points are flattened along the surface. In fact, lets study the probability that an uniform variable on the unit sphere belongs to the shell (circular crown) between the spheres of radius 0.9 and 1, specifically $P(X \in S_p(0.9)) = 1 - 0.9^p \to 1$ as $p \to +\infty$. Then, the probability of picking a uniform random point on the shell tends to 1.

Now consider $X_1, \dots, X_n$ i.i.d random variables in dimension p, with uniform distribution on the unit ball. The median distance from the origin to the closest data point is given by

$$med(p,n) = \left(1 - \frac{1}{2^{1/n}}\right)^{\frac{1}{p}}$$

Then, for $n = 500$ and $p = 10$ we have $med = 0.52$, which means that most data points are more closer to the surface rather than to the center of the hyper-sphere.

In general, samples are closer to the boundary of the sample space, which makes predictions much more difficult to be performed. Indeed, near the edges of the training samples, one must extrapolate from neighboring sample points rather than interpolate between them. This tells us that problems which arise with high dimensional data cannot be solved with smart algorithms or powerful machines.

Example: Assume n data points are independently sampled from an uniform distribution on $[-1,1]^p$. Suppose one wants to estimate $e^{-||x||^2/8}$ in 0 from the data and the chosen estimator is the observed value in x_i, hence the nearest neighbor of 0. For $n = 1000$ samples and $p = 10$, the probability that this neighbor is located at a distance larger than 0.5 from 0 is around 0.99. So it is impossible to estimate the real zero value.

7.1.4 What happens to the multivariate Gaussian distribution?

Most of the mass of a Gaussian distribution is located in areas where the density is extremely small compared to its maximum value. It means that most of the probability goes in the tail of the Gaussian, far away from its expected value. Consequently, one can no longer define the concept of rare events using the distance from the average value.

7.1.5 Surprising asymptotic properties for covariance matrices with highly dimensional data

Sample covariance matrices appear everywhere in statistics, for example in classification with Gaussian mixture models, but also in principal component analysis (PCA) and in linear regression with least squares.

$X \in \mathbb{R}^{m\times d}$, where each row is a sample, is the **data matrix**. The **average vector** is defined as

$$\mu = (\frac{1}{m}\sum_{j=1}^{m} x_{j1}, \ldots, \frac{1}{m}\sum_{j=1}^{m} x_{jd}) = (\mu_1, \ldots, \mu_d) \in \mathbb{R}^d$$

By centering the data, i.e. by subtracting the average vector to each row of X, the **centered data matrix** $\overline{X}$ is obtained. Then, the **sample covariance matrix** can be computed as $\hat{\Sigma}_X = \frac{1}{m}\overline{X}^T\overline{X} \in \mathbb{R}^{d\times d}$. Here some problems may arise:

- Often is necessary to invert Σ_X.
- If m is not large enough, the estimates of $\hat{\Sigma}_X$ are ill-conditioned or singular.
- Sometimes it is necessary to estimate the eigenvalues of Σ_X (in PCA for example).

Note that if we compute $\overline{X}\overline{X}^T \in \mathbb{R}^{m\times m}$, this matrix has the same rank as $\hat{\Sigma}_X$ and the same eigenvalues. Then, if $m < d$ it happens that the sample covariance matrix has a rank$(\hat{\Sigma}_X) \leq m$, so we want $m > d$ to get a full rank and invertible $\hat{\Sigma}_X$.

Context: $x_1, \ldots, x_m \in \mathbb{R}^d$ are i.i.d. samples from a gaussian multivariate distribution $\mathcal{N}_d(\bar{0}, \Sigma_d)$, where $\Sigma_d \in \mathbb{R}^{d\times d}$ is the covariance matrix, i.e., the entry at row i, column j is the value

$$Cov(X_i, X_j) = \mathbb{E}[(X_i - \mathbb{E}[X_i])(X_j - \mathbb{E}[X_j])]$$

Then, the maximum likelihood estimator for Σ_d is the sample covariance matrix

$$\hat{\Sigma}_X = \frac{1}{m}\sum_{k=1}^{m} x_k x_k^T \approx \Sigma_X$$

If d is fixed and $m \to +\infty$, then, for the strong law of large numbers, $||\hat{\Sigma}_d - \Sigma_d|| \to 0$ almost surely (i.e., with probability 1) for any matrix norm. However, if both d and m increase at the same rate, the spectral norm does not converge.

Let's consider $\Sigma_X = I_d$, i.e., all variables are independent, and $d > m$. Again, given that d and m increase with the same rate:

$$||\hat{\Sigma}_X - \Sigma_X||_\infty = \max_{i,j} |\hat{\sigma}_{i,j} - \sigma_{i,j}| \to 0$$

However the convergence in spectral norm is lost, since:

$$rank(X) \leq d \implies \lambda_{min}(\hat{\Sigma}_X) = 0 < 1 = \lambda_{min}(\Sigma_X)$$

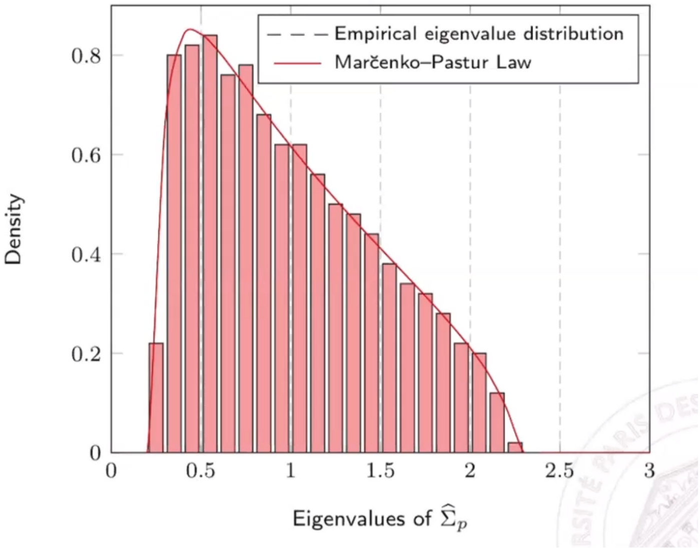
since the identity matrix I_d has all eigenvalues equal to 1. To conclude, the spectral norm of the sample covariance matrix (i.e., the set of its eigenvalues) does not converge to the one of the covariance matrix, and the sample covariance matrix cannot be used to estimate the spectrum of the real covariance matrix.

Definition 7.1 (Marcenko-Pastur Theorem) *Given $m, d \to \infty$ with $d/m \to c > 0$ (same order of infinite except than for a constant c), then*

$$\frac{1}{d}\sum_{k=1}^{d} \delta_{\lambda_k(\hat{\Sigma}_X)} \to \mu \text{ weakly}$$

*with μ **the Marcenko-Pastur law** of parameter $c = d/m$, which satisfies*

- $\mu(\{0\}) = \max(0, 1 - c^{-1})$
- *on $(0, \infty)$, μ has a continuous density supported on $[(1-\sqrt{c})^2, (1+\sqrt{c})^2]$*



In the figure, the distribution of the eigenvalues of $\hat{\Sigma}_X$ for $d = 500, m = 2000, \Sigma_X = I_d$ is pictured. The red line is the distribution predicted using the Marcenko-Pastur law. Then, if some of the eigenvalues are out of the distribution, it means that in those direction it is happening something interesting and there is some information to be detected.

7.1.6 Classical ways to avoid the curse of dimensionality

- Regularize the unstable parameter estimates until they are correctly estimated.
- Make more restrictive assumptions on the model in order to get a lower number of parameters to be estimated.
- Dimensionality reduction

7.2 Principal Component Analysis

The main idea of principal component analysis (PCA) is to reduce the dimensionality of a data set consisting of many variables. PCA is a feature reduction (or feature extraction) mechanism, that helps in handling high-dimensional data with more features than is convenient to interpret.

A d-dimensional normal distribution with mean vector $\bar{0}$ and covariance matrix Σ has the following probability density function (PDF)

$$f(x) = \frac{1}{\sqrt{(2\pi)^d|\Sigma|}} \cdot \exp\{-\frac{1}{2}x^T\Sigma^{-1}x\},\ x \in \mathbb{R}^d$$

where $|\Sigma| = det(\Sigma)$, $\Sigma^T = \Sigma \in \mathbb{R}^{d\times d}$ (symmetric). Now let's look at the exponent, here are some observations:

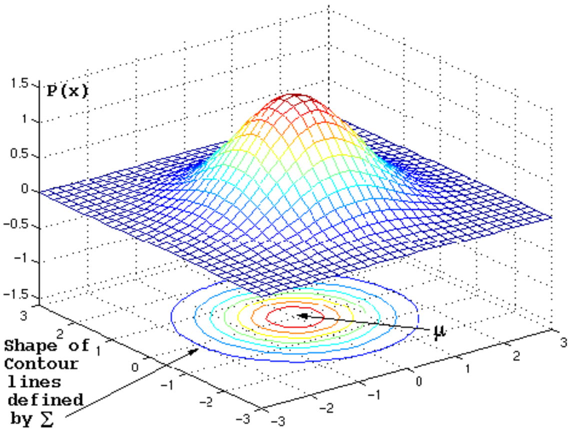
1. The equation x^TAx can be seen as a generalization of the scalar product. Suppose $A = I_d$, then $x^TAx = x^Tx =< x, x >= ||x||_2^2$: so, one can think the matrix A as a way of measuring the angle between vectors.
2. The matrix Σ is symmetric positive definite, i.e., $\forall x \neq \bar{0}\ x^T\Sigma x > 0$ and its eigenvalues are all strictly positive reals.
3. $\exists U \in \mathbb{R}^{d\times d} : U^TU = UU^T = I_d$ (i.e., U is orthonormal) such that $\exists \Lambda = diag(\lambda_1, \dots, \lambda_d)$ with $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_d > 0$. Then, $\Sigma = U^T\Lambda U \iff U\Sigma U^T = \Lambda$, i.e., there exists an orthonormal basis U which diagonalizes the covariance matrix. Its existence comes from the symmetry of Σ, while the fact that the eigenvalues are positives comes from its positive definiteness.
4. The term $x^T\Sigma^{-1}x = c$ is the representation of the contours of the normal distribution at level c, hence an ellipsoid. In fact, considering $c = 1$ one might have the following change of basis:

$$y = Ux \iff y^T = x^TU^T \iff x = U^Ty \iff x^T = y^TU$$

then, if substituting the new basis and the result of point (3):

$$x^T\Sigma^{-1}x = 1 \iff y^TU\Sigma^{-1}U^Ty = y^T\Lambda y = \underbrace{\lambda_1y_1^2 + \lambda_2y_2^2 + \dots + \lambda_dy_d^2 = 1}_{\text{ellipsoid of axes } \lambda_i}$$

The last term is indeed the equation of an ellipsoid.



Let $\Sigma^{-1} = BB^T$, then the ellipsoid can be viewed as the linear transformation of d-dimensional unit sphere via matrix B, since the equation $x^T\Sigma^{-1}x = (x^TB)(B^Tx) = 1$, B moves the points in a space where they have norm one. Moreover, the principal axes of the ellipsoid can be found via a singular value decomposition (SVD) of B (or Σ).

Suppose that an SVD of B is $B = UDV^T$ (and therefore the SVD of Σ is UD^2U^T), then:

- The columns of the matrix UD correspond to the principal axes of the ellipsoid.
- The relative magnitudes of the axes are given by the elements of the diagonal matrix D, i.e., the eigenvalues.

If some of the magnitudes are small compared to the others (eigengap), a reduction in the dimension of the space may be achieved by projecting each data point $x \in \mathbb{R}^d$ onto the subspace by the main (say $k << d$) columns of U, the so-called **principal components**.

Suppose, without loss of generality, that the first k principal components are given by the first columns of U, and let U_k be the corresponding $d \times k$ matrix. The vector $x = x_1e_1 + \cdots + x_de_d$ in the canonical basis is represented by the d-dimensional vector $[x_1, \ldots, x_d]^T$, while in the orthonormal basis $\{u_1, \ldots, u_d\}$ formed by the columns of the matrix U the representation is given by U^Tx. Similarly, the projection of any data point x onto the subspace spanned by the first k principal vectors is represented by the k-dimensional vector $U_k^Tx \in \mathbb{R}^k$ w.r.t. the orthonormal basis formed by the columns of U_k.

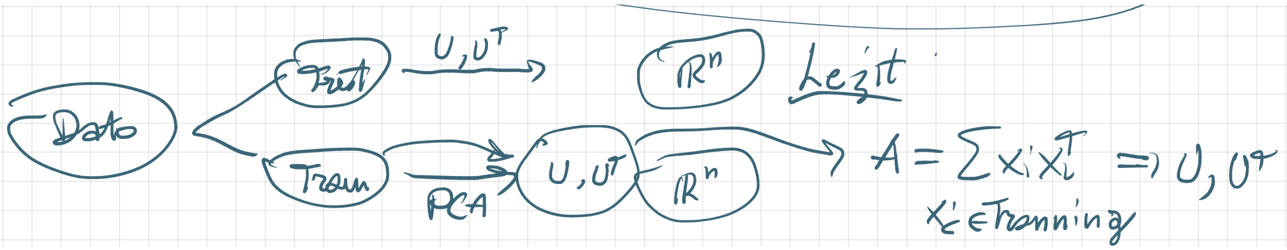
So, the idea is that if a point x lies close to its projection $U_kU_k^Tx \in \mathbb{R}^d$, we may represent it via k numbers instead of d, using the combined features given by the k principal components.

7.2.1 Some Practical notes on PCA

- Before to apply PCA, the matrix X should be centered by subtracting the column mean to every column:

$$x'_{ij} = x_{ij} - \bar{x_j}, \;\; \bar{x_j} = \frac{1}{n}\sum_{i=1}^{n} x_{ij}$$

- When the holdout technique is used, i.e., the dataset is splitted into a Training set and a Test set, PCA should be performed on the Training set and then it is applied to the Test set, for example before performing classification.



8 Lesson 8 - May 13

8.1 Recap and extensions of PCA

The objective of Principal Component Analysis is to reduce the dimensionality of the data while preserving the maximal possible variance. Formally, if the data are in a d dimensional real space $\mathbb{R}^d$ we want to find a matrix $W \in \mathbb{R}^{n\times d}$ which brings data from the original space to a new space $\mathbb{R}^n$ with $n < d$ and another matrix $U \in \mathbb{R}^{d\times n}$ which constructs an approximation of the original data. Thus, W is called the **compression** or **encoding matrix**, while U is the **decoding matrix**, and they both are linear transformations (i.e., a map between two vector spaces, in this case $\mathbb{R}^d$ and $\mathbb{R}^n$, that preserves the operations of vector addition and scalar multiplication).

Another point of view is to interpret the PCA as a Multi-Layer Perceptron (MLP) with biases equal to 0 and an identity activation function:

- The input layer is composed of d neurons, where d is the dimension of the source vector space.
- The weight matrix between the first and the second layer corresponds to the matrix W (the encoder) introduced before.
- The second layer is composed of n neurons, where $n < d$ is the dimension of the destination vector space.
- The weight matrix between the second and the third layer corresponds to the matrix U (the decoder) introduced before.
- The third and last layer is composed again of d neurons and it also corresponds to the output layer.

The aim is to minimize the difference between the output layer and the input one for each sample in the data. Given m samples $\{x_1, \ldots, x_m\} \in \mathbb{R}^d$ the least square problem for minimization is reported in equation 10.

$$\operatorname*{arg\,min}_{W\in\mathbb{R}^{n\times d},\ U\in\mathbb{R}^{d\times n}} \sum_{i=1}^{m} ||x_i - UWx_i||_2^2 \tag{10}$$

Of course we can apply gradient descent (since it is a convex problem, a sum of squares) and all the techniques for neural networks, but theorem 2 provide the solution.

Theorem 2 *Let $A = \sum_{i=1}^m x_i \cdot x_i^T = X^T X$ where $X \in R^{m\times d}$ is the centered data matrix in which each row is a data sample with the mean subtracted. Then, the solution to the problem 10 is given by $U = (u_1, \ldots, u_n)$ and $W = U^T$ where u_i is the i-th leading eigenvector of A (i.e., the eigenvector corresponding to the i-th highest in modulus eigenvalue of A).*

Corollary 8.1 *The residual with the optimal solution provided in the previous theorem is the following.*

$$\min_{W\in\mathbb{R}^{n\times d},\ U\in\mathbb{R}^{d\times n}} \sum_{i=1}^{m} ||x_i - UWx_i||_2^2 = \sum_{i=n+1}^{d} \lambda_i(A) \tag{11}$$

where $\lambda_i(A)$ is the i-th highest in modulus eigenvalue of the matrix A.

Furthermore, note that the eigenvectors $\{u_1, \ldots, u_n\}$ are taken orthonormal (i.e., orthogonal to each other and all with norm $||u_i||_2 = 1\ \forall i \in [n]$). In this way the scalar product becomes $< u_i, u_j >= \delta_{ij}$, where $\delta_{ij} = 1$ if and only if $i = j$, 0 otherwise.

8.2 An interpretation of PCA as Variance Maximization

Let $\{x_1, \ldots, x_m\} \in \mathbb{R}^d$, let x be a random vector distributed according to the uniform distribution over $x_1, \ldots, x_m$, assume $\mathbb{E}[x] = 0$.

8.2.1 Part 1

Find $w \in \mathbb{R}^d$ such that $< x, w >$ (which is a random variable since x is a random vector) has maximal variance, under the condition that w is a unitary vector, $||w||_2 = 1$.

More formally, we want to solve problem 12.

$$\operatorname*{arg\,max}_{w:||w||=1} \mathrm{Var}[< w, x >] = \operatorname*{arg\,max}_{w:||w||=1} \frac{1}{m} \cdot \sum_{i=1}^{m} (< w, x_i >)^2 \tag{12}$$

Show that the solution is the first principal component of $\{x_1, \ldots, x_m\}$.

$\forall w \in \mathbb{R}^d$, $||w|| = 1$ we have that equation 13 holds.

$$(< w, x_i >)^2 = \text{trace}(w^T \cdot x_i \cdot x_i^T \cdot w) \ \forall i \in [m] \tag{13}$$

where the trace is the sum of all components in the main diagonal of a square matrix. Proof is omitted since it is not necessary for the exam. Then:

$$||x - UU^T x||^2 = ||x||^2 - x^T UU^T x = ||x||^2 - \text{trace}(U^T x x^T U)$$

N.B: $||UU^T x||^2 = (x^T UU^T)(UU^T x) = (x^T U \underbrace{U^T U}_{\text{Identity}} U^T x) = x^T UU^T x.$

So one can reformulate the problem of minimizing $||x - UU^T x||^2$ with maximizing $\text{trace}(U^T x x^T U)$, where $W = U^T$ is the encoder, U is the decoder (in the PCA setting). Since it is the same problem of PCA, the solution is the first eigenvector of the sample covariance matrix $\hat{\Sigma}_X = X^T X$. □

8.2.2 Part 2

Assume w_1 to be the first principal component found in Part 1. Find $w_2 \in \mathbb{R}^d$ with unitary norm that maximizes the variance of $< w_2, x >$ and it is also uncorrelated to $< w_1, x >$ (i.e., w_2 is orthogonal to w_1). Note that the variance is a quadratic formula, hence it is affected by the size of the vector (if the vector has a large length, then the variance increases). However our focus is on the direction in which the variance is maximal, therefore, the norm of w_2 is imposed to be unitary. Formally

$$\begin{aligned} w^* = & \underset{w:||w||=1, \mathbb{E}[(<w_1,x>)(<w,x>)]=0}{\arg\max} Var[< w, x >] = \\ = & \underset{w:||w||=1, <w,w_1>=0}{\arg\max} \frac{1}{m} \sum_{i=1}^{m} (< w, x_i >)^2 \\ = & \underset{w:||w||=1, <w,w_1>=0}{\arg\max} \text{trace}(w^T (\frac{1}{m} \sum_{i=1}^{m} x_i x_i^T) w) \ \text{ from equation 13} \end{aligned}$$

Now consider the case with $n = 2$, in PCA we find $W \in \mathbb{R}^{d \times 2}$ the encoding matrix, such that $W^T \frac{1}{m}(\sum_{i=1}^m x_i x_i^T) W$ is the variance kept in the dimensionality reduction. By comparing this value with the ones obtained above (i.e., w_1 from the first part and $w_2 = w^*$ obtained in this section) the following result holds: denote with w_1, w_2 the first two columns of W, then

$$\text{trace}(W^T \frac{1}{m} (\sum_{i=1}^{m} x_i x_i^T) W) = w_1^T (\frac{1}{m} \sum_{i=1}^{m} x_i x_i^T) w_1 + w_2^T (\frac{1}{m} \sum_{i=1}^{m} x_i x_i^T) w_2$$

But w^* and w_1 are orthonormal, then we have

$$\begin{aligned} & w_1^T (\frac{1}{m} \sum_{i=1}^{m} x_i x_i^T) w_1 + w_2^T (\frac{1}{m} \sum_{i=1}^{m} x_i x_i^T) w_2 \\ \geq & w_1^T (\frac{1}{m} \sum_{i=1}^{m} x_i x_i^T) w_1 + w^{*T} (\frac{1}{m} \sum_{i=1}^{m} x_i x_i^T) w^{*T} \end{aligned}$$

since the variance given by w_2 is greater or equal than the optimum one given by w^*. But, by definition of PCA, the left-hand term in the inequality is a minimum, consequently the two terms must be equal. Thus $w_2 = w^*$, i.e., the optimal value is the second principal component. □

8.2.3 Another formulation

Find $w_1, \ldots, w_n \in \mathbb{R}^d$ such that

$$\begin{aligned} & w_1 \in \underset{||w||=1}{\arg\max} Var[< w, x >] \\ & w_2 \in \underset{||w||=1, <w,w_1>=0}{\arg\max} Var[< w, x >] \\ & \vdots \\ & w_n \in \underset{||w||=1, <w,w_i>=0}{\arg\max} Var[< w, x >] \ \forall i \in [n-1] \end{aligned}$$

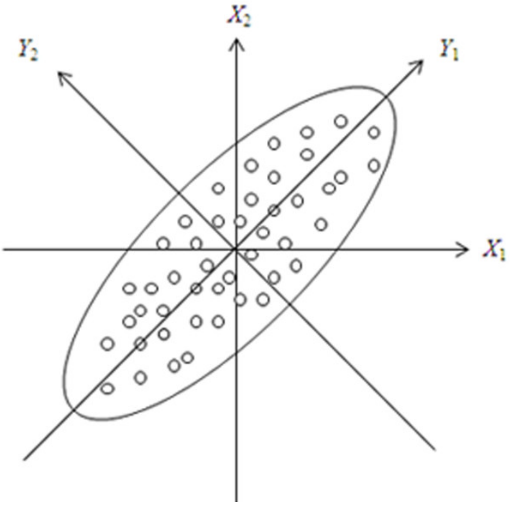


Figure 2: PCA geometric interpretation in two dimensions

These correspond to $u_1, \dots, u_n$ the principal components.

PCA also provides information about the data distribution, since it identifies the orthogonal directions along which our data has maximal variance in decreasing order. It can be seen that given the matrix $A = X^T X$, where the rows of $X \in \mathbb{R}^{m\times d}$ are the samples, and an arbitrary point $v \in \mathbb{R}^d$ in the same space of the data, $v^T(\sum_{i=1}^m x_i x_i^T)v = 1$ is an ellipsoid centered in $\mu_X = \frac{1}{m}\sum_{i=1}^m x_i$ (the sample mean) and whose axes are parallel to the principal components.

By diagonalizing and centering the matrix A, the following equation holds:

$$\lambda_1 x_1^2 + \lambda_2 x_2^2 + \cdots + \lambda_d x_d^2 = 1$$

So the eigenvalues (or the singular values) give the length of the semi-axis of the ellipsoid, while the eigenvectors give the direction. An example in $\mathbb{R}^2$ is reported in Figure 2: the eigenvectors (principal components) are the axis Y_1 and Y_2, while the eigenvalue associated with Y_1 is clearly higher then the other, since there is more variance along that direction.

8.3 Random projections

In PCA, the measure of success is the reconstruction error, which is equivalent to the amount of lost variance in our data $X \in \mathbb{R}^{m\times d}$ (each row is a sample). Instead of finding the best n-dimensional subspace of $\mathbb{R}^d$, one tries to find the minimal $n < d$ such that there is a map from $\mathbb{R}^d$ to $\mathbb{R}^n$ which preserves the property:

$$||x_i - x_j|| \approx ||y_i - y_j||$$

where $x_i, x_j \in \mathbb{R}^d$ are mapped respectively to $y_i, y_j \in \mathbb{R}^n$ $\forall i, j \in [m]$. So, the position into space of the data points is not preserved, but their distance is. Let's look then for W (the encoding matrix) such that

$$\frac{||Wx_i - Wx_j||}{||x_i - x_j||} \approx 1 \ \forall i, j \in [m]$$

Now, consider $Q = \{x_i - x_j | i, j \in [m]\}$ as the set of differences between points in the dataset (note that a difference between vectors is again a vector). Then let's look for W such that $\frac{||Wx||}{||x||} \approx 1$ with $x \in Q$.

To reach this scope, a **random projection** is used, hence a random matrix $W \in \mathbb{R}^{n\times d}$ in which every entry w_{ij} is an indipendent and identically distributed (i.i.d.) continuous random variable with normal distribution $\mathcal{N}(0, \frac{1}{n})$, where n is the dimension of the destination subspace. Then one has the following consideration:

$$\begin{aligned}\mathbb{E}\left[||Wx||^2\right] &= \sum_{i=1}^n \mathbb{E}\left[< w_i, x >^2\right] = \sum_{i=1}^n x^T \mathbb{E}\left[w_i w_i^T\right] x \\ &= n x^T \left(\frac{1}{n} \cdot I\right) x = n \cdot \frac{1}{n} x^T x = ||x||^2\end{aligned}$$

where $\mathbb{E}[w_i w_i^T]$ is the covariance matrix. Actually $||Wx||^2$ has a $\mathcal{X}_n^2$ distribution (since it is a linear combination of squared independent Gaussians), then, using concentration measures:

$$\mathbb{P}\left[\left|\frac{||Wx||^2}{||x||^2} - 1\right| > \varepsilon\right] \le 2 \cdot \exp\left\{-\frac{\varepsilon^2 n}{6}\right\}$$

This result states that the higher is the threshold ε or the dimension n, the lower is the probability to have a big distortion of the distance.

Lemma 8.1 *Let Q be a finite set of vectors in $\mathbb{R}^d$, let $\delta \in (0,1)$ and let n be an integer such that*

$$\varepsilon = \sqrt{\left(\frac{6 \cdot log(2|Q|/\delta)}{n}\right)} \leq 3$$

Then, with probability of at least $1-\delta$ over a choice of a random matrix $W \in \mathbb{R}^{n\times d}$ with $w_{ij} \sim \mathcal{N}(0, \frac{1}{n})$ we have that

$$\max_{x\in Q}\left|\frac{||Wx||^2}{||x||^2} - 1\right| < \varepsilon$$

□

Note that the previous result does not depend on d, the dimension of the original data. Let's see another formulation of the same thing.

Lemma 8.2 *(Johnson-Lindenstrauss) Let $x_i \in \mathbb{R}^d$ with $i \in [m]$. Then, for some $n \in O(log(m)/\varepsilon^2)$ there exist points $y_1, \ldots, y_m, y_i \in \mathbb{R}^n$ such that*

$$\begin{aligned} &(1-\varepsilon)||x_j|| \leq ||y_j|| \leq ||x_j||(1+\varepsilon) && \forall j \in [m] \\ &(1-\varepsilon)||x_i - x_j|| \leq ||x_i - x_j|| \leq ||x_i - x_j||(1+\varepsilon) && \forall i,j \in [m] \end{aligned}$$

Moreover, in polynomial time, one can compute a linear mapping $\mathcal{L} : \mathbb{R}^d \to \mathbb{R}^n$ such that $y_j = \mathcal{L}(x_j)$ and the inequalities above are satisfied with probability at least $1 - \frac{2}{m}$. □

Note that random projections have some advantages. For example:

- The dimensionality reduction made in PCA is not applicable to data streams, since new data may be affected by data shift (i.e., the new data is not compliant with the distribution in the training set which has been used to infer matrix W). In such a case, the covariance matrix changes over the time, hence PCA cannot be applied on the stream. Instead, the dimensionality reduction dependency on the training data does not affect the random projector $\mathcal{L}$.
- If clustering is involved, then the random projector $\mathcal{L}$ is a suitable way to achieve data anonymization, since it is practically impossible to reconstruct the original data from the projections.

Moreover, the concept of random projections is somehow "dual" to the one of PCA: in PCA the dimension is fixed and one looks for the best minimizing subspace, while in Johnson-Lindenstrauss (JL) the distorsion ε is fixed (the reconstruction error) and one looks for the minimal dimension. PCA has therefore one and only one solution, while JL has infinite solutions $\forall n, \varepsilon$ compatible.

8.4 Exercise 1 - Chapter 23

Text

In this exercise we show that, in the general case, exact recovery of a linear compression scheme is impossible.

Let $A \in \mathbb{R}^{n\times d}$ be an arbitrary compression matrix where $n < d$. Show that there exists $u, v \in \mathbb{R}^n$ such that $u \neq v$ and $Au = Av$. Conclude that the exact recovery of a linear compression scheme is impossible.

Solution

We know that $rank(A) \leq n$, but we know that $rank(A) + null(A) = d$ (i.e., $dim(Im(A)) + dim(ker(A)) = d$). Then $ker(A) \neq \emptyset$ and the map $x \to Ax$ is not injective (remember that $dim(ker(A)) > 0$ implies that $Ax = 0$ form some $x \neq 0$, then $Ax = A0$ we have that 0 is an eigenvalue of A, with x as associated eigenvector).

Let $u, v \in \mathbb{R}^d$ such that $u \neq v$ and $Au = Av$, then we have that $f(Aw) = f(Av)$ where $f : \mathbb{R}^n \to \mathbb{R}^d$ is the decompression function, but since $u \neq v$ we have that $f(Av) \neq v$ and/or $f(Au) \neq u$ (i.e., at least one vector has not been reconstructed correctly).

8.5 Exercise 2 - Chapter 23

Text

Let $\alpha \in \mathbb{R}^d$ such that $\alpha_1 \geq \alpha_2 \geq \cdots \geq \alpha_d \geq 0$. Show that:

$$\max_{\beta\in[0,1]^d : ||\beta||_1 \leq n} \sum_{j=1}^{d} \alpha_j \beta_j = \sum_{j=1}^{n} \alpha_j$$

Solution

Take a vector $\beta \in [0,1]^d$ constructed in this way: $(1,1,\dots,1|\beta_i,\dots)$ where $\beta_i < 1$, all the elements before are equal to 1 and on the elements after no assumption is made. If we take $i = n+1$ we have that $\beta_1,\dots,\beta_n$ are all equal to 1. When we compute $<\alpha,\beta>= \sum_{j=1}^{d} \alpha_j\beta_j$ we have that this corresponds to the sum of the first n components of α. Thus, by the definition of the vector α (components in α are sorted descendently), the lemma holds.

Otherwise, if $i \neq n+1$, it implies that $i < n+1$, because otherwise $||\beta||_1$ would be greater than n (which goes against the constraint in the maximization problem). In this situation, you can increase β_i and decrease the value of element after (i+1), in order to respect the constraint on the norm. Then the dot product will become:

$$\alpha_1 + \alpha_2 + \dots + \alpha_{i-1} + \beta_i\alpha_i + \beta_{i+1}\alpha_{i+1} + \dots + \beta_d\alpha_d$$
$$\leq \alpha_1 + \alpha_2 + \dots + \alpha_{i-1} + \beta'_i\alpha_i + \beta'_{i+1}\alpha_{i+1} + \dots + \beta'_d\alpha_d$$

because $\beta'_i > \beta_i$, $\beta'_{i+1} < \beta_{i+1}$ and $\alpha_i \geq \alpha_{i+1}$.

Then we conclude that the best possible choice is to set $\beta_1 = \beta_2 = \dots = \beta_n = 1$ and $\beta_{n+1} = \dots = \beta_d = 0$.

9 Lesson 9 - May 20

9.1 Hilbert Spaces

Euclidean vector spaces can be generalized to vector spaces of functions. Every element of a (real-valued) function space $\mathcal{H}$ is a function from some set $\mathcal{X} \to \mathbb{R}$, and elements can be added and scalar multiplied as if they were vectors. In other words, if $f \in \mathcal{H}$ and $g \in \mathcal{H}$, then $\alpha f + \beta g \in \mathcal{H}$ for all $\alpha, \beta \in \mathbb{R}$. On $\mathcal{H}$ we can impose an inner product as a mapping $\langle \cdot, \cdot \rangle$ from $\mathcal{H} \times \mathcal{H} \to \mathbb{R}$ that satisfies

1. $\langle \alpha f_1 + \beta f_2, g \rangle = \alpha \langle f_1, g \rangle + \beta \langle f_2, g \rangle$
2. $\langle f, g \rangle = \langle g, f \rangle$
3. $\langle f, f \rangle \geq 0$
4. $\langle f, f \rangle = 0 \iff f = 0$ (the zero function)

Definition 9.1 (Norm) *Two elements $f, g \in \mathcal{H}$ are orthogonal to each other with respect to this inner product if $\langle f, g \rangle = 0$. Given an inner product, we can measure distances between elements of the function space $\mathcal{H}$ using the norm*

$$\|f\| = \sqrt{\langle f, f \rangle} \tag{14}$$

Definition 9.2 (Completeness) *The distance between two functions f_m, f_n is given by $\|f_m - f_n\|$. The space $\mathcal{H}$ is said to be complete with respect to the inner product $\|\cdot\|_{\mathcal{H}}$ if every sequence of functions $f_1, f_2, ... \in \mathcal{H}$ for which $\|f_m - f_n\| \to 0$ as $m, n \to \infty$ (Cauchy sequence) there is $f \in \mathcal{H}$ such that*

$$\lim_{n \to \infty} \|f_n - f\|_{\mathcal{H}} = 0$$

A complete inner product space is called a **Hilbert space**. The most fundamental Hilbert space of functions is the space L^2

Definition 9.3 (L^2 space) *Let $\mathcal{X} \subseteq \mathbb{R}^d$ with measure $\mu(dx) = w(x)dx, w(x) \geq 0$ and for $A \subseteq \mathcal{X}$ then $\mu(A) = \int_A w(x)dx$. The Hilbert space $L^2(\mathcal{X}, \mu)$ is the linear space of functions $f : \mathcal{X} \to \mathbb{R}$ that satisfy*

$$\int_{\mathcal{X}} f(x)^2 w(x) dx < \infty \tag{15}$$

and with inner product

$$\langle f, g \rangle = \int_{\mathcal{X}} f(x) \cdot g(x) \cdot w(x) dx \tag{16}$$

Definition 9.4 (Orthonormal system) *Let $\mathcal{H}$ be a Hilbert space. A set of functions $\{f_i, i \in \mathcal{I}\}$ is called an orthonormal system if*

1. *$\langle f_i, f_i \rangle = 1, \forall i \in \mathcal{I}$ (unitary norm)*
2. *$\langle f_i, f_j \rangle = 0, i \neq j$ (orthogonal f_i)*

Theorem 3 (Cauchy-Schwarz) *Let $\mathcal{H}$ be a Hilbert space. For every $f, g \in \mathcal{H}$ it holds that*

$$|\langle f, g \rangle| \leq \|f\| \, \|g\| \tag{17}$$

Definition 9.5 (Operator) *Let $\mathcal{V}$ and $\mathcal{W}$ be two linear vector spaces (for example, Hilbert spaces) on which norms $\|\cdot\|_{\mathcal{V}}$ and $\|\cdot\|_{\mathcal{W}}$ are defined. Suppose $A : \mathcal{V} \to \mathcal{W}$ is a mapping $\mathcal{V} \to \mathcal{W}$. When $\mathcal{W} = \mathcal{V}$, such a mapping is called an operator. When $\mathcal{W} = \mathbb{R}$ it is called a functional.*

Mapping A is said to be **linear** if $A(\alpha f + \beta g) = \alpha A(f) + \beta A(g)$. In this case we write Af instead of $A(f)$. If there exists $\gamma < \infty$ such that

$$\|Af\|_{\mathcal{W}} \leq \gamma \|f\|_{\mathcal{V}}, \quad f \in \mathcal{V} \tag{18}$$

then A is said to be a **bounded mapping**. The smallest γ for which 9.1 holds is called the **norm** of A; denoted by $\|A\|$.

A (not necessarily linear) mapping $A : \mathcal{V} \to \mathcal{W}$ is said to be continuous at f if for any sequence $f_1, f_2, ...$ converging to f the sequence $A(f_1), A(f_2), ...$ converges to $A(f)$. In other terms if $\lim_{n \to \infty} f_n = f \in \mathcal{V}$ it holds that $\lim_{n \to \infty} A(f_n) = A(f) \in \mathcal{W}$. That is, if

$$\forall \varepsilon > 0, \exists \delta > 0 : \forall g \in \mathcal{V}, \|f - g\|_{\mathcal{V}} < \delta \Rightarrow \|A(f) - A(g)\|_{\mathcal{W}} < \varepsilon \tag{19}$$

If the above property holds for every $f \in \mathcal{V}$, then the mapping A itself is called **continuous**.

Theorem 4 (Continuity and Boundness for Linear Mappings) *For a linear mapping, continuity and boundedness are equivalent.*

Theorem 5 (Riesz Representation) *Any bounded linear functional ϕ on a Hilbert space $\mathcal{H}$ can be represented as $\phi(h) = \langle h, g \rangle$, for some $g \in \mathcal{H}$ (depending on ϕ).*

Example
Let the Hilbert space be $\mathcal{H} : (\mathbb{R}^n, \langle \cdot, \cdot \rangle)$ where $\langle a, b \rangle = a^T b$ is the usual inner product.
Given a linear mapping $\phi : \mathbb{R}^n \to \mathbb{R}$ then it could be written with the matrix notation where ϕ is a $1 \times n$ matrix and $\phi(x) = a_1 x_1 + ... + a_n x_n$ with $g = (a_1, ..., a_n)^T$ therefore the inner product $\langle x, g \rangle = x^T g$ can be written as

$$\begin{pmatrix} x_1 & \dots & x_n \end{pmatrix} \begin{pmatrix} a_1 \\ \vdots \\ a_n \end{pmatrix} = \phi(x)$$

9.2 Regularization

The aim of regularization is to improve the predictive perfomance of the best learner in some class of functions $\mathcal{G}$ by adding a penalty term to the training loss that penalizes learners that tend to overfit the data.

Let $\mathcal{G}$ be a Hilbert space of functions. One way to avoid overfitting is to introduce a non-negative functional $J : \mathcal{G} \mapsto \mathbb{R}_+$ which penalizes complex models.

9.2.1 Ridge Regression

Ridge regression is a linear regression with a squared-norm penalty function. The squared loss is combined with a squared-norm penalty with regularization paramter $\gamma > 0$

$$\min_{g \in \mathcal{G}} \frac{1}{n} \sum_{i=1}^{n} (y_i - g(x_i))^2 + \gamma \left\| g \right\|_2^2 \tag{20}$$

By associating each $g \in \mathcal{G}$ with a vector of parameters $\beta \in \mathbb{R}^p$ equation 20 can be rewritten as

$$\min_{\beta \in \mathbb{R}^p} \frac{1}{n} \left\| y - X\beta \right\|^2 + \gamma \left\| \beta \right\|_2^2 \tag{21}$$

As $\gamma \to \infty$, the regularization term becomes dominant while for $\gamma = 0$ the normal squared loss regression is obtained.

9.2.2 Lasso Regression

Replacing the squared 2-norm with a 1-norm gives the lasso regression

$$\min_{\beta \in \mathbb{R}^p} \frac{1}{n} \left\| y - X\beta \right\|^2 + \gamma \left\| \beta \right\|_1 \tag{22}$$

Both Ridge and Lasso regression are convex optimization problems, the main difference is that while Ridge is differentiable, the absolute value in Lasso regression is not differentiable, therefore numerical optimization approaches are need to apply Lasso regularization.

9.3 Reproducing Kernel Hilbert Spaces

To evaluate the loss of a learner $f \in \mathcal{G}$ it is only required to evaluate g at all the feature vectors x_i (i.e., the samples). A defining property of a *Reproducing Kernel Hilbert Space* $\mathcal{G}$ is that function evaluation at point x can be performed by taking the inner product of g with the feature function $\kappa_x \in \mathcal{G}$, $g(x) = \langle g, \kappa_x \rangle$

Definition 9.6 (Reproducing Kernel Hilbert Space) *For a non-empty set $\mathcal{X}$, a Hilbert space $\mathcal{G}$ of functions $g : \mathcal{X} \to \mathbb{R}$ with inner product $\langle \cdot, \cdot \rangle_{\mathcal{G}}$ is called a Reproducing Kernel Hilbert Space (RKHS) with reproducing kernel $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ if*

1. *for every $x \in \mathcal{X}, \kappa_x \overset{def}{=} \kappa(x, \cdot)$ is in $\mathcal{G}$*
2. *$\kappa(x, x) < \infty$ for all $x \in \mathcal{X}$ (Boundedness)*
3. *for every $x \in \mathcal{X}$ and $g \in \mathcal{G}, g(x) = \langle g, \kappa_x \rangle_{\mathcal{G}}$ (reproducing property)*

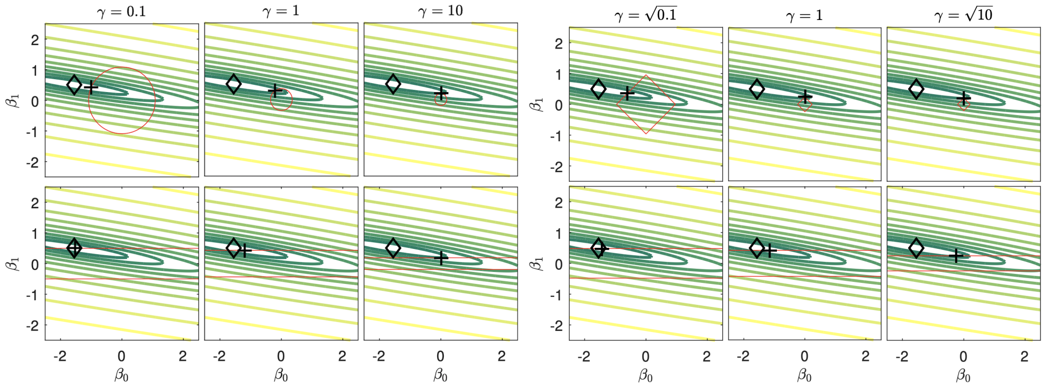


Figure 3: Ridge (left) and Lasso (right) Regression Regularization with different values of γ. *Diamonds* represent the minimizers, *Plusses* show the minimizers of the regularized problems.

The matrix $K = XX^T$ collects the inner products: $K = [\kappa(x_i, x_j)|i, j = 1, ..., n]$ and it is called **Gram Matrix**. A reproducing kernel κ is a positive semidefinite function (For any $n \geq 1$ and every choice of $\alpha_1, \dots, \alpha_n \in \mathbb{R}$ and $x_1, \dots, x_n \in \mathcal{X}$, than it holds $\sum_{i=1}^n \sum_{j=1}^n \alpha_i \kappa(x_i, x_j)\alpha_j \geq 0$. Thus, every Gram Matrix K associated with κ is a positive semidefinite matrix.

Theorem 6 (Continuos Evaluation Functional Characterize a RKHS) *An RKHS $\mathcal{G}$ on a set $\mathcal{X}$ is a Hilbert space in which every evaluation function $\sigma_x : g \mapsto g(x)$ is bounded.*

Conversely, a Hilbert space $\mathcal{G}$ of functions $\mathcal{X} \to \mathbb{R}$ for which every evaluation functional is bounded is an RKHS.

Practically if two functions have similar norm, their outputs will have similar norm too.

Theorem 7 (Riesz Representation Theorem) *Any bounded linear functional ϕ on a Hilbert space $\mathcal{H}$ can be represented as $\phi(h) = \langle h, g \rangle$ for some $g \in \mathcal{G}$*

Any finite function κ with certain characteristics can serve as a reproducing kernel.

Theorem 8 (Moore-Aronszajn) *Given a non-empty set $\mathcal{X}$ and any finite symmetric positive semidefinite function $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, there exists an RKHS $\mathcal{G}$ of functions $g : \mathcal{X} \to \mathbb{R}$ with reproducing kernel κ. Moreover, $\mathcal{G}$ is unique.*

Example: Linear Kernel

A simple RKHS example is defined by the feature map $\phi : \mathcal{X} \to \mathbb{R}^p$ with $\kappa(x, x') \overset{def}{=} \langle \phi(x), \phi(x') \rangle$ where $\langle \cdot, \cdot \rangle$ is the Euclidian inner product. In particular, the identity feature map $\phi(x) = x$ is called linear kernel

$$\kappa(x, x') = \langle x, x' \rangle = x^T x' \tag{23}$$

The RKHS of functions corresponding to the linear kernel is the space of linear functions on $\mathbb{R}^p$

9.4 Constructing Kernels

9.4.1 Gaussian Kernel

One of the most popular kernel is the Gaussian one, also called Radial Basis Function (RBF). Its characteristic function is

$$\kappa(x, x') = \exp\left(-\frac{1}{2}\frac{\|x - x'\|^2}{\sigma^2}\right) \tag{24}$$

σ is called *bandwidth* and it is an hyperparameter that controls the sensitivity of the kernel away from the center.

10 Lesson 10 - May 26

10.1 Representer theorem

Let's have a brief recap:

- We have our training set τ and a loss function that measures fitness to data
- We wish to find a function $g \in \mathcal{G}$ that minimizes the training loss with the addition of the regularization penalty term. Working with regularization reduces the variance of the solutions, they are more stable with respect to data variance.
- We assume that the class $\mathcal{G}$ of prediction functions can be express as the direct sum of an RKHS $\mathcal{H}$, defined by a kernel function $k : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$, and another linear space of real-valued functions $\mathcal{H}_0$ on $\mathcal{X}$. This way $\mathcal{G}$ can be expressed as a direct sum:

  $$\mathcal{G} = \mathcal{H} \oplus \mathcal{H}_0$$

  this means that $\forall g \in \mathcal{G}, \exists$ unique $h \in \mathcal{H}, h_0 \in \mathcal{H}_0$ such that $g = h + h_0$. We wish to penalize the h term of g but not h_0.

As an example in the case of ridge regression, the kernel was the usual scalar product, $\mathcal{H}$ the space of linear functions in $\mathbb{R}^d$, and $\mathcal{H}_0$ was the space of constant functions, practically:

$$\min_{g \in \mathcal{H} \oplus \mathcal{H}_0} \frac{1}{n} \sum_{i=1}^{n} \text{Loss}(y_i, g(x_i)) + \gamma ||g||_{\mathcal{H}}^2 \tag{25}$$

Where $||g||_{\mathcal{H}} = ||h||_{\mathcal{H}}$. $\mathcal{H}_0$ is seen as the null space of the functional $g \mapsto ||g||_{\mathcal{H}}$ which is a mapping $\mathcal{H} \oplus \mathcal{H}_0 \mapsto \mathbb{R}$, Which practically translates to all functions that are mapped to zeros, i.e. $||g||_{\mathcal{H}} = 0$.

The null space may be empty, but typically has a small dimension. For example in the ridge regression it is the set of constant functions, which has dimension 1.
In the ridge regression the feature vector $\tilde{x} = [1, x]$ and $\mathcal{G}$ consists of function in the form $g : \tilde{x} \mapsto \beta_0 + x^\top \beta$ with $h : \tilde{x} \mapsto x^\top \beta$ and $h_0 : \tilde{x} \mapsto \beta_0$.
Since γ can go from zero to infinity, we can regulate g to be simple (large γ) or complex (small γ).
Since we took $\mathcal{H}$ to be an RKHS we can see functional optimization as parametric optimization. This implies that we can find a solution numerically with a closed formula, for example using least square problem.

Theorem 9 (Representer theorem) *Let $\mathcal{H}$ be an RKHS with kernel k. The solution to the penalized optimization problem presented in equation 25 can be expressed as: a linear combination of data with the variable x passed through the kernel function and an element expressed as a combination of a basis of $\mathcal{H}_0$:*

$$g(x) = \sum_{i=1}^{n} \alpha_i \kappa(x_i, x) + \sum_{j=1}^{m} \eta_j q_j(x)$$

where $\{q_1, \ldots, q_m\}$ is a basis of $\mathcal{H}_0$.

Proof
recall that $\kappa_{x_i} = \kappa(x_i, x)$, Let $\mathcal{F} = \mathit{span}\{\kappa_{x_i}, i \in [n]\}$, clearly $\mathcal{F} \subseteq \mathcal{H}$. let $\mathcal{F}^\perp$ the orthogonal complement of $\mathcal{F}$ defined as:

$$\{f^\perp \in \mathcal{H} : \langle f^\perp, f \rangle_{\mathcal{H}} = 0, f \in \mathcal{F}\} \equiv \{f^\perp : \langle f^\perp, \kappa_{x_i} \rangle_{\mathcal{H}} = 0, i \in [n]\}$$

Since this is the complement we have that $\mathcal{H} = \mathcal{F} \oplus \mathcal{F}^\perp$ By reproducing the kernel property, $\forall f^\perp \in \mathcal{F}^\perp$:

$$f^\perp(x_i) = \langle f^\perp, \kappa_{x_i} \rangle_{\mathcal{H}} = 0$$

lets take $g \in \mathcal{H} \oplus \mathcal{H}_0 = \mathcal{F} \oplus \mathcal{F}^\perp \oplus \mathcal{H}_0$ and lets write it as $g = f + f^\perp + h_0$.
By the definition of the null space $\mathcal{H}_0$ we have $||g||_{\mathcal{H}} = ||f + f^\perp||_{\mathcal{H}}$. By Pythagoras theorem $||f + f^\perp||_{\mathcal{H}}^2 = ||f||_{\mathcal{H}}^2 + ||f^\perp||_{\mathcal{H}}^2$ we have that:

$$\frac{1}{n} \sum_{i=1}^{n} \text{Loss}(y_i, g(x_i)) + \gamma ||g||_{\mathcal{H}}^2 = \frac{1}{n} \sum_{i=1}^{n} \text{Loss}(y_i, f(x_i) + h_0(x_i)) + \gamma(||f||_{\mathcal{H}}^2 + ||f^\perp||_{\mathcal{H}}^2) \tag{26}$$

$$\geq \frac{1}{n} \sum_{i=1}^{n} \text{Loss}(y_i, f(x_i) + h_0(x_i)) + \gamma ||f||_{\mathcal{H}}^2 \tag{27}$$

And so since the first quantity is greater than the other, by taking and we obtain an equality for $f^{\perp} = 0$. This implies that the minimizer lies it the subspace $\mathcal{F} \oplus \mathcal{H}_0$ of $\mathcal{G}$. □

By substituting the representation of g we obtain the following parametric optimization:

$$\min_{\alpha \in \mathbb{R}^n, \eta \in \mathbb{R}^m} \frac{1}{n} \sum_{i=1}^{n} \text{Loss}(y_i, (\mathbf{K}\alpha + \mathbf{Q}\eta)_i) + \gamma \alpha^\top \mathbf{K} \alpha$$

where $\mathbf{K}$ is the $n \times n$ Gram matrix with entries $\kappa(x_i, x_j), i \in [n], j \in [n]$ and $\mathbf{Q}$ the $n \times m$ matrix with entries $q_j(x_i), i \in [n], j \in [m]$

In particular for the squared-error loss we have:

$$\min_{\alpha \in \mathbb{R}^n, \eta \in \mathbb{R}^m} \frac{1}{n} ||y - (\mathbf{K}\alpha + \mathbf{Q}\eta))||^2 + \gamma \alpha^\top \mathbf{K} \alpha$$

Which is a convex optimization problem and the solution can be found by differentiating with respect to α and η and equating them to zero in a linear system which has a unique solution if $\mathbf{Q}$ is full rank. In the example of ridge regression $\mathbf{K} = XX^\top$ and $\mathbf{Q} = 1$.
Our goal is to learn the Peaks function fig.4(which has a complicated formula that you can find on wikipedia) by means of a Gaussian kernel on $\mathbb{R}^2$, if we omit the regularization term that the objective becomes:

$$\min_{g \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} (y_i - g(x_i))^2$$

by the representer theorem we can express the optimal function as:

$$g(x) = \sum_{i=1}^{n} \alpha_i exp(-\frac{||x - x_i||^2}{2\sigma^2})$$

where α, is the solution of the linear system $\mathbf{K}\mathbf{K}^\top \alpha = \mathbf{K}y$ (the same as least square approximation).
With $\mathbf{K}_{i,j} = \kappa(x_i, x_j) = exp(-\frac{||x_i - x_j||^2}{2\sigma^2})$

10.2 Support Vector Machines

We put our selves in the binary classification case with training set $\tau = \{(\boldsymbol{x}, y) \mid \boldsymbol{x} \in \mathcal{X}, y \in \mathcal{Y}\}$, with $\mathcal{Y} = \{-1, 1\}$.

Definition 10.1 (Hinge Loss)

$$Loss(y, \hat{y}) = (1 - y\hat{y})_+ := \max\{0, 1 - y\hat{y}\}$$

When the prediction is correct $1 - y\hat{y}$ is 0 otherwise is 2.
The optimal classifier $g^\star$ can be viewed as a minimizer for the hinge loss over all prediction functions g:

$$g^\star = \arg\min_{g} \mathbb{E}[1 - Yg(X)]$$

Let g_τ be the ERM predictor on our dataset τ. Generally it is not a classifier by itself, so to produce values in $\{-1, 1\}$ we do the following:

$$\text{sign } g_\tau(\boldsymbol{x})$$

Therefore, a feature vector $\boldsymbol{x}$ is classified according to 1 or -1 depending on whether $g_\tau(\boldsymbol{x}) \geq 0$ or < 0, respectively. If g_τ is a linear function $g_\tau(x) = 0$ defines the optimal hyper plane decision boundary facing the positive samples.

In a RKHS setting we can find the best classifier $g^\star$ by solving the following optimization problem:

$$\min_{g \in \mathcal{H} \oplus \mathcal{H}_0} \frac{1}{n} \sum_{i=1}^{n} (1 - y_i g(\boldsymbol{x}_i))_+ + \tilde{\gamma} ||g||^2_{\mathcal{H}}$$

for some regularization parameter $\tilde{\gamma}$.
To simplify our computation we impose $\gamma := 2n\tilde{\gamma}$, which is equivalent to solving:

$$\min_{g \in \mathcal{H}} \sum_{i=1}^{n} (1 - y_i g(\boldsymbol{x}_i))_+ + \frac{\gamma}{2} ||g||^2_{\mathcal{H}} \tag{28}$$

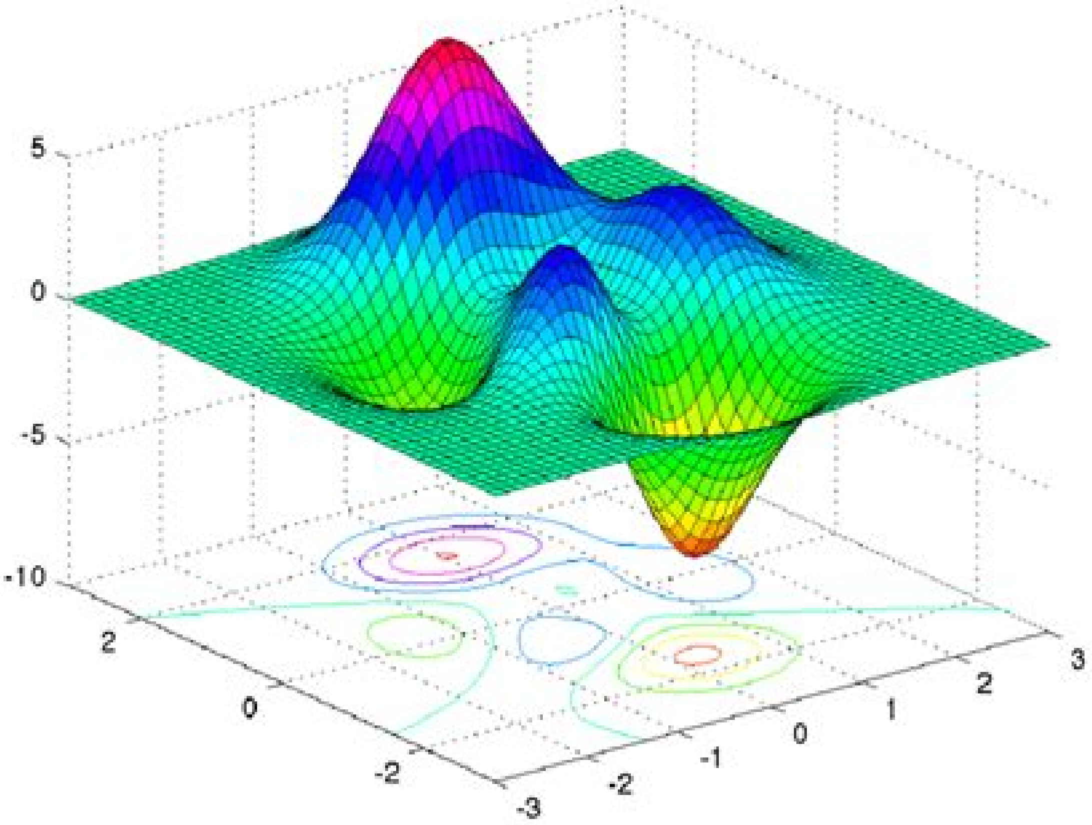


Figure 4: Peaks function

Using the representer theorem we know that if κ is the reproducing kernel corresponding to $\mathcal{H}$ (assuming $\mathcal{H}_0$ has a constant term only), then the solution is of the form:

$$g(\boldsymbol{x}) = \alpha_0 + \sum_{i=1}^{n} \alpha_i \kappa(\boldsymbol{x}_i, \boldsymbol{x})$$

Substituting this result into the minimization problem (28) we obtain the following expression:

$$\min_{\boldsymbol{\alpha}, \alpha_0} \sum_{i=1}^{n} [1 - y_i(\alpha_0 + \{\mathbf{K}\boldsymbol{\alpha}\}_i)]_+ + \frac{\gamma}{2} \boldsymbol{\alpha}^\top \mathbf{K} \boldsymbol{\alpha}$$

where $\mathbf{K}$ is the Gram Matrix.

Unfortunately this function cannot be solved with the usual closed form because the hinge loss is not differentiable, there is the need for a new approach: we are going to solve the problem with its dual optimization.

Let's define $\lambda_i := \gamma \alpha_i / y_i, i \in [n]$ and $\lambda = [\lambda_1, \ldots, \lambda_n]^\top$ it can be shown that the optimal $\boldsymbol{\alpha}$ can be obtained by solving the "dual" convex constrained optimization problem:

$$\max_{\lambda} \sum_{i=1}^{n} \lambda_i - \frac{1}{2\gamma} \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \lambda_j y_i y_j \kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) \tag{29}$$

$$\text{s.t. } \boldsymbol{\lambda}^\top \boldsymbol{y} = 0 \tag{30}$$

$$\mathbf{0} \leq \boldsymbol{\lambda} \leq \mathbf{1} \tag{31}$$

The optimal prediction function is then given by:

$$g_\tau(\boldsymbol{x}) = \alpha_0 + \frac{1}{\gamma} \sum_{i=1} n y_i \lambda_i \kappa(\boldsymbol{x}_i, \boldsymbol{x})$$

please note that when $\lambda_i = 0$ then the corresponding training point x_i is not useful to the solution. The one participating in our solution are called support vectors. The dual problem can be seen as maximizing the margin, which is the distance of the support vectors from the decision boundary.

11 Lesson 11 - May 27

11.1 Linear Separators

Notation for 11.1

- $\boldsymbol{x}_i$ is the i^{th} instance (**bold** for vectors)
- y_i is the i^{th} instance label (non-bold for scalars)
- x_{ij} is the j^{th} feature of the i^{th} instance

11.1.1 Linear Separators

Two classes are linearly separable if

$$\exists(\boldsymbol{\theta}, b) \in \mathbb{R}^d \times \mathbb{R} : y_i \cdot (\langle \boldsymbol{\theta}, \boldsymbol{x}_i \rangle + b) \geq 0, \ \forall i = 1, \dots, m$$

where $\pi : (\boldsymbol{\theta}, b)$ is an hyperplane and $\{(\boldsymbol{x}_i, y_i) \mid i = 1, \dots, m\}$ are the data points, having label $y = \{-1, 1\}$.
A good separator is a hyperplane for which the margin is maximized (”fattest” hyperplane). The margin is defined as the minimum distance between hyperplane and data:

$$\min\{d(\boldsymbol{\theta}, \boldsymbol{x}) \mid x \in Dataset\}$$

Let p be the projection of $\boldsymbol{x}_i$ onto the vector $\boldsymbol{\theta}$:

$$\boldsymbol{\theta}^\top \boldsymbol{x} = \|\boldsymbol{\theta}\|_2 \|\boldsymbol{x}\|_2 \cos\hat{\theta} = p \|\boldsymbol{\theta}\|_2$$

where $\hat{\theta}$ is the angle between the two vectors.

We define the *support vectors* as the points satisfying $\boldsymbol{\theta}^\top \boldsymbol{x} = 1$ or -1. In practice they are the points that lie exactly in the margin and they are the only ones used for computing the best margin. For support vectors holds $p = \frac{1}{\|\boldsymbol{\theta}\|_2}$.
With some geometric considerations (omitted) we obtain the expression of the margin width, that is $2p = \frac{2}{\|\boldsymbol{\theta}\|}$. Therefore the problem of finding the best margin can be seen as a the problem of maximizing p. Alternatively, we solve the similar problem of minimizing $\|\boldsymbol{\theta}\|$:

$$\max(\text{margin width}) = \max_p 2p = \max_{\boldsymbol{\theta}} \frac{2}{\|\boldsymbol{\theta}\|} \ \rightarrow \ \min_{\boldsymbol{\theta}} \|\boldsymbol{\theta}\|$$

that, for mathematical convenience, becomes:

$$\begin{aligned} \min_{\boldsymbol{\theta}} \quad & \frac{1}{2}\sum_{j=1}^{d} \theta_j^2 \\ \text{s.t.} \quad & \boldsymbol{\theta}^\top \boldsymbol{x}_i \geq 1 \quad \text{if} \quad y_i = 1 \\ & \boldsymbol{\theta}^\top \boldsymbol{x}_i \leq -1 \quad \text{if} \quad y_i = -1 \end{aligned}$$

which is the Primal Problem, with the assumption that the classes are linearly separable.
It can be solved more efficiently by taking the Lagrangian dual, introducing slack variables α_i, each one indicating how important a particular constraint is to the solution. This allows to transform a constrained problem to an unconstrained one (Lagrangian relaxation).
The Lagrangian is given by:

$$\begin{aligned} L(\boldsymbol{\theta}, \boldsymbol{\alpha}) =& \frac{1}{2}\sum_{j=1}^{d} \theta_j^2 - \sum_{i=1}^{n} \alpha_i (y_i \boldsymbol{\theta}^\top \boldsymbol{x} - 1) \\ \text{s.t.} \ & \alpha_i \geq 0 \quad \forall i \end{aligned}$$

We must minimize over $\boldsymbol{\theta}$ and maximize over $\boldsymbol{\alpha}$. At first we minimize over $\boldsymbol{\theta}$ computing the gradient and looking for partial derivatives equal to 0 (omitted), obtaining:

$$\begin{aligned} J(\boldsymbol{\alpha}) =& \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle \\ \text{s.t.} \ & \alpha_i \geq 0 \quad \forall i \\ & \sum_i \alpha_i y_i = 0 \end{aligned} \tag{32}$$

Considerations:

- $\alpha_i \geq 0 \quad \forall i$ → constraint weights cannot be negative;
- $\sum_i \alpha_i y_i = 0$ → balances between the weight of contraints for different classes;
- $y_i y_j$ → points with different labels increase the sum, those with the same label decrease it;
- $\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle$ → is the Gram matrix and it measures the similarity between points. Substituting this inner product with a kernel we can obtain non-linear separators.

Solving this problem is not easy and we are not able to find a close form for the solution (we can use computational analysis, SGD for SVM, ...), but it is easy to formulate, as it is a quadratic form with the Gram Matrix. In the solution, either:

- $\alpha_i > 0$ and the constraint is tight $(y_i(\boldsymbol{\theta}^\top x_i) = 1)$ → point is a support vector;
- $\alpha_i = 0$ → point is not a support vector.

Support vectors are those points that lie in the margin: $\mathcal{SV} = \{\boldsymbol{x} : |\langle \boldsymbol{\theta}, \boldsymbol{x} \rangle| = 1\}$

Theorem 10 (Fritz John Optimality Condition) *Let $\boldsymbol{w}_0 \in \arg\min_{\boldsymbol{w}} \|\boldsymbol{w}\|^2$ such that $\forall i, y_i \cdot (\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle) \geq 1$ (linear separability condition) and let $I = \{i : |\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle| = 1\}$ be the set of indices of the support vectors. Then:*

$$\exists \alpha_1, \dots, \alpha_m \in \mathbb{R} \quad \textit{such that} \quad \boldsymbol{w}_0 = \sum_{i \in I} \alpha_i \boldsymbol{x}_i$$

It says that the solution of the minimization problem, in case of separability, is a linear combination of the support vectors. This implies that support vectors are the only points that are relevant for the problem of finding the margin. The proof of this condition makes use of the following lemma.

Lemma 11.1 (Fritz John) *Let $w^* \in \arg\min_w f(w)$ such that $\forall i \in [m], g_i \leq 0$ where $f, g_1, \dots, g_m$ are differentiable. Then $\exists \alpha \in \mathbb{R}^m$ such that:*

$$\nabla f(w^*) + \sum_{i \in I} \alpha_i \nabla g_i(w^*) = 0$$

where $I = \{i \in [m] : g_i(w^*) = 0\}$

In our case $g_i(w^*) = 1 - y_i \cdot (\langle w, x_i \rangle) \leq 0$ are the support vectors, $f(w) = \|w\| = \sum_j w_j^2$ and $\nabla f = 2w$. As a consequence of the lemma, given the optimal solution $\boldsymbol{\alpha}^*$, optimal weights are:

$$\boldsymbol{\theta}^* = \sum_{i \in \mathcal{SV}} \alpha_i^* y_i \boldsymbol{x}_i$$

Therefore we can solve one of the SV constraints $y_i(\boldsymbol{\theta}^* \cdot \boldsymbol{x}_i + \theta_0) = 1$ to obtain θ_0 or, more commonly, take the average solution over all support vectors (and use it as an approximation or perform SGD).

11.1.2 Soft Margins

If data are not linearly separable, we cannot find a $\boldsymbol{\theta}$ that satisfies $y_i(\boldsymbol{\theta}^\top \boldsymbol{x}_i) \geq 1 \quad \forall i$. One solution could be to relax the constraint introducing some slack variables ξ_i and obtaining the following new problem:

$$\min_{\boldsymbol{\theta}} \frac{1}{2} \sum_{j=1}^{d} \theta_j^2 + C \sum_i \xi_i$$
$$\text{s.t.} \quad y_i(\boldsymbol{\theta}^\top \boldsymbol{x}_i) \geq 1 - \xi_i \quad \forall i$$

In this way we allow the points to penetrate into the margins of a quantity ξ_i that we can control introducing a cost C, which is a hyperparameter (to be tuned). This is the concept of Soft Margin. The quantity $C \sum_i \xi_i$ is a regularization term. For $C \to \infty$, $\xi_i \to 0$, while if $C \to 0$, ξ_i tend to push the boundary towards including all points.

11.1.3 Non-linear separators

Even if we allow margin violation, some data are not linearly separable due to the intrinsic geometry. To solve this problem, we can try mapping the data into a higher dimensional space and fit a linear separator into that space, finding the margin. Then we map back that margin into the original space to find the interested decision boundary.

- define a feature map $\Phi : \mathcal{X} \to \hat{\mathcal{X}}$;
- rather than run SVM on $\boldsymbol{x}_i$, run it on $\Phi(\boldsymbol{x}_i)$ to find a non-linear separator in input space;
- if $\Phi(\boldsymbol{x}_i)$ is really big, then use kernels. In fact computing $\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \langle \Phi(\boldsymbol{x}_i), \Phi(\boldsymbol{x}_j) \rangle$ should be much more efficient than computing $\Phi(\boldsymbol{x}_i)$ and $\Phi(\boldsymbol{x}_j)$.

Substituting a kernel into the objective function of (32) we obtain:

$$\bar{J}(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \langle \Phi(\boldsymbol{x}_i), \Phi(\boldsymbol{x}_j) \rangle = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

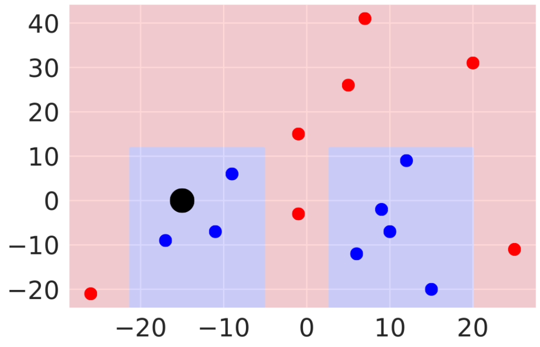
kernel is the dot product of mapped arguments

11.1.4 Some examples of Kernels

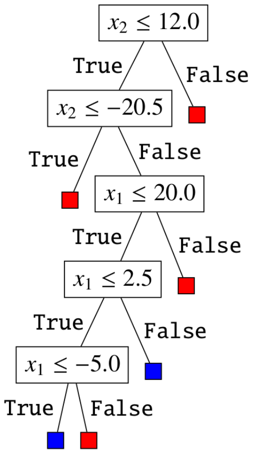
- Linear Kernel $\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle$
- Polynomial Kernel $\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle + c)^d$
- Gaussian Kernel $\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2}{2\sigma^2}\right)$
- Sigmoid Kernel $\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \tanh\left(\alpha \boldsymbol{x}_i^\top \boldsymbol{x}_j + c\right)$
- Cosine similarity
- Chi-squared
- String/tree/graph/wavelet/etc

11.2 Decision Trees

Suppose we want to classify two-dimensional points into two classes. We can partition the feature space $\mathcal{X} = \mathbb{R}^2$ into rectangular regions. The partition defines a classifier g that assigns to each feature vector $\boldsymbol{x}$ a class (red or blue). The classification procedure and the partitioning of the feature space can be represented by a decision tree.



(a) Partition of the the feature space $\mathcal{X} = \mathbb{R}^2$ defined by the decision tree g



(b) Classification of a new point $\boldsymbol{x} = [x_1, x_2]$ by means of the decision tree g

The space is subdivided only into regions of which boundaries are orthogonal to the axis. A region in $\mathbb{R}^d$ will have shape $[a_1, b_1] \times [a_2, b_2] \times \ldots \times [a_d, b_d]$ (d-dimensional boxes).

Each internal node ν contains a logical condition that divides $\mathcal{R}_\nu$ into two disjoint subregions. Each leaf node w is a regional prediction function g^w on $\mathcal{R}_w$.
A binary tree $\mathbb{T}$ partitions the feature space $\mathcal{X}$ into as many regions as there are leaf nodes. Denote the set of leaf nodes with $\mathcal{W}$. The overall prediction function that corresponds to the tree can be written as

$$g(\boldsymbol{x}) = \sum_{w \in \mathcal{W}} g^w(\boldsymbol{x}) \cdot \mathbb{1}_{\boldsymbol{x} \in \mathcal{R}_w},$$

where $\mathbb{1}$ denotes the indicator function. Constructing a tree with a training set $\tau = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ and a given loss function, amounts to minimizing the training loss

$$l_\tau(g) = \sum_{w \in \mathcal{W}} \frac{1}{n} \sum_{i=1}^n \mathbb{1}_{\boldsymbol{x}_i \in \mathcal{R}_w} \cdot Loss(y_i, g^w(\boldsymbol{x}_i)),$$

that is the sum of the contributions of the regional prediction functions g^w to the overall training loss.
Finding the partition that minimizes the training loss is a NP-hard problem, therefore a greedy algorithm is employed to construct the tree. At each step, given a splitting rule, it is selected the split that minimizes the most the loss function. The rule splits the data set σ associated to the node ν into σ_T and σ_F. For a classification problem the aim is to choose a splitting rule that minimizes

$$\frac{1}{n} \sum_{(\boldsymbol{x},y) \in \sigma_T} \mathbb{1}_{y \neq y_T^*} + \frac{1}{n} \sum_{(\boldsymbol{x},y) \in \sigma_F} \mathbb{1}_{y \neq y_F^*}$$

where $y_T^* = g^\top(\boldsymbol{x})$ is the most prevalent class in σ_T and y_F^* the most prevalent in σ_F. We can also view the minimization as minimizing a weighted average of impurities of nodes σ_T and σ_F. The two main impurity measures used are Entropy and Gini, which numerically are pretty similar and exchangeable. The termination conditions could be, for example:

- stop when the number of points in a node is less than or equal to some predefined number;
- stop when the tree has reached a maximal predefined depth;
- stop when there is no significant advantage in splitting in terms of training loss.

The quality of a tree is determined by its predictive performance (generalization risk) and the termination condition should strike a balance between minimizing the approximation error and minimizing the statistical error.

11.3 Ensemble Methods

The training procedure of the decision trees leads to a weak learner that easily overfits the data. Nevertheless they are quick to compute and they can be used on categorical values without one-hot encoding. Note that one-hot encoding drives toward an explosion of the problem dimension, making SVM work very bad on the data. Splits in decision trees, on the other hand, codify implicitly the one-hot encoding. To overcome this problem we introduce the Bootstrap Aggregation.

11.3.1 Bootstrap Aggregation

The major idea behind bootstrap aggregation or *bagging* method is to combine predictions learned from multiple data sets, with a view to improving overall prediction accuracy.

Suppose we have B iid copies $\mathcal{T}_1, \dots, \mathcal{T}_B$ of a training set $\mathcal{T}$ and that we train B separate regression models (decision trees) using these sets, obtaining the learners $g_{\mathcal{T}_1}, \dots, g_{\mathcal{T}_B}$. Taking their average we get

$$g_{avg}(\boldsymbol{x}) = \frac{1}{B} \sum_{b=1}^B g_{\mathcal{T}_b}(\boldsymbol{x}).$$

By the law of large numbers, as $B \to \infty$, the average prediction function converges to the expected prediction function $g^\dagger := \mathbb{E}_{g_\mathcal{T}}$. The following results shows that using $g^\dagger$ as a prediction function would result in a lower or equal expected squared-error generalization risk with respect to a general prediction function $g_\mathcal{T}$.

Theorem 11 (Expected Squared-Error Generalization Risk) *Let $\mathcal{T}$ be a random training set and let $\boldsymbol{X}$, Y be a random feature vector and response that are independent of $\mathcal{T}$. Then,*

$$\mathbb{E}\left(Y - g_\mathcal{T}(\boldsymbol{X})\right)^2 \geq \mathbb{E}\left(Y - g^\dagger(\boldsymbol{X})\right)^2 \tag{33}$$

Unfortunately multiple independent data sets are rarely available, but we can substitute them by *bootstrapped* ones. Specifically, instead of the $\mathcal{T}_1, \ldots, \mathcal{T}_B$ sets, we can obtain random training sets $\mathcal{T}_1^*, \ldots, \mathcal{T}_B^*$ by resampling them (**with replacement**) from a single fixed training set τ and use them to train B separate models. In this way we obtain the bootstrapped aggregated estimator (or bagged estimator) of the form:

$$g_{bag}(\boldsymbol{x}) = \frac{1}{B}\sum_{b=1}^{B} g_{\mathcal{T}_b^*}(\boldsymbol{x}).$$

The number of bootstrapped set that we can construct from a data set τ is n^n where $n = |\tau|$. Note: bagging is not performed by choosing random subsets of the training set, but instead by sampling points with replacement to create subset with the same size of the original training set.

(33) is suitable for regression problems, but the bagging idea can be extended to handle classification as well. For example g_{bag} can take the majority vote among $\{g_{\mathcal{T}_b^*}\}_{b=1}^{B}$. It is more effective for predictors that are sensitive to small changes in the training set. Average and bagging in general are only useful for predictors with a large expected variance. Examples of such unstable predictors are decision trees or neural networks, while stable one is, for example, the K-nearest neighbors method. Note that for independent training set $\mathcal{T}_1, \ldots, \mathcal{T}_B$ a reduction of the variance by a factor B is achieved:

$$Var\ g_{bag}(\boldsymbol{x})\ = \frac{1}{B} Var\ g_{\mathcal{T}}(\boldsymbol{x})$$

11.3.2 Out of bag observations

On average about a third $(1/e)$ of the original sample points are not included in bootstrapped set $\mathcal{T}_b^*$ for $1 \leq b \leq B$. These samples can be used for the loss estimation and are called *out-of-bag* (OOB) observations. The OOB score is a way to validate the bagging classifier (even if it is different from the validation score): each point of the original training set will not be included into a subset of predictors, therefore we test each point on that subset and we obtain (using majority vote, average, ...) the OOB score, computed as the number of correctly predicted points from the OOB sample. Note that OOB score is calculated using only a subset of predictors not containing the OOB sample in their bootstrap training dataset, while the validation score is calculated using all the predictors of the ensemble.

11.3.3 Correlated Predictions

Let $Z_b = g_{\mathcal{T}_b}(\boldsymbol{x}),\ b = 1, \ldots, B$ be iid prediction values obtained from independent training sets $\mathcal{T}_1, \ldots, \mathcal{T}_B$ with $Var\ Z_b = \sigma^2, \forall b$. Then the variance of the average prediction is $\bar{Z}_B = \sigma^2/B$. However, if bootstrapped sets $\{\mathcal{T}_b^*\}$ are used, the random variables $\{Z_b\}$ will be correlated. In particular they are identically distributed (but not independent!) with some positive pairwise correlation ρ. It holds that:

$$Var\ \bar{Z}_b = \rho\sigma^2 + \sigma^2\frac{(1-\rho)}{B}.$$

While the second term goes to zero as B increases, the first term remains constant.

11.3.4 Random Forests

This correlated prediction issues is particularly relevant for bagging with decision trees. Suppose there is a feature that provides a very good split of the data. That feature would be selected at root level by every decision tree, leading to highly correlated predictions. In this case averaging will not introduce the desired improvement in the performance of the bagged predictor.

The major idea of random forest is to bagging in combination with decorrelation of the trees by including only a subset of features during the tree construction. For each training set $\mathcal{T}_b^*$ we build a decision tree using a subset of $m \leq p$ features for the splitting rules. A more sophisticated way is to choose randomly m predictors out of p for each split of each bagged tree (choose one random subset of features at each split).

For addressing the interpretability issue, we can evaluate the *feature importance* of features and of the random forest. For splitting rules of the type $\mathbb{1}_{x_j \leq \xi}$ each node ν is associated with a feature x_j that determines the split. Also each internal node ν induces a training loss decrease $\Delta_{Loss}(\nu)$. We can define the feature importance of x_j as:

$$\mathcal{I}_{\mathbb{T}}(x_j) = \sum_{\nu_{internal} \in \mathbb{T}} \Delta_{Loss}(\nu) \cdot \mathbb{1}_{[x_j \text{ is associated with } \nu]}, \quad 1 \leq j \leq p.$$

The feature importance of x_j of the random forest is the average feature importance of x_j of the trees:

$$\mathcal{I}_{RF}(x_j) = \frac{1}{B}\sum_{b=1}^{B} \mathcal{I}_{\mathbb{T}_b}(x_j), \quad 1 \leq j \leq p.$$

12 Lesson 12 - June 03

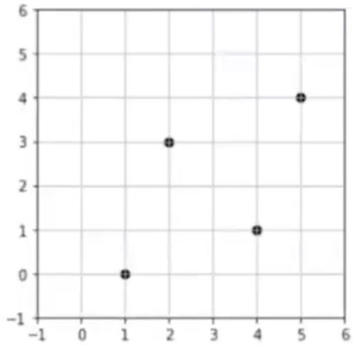
12.1 Exercise 1

Text Consider the following design matrix, representing four sample points $x_i \in \mathbb{R}^2$.

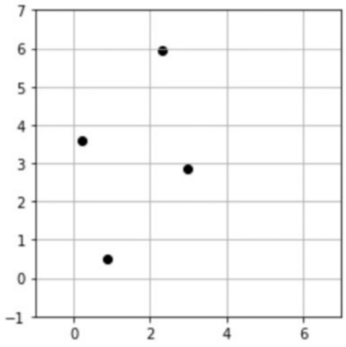
$$X = \begin{bmatrix} 4 & 1 \\ 2 & 3 \\ 5 & 4 \\ 1 & 0 \end{bmatrix}$$

We want to represent the data in only one dimension, so we turn to principal components analysis (PCA).

1. Compute the unit-length principal component directions of X, and state which one the PCA algorithm would choose if you request just one principal component. Please provide an exact answer without approximations.
2. The plot below depicts the sample points from X. We want a one-dimensional representation of the data, so draw the principal component direction (as a line) and the projections of all four sample points onto the principal direction. Label each projected point with its principal coordinate value (where the origin's principal coordinate is zero). Give the principal coordinate values exactly.



3. The plot below depicts the sample points from X rotated 30 degrees counterclockwise about the origin. Identify the principal component direction that the PCA algorithm would choose and draw it (as a line) on the plot. Also draw the projections of the rotated points onto the principal direction. Label each projected point with the exact value of its principal coordinate.



Solution part 1 First of all we center the data. To do so, we first compute the average of each column of X, and then we subtract the mean to each sample. We have that $\mu_1 = (4 + 2 + 5 + 1)/4 = 3$ and $\mu_2 = (1 + 3 + 4 + 0)/4 = 2$, then the centered data matrix is the following.

$$\overline{X} = \begin{bmatrix} +1 & -1 \\ -1 & +1 \\ +2 & +2 \\ -2 & -2 \end{bmatrix}$$

Now we compute the centered scatter.

$$\Sigma = \overline{X}^T\overline{X} = \begin{bmatrix} +1 & -1 & +2 & -2 \\ -1 & +1 & +2 & -2 \end{bmatrix} \cdot \begin{bmatrix} +1 & -1 \\ -1 & +1 \\ +2 & +2 \\ -2 & -2 \end{bmatrix} = \begin{bmatrix} 10 & 6 \\ 6 & 10 \end{bmatrix}$$

To obtain the sample covariance, we divide the centered scatter matrix by the number of samples. We obtain

$$\frac{1}{4} \cdot \begin{bmatrix} 10 & 6 \\ 6 & 10 \end{bmatrix}$$

In the following steps, we will use the centered scatter matrix because we do not care about the magnitude and computations are easier with integers. Now we find eigenvalues and eigenvectors of Σ, so we have the principal components and the related variance. Here the computation of the eigenvalues.

$$\mathcal{X}_\Sigma(\Sigma - t \cdot I_2) = \det \begin{pmatrix} 10-t & 6 \\ 6 & 10-t \end{pmatrix} = (10-t)^2 - 36 = t^2 - 20t + 64$$

$$t = \frac{20 \pm \sqrt{400 - 4 \cdot 64}}{2} = \frac{20 \pm 12}{2} = 16, \ 4$$

Now we compute the associated eigenvector to the eigenvalue 4.

$$\ker(\Sigma - 4 \cdot I_2) = \ker \begin{pmatrix} 6 & 6 \\ 6 & 6 \end{pmatrix}$$

We know that the kernel is the set of vectors which multiplied by the matrix return the null vector. Thus, we need to solve the following system.

$$\begin{bmatrix} 6 & 6 \\ 6 & 6 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \iff \begin{cases} 6x_1 + 6x_2 = 0 \\ 6x_1 + 6x_2 = 0 \end{cases} \iff x_1 = -x_2$$

The eigenvector associated with 4 is $a = [1, -1]^T$. Now we do the same for 16.

$$\ker(\Sigma - 16 \cdot I_2) = \ker \begin{pmatrix} -6 & +6 \\ +6 & -6 \end{pmatrix}$$

$$\begin{bmatrix} -6 & +6 \\ +6 & -6 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \iff \begin{cases} -6x_1 + 6x_2 = 0 \\ 6x_1 - 6x_2 = 0 \end{cases} \iff x_1 = x_2$$

Thus the eigenvector associated with 16 is $b = [1, 1]^T$. Now we want an orthonormal basis, so the two eigenvectors must be orthogonal and normal.

$$\begin{bmatrix} 1 & -1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 1 \end{bmatrix} = 0 \implies a \perp b$$

but $||a||_2 = ||b||_2 = \sqrt{2}$, then our principal components are $a/\sqrt{2}$ and $b/\sqrt{2}$. We order the principal components in descending order according to their magnitude, so

$$\text{pc1} \ = \begin{pmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{pmatrix} \ \text{pc2} \ = \begin{pmatrix} 1/\sqrt{2} \\ -1/\sqrt{2} \end{pmatrix}$$

PCA will project the points on pc1.

Solution part 2 You just need to compute the scalar products between the **centered** points and the direction pc1. Then you can make a plot of the centered points with pc1 and their projections onto it.

$$\begin{bmatrix} +1 & -1 \\ -1 & +1 \\ +2 & +2 \\ -2 & -2 \end{bmatrix} \cdot \begin{bmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 4/\sqrt{2} \\ -4/\sqrt{2} \end{bmatrix}$$

You could also project the original data matrix using the same principal component.

Solution part 3 We rotate the points of an angle $\theta = 30° = \pi/6$. We have that $\cos\theta = \sqrt{3}/2$ and $\sin\theta = 1/2$, then the rotation matrix is

$$\rho_\theta = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} = \frac{1}{2}\begin{pmatrix} \sqrt{3} & -1 \\ 1 & \sqrt{3} \end{pmatrix}$$

The rotated data matrix is $Y = \overline{X} \cdot \rho_\theta$, then

$$\Sigma_Y = Y^T Y = \rho_\theta^T \overline{X}^T \overline{X} \rho_\theta = \rho_\theta^T \Sigma_X \rho_\theta$$

Note that $\rho_\theta^T = \rho_\theta^{-1}$. **Under rotation, the value of the projection on pc1 remain the same**, because Σ_Y has the same eigenvalues of Σ_X. The PCA is invariant under rotation. Note that the principal components are different, but the coordinate of the projected points remain the same.

12.2 Exercise 2

Text Recall that PCA transorms zero-mean data into low-dimensional reconstructions that lie in the span of the top k eigenvectors of the sample covariance matrix. Let $\mathbf{U}_k$ denote the $d \times k$ matrix of the top k eigenvectors of the covariance matrix ($\mathbf{U}_k$ is a truncated version of $\mathbb{U}$, which is the matrix of eigenvectors of the covariance matrix). There are two approaches to computing the low-dimensional reconstruction of $\mathbf{w} \in \mathbb{R}^k$ of a data point $\mathbf{x} \in \mathbb{R}^d$:

1. Solve a least squares problem to minimize the reconstruction error.
2. Project $\mathbf{x}$ onto the span of the columns of $\mathbf{U}_k$.

In this problem, you will show that these approaches are equivalent.

1. Formulate least squares problem in terms of $\mathbf{U}_k, \mathbf{x}$, and the variable $\mathbf{w}$. (Hint: this optimization problem should resemble the linear regression).
2. Show that the solution of the least squares problem is equal to $\mathbf{U}_k^T\mathbf{x}$, which is the projection of $\mathbf{x}$ onto the span of the columns of $\mathbf{U}_k$.

Solution part 1 The reconstruction error for the point i is $||U_k w_i - x_i||_2^2$, therefore we want to minimize

$$\min_w ||U_k w - x||^2$$

The minimizer of a least squares problem is the solution of a multi-linear regression, i.e., $w^* = (U_k^T U_k)^{-1} U_k^T x$ (this holds when U_k is full rank, but since U is orthonormal it is always full rank). Or, since it is a convex problem, you can find the gradient and imposing it equal to zero, to find the only stationary point, which is a global minima.

Solution part 2 We show that the solution to the least squares problem is $U_k^T x$.

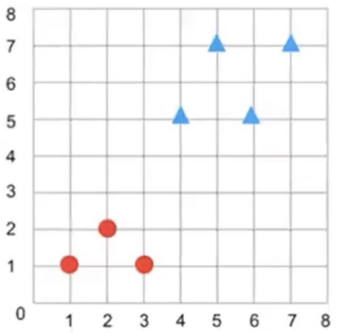
$$w^* = (U_k^T U_k)^{-1} U_k^T x$$
$$U_k^T U_k = I_k \implies w^* = U_k^T x \ \square$$

12.3 Exercise 3

Text Recall that a maximum margin classifier, also known as a hard-margin support vector machine (SVM), takes n training points $X_1, X_2, \ldots, X_n \in \mathbb{R}^d$ with labels $y_1, y_2, \ldots, y_n \in \{+1, -1\}$, and finds parameters $w \in \mathbb{R}^d$ and $\alpha \in \mathbb{R}$ that satisfy a certain objective function to the constraints

$$y_i(X_i \cdot w + \alpha) \geq 1, \ \forall i \in \{1, \ldots, n\}$$

For parts (1) and (2), consider the following training points. Circles are classified as positive examples with label +1 and triangles are classified as negative examples with label -1.



For parts (3)-(6) forget about the figure above, but assume that there is at least one sample point in each class and that the sample points are linearly separable.

1. Which are the support vectors? Write as $\begin{bmatrix}\text{horizontal}\\\text{vertical}\end{bmatrix}$. E.g., the bottom right circle is $\begin{bmatrix}3\\1\end{bmatrix}$.
2. If we add the sample point $x = \begin{bmatrix}5\\1\end{bmatrix}$ with label -1 (triangle) to the training set, which points are the support vectors?
3. Describe the geometric relationship between w and the decision boundary.
4. Describe the relationship between w and the margin. (For the purposes of this question, the margin is just a number).
5. Knowing what you know about the hard-margin SVM objective function, explain why for the optimal (w, α), there must be at least one sample point for which $X_i \cdot w + \alpha = 1$ and one sample point for which $X_i \cdot w + \alpha = -1$.
6. If we add new features to the sample points (while retaining all the original features), can be the optimal $||w_{new}||$ in the enlarged SVM be greater than the optimal $||w_{old}||$ in the original SVM? Can it be smaller? Can it be the same? Explain why.

Solution part 1 The support vectors are

$$\begin{bmatrix}2\\2\end{bmatrix}, \begin{bmatrix}4\\5\end{bmatrix}$$

because they are the nearest points to the solution hyperplane.

find two closest point from different classes (finding minimum margin) and get the middle of them to draw the hyper-plane

Solution part 2 The new support vectors are

$$\begin{bmatrix}3\\1\end{bmatrix}, \begin{bmatrix}5\\1\end{bmatrix}$$

dmin/2 is the margin and hyperplane direction,then any point with this distance to hyperplane are the support vectors

Random theory added to solve this exercise during the lecture. We can express the SVM in terms of convex geometry. m is the number of samples and $< \bullet, \bullet >$ is the scalar product.

$$\begin{cases}\min_\alpha \frac{1}{2}\sum_{i=1}^m \sum_{j=1}^m y_i \cdot y_j \cdot \alpha_i \cdot \alpha_j \cdot < x_i, x_j > - \sum_{i=1}^m \alpha_i \\ \sum_{i=1}^m \alpha_i y_i = 0 \quad 0 \leq \alpha_i \leq C\end{cases}$$

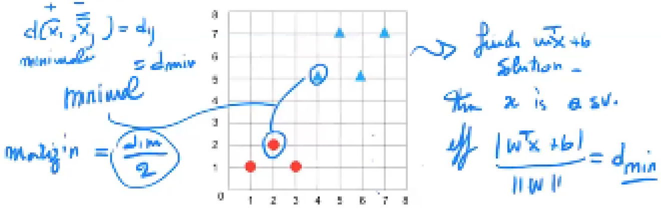
A set A is convex if and only if $\forall t \in [0,1) : ta + (1-t)a' \in A \forall a, a' \in A$ (i.e., the hyperplane joining the two points is contained in the set).

Proposition: The distance between a point $p \in \mathbb{R}^d$ and a convex set C, $p \notin C$, is the maximal of the distances between p and the hyper-planes separating p from C.

i. If C, D are two disjointed convex sets, $C \cap D = 0$, then exists an hyper-plane $w^T x + b = 0$ separating them, i.e., $w^T x + b \geq 0$ if $x \in C$, $w^T x + b < 0$ if $x \in D$.

ii. if $p \notin C$, then $\exists w x^T + b \mid w^T p + b > 0$ and $w^T x + b < 0 \ \forall x \in C$.

iii. $\forall A \subset \mathbb{R}^d \ \exists!$ minimal convex subset $H(A) \subset \mathbb{R}^d$ such that $A \subset H(A)$, i.e., if $A \subset S$ then $H(A) \subset S \ \forall S$. Then $H(A)$ is the *convex hull* of A, i.e., $H(A) = \{\sum_{a \in A} \mid \sum t_a = 1, t_a \in [0,1], \forall a\}$. This means that if you have a set composed of two points, its convex hull is a segment; if you take 3 points, $H(A)$ is a simplex (triangle), etc...



Then SVM is rewritten as

$$\begin{cases} min_{\alpha} \frac{1}{2}||\tilde{x} - \bar{x}|| \\ \sum_{i \in S_{-1}} \alpha_i = \sum_{j \in S_{+1}} \alpha_j = 1 \\ S_{+1} = \{i \mid y_i = +1\} \\ S_{-1} = \{i \mid y_i = -1\} \\ \tilde{x} \in H(S_{-1}) \\ \bar{x} \in H(S_{+1}) \\ \alpha_i \geq 0 \; \forall i \in [m] \end{cases}$$

and the solutions to this problem are contained in the set of solutions to the original SVM problem. Considering π as the solution hyper-plane, then

i. $\pi \perp w = \sum_{i=1}^{m} y_i \alpha_i x_i$

ii. π is at the halfway between the convex hulls of the two classes

Solution part 3 The decision boundary π is orthogonal to w. Note that w is the vector joining the support vectors. Furthermore, the decision boundary can be rewritten as $\text{sign}(w^T x + b)$.

Solution part 4 The distance from π (the decision boundary) to the margins is $1/||w||$.

Solution part 5 The objective is to minimize $||w||^2$ (or $||w||$). If for every sample point we have $y_i(X_i \cdot w + \alpha) > 1$ we can simply scale w to make it smaller until there is a point such that $y_i(X_i \cdot w + \alpha) = 1$ (improve the solution, because we are minimizing, we take w smaller). Then there should be some (X_i, y_i) such that $y_i(X_i \cdot w + \alpha) = 1$, but for every negative sample $y_i(X_i \cdot w + \alpha) < -1$ we can make α greater such that we have for all samples $y_i(X_i \cdot w + \alpha) > 1$, and then we can shrink more w, then no solution is optimal (you can continue shrinking $||w||$).

Key point: if no points lies on the margins, you can continue increasing α or decreasing $||w||$ and you will not find a unique solution, but there is one because it is a convex problem.

Solution part 6 We add a feature independent to the other in the dataset $\{x_1, \ldots, x_m\} \subset \mathbb{R}^d$, so now it is in $\mathbb{R}^{d+1}$. Can be the optimal $||w_{new}||$ in this enlarged SVM greater of the $||w_{old}||$ (the solution to the SVM on the original dataset)? It can be **equal or smaller** but not greater. It can be the same only if the new feature has only zero values. The smaller is $||w||$ the bigger is the margin since $M = 1/||w||$, so the new feature can by chance take the two classes arbitrarily apart (i.e., with a larger margin).

The margin cannot be smaller (i.e., $||w||$ cannot be bigger). When we add the new feature we create the new solution (w_{new}, α) which has the same value as w_{old} if the new feature has only zero value. In this case, you have that $w_{new} \cdot x_{new} = w_{old} \cdot x_{old}$ and the optimal for $||w_{new}||$ must satisfy also $||w_{old}||$ (because the latter is a sub-problem of the first). Then the solution to $||w_{new}||$ cannot be greater than the solution of $||w_{old}||$, otherwise it would not be a solution to $||w_{old}||$ (since it is a minimization problem).

A Exercises on VC dimension

For the following classes of functions F, compute the VC dimension. Note that $\mathbb{1}$ is the indicator function, i.e., 1 if the condition is satisfied, 0 otherwise. The domain is $\chi = (0, 1)$.

A.1 Exercise 1

Text

$$F = \{f : \chi \to \{0, 1\}, f(x) = \mathbb{1}_{x<t}, t \in [0, 1]\}$$

Solution You can easily see that this class shutters the set composed of 1 sample. If the only sample is $x_1 \in \chi$, then it can have label 0 or 1. For label 0 just pick $t \leq x_1$, while for label 1 $t > x_1$. Thus, VCdim$(F) \geq 1$ □. Now, consider a set composed of two points, $x_1 < x_2$ (without loss of generality). If the corresponding labels are $y_1 = 0$ and $y_2 = 1$, then there is no function $f \in F$ which shutters this combination. In fact you have three option for choosing t:

- $t > x_2 > x_1$ then $y_1 = y_2 = 1$
- $x_1 < t \leq x_2$ then $y_1 = 1$ and $y_2 = 0$
- $t \leq x_1 \leq x_2$ then $y_1 = y_2 = 0$

As you can see, the combination $y_1 = 0$ and $y_2 = 1$ never appears, thus VCdim$(F) \leq 1$ □.
We can conclude that VCdim$(F) = 1$.

A.2 Exercise 2

Text

$$F = \{f : \chi \to \{0, 1\},\ f(x) = \mathbb{1}_{x<t} \cup f(x) = 1 - \mathbb{1}_{x<t} = \mathbb{1}_{x\geq t},\ t \in [0, 1]\}$$

Solution First of all, we show that each possible labelling with a set of two samples $x_1 < x_2$ is shuttered.

y_1	y_2	t	$f(x)$
0	0	1	$f(x) = \mathbb{1}_{x\geq t}$
0	1	$x_1 < t < x_2$	$f(x) = \mathbb{1}_{x\geq t}$
1	0	$x_1 < t < x_2$	$f(x) = \mathbb{1}_{x<t}$
1	1	1	$f(x) = \mathbb{1}_{x<t}$

Then VCdim$(F) \geq 2$ □ Now we show that when the set is composed of three samples $x_1 < x_2 < x_3$ the class does not shutter any possible combination of labelling. Thus, we conclude that VCdim$(F) = 2$ □

y_1	y_2	y_3	Shuttered?
0	0	0	YES
0	0	1	YES
0	1	0	NO
0	1	1	YES
1	0	0	YES
1	0	1	NO
1	1	0	YES
1	1	1	YES

A.3 Exercise 3

Text

$$F = \{f : \chi \to \{0,1\}, f(x) = \mathbb{1}_{t_1 \leq x < t_2},\ t_1 < t_2 \in [0,1]\}$$

Solution We show that when the dataset is composed by two samples $x_1 < x_2$ all the possible labelling are covered by the class.

y_1	y_2	t_1 & t_2
0	0	$t_1 < t_2 < x_1 < x_2$
0	1	$x_1 < t_1 < x_2 < t_2$
1	0	$t_1 < x_1 < t_2 < x_2$
1	1	$t_1 < x_1 < x_2 < t_2$

Then the VC dimension of this class is at least 2. Consider now a dataset composed of three samples $x_1 < x_2 < x_3$. We have the following table. As you can see, there are no parameters t_1, t_2 which cover the labelling $y_1 = y_3 = 1$ and $y_2 = 0$, therefore the VC dimension must be less than 3. Joining this statement with the previous, we conclude that $\text{VCdim}(F) = 2$ □

y_1	y_2	y_3	t_1 & t_2
0	0	0	$t_1 < t_2 < x_1 < x_2 < x_3$
0	0	1	$x_1 < x_2 < t_1 < x_3 < t_2$
0	1	0	$x_1 < t_1 < x_2 < t_2 < x_3$
0	1	1	$x_1 < t_1 < x_2 < x_3 < t_2$
1	0	0	$t_1 < x_1 < t_2 < x_2 < x_3$
1	0	1	$\nexists\ t_1, t_2$
1	1	0	$t_1 < x_1 < x_2 < t_2 < x_3$
1	1	1	$t_1 < x_1 < x_2 < x_3 < t_3$

A.4 Exercise 4

Text

$$F = \{f : \chi \to \{0,1\},\ f_1(x) = \mathbb{1}_{t_1 \leq x < t_2} \cup f_2(x) = 1 - \mathbb{1}_{t_1 \leq x < t_2} = \mathbb{1}_{x < t_1 \cup x \geq t_2},\ t_1 < t_2 \in [0,1]\}$$

Solution You can easily see that this class cover also the case left behind by the previous exercise. In fact, when $y_1 = y_3 = 1$ and $y_2 = 0$ we can choose the function $f_2(x)$ with parameters $x_1 < t_1 < x_2 < t_2 < x_3$. Then the VC dimension is at least 3. Anyway, we show now that when we have four samples $x_1 < x_2 < x_3 < x_4$ not all possible labelling are covered. As you can see, two combination are not reached. Therefore the VC dimension is exactly 3.

y_1	y_2	y_3	y_4	t_1 & t_2	f_1 or f_2
0	0	0	0	$t_1 < t_2 < x_1 < x_2 < x_3 < x_4$	1
0	0	0	1	$x_1 < x_2 < x_3 < t_1 < x_4 < t_2$	1
0	0	1	0	$x_1 < x_2 < t_1 < x_3 < t_2 < x_4$	1
0	0	1	1	$x_1 < x_2 < t_1 < x_3 < x_4 < t_2$	1
0	1	0	0	$x_1 < t_1 < x_2 < t_2 < x_3 < x_4$	1
0	1	0	1	$\nexists\ t_1, t_2$	/
0	1	1	0	$x_1 < t_1 < x_2 < x_3 < t_2 < x_4$	1
0	1	1	1	$x_1 < t_1 < x_2 < x_3 < x_4 < t_2$	1
1	0	0	0	$t_1 < x_1 < t_2 < x_2 < x_3 < x_4$	1
1	0	0	1	$x_1 < t_1 < x_2 < x_3 < t_2 < x_4$	2
1	0	1	0	$\nexists\ t_1, t_2$	/
1	0	1	1	$x_1 < t_1 < x_2 < t_2 < x_3 < x_4$	2
1	1	0	0	$t_1 < x_1 < x_2 < t_2 < x_3 < x_4$	1
1	1	0	1	$x_1 < x_2 < t_1 < x_3 < t_2 < x_4$	2
1	1	1	0	$t_1 < x_1 < x_2 < x_3 < t_2 < x_4$	1
1	1	1	1	$t_1 < x_1 < x_2 < x_3 < x_4 < t_2$	1

A.5 Exercise 5

Text

$$F = \{f : \chi \to \{0,1\},\ f(x) = \sum_{i=0}^{k} \mathbb{1}_{t_{2i}<x<t_{2i+1}},\ 0 \leq t_0 \leq t_1 \leq \cdots \leq t_{2k+1}\}\ \forall k \geq 1$$

Solution With this class of function, we have $k+1$ intervals which produce the label 1 when a sample is placed inside. Those intervals are:

- $i = 0 \to (t_0, t_1)$
- $i = 1 \to (t_2, t_3)$
- $i = 2 \to (t_4, t_5)$
- $\dots$
- $i = k \to (t_{2k}, t_{2k+1})$

As you can see from previous exercise, the case which fucks up the situation is always when labels are alternating, i.e., $(1, 0, 1, 0, 1, 0, \dots)$. Then this is the worst possible labeling, and if we show that the class reach this one with size m then it reach all the labelling of the same size. If there are $k+1$ labels equal to one and as many others equal to zero alternating, you can find a function in the class which covers this situation, choosing as parameters $t_0 < x_1 < t_1 < x_2 < t_2 < x_3 < t_3 < x_4 < \dots$ (if the first sample has label 1). Otherwise if the first sample has label 0, then you should choose $x_1 < t_0 < x_2 < t_1 < x_3 < t_2 < \dots$ Thus the VC dimension is at least $2(k+1)$.

Now consider a bigger dataset with size $2k+3$. You can shutter the alternating combination which starts and ends with 0, but you cannot shutter it if all the values are flipped, i.e., the front and the tail are equal to 1, because in this case you have a number of positive labels equal to $k+2$. Then the VC dimension of this class is equal to $2k+2$ □