Lecture Notes for

Mathematics of Machine Learning (401-2684-00L at ETH Zurich)

Afonso S. Bandeira & Nikita Zhivotovskiy
ETH Zurich

Last update on August 16, 2021

SYLLABUS

Introductory course to Mathematical aspects of Machine Learning, including Supervised Learning, Unsupervised Learning, Sparsity, and Online Learning.

Course Coordinator: Pedro Abdalla Teixeira
pedro.abdallateixeira@ifor.math.ethz.ch

The contents of the course will depend on the speed and feedback received during the semester, a tentative plan is:

(1) **Unsupervized Learning and Data Parsimony:**
- Clustering and k-means
- Singular Value Decomposition
- Low Rank approximations and Eckart–Young–Mirsky Theorem
- Dimension Reduction and Principal Component Analysis
- Matrix Completion and the Netflix Prize
- Overcomplete Dictionaries and Finite Frame Theory
- Sparsity and Compressed Sensing
- Introduction to Spectral Graph Theory.

(2) **Supervized and Online Learning:**
- Introduction to Classification and Generalization of Classifiers
- Some concentration inequalities
- Stability and VC Dimension
- Online Learning: Learning with expert advice and exponential weights
- A short introduction to Optimization and Gradient Descent

Note for non-Mathematics students: this class requires a certain degree of mathematical maturity–including abstract thinking and the ability to understand and write proofs.

Please visit the Forum at
https://forum.math.ethz.ch/c/spring-2021/mathematics-of-machine-learning/49
for more information.

Please excuse the lack of polishing and typos in this draft. If you find any typos, please let us know! This draft was last updated on August 16, 2021.

You will notice several questions along the way, separated into Challenges (and Exploratory Challenges.

- **Challenges** are well-defined mathematical questions, of varying level of difficulty. Some are very easy, and some are much harder than any homework problem.
- **Exploratory Challenges** are not necessarily well defined, but thinking about them should improve your understanding of the material.

We also include a few "Further Reading" references in case you are interested in learning more about a particular topic.

Some of the material in the first half of the notes is adapted from [BSS]. The book [BSS] is more advanced and tends to have a probabilistic viewpoint, but you might enjoy reading through it. The first author has written a set of lecture notes for a similar advanced course that contains many open problems [Ban16].

The authors would like to thank the participants of the course for their useful comments and valuable feedback. We are also grateful to Daniel Paleka for his help in proofreading the initial draft.

CONTENTS

1. INTRODUCTION (25.02.2021)

We will study four areas of Machine Learning and Analysis of Data, focusing on the mathematical aspects. The four areas are:

- Unsupervised Learning: The most common instance in exploratory data analysis is when we receive data points without a priori known structure, think e.g. unlabeled images from a databased, genomes of a population, etc. The natural first question is to ask if we can learn the geometry of the data. Simple examples include: Does the data separate well into clusters? Does the data naturally live in a smaller dimensional space? Sometimes the dataset comes in the form of a network (or graph) and the same questions can be asked, an approach in this case is with Spectral Graph Theory which we will cover if time permits.
- Parsimony and Sparsity: Sometimes, the information/signal we are after has a particular structure. In the famous Netflix prize the goal is to predict the user-movie preferences (before the user watches that particular movie) from ratings from other user-movie pairs; the matrix of ratings is low-rank and this structure can be leveraged. Another common form of parsimony is sparsity in a particular linear dictionary, such as natural images in the Wavelet basis. We will present the basics of sparsity, Compressed Sensing, and Matrix Completion, without requiring any advanced Probability Theory.
- Supervised Learning: Another common problem is that of classification. As an illustrative example, consider one receives images of cats and dogs with the respective correct labels, the goal is then to construct a classifier that generalises well, i.e. given a new unseen image predicts correctly whether it is a cat or a dog. Here basic notions of probability and concentration of measure are used to understand generalisation via stability and VC dimension. If time permits we will describe Neural networks and the back-propagation algorithm.
- Online Learning: Some tasks need to be performed, and learned, in real time rather than receiving all the data before starting the analysis. Examples include much of the advertisement auctioning online, real time update of portfolios, etc. Here we will present the basic of the Mathematics of Online learning, including learning with expert advice and the multiplicative weights algorithm.

2. CLUSTERING AND k-MEANS (25.02.2021)

Clustering is one of the central tasks in machine learning. [1] Given a set of data points, the purpose of clustering is to partition the data into a set of clusters where data points assigned to the same cluster correspond to similar data (for example, having small distance to each other if the points are in Euclidean space).

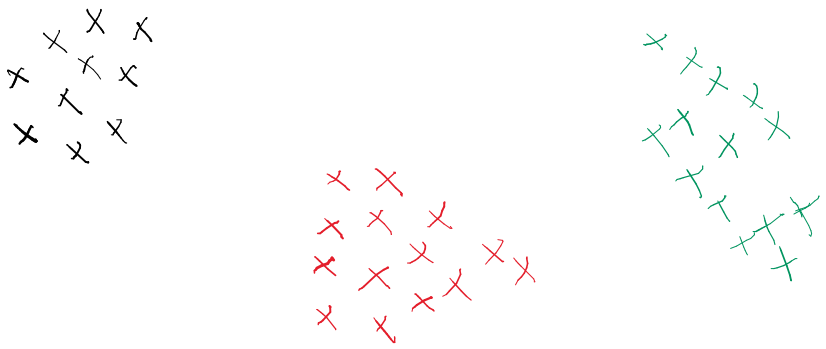


FIGURE 1. Examples of points separated in clusters.

2.0.1. *k-means Clustering.* One the most popular methods used for clustering is k-means clustering. Given $x_1, \dots, x_n \in \mathbb{R}^p$, the k-means clustering partitions the data points in clusters $S_1, \dots, S_k$ with centers $\mu_1, \dots, \mu_k \in \mathbb{R}^p$ as the solution to:

$$\min_{\substack{\text{partition } S_1,\dots,S_k \\ \mu_1,\dots,\mu_k}} \sum_{l=1}^{k} \sum_{i \in S_l} \|x_i - \mu_l\|^2 . \tag{1}$$

A popular algorithm attempts to minimize (1), Lloyd's Algorithm [Llo82] (this is also sometimes referred to as simply "the k-means algorithm"). It relies on the following two observations

Proposition 2.1.

- *Given a choice for the partition $S_1 \cup \cdots \cup S_k$, the centers that minimize* (1) *are given by*
$$\mu_l = \frac{1}{|S_l|} \sum_{i \in S_l} x_i.$$
- *Given the centers $\mu_1, \dots, \mu_k \in \mathbb{R}^p$, the partition that minimizes* (1) *assigns each point x_i to the closest center μ_k.*

Challenge 2.1. *Prove this fact.*

Lloyd's Algorithm is an iterative algorithm that starts with an arbitrary choice of centers and iteratively alternates between

- Given centers $\mu_1, \ldots, \mu_k$, assign each point x_i to the cluster
$$l = \text{argmin}_{l=1,\ldots,k} \|x_i - \mu_l\| .$$
- Update the centers $\mu_l = \frac{1}{|S_l|} \sum_{i \in S_l} x_i$,

until no update is taken.

Unfortunately, Lloyd's algorithm is not guaranteed to converge to the solution of (1). Indeed, Lloyd's algorithm oftentimes gets stuck in local optima of (1). In fact optimizing (1) is *NP*-hard and so there is no polynomial time algorithm that works in the worst-case (assuming the widely believed conjecture $P \neq NP$).

Challenge 2.2. *Show that Lloyd's algorithm converges*[2] *(even if not always to the minimum of* (1)*).*

Challenge 2.3. *Can you find an example of points and starting centers for which Lloyd's algorithm does not converge to the optimal solution of* (1)*?*

Exploratory Challenge 2.4. *How would you try to "fix" Lloyd's Algorithm to avoid it getting stuck in the example you constructed in Challenge 2.3?*

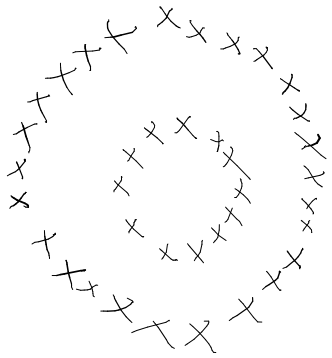


FIGURE 2. Because the solutions of k-means are always convex clusters, it is not able to handle some cluster structures.

While popular, k-means clustering has some potential issues:

- One needs to set the number of clusters a priori. . A typical way to overcome this issue is to try the algorithm for different numbers of clusters.

[1]This section is less mathematical, it is a "warm-up" for the course.

[2]In the sense that it stops after a finite number of iterations.

- The way the formula (1) is defined needs the points to be defined in an Euclidean space. Often we are interested in clustering data for which we only have some measure of affinity between different data points, but not necessarily an embedding in $\mathbb{R}^p$ (this issue can be overcome by reformulating (1) in terms of distances only — *you will do this on the first homework problem set.*).
- The formulation is computationally hard, so algorithms may produce suboptimal instances.
- The solutions of k-means are always convex clusters. This means that k-means cannot find clusters such as in Figure 2.

Further Reading 2.5. *On the computational side, there are many interesting questions regarding when the k-means objective can be efficiently approximated, you can see a few open problems on this in* [Ban16] *(for example Open Problem 9.4).*

3. THE SINGULAR VALUE DECOMPOSITION (04.03.2021)

Data is often presented as a $d \times n$ matrix whose columns correspond to n data points in $\mathbb{R}^d$. Other examples include matrices of interactions where the entry (i, j) of a matrix contains information about an interaction, or similarity, between an item (or entity) i and j.

The Singular Value Decomposition is one of the most powerful tools to analyze matrix data.

Given a matrix $X \in \mathbb{R}^{n\times m}$, its Singular Value Decomposition is given by

$$X = U\Sigma V^T,$$

where $U \in O(n)$ and $V \in O(m)$ are orthogonal matrices, and $\Sigma \in \mathbb{R}^{n\times m}$ is a rectangular diagonal matrix, in the sense that $\Sigma_{ij} = 0$ for $i \neq j$, and whose diagonal entries are non-negative.

The diagonal entries $\sigma_1 \geq \sigma_2 \geq \ldots, \sigma_{\min\{n,m\}}$ of Σ are called the singular values[3] of X. Recall that unlike eigenvalues they must be real and non-negative. The columns u_k and v_k of respectively U and V are called, respectively, the left and right singular vectors of X.

Proposition 3.1 (Some basic properties of SVD)**.**

- $\operatorname{rank}(X)$ *is equal to the number of non-zero singular values of* X.
- *If* $n \leq m$*, then the singular values of* X *are the square roots of the eigenvalues of* XX^T*. If* $m \leq n$ *they are the square roots of the eigenvalues of* X^TX.

Challenge 3.1. *Prove this fact.*

The SVD can also be written in more economic ways. For example, if $\operatorname{rank}(X) = r$ then we can instead write

$$X = U\Sigma V^T,$$

where $U^TU = I_{r\times r}$, $V^TV = I_{r\times r}$, and Σ is a non-singular $r \times r$ diagonal matrix matrix. Note that this representation only requires $r(n+m+1)$ numbers, which if $r \ll \min\{n,m\}$, is considerable savings when compared to nm.

It is also useful to write the SVD as

$$X = \sum_{k=1}^{r} \sigma_k u_k v_k^T,$$

where σ_k is the k-th largest singular value, and u_k and v_k are the corresponding left and right singular vectors.

[3]The most common convention is that the singular values are ordered in decreasing order, it is the convention we observe here.

4. Low rank approximation of matrix data (04.03.2021)

A *key observation in Machine Learning and Data Science* is that (matrix) data is oftentimes well approximated by low-rank matrices. You will see examples of this phenomenon both in the lecture and the code simulations available on the class webpage.

In order to talk about what it means for a matrix B to approximate another matrix A, we need to have a notion of distance between matrices of the same dimensions, or equivalently a notion of norm of $A-B$. Let us start with some classical norms.

Definition 4.1 (Spectral Norm)**.** *The spectral norm of $X \in \mathbb{R}^{n\times m}$ is given by*

$$\|X\| = \max_{v:\|v\|=1} \|Xv\|_2,$$

or equivalently $\|X\| = \sigma_1(X)$.

Challenge 4.1. *Show that the two definitions above are equivalent.*

Another common matrix norm is the Frobenius norm.

Definition 4.2 (Frobenius norm (or Hilbert-Schmidt norm))**.** *The Frobenius norm of $X \in \mathbb{R}^{n\times m}$ is given by*

$$\|X\|_F^2 = \sum_{i=1}^{n}\sum_{j=1}^{m} X_{ij}^2.$$

Challenge 4.2. *Show that*

$$\|X\|_F^2 = \sum_{i=1}^{\min\{n,m\}} \sigma_i(X)^2.$$

Challenge 4.3. *Show that the two norms defined above are indeed norms.*

Note that by solving Challenges 4.1 and 4.3 you have shown also that for any two matrices $X, Y \in \mathbb{R}^{n\times n}$,

$$\sigma_1(X+Y) \le \sigma_1(X) + \sigma_1(Y). \tag{2}$$

There is a natural generalization of the two norms above, the so called Schatten p-norms.

Definition 4.3 (Schatten p-norm)**.** *Given a matrix $X \in \mathbb{R}^{n\times m}$ and $1 \le p \le \infty$, the Schatten p-norm of X is given by*

$$\|X\|_{(S,p)} = \left(\sum_{i=1}^{\min\{n,m\}} \sigma_i(X)^p\right)^{1/p} = \|\sigma(X)\|_p,$$

where $\sigma(X)$ corresponds to the vector whose entries are the singular values of X.

For $p=\infty$, this corresponds to the spectral norm and we often simply use $\|X\|$ without a subscript,

$$\|X\| = \|X\|_{(S,\infty)} = \sigma_1(X).$$

For $p=2$ this corresponds to the commonly used Frobenius norm.

Challenge 4.4. *Show that the Schatten p-norm is a norm. (proving triangular inequality for general p is non-trivial).*

For $p=2$ and $p=\infty$ this corresponds to the familiar Frobenius and spectral/operator norms, as justified in the proposition below.

Another key insight in this section is that, since the rank of a matrix X is the number of non-zero singular values, a natural rank r approximation for a matrix X is to replace all singular values but the largest r singular values of X with zero. This is often referred to as the **truncated SVD**. Let us be more precise.

Definition 4.4. *Let $X \in \mathbb{R}^{n\times m}$ and $X = U\Sigma V^T$ be its SVD. We define $X_r = U_r\Sigma_r V_r$ the truncated SVD of X by setting $U_r \in \mathbb{R}^{n\times r}$ and $V_r \in \mathbb{R}^{m\times r}$ to be, respectively, the first r columns of U and V; and $\Sigma_r \in \mathbb{R}^{r\times r}$ to be a diagonal matrix with the first r singular values of X (notice this are the largest ones, due to the way we defined SVD).*

Warning: The notation X_r for low-rank approximations is not standard.

Note that $\text{rank}(X_r) = r$ and $\sigma_1(X - X_r) = \sigma_{r+1}(X)$.

It turns out that this way to approximate a matrix by a low rank matrix is optimal is a very strong sense, this is captured by the celebrated Eckart–Young–Mirsky Theorem, we start with a particular case of it.

Lemma 4.5 (Eckart–Young–Mirsky Theorem for Spectral norm)**.** *The truncated SVD provides the best low rank approximation in spectral norm. In other words:*

Let $X \in \mathbb{R}^{n\times m}$ and $r < \min\{n,m\}$. Let X_r be as in Definition 4.4, then:

$$\|X - B\| \geq \|X - X_r\|,$$

for any rank r matrix B.

Proof. Let $X = U\Sigma V^T$ be the SVD of X. Since $\text{rank}(B) = r$ there must exist a vector w in the span of the first $r+1$ right singular vectors $v_1, \ldots, v_{r+1}$ of X in the kernel of B. Without loss of generality let w have unit norm.

Let us write $w = \sum_{k=1}^{r+1} \alpha_k v_k$. Since w is unit-norm and the v_k's are orthonormal we have $\alpha_k = v_k^T w$ and $\sum_{k=1}^{r+1} \alpha_k^2 = 1$.

Finally,

$$\|X-B\| \geq \|(X-B)w\|_2 = \|Xw\|_2 = \|\Sigma V^T w\|_2 = \sqrt{\sum_{k=1}^{r+1} \sigma_k^2(X)\alpha_k^2} \geq \sigma_{r+1}(X) = \|X - X_r\|.$$

Challenge 4.5. *If you think the existence of the vector w in the start of the proof above is not obvious (or any other step), try to prove it.*

The inequality (2) is a particular case of a more general set of inequalities, the Weyl inequalities, named after Hermann Weyl (a brilliant Mathematician who spent many years at ETH). Here we focus on the inequalities for singular values, the more classical ones are for eigenvalues; it is worth noting also that these follow from the ones for eigenvalues since the singular values of X are the square-roots of the eigenvalues of $X^T X$.

Theorem 4.6 (Weyl inequalities for singular values)**.**

$$\sigma_{i+j-1}(X+Y) \leq \sigma_i(X) + \sigma_j(Y),$$

for all $1 \leq i, j, \leq \min\{n,m\}$ *satisfying* $i+j-1 \leq \min\{n,m\}$

Proof. Let X_{i-1} and Y_{j-1} be, respectively, the rank $i-1$ and $j-1$ approximation of X and Y (as in Definition 4.4). By (2) we have

$$\sigma_1\left((X-X_{i-1}) + \left(Y-Y_{j-1}\right)\right) \leq \sigma_1\left(X-X_{i-1}\right) + \sigma\left(Y-Y_{j-1}\right) = \sigma_i(X) + \sigma_j(Y).$$

Since $X_{i-1}+Y_{j-1}$ has rank at most $i+j-2$, Lemma 4.5 implies that

$$\sigma_{i+j+1}(X+Y) = \sigma_1(X+Y-(X+Y)_{i+j-2}) \leq \sigma_1\left(X+Y-\left(X_{i-1}+Y_{j-1}\right)\right).$$

Putting both inequalities together we get

$$\sigma_{i+j+1}(X+Y) \leq \sigma_1\left(X+Y-X_{i-1}-Y_{j-1}\right) \leq \sigma_i(X) + \sigma_j(Y).$$

Challenge 4.6. *There is another simple proof of this Theorem based on the Courant-Fischer minimax variational characterization of singular values:*

$$\sigma_k(X) = \max_{V \subseteq R^m, \dim(V)=k} \min_{v \in V, \|v\|=1} \|Xv\|, \tag{3}$$

$$\sigma_{k+1}(X) = \min_{V \subseteq R^m, \dim(V)=m-k} \max_{v \in V, \|v\|=1} \|Xv\|. \tag{4}$$

try to prove it that way.

We are now ready to prove the main Theorem in this Section

Theorem 4.7 (Eckart–Young–Mirsky Theorem)**.** *The truncated SVD provides the best low rank approximation in any Schatten p-norm. In other words:*

Let $X \in \mathbb{R}^{n\times m}$, $r < \min\{n,m\}$, and $1 \le p \le \infty$. Let X_r be as in Definition 4.4, then:

$$\|X-B\|_{(S,p)} \ge \|X-X_r\|_{(S,p)}.$$

for any rank r matrix B.

Proof. We have already proved this for $p = \infty$ (Lemma 4.5). The proof of the general result follows from Weyl's inequalities (Theorem 4.6).

Let $X \in \mathbb{R}^{n\times m}$ and B a rank r matrix. We use Theorem 4.6 for $X-B$ and B:

$$\sigma_{i+j-1}(X) \le \sigma_i(X-B) + \sigma_j(B),$$

Taking $j = r+1$, for and $i > 1$ satisfying $i+(r+1)-1 \le \min\{n,m\}$ we have

$$\sigma_{i+r}(X) \le \sigma_i(X-B), \tag{5}$$

since $\sigma_{r+1}(B) = 0$. Thus

$$\|X-B\|^p_{(S,p)} = \sum_{k=1}^{\min\{n,m\}} \sigma_k^p(X-B) \ge \sum_{k=1}^{\min\{n,m\}-r} \sigma_k^p(X-B).$$

Finally, by (5):

$$\sum_{k=1}^{\min\{n,m\}-r} \sigma_k^p(X-B) \ge \sum_{k=1}^{\min\{n,m\}-r} \sigma_{k+r}^p(X) = \sum_{k=r+1}^{\min\{n,m\}} \sigma_k^p(X) = \|X-X_r\|^p_{(S,p)}.$$

5. Dimension Reduction and Principal Component Analysis (11.03.2021)

A classical problem in data analysis is that of dimension reduction. Given m points in $\mathbb{R}^p$,

$$y_1, \ldots, y_m \in \mathbb{R}^p,$$

the goal is to find their best d-dimensional representation. In this Section we will describe Principal Component Analysis, connect it to truncated SVD, and describe an interpretation of the left and right singular vectors in the truncated SVD.

Let us start by centering the points (so that the approximation is by a subspace and not by an affine subspace), consider:

$$x_k := y_k - \mu,$$

where $\mu = \frac{1}{m}\sum_{i=1}^m y_k$ is the empirical average of $y_1, \ldots, y_k$.

We are looking for points $z_1, \ldots, z_m$ lying in a d-dimensional subspace such that $\sum_{k=1}^m \|z_k - x_k\|^2$ is minimum. Let us consider the matrices

$$Z = \begin{bmatrix} | & & | \\ z_1 & \cdots & z_m \\ | & & | \end{bmatrix} \quad \text{and} \quad X = \begin{bmatrix} | & & | \\ x_1 & \cdots & x_m \\ | & & | \end{bmatrix}.$$

The condition that $z_1, \ldots, z_m$ are in a d-dimensional is equivalent to $\operatorname{rank}(Z) \leq d$ and

$$\sum_{k=1}^m \|z_k - x_k\|^2 = \|Z - X\|_F^2.$$

Hence we are looking for the solution of

$$\min_{Z:\ \operatorname{rank}(Z)\leq d} \|Z - X\|_F,$$

which, by Theorem 4.7, corresponds to the truncated SVD of X. This means that

$$Z = U_d \Sigma_d V_d^T,$$

where $U_d \in \mathbb{R}^p \times d$, $\Sigma_d \in \mathbb{R}^{d\times d}$, and $V \in \mathbb{R}^{m\times d}$. This means that

$$y_k \sim U_d \beta_k + \mu, \tag{6}$$

is a d-dimensional approximation of the original dataset, where $\beta_k = \Sigma_d v_k$ and $v_k \in \mathbb{R}^d$ is the k-th row of V_d. This is *Principal Component Analysis* (PCA) (see, for example, Chapter 3.2 in [BSS] for an alternative derivation, not based on SVD).

Remark 5.1. *Notice how the left singular vectors U_d and the right singular vectors V_d have two different interpertations, each of one corresponds to a different way of presenting PCA, as we will do below.*

- *The singular vectors U_d correspond to the basis in which to project the original points (after centering).*
- *The singular vectors V_d (after scaling entries by Σ_d) correspond to low dimensional coordinates for the points.*

Challenge 5.1. *Instead of centering the points at the start, we could have asked for the best approximation in the sense of picking $\beta_k \in \mathbb{R}^d$, a matrix U_d whose columns are a basis for a d-dimensional subspace, and $\mu \in \mathbb{R}^d$ such that* (6) *is the best possible approximation (in the sense of sum of squares of distances). Show that the vector μ in this case is indeed the empirical mean.*

5.1. **Principal Component Analysis - description 1.** One way to describe PCA (see, for example, Chapter 3.2. of [BSS]) is to build the sample covariance matrix of the data:

$$\frac{1}{m-1}XX^T = \frac{1}{m-1}\sum_{k=1}^{m}(y_k-\mu)(y_k-\mu)^T,$$

where μ is the empirical mean of the y_k's.

PCA then consists in writing the data in the subspace generated by the leading eigenvectors of XX^T (recall that the scaling of $\frac{1}{m-1}$ is not relevant). This is the same as above:

$$XX^T = U\Sigma V^T\left(U\Sigma V^T\right)^T = U\Sigma^2 U^T,$$

where $X = U\Sigma V^T$ is the SVD of X. Thus the leading eigenvectors of XX^T correspond to the leading *left* singular vectors of X.

5.2. **Principal Component Analysis - description 2.** The second interpretation of Principal Component Analysis will be in terms of the right singular vectors of X. Although, without the connection with truncated PCA, this description seems less natural, we will see in Subsection 5.3 it has a surprising advantage.

For simplicity let us consider the centered points $x_1,\ldots,x_m$. The idea is to build a matrix $M \in \mathbb{R}^{m\times m}$ whose entries are

$$M_{ij} = \langle x_i, x_j\rangle, \tag{7}$$

and use its leading eigenvectors as low dimensional coordinates. To be more precise if λ_i is the i-th largest eigenvalue of M, and v_i the corresponding eigenvector, for each point y_j we use the

following d-dimensional coordinates

$$\begin{bmatrix} \sqrt{\lambda_1} v_1(j) \\ \sqrt{\lambda_2} v_2(j) \\ \vdots \\ \sqrt{\lambda_d} v_d(j) \end{bmatrix}, \tag{8}$$

sometimes the scaling by the eingenvalues is a different one (since it is only shrinking or expanding coordinates it is usually not important).

Since $X^T X = \left(U\Sigma V^T\right)^T U\Sigma V^T = V\Sigma^2 V^T$ this is equivalent to the interpertation of the right singular vectors of X as (perhaps scaled) low dimensional coordinates.

5.3. **A short mention of Kernel PCA.** One can interpret the matrix M in (7) as M_{ij} measuring affinity between point i and j; indeed $x_i^T x_j$ is larger if x_i and x_j are more similar. The advantage is that with this interpretation is that it allows to perform versions of PCA with other notions of affinity

$$M_{ij} = K(x_i, x_j),$$

where the affinity function K is often called a Kernel. This is the idea behind *Kernel PCA*. Notice that this can be defined even when the data points are not in Euclidean space (see Remark 5.2)

Further Reading 5.2. *A common choice of Kernel is the so-called Gaussian kernel* $K(x_i, x_j) = \exp\left(-\|x_i - x_j\|^2/\varepsilon^2\right)$*, for* $\varepsilon > 0$*. The intuition of why one would use this notion of affinity is that it tends to ignore distances at a scale larger than* ε*; if data has a low dimensional structure embedded, with some curvature, in a larger dimensional ambient space then small distances in the ambient space should be similar to intrinsic distances, but larger distances are less reliable (recall Figure 2); see Chapter 5 in* [BSS] *for more on this, and some illustrative pictures.*

In order for the interpretation above to apply we need $M \succeq 0$ ($M \succeq 0$ means M is positive semidefinite, all eigenvalues are non-negative; we only use the notation $M \succeq 0$ for symmetric matrices). When this is the case, we can write the Cholesky decomposition of M as

$$M = \Phi^T \Phi,$$

for some matrix Φ. If φ_i is the i-th column of Φ then

$$M_{ij} = \langle \varphi_i, \varphi_j \rangle,$$

for this reason φ_i is commonly referred to, in the Machine Learning community, referred to as the feature vector of i.

Challenge 5.3. *Show that the resulting matrix* M *is always positive definite for the Gaussian Kernel* $K(x_i, x_j) = \exp\left(-\|x_i - x_j\|^2/\varepsilon^2\right)$.

Further Reading 5.4. *There are several interesting Mathematical questions arising from this short section, unfortunately their answers need more background than the pre-requisites in this course, in any case we leave a few notes for further reading here:*

- *A very natural question is whether the features φ_i depend only on the kernel K and not on the data points (as it was described), this is related to the celebrated* Mercer Theorem *(essentially a spectral theorem for positive semidefinite kernels).*
- *A beautiful theorem in this area is* Bochner's Theorem. *In the special case of kernels that are a function of the difference between the points, relates a kernel being positive with properties of its Fourier Transform. This theorem can be used to solve Challenge 5.3 (but there are other ways).*

Remark 5.2. *In the next section we do, in a sense, a version of the idea described here for a network, where the matrix M will simply discriminate whether nodes i and j are, or not, connected in a network.*

6. THE GRAPH LAPLACIAN (11.03.2021)

In this section we will study networks, also called graphs.

Definition 6.1 (Graph)**.** *A graph is a mathematical object consisting of a set of vertices V and a set of edges $E \subseteq \binom{V}{2}$. We will focus on undirected graphs. We say that $i \sim j$, i is connected to j, if $(i,j) \in E$. We assume graphs have no loops, i.e. $(i,i) \notin E$ for all i.*

In what follows the graph will have n nodes ($|V| = n$). It is sometimes useful to consider a weighted graph, in which and edge (i,j) has a non-negative weight w_{ij} . Essentially everything remains the same if considering weighted graphs, we focus on unweighted graphs to lighten the notation (See Chapter 4 in [BSS] for a similar treatment that includes weighted graphs).

A useful way to represent a graph is via its adjacency matrix. Given a graph $G = (V,E)$ on n nodes ($|V| = n$), we define its adjacency matrix $A \in \mathbb{R}^{n \times n}$ as the symmetric matrix with entries

$$A_{ij} = \begin{cases} 1 & \text{if } (i,j) \in E, \\ 0 & \text{otherwise.} \end{cases}$$

Remark 6.2. *As foreshadowed on Remark 5.2 this can be viewed in the same way as the kernels above (ignoring the diagonal of the matrix). In fact, there are many ways to transform data into a graph: examples include considering it as a weighted graph where the weights are given by a kernel, or connecting data points if they correspond to nearest neighbors.*

A few definitions will be useful.

Definition 6.3 (Graph Laplacian and Degree Matrix)**.** *Let $G = (V,E)$ be a graph and A its adjacency matrix. The degree matrix D is a diagonal matrix with diagonal entries*

$$D_{ii} = \deg(i),$$

where $\deg(i)$ *is the degree of node i, the number of neighbors of i.*

The graph Laplacian of G is given by

$$L_G = D - A.$$

Equivalently

$$L_G := \sum_{(i,j) \in E} \left(e_i - e_j\right)\left(e_i - e_j\right)^T.$$

Definition 6.4 (Cut, Volume, and Connectivity)**.** *Given a subset $S \subseteq V$ of the vertices, we call $S^c = V \setminus S$ the complement of S and we define*

$$\text{cut}(S) = \sum_{i \in S} \sum_{j \in S^c} 1_{(i,j) \in E},$$

as the number of edges "cut" by the partition (S, S^c)*, where* 1_X *is the indicator of* X*. Also,*

$$\mathrm{vol}(S) = \sum_{i \in S} \deg(i),$$

we sometimes abuse notation and use $\mathrm{vol}(G)$ *to denote the total volume.*

Furthermore, we say that a graph G *is disconnected if there exists* $\emptyset \subsetneq S \subsetneq V$ *such that* $\mathrm{cut}(S) = 0$*.*

The following is one of the most important properties of the graph Laplacian.

Proposition 6.5. *Let* $G = (V, E)$ *be a graph and* L_G *its graph Laplacian, let* $x \in \mathbb{R}^n$*. Then*

$$x^T L_G x = \sum_{(i,j) \in E} (x_i - x_j)^2$$

Note that each edge is appearing in the sum only once.

Proof.

$$\begin{aligned}
\sum_{(i,j) \in E} \left(x_i - x_j\right)^2 &= \sum_{(i,j) \in E} \left[\left(e_i - e_j\right)^T x\right]^T \left[\left(e_i - e_j\right)^T x\right] \\
&= \sum_{(i,j) \in E} x^T \left(e_i - e_j\right) \left(e_i - e_j\right)^T x \\
&= x^T \left[\sum_{(i,j) \in E} \left(e_i - e_j\right) \left(e_i - e_j\right)^T\right] x
\end{aligned}$$

□

An immediate consequence of this is that

$$L \succeq 0.$$

Note also that, due to Proposition 6.5 we have

$$L_G \mathbf{1} = 0,$$

where $\mathbf{1}$ is the all-ones vector (notice that the Proposition implies that $\mathbf{1}^T L_G \mathbf{1} = 0$ but since $L_G \succeq 0$ this implies that $L_G \mathbf{1} = 0$). This means that $\frac{1}{\sqrt{n}}\mathbf{1}$ is the eigenvector of L_G corresponding to its smallest eigenvalue.

Because in the definition of graph Laplacian, the matrix A appears with a negative sign, the important eigenvalues become the smallest ones of L, not the largest ones as in the previous

section. Since $L_G \succeq 0$ we can order them[4]

$$0 = \lambda_1(L_G) \leq \lambda_2(L_G) \leq \cdots \leq \lambda_n(L_G).$$

Now we prove the first Theorem relating the geometry of a graph with the spectrum of its Laplacian.

Theorem 6.6. *Let $G = (V,E)$ be a graph and L_G its graph Laplacian. $\lambda_2(L_G) = 0$ if and only if G is disconnected.*

Proof. Since the eigenvector corresponding to $\lambda_1(L_G)$ is $v_1(L_G) = \frac{1}{\sqrt{n}}$ then $\lambda_2(L_G) = 0$ if and only if there exists nonzero $y \in \mathbb{R}^n$ such that $y \perp \mathbf{1}$ and $y^T L y = 0$.

Let us assume that $\lambda_2(L_G) = 0$, and let y be a vector as described above. Let $S = \{i \in V | y_i \geq 0\}$, because y is non-zero and $y \perp 1$ we have that $\emptyset \subsetneq S \subsetneq V$. Also,

$$y^T L_G y = \sum_{(i,j)\in E} (y_i - y_j)^2,$$

so for this sum to be zero, all its terms need to be zero. For $i \in S$ and $j \neq S$ we have $(y_i - y_j)^2 > 0$ thus $(i,j) \neq E$; this means that $\text{cut}(S) = 0$.

For the converse, suppose that there exists $\emptyset \subsetneq S \subsetneq V$ such that $\text{cut}(S)$ and take $y = 1_S$ the indicator of S. It is easy to see that $1_S \in \ker(L_G)$ and $1_S \perp \mathbf{1}$, thus $\dim \ker(L_G) \geq 2$. □

This already suggests that eigenvalues of L_G contain information about the graph, in particular about whether there are two sets of nodes in the graph that cluster, without there being edges between clusters. In the next section we will show a quantitative version of this theorem, which is the motivation behind the popular method of *Spectral Clustering*.

Challenge 6.1. *Prove a version of Theorem 6.6 for other eigenvalues $\lambda_k(L_G)$. What does $\lambda_k(L_G) = 0$ say about the geometry of the graph G for $k > 2$?*

[4]Notice the different ordering than above, it is because the smallest eigenvalue of L is the one associated with the largest one of A, however in Spectral Graph Theory is it more convenient to work with the graph Laplacian.

7. Cheeger Inequality and Spectral Clustering (18.03.2021)

Let us suppose that the graph G does indeed have a non-trivial[5] partition of its nodes (S, S^c) with a small number of edges connecting nodes in S with nodes in S^c. We know that if the number of edges is zero this implies that $\lambda_2(G) = 0$ (by Theorem 6.6), in this section we will investigate what happens if the cut is small, but not necessarily zero.

We start by normalizing the graph Laplacian to alleviate the dependency of the spectrum on the number of edges. We will assume throughout this section that the graph has no isolated nodes, i.e. $\deg(i) \neq 0$ for all $i \in V$. We define the normalized graph Laplacian as

$$\mathscr{L}_G = D^{-\frac{1}{2}} L_G D^{-\frac{1}{2}} = I - D^{-\frac{1}{2}} A D^{-\frac{1}{2}}.$$

Note that $\mathscr{L}_G \succeq 0$ and that $D^{\frac{1}{2}}\mathbf{1} \in \ker(\mathscr{L}_G)$. Similarly we consider the eigenvalues ordered as

$$0 = \lambda_1(\mathscr{L}_G) \leq \lambda_2(\mathscr{L}_G) \leq \cdots \leq \lambda_n(\mathscr{L}_G).$$

We will show the following relationship

Theorem 7.1. *Let $G = (V, E)$ be a graph without isolated nodes and $\mathscr{L}_G$ its normalized Laplacian $\mathscr{L}_G = I - D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$ then*

$$\lambda_2(\mathscr{L}_G) \leq \min_{\emptyset \subsetneq S \subsetneq V} \frac{\mathrm{cut}(S)}{\mathrm{vol}(S)} + \frac{\mathrm{cut}(S)}{\mathrm{vol}(S^c)}.$$

The quantity on the RHS is often referred to as the Normalized Cut of G.

Proof.

The key idea in this proof is that of a *relaxation* — when a complicated minimization problem is lower bounded by taking the minimization over a larger, but simpler, set.

By the Courant-Fischer variational principal of eigenvalues we know that

$$\lambda_2(\mathscr{L}_G) = \min_{\substack{\|z\|=1, \\ z \perp v_1(\mathscr{L}_G)}} z^T \mathscr{L}_G z,$$

where $v_1(\mathscr{L}_G)$ is the eigenvector corresponding to the smallest eigenvalue of $\mathscr{L}_G$, which we know is a multiple of $D^{\frac{1}{2}}\mathbf{1}$. It will be helpful to write $y = D^{-\frac{1}{2}} z$, we then have:

$$\lambda_2(\mathscr{L}_G) = \min_{\substack{y^T D y = 1, \\ y^T D \mathbf{1} = 0}} y^T D^{\frac{1}{2}} \mathscr{L}_G D^{\frac{1}{2}} y = \min_{\substack{y^T D y = 1 \\ y^T D \mathbf{1} = 0}} y^T L_G y = \min_{\substack{y^T D y = 1 \\ y^T D \mathbf{1} = 0}} \sum_{(i,j) \in E} (y_i - y_j)^2.$$

[5] Non-trivial here simply means that neither part is the empty set.

The key argument is that the Normalized Cut will correspond to minimum of this when we restrict the vector y to take only two different values, one in S and another in S^c.

For a non-trivial subset $S \subset V$, let us consider the vector $y \in \mathbb{R}^n$ such that

$$y_i = \begin{cases} a & \text{if } i \in S \\ b & \text{if } i \in S^c. \end{cases}$$

For the constraints $y^T Dy = 1$ and $y^T D\mathbf{1} = 0$ to be satisfied we must have (see Challenge 7.1)

$$y_i = \begin{cases} \left(\frac{\text{vol}(S^c)}{\text{vol}(S)\,\text{vol}(G)}\right)^{\frac{1}{2}} & \text{if } i \in S \\ -\left(\frac{\text{vol}(S)}{\text{vol}(S^c)\,\text{vol}(G)}\right)^{\frac{1}{2}} & \text{if } i \in S^c. \end{cases} \tag{9}$$

The rest of the proof proceeds by computing $y^T L_G y$ for y of the form (9):

$$\begin{aligned}
y^T L_G y &= \frac{1}{2}\sum_{(i,j)\in E}(y_i - y_j)^2 \\
&= \sum_{i\in S}\sum_{j\in S^c} 1_{(i,j)\in E}(y_i - y_j)^2 \\
&= \sum_{i\in S}\sum_{j\in S^c} 1_{(i,j)\in E}\left[\left(\frac{\text{vol}(S^c)}{\text{vol}(S)\,\text{vol}(G)}\right)^{\frac{1}{2}} + \left(\frac{\text{vol}(S)}{\text{vol}(S^c)\,\text{vol}(G)}\right)^{\frac{1}{2}}\right]^2 \\
&= \sum_{i\in S}\sum_{j\in S^c} 1_{(i,j)\in E}\frac{1}{\text{vol}(G)}\left[\frac{\text{vol}(S^c)}{\text{vol}(S)} + \frac{\text{vol}(S)}{\text{vol}(S^c)} + 2\right] \\
&= \sum_{i\in S}\sum_{j\in S^c} 1_{(i,j)\in E}\frac{1}{\text{vol}(G)}\left[\frac{\text{vol}(S^c)}{\text{vol}(S)} + \frac{\text{vol}(S)}{\text{vol}(S^c)} + \frac{\text{vol}(S)}{\text{vol}(S)} + \frac{\text{vol}(S^c)}{\text{vol}(S^c)}\right] \\
&= \sum_{i\in S}\sum_{j\in S^c} 1_{(i,j)\in E}\left[\frac{1}{\text{vol}(S)} + \frac{1}{\text{vol}(S^c)}\right] \\
&= \text{cut}(S)\left[\frac{1}{\text{vol}(S)} + \frac{1}{\text{vol}(S^c)}\right]
\end{aligned}$$

Finally

$$\lambda_2(\mathscr{L}_G) = \min_{\substack{y^T Dy=1 \\ y^T D\mathbf{1}=0}} \sum_{(i,j)\in E}(y_i - y_j)^2 \leq \min_{\substack{y^T Dy=1,\ y^T D\mathbf{1}=0 \\ y\in\{a,b\}^n \text{ for } a,b\in\mathbb{R}}} \sum_{(i,j)\in E}(y_i - y_j)^2 = \min_{\emptyset\subsetneq S\subsetneq V} \frac{\text{cut}(S)}{\text{vol}(S)} + \frac{\text{cut}(S)}{\text{vol}(S^c)} \tag{10}$$

□

Challenge 7.1. *Show that for any non-trivial subset $S \subset V$, the vector $y \in \mathbb{R}^n$ such that*

$$y_i = \begin{cases} a & \text{if } i \in S \\ b & \text{if } i \in S^c \end{cases}$$

satisfies $y^T D y = 1$ and $y^T D\mathbf{1} = 0$ if and only if

$$y_i = \begin{cases} \left(\frac{\mathrm{vol}(S^c)}{\mathrm{vol}(S)\,\mathrm{vol}(G)}\right)^{\frac{1}{2}} & \text{if } i \in S \\ -\left(\frac{\mathrm{vol}(S)}{\mathrm{vol}(S^c)\,\mathrm{vol}(G)}\right)^{\frac{1}{2}} & \text{if } i \in S^c. \end{cases}$$

Hint: Recall that $\mathrm{vol}(G) = \mathrm{vol}(S) + \mathrm{vol}(S^c)$.

There are (at least) two consequential ideas illustrated in (10):

(1) The way cuts of partitions are measured in (10) promotes somewhat balanced partitions (so that neither $\mathrm{vol}(S)$ nor $\mathrm{vol}(S^c)$ are too small), this turns out to be beneficial and to avoid trivial solutions such as partition a graph by splitting just one nodes from all the others.

(2) There is an important algorithmic consequence of (10): when we want to cluster a network, what we want to minimize is the RHS of (10), this is unfortunately computationally intractable (in fact, it is known to be NP-hard). However, the LHS of the inequality is a spectral problem and so computationally tractable (we would compute z the eigenvector of $\mathscr{L}_G$ and then compute $y = D^{-\frac{1}{2}}z$). This is the idea behind the popular algorithm of *Spectral clustering* (Algorithm 1).

Algorithm 1 Spectral Clustering

Given a graph $G = (V, E, W)$, let v_2 be the eigenvector corresponding to the second smallest eigenvalue of the normalized Laplacian $\mathscr{L}_G$. Let $\varphi_2 = D^{-\frac{1}{2}}v_2$. Given a threshold τ (one can try all different possibilities, or run k-means in the entries of φ_2 for $k = 2$), set

$$S = \{i \in V : \varphi_2(i) \leq \tau\}.$$

A natural question is whether one can give a guarantee for this algorithm: "Does Algorithm 1 produce a partition whose normalized cut is comparable with $\lambda_2(\mathscr{L}_G)$?", although the proof of such a guarantee is outside the scope of this course, we will briefly describe it below, it corresponds to the celebrated Cheeger's Inequality.

It is best formulated in terms of the so called Cheeger cut and Cheeger constant.

Definition 7.2 (Cheeger's cut)**.** *Given a graph and a vertex partition (S, S^c), the Cheeger cut (also known as conductance, or expansion) of S is given by*

$$h(S) = \frac{\text{cut}(S)}{\min\{\text{vol}(S), \text{vol}(S^c)\}},$$

where $\text{vol}(S) = \sum_{i \in S} \deg(i)$.

The Cheeger constant of G is given by

$$h_G = \min_{S \subset V} h(S).$$

Note that the normalized cut and $h(S)$ are tightly related, in fact it is easy to see that:

$$h(S) \leq \min_{\emptyset \subsetneq S \subsetneq V} \frac{\text{cut}(S)}{\text{vol}(S)} + \frac{\text{cut}(S)}{\text{vol}(S^c)} \leq 2h(S).$$

This means that we proved in Theorem 7.1 that

$$\frac{1}{2}\lambda_2(\mathscr{L}_G) \leq h_G.$$

This is often referred to as the easy side of Cheeger's inequality.

Theorem 7.3 (Cheeger's Inequality)**.** *Recall the definitions above. The following holds:*

$$\frac{1}{2}\lambda_2(\mathscr{L}_G) \leq h_G \leq \sqrt{2\lambda_2(\mathscr{L}_G)}.$$

Cheeger's inequality was first established for manifolds by Jeff Cheeger in 1970 [Che70], the graph version is due to Noga Alon and Vitaly Milman [Alo86, AM85] in the mid 80s. The upper bound in Cheeger's inequality (corresponding to Lemma 7.4) is more difficult to prove and outside of the scope of this course, it is often referred to as the "the difficult part" of Cheeger's inequality. There are several proofs of this inequality (see [Chu10] for four different proofs! You can also see [BSS] for a proof in notation very close to these notes). We just mention that this inequality can be proven via a guarantee to spectral clustering.

Lemma 7.4. *There is a threshold τ, in Algorithm 1 producing a partition S such that*

$$h(S) \leq \sqrt{2\lambda_2(\mathscr{L}_G)}.$$

This implies in particular that

$$h(S) \leq \sqrt{4h_G},$$

meaning that Algorithm 1 is suboptimal at most by a square-root factor.

Remark 7.5. *Algorithm 1 can be used to cluster data into $k > 2$ clusters. In that case one considers the $k-1$ eigenvectors (from the 2nd to the kth) and to each nodes i we associate the*

$k-1$ *dimensional representation*

$$v_i \to [\varphi_2(i), \varphi_3(i), \cdots, \varphi_k(i)]^T,$$

and uses k-means on this representation.

Remark 7.6. *There is another powerful tool that follows from this: one can use the representation described in Remark 7.5 to embed the graph in Euclidean space. This is oftentimes referred to as "Diffusion Maps" or "Spectral Embedding", see for example Chapter 5 in* [BSS].

Notice also the relationship with Kernel PCA as described in the section above.

8. Introduction to Finite Frame Theory (18.03.2021)

We will now start the portion of the course on Parsimony, focusing on sparsity and low rank matrix completion. Before introducing those objects, it will be useful to go over the basics of Finite dimensional frame theory, this is what we will do in this section. For a reference on this topic, see for example the first Chapter of the book [Chr16].

Throughout this section we will use $\mathbb{K}^d$ to refer either $\mathbb{R}^d$ or $\mathbb{C}^d$. When the field matters, we will point this out explicitly.

If $\varphi_1, \ldots, \varphi_d \in \mathbb{K}^d$ are a basis then any point $x \in \mathbb{K}^d$ is uniquely identified by the inner products $b_k = \langle \varphi_k, x \rangle$. In particular if $\varphi_1, \ldots, \varphi_d \in \mathbb{K}^d$ are an orthonormal basis this representation satisfies a Parseval identity

$$\left\| [\langle \varphi_k, x \rangle]_{k=1}^d \right\| = \|x\|.$$

Notice that in particular by using this identity on $x - y$ this ensures stability in the representation.

Redundancy. : In signal processing and communication it is useful to include redundancy. If instead of a basis one considers a redundant spanning set $\varphi_1, \ldots, \varphi_m \in \mathbb{K}^d$ with $m > d$ a few advantages arise: for example, if in a communication channel one of the coefficients b_k gets erased, it might still be possible to reconstruct x. Such sets are sometimes called **redundant dictionaries** or **overcomplete dictionaries**.

Stability. : It is important to keep some form of stability of the type of the Parseval identity above. While this is particularly important for infinite dimensional vector spaces (more precisely Hilbert spaces) we will focus our exposition on finite dimensions.

Definition 8.1. *A set $\varphi_1, \ldots, \varphi_m \in \mathbb{K}^d$ is called a frame of $\mathbb{K}^d$ if there exist non-zero finite constants A and B such that, for all $x \in \mathbb{K}^d$*

$$A\|x\|^2 \leq \sum_{k=1}^m |\langle \varphi_k, x \rangle|^2 \leq B\|x\|^2.$$

A and B are called respectively the lower and upper frame bound. The maximum A and the minimum B are called the optimal frame bounds.

Challenge 8.1. *Show that $\varphi_1, \ldots, \varphi_m \in \mathbb{K}^d$ is a frame if and only if it spans all of $\mathbb{K}^d$.*

Further Reading 8.2. *In infinite dimensions the situation is considerably more delicate than suggested by Challenge 8.1, and it is tightly connected with the notion of stable sampling from signal processing. You can see, e.g.,* [Chr16].

Given a frame $\varphi_1, \ldots, \varphi_m \in \mathbb{K}^d$, let

$$\Phi = \begin{bmatrix} | & & | \\ \varphi_1 & \cdots & \varphi_m \\ | & & | \end{bmatrix}. \tag{11}$$

The following are classical definitions in the frame theory literature (although for finite dimensions the objects are essentially just matrices involving Φ and so the definitions are not as important; also not that we are doing a slight abuse of notation using the same notation for a matrix and the linear operator it represents – it will be clear from context which object we mean.)

Definition 8.2. *Given a frame $\varphi_1, \ldots, \varphi_m \in \mathbb{K}^d$. It is classical to call:*

- *The operator $\Phi : \mathbb{K}^m \to \mathbb{K}^d$ corresponding to the matrix Φ, meaning $\Phi(c) = \sum_{k=1}^n c_k \varphi_k$, is often called the* Synthesis Operator.
- *Its adjoint operator $\Phi^* : \mathbb{K}^d \to \mathbb{K}^m$ corresponding to the matrix Φ^*, meaning $\Phi(x) = \{\langle x, \varphi_k \rangle\}_{k=1}^m$, is often called the* Analysis Operator.
- *The self-adjoint operator $S : \mathbb{K}^d \to \mathbb{K}^d$ given by $S = \Phi\Phi^*$ is often called the* Frame Operator.

Challenge 8.3. *Show that $S \succeq 0$ and that S is invertible.*

The following are interesting (and useful) definitions:

Definition 8.3. *A frame is called a tight frame if the frame bounds can be taken to be equal $A = B$.*

Challenge 8.4. *What can you say about the Frame Operator S for a tight frame?*

Definition 8.4. *A frame $\varphi_1, \ldots, \varphi_m \in \mathbb{K}^d$ is said to be unit normed (or unit norm) if for all $k \in [m]$ we have $\|\varphi_k\| = 1$.*

Definition 8.5. *The spark of a frame $\varphi_1, \ldots, \varphi_m \in \mathbb{K}^d$, or matrix Φ, is the minimum number of elements of the frame, of columns of the matrix, that make up a linearly dependent set.*

Challenge 8.5. *For a matrix Φ, show that* $\operatorname{spark}(\Phi) \leq \operatorname{rank}(\Phi) + 1$. *Can you prove it in a single line?*

Definition 8.6. *Given a unit norm frame $\varphi_1, \ldots, \varphi_m \in \mathbb{K}^d$ we call the worst-case coherence (sometimes also called dictionary coherence) the quantity*

$$\mu = \max_{i \neq j} |\langle \varphi_i, \varphi_j \rangle|.$$

Challenge 8.6. *Can you give a relationship between the spark and the worst-case coherence of a frame?*

9. PARSIMONY (25.03.2021)

Parsimony is an important principle in machine learning. The key idea is that oftentimes one wants to learn (or recover) and object with special structure. As we will see in the second half of the course, it is also important in supervised learning, the key idea there being that classifiers (or regression rules, as you will see in a Statistics course) that are simple are in theory more likely to generalize to unseen data. Observations of this type date back at least to eight centuries ago, the most notable instance being William of Ockham's celebrated *Occam's Razor*: "Entia non-sunt multiplicanda praeter necessitatem (Entities must not be multiplied beyond necessity)", which is today used as a synonim for parsimony.

One example that we will discuss is recommendation systems, in which the goal is to make recommendations of a product to users based both on the particular user scores of other items, and the scores other users gives to items. The score matrix whose rows correspond to users, columns to items, and entries to scores is known to be low rank and this form of parsimony is key to perform "matrix completion", meaning to recover (or estimate) unseen scores (matrix entries) from the ones that are available; we will revisit this problem in a couple of lectures.

A simpler form of parsimony is sparsity. Not only is sparsity present in many problems, including signal and image processing, but the mathematics arising from its study are crucial also to solve problems such as matrix completion. In what follows we will use image processing as the driving motivation. [6]

9.1. **Sparse Recovery.** Most of us have noticed how saving an image in JPEG dramatically reduces the space it occupies in our hard drives (as oppose to file types that save the pixel value of each pixel in the image). The idea behind these compression methods is to exploit known structure in the images; although our cameras will record the pixel value (even three values in RGB) for each pixel, it is clear that most collections of pixel values will not correspond to pictures that we would expect to see. This special structure tends to exploited via sparsity. Indeed, natural images are known to be sparse in certain bases (such as the wavelet bases) and this is the core idea behind JPEG (actually, JPEG2000; JPEG uses a different basis). There is an example illustrating this in the jupyter notebook accompanying the class.

Let us think of $x \in \mathbb{R}^N$ as the signal corresponding to the image already in the basis for which it is sparse, meaning that it has few non-zero entries. We use the notation $\|x\|_0$ for the number of non-zero entries of x, it is common to refer to this as the ℓ_0 norm, even though it is not actually a norm. Let us assume that $x \in \mathbb{R}^N$ is s-sparse, or $\|x\|_0 \leq s$, meaning that x has, at most, s non-zero components and, usually, $s \ll N$. This means that, when we take a picture, our camera makes N

[6] In this Section we follow parts of Section 6 in [Ban16], the exposition in [Ban16] is more advanced.

measurements (each corresponding to a pixel) but then, after an appropriate change of basis, it keeps only $s \ll N$ non-zero coefficients and drops the others. This motivates the question: "If only a few degrees of freedom are kept after compression, why not measure in a more efficient way and take considerably less than N measurements?". This question is in the heart of Compressed Sensing. It is particularly important in MRI imaging as less measurements potentially means less measurement time. The following book is a great reference on Compressed Sensing [FR13].

More precisely, given a s-sparse vector x, we take $s < M \ll N$ linear measurements $y_i = a_i^T x$ and the goal is to recover x from the underdetermined system:

$$\begin{bmatrix} \\ y \\ \\ \end{bmatrix} = \begin{bmatrix} & & \\ & \Phi & \\ & & \end{bmatrix} \begin{bmatrix} \\ \\ x \\ \\ \\ \end{bmatrix}.$$

10. Compressed Sensing and Sparse Recovery (25.03.2021)

Recall the setting and consider again $\mathbb{K}$ to mean either the real numbers of the complex numbers: given an s-sparse vector $x \in \mathbb{K}^N$, we take $s < d \ll N$ linear measurements and the goal is to recover x from the underdetermined system:

$$\begin{bmatrix} \\ y \\ \\ \end{bmatrix} = \begin{bmatrix} & & \\ & \Phi & \\ & & \end{bmatrix} \begin{bmatrix} \\ \\ x \\ \\ \\ \end{bmatrix}.$$

Since the system is underdetermined and we know x is sparse, the natural thing to try, in order to recover x, is to solve

$$\begin{aligned} \min \quad & \|z\|_0 \\ s.t. \quad & \Phi z = y, \end{aligned} \tag{12}$$

and hope that the optimal solution z corresponds to the signal in question x.

Remark 10.1. *There is another useful way to think about* (12)*. We can think of the columns of* Φ *as a redundant dictionary or frame. In that case, the goal becomes to represent a vector* $y \in \mathbb{K}^d$ *as a linear combination of the dictionary elements. Due to the redundancy, a common choice is to use the sparsest representation, corresponding to solving problem* (12)*.*

Proposition 10.2. *If* x *is* s*-sparse and* $\operatorname{spark}(\Phi) > 2s$ *then* x *is the unique solution to* (12) *for* $y = \Phi x$*.*

Challenge 10.1. *Can you construct* Φ *with large spark, and small number of measurements* d*?*

There are two significant issues with (12), stability (as the ℓ_0 norm is very brittle) and computation. In fact, (12) is known to be a computationally hard problem in general (provided $\mathtt{P} \neq \mathtt{NP}$). Instead, the approach usually taken in sparse recovery is to consider a convex relaxation of the ℓ_0 norm, the ℓ_1 norm: $\|z\|_1 = \sum_{i=1}^N |z_i|$. Figure 3 depicts how the ℓ_1 norm can be seen as a convex relaxation of the ℓ_0 norm and how it promotes sparsity.

This motivates one to consider the following optimization problem (surrogate to (12)):

$$\begin{aligned} \min \quad & \|z\|_1 \\ s.t. \quad & \Phi z = y, \end{aligned} \tag{13}$$

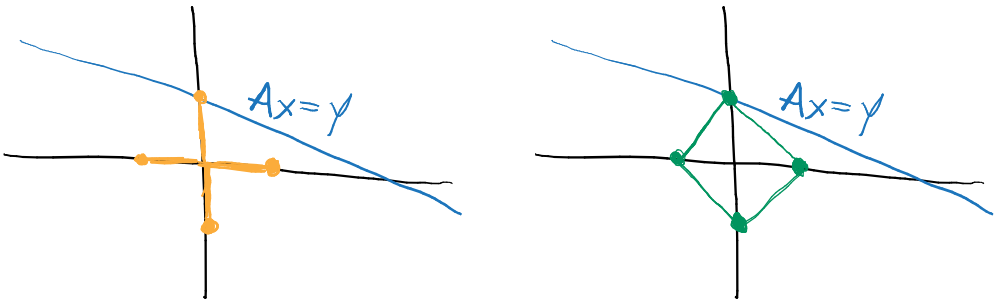


FIGURE 3. A two-dimensional depiction of ℓ_0 and ℓ_1 minimization. In ℓ_1 minimization (the picture of the right), one inflates the ℓ_1 ball (the diamond) until it hits the affine subspace of interest, this image conveys how this norm promotes sparsity, due to the pointy corners on sparse vectors.

For (13) to be useful, two things are needed: (1) the solution of it needs to be meaningful (hopefully to coincide with x) and (2) (13) should be efficiently solved.

10.1. **Computational efficiency.** To address computational efficiency we will focus on the real case ($\mathbb{K} = \mathbb{R}$) and formulate (13) as a Linear Program (and thus show that it is efficiently solvable). Let us think of ω^+ as the positive part of x and ω^- as the symmetric of the negative part of it, meaning that $x = \omega^+ - \omega^-$ and, for each i, either ω_i^- or ω_i^+ is zero. Note that, in that case (for $x \in \mathbb{R}^N$),

$$\|x\|_1 = \sum_{i=1}^{N} \omega_i^+ + \omega_i^- = \mathbf{1}^T \left(\omega^+ + \omega^-\right).$$

Motivated by this we consider:

$$\begin{array}{ll} \min & \mathbf{1}^T \left(\omega^+ + \omega^-\right) \\ s.t. & A\left(\omega^+ - \omega^-\right) = y \\ & \omega^+ \geq 0 \\ & \omega^- \geq 0, \end{array} \tag{14}$$

which is a linear program. It is not difficult to see that the optimal solution of (14) will indeed satisfy that, for each i, either ω_i^- or ω_i^+ is zero and it is indeed equivalent to (13). Since linear programs are efficiently solvable [VB04], this means that (13) can be solved efficiently.

Remark 10.3. *While* (13) *does not correspond to a linear program in the Complex case* $\mathbb{K} = \mathbb{C}$ *it is nonetheless efficient to solve, the key property is that it is a convex problem, but a general discussion about convexity is outside the scope of this course.*

10.2. **Exact recovery via ℓ_1 minimization.** The goal now is to show that, under certain conditions, the solution of (13) indeed coincides with x. There are several approaches to this, we refer to [BSS] for a few alternatives.[7] Here we will discuss a deterministic approach based on coherence (from a couple of lecture ago).

Given x a sparse vector, we want to show that x is the unique optimal solution to (13) for $y = \Phi x$,

$$\begin{aligned} \min \quad & \|z\|_1 \\ s.t. \quad & \Phi z = y, \end{aligned}$$

Let $S = \text{supp}(x)$ and suppose that $z \neq x$ is an optimal solution of the ℓ_1 minimization problem. Let $v = z - x$, so $z = v + x$ and

$$\|v+x\|_1 \leq \|x\|_1 \quad \text{and} \quad \Phi(v+x) = \Phi x,$$

this means that $\Phi v = 0$. Also,

$$\|x\|_S = \|x\|_1 \geq \|v+x\|_1 = \|\,(v+x)_S\,\|_1 + \|v_{S^c}\|_1 \geq \|x\|_S - \|v_S\|_1 + \|v\|_{S^c},$$

where the last inequality follows by triangular inequality. This means that $\|v_S\|_1 \geq \|v_{S^c}\|_1$, but since $|S| \ll N$ it is unlikely for Φ to have vectors in its nullspace that are this concentrated on such few entries. This motivates the following definition.

Definition 10.4 (Null Space Property)**.** *Φ is said to satisfy the s-Null Space Property if, for all v in* $\ker(\Phi)$ *(the nullspace of Φ) and all $|S| = s$ we have*

$$\|v_S\|_1 < \|v_{S^c}\|_1.$$

In the argument above, we have shown that if Φ satisfies the Null Space Property for s, then x will indeed be the unique optimal solution to (13). In fact, the converse also holds

Theorem 10.5. *The following are equivalent for $\Phi \in \mathbb{K}^{d\times N}$:*

(1) For any s-sparse vector x, x is the unique optimal solution of (13) *for $y = \Phi x$.*
(2) Φ satisfies the s-Null Space Property.

Challenge 10.2. *We proved* $(1) \Leftarrow (2)$ *in Theorem 10.5. Can you prove* $(1) \Rightarrow (2)$*?*

We now prove the main Theorem of this section, which gives a sufficient condition for exact recovery via ℓ_1 minimization based on the worst case coherence of a matrix, or more precisely of its columns (recall Definition 8.6).

[7] We thank Helmut Bölcskei for suggesting this approach.

Theorem 10.6. *If the worst case coherence μ of a matrix Φ with unit norm vectors satisfies*

$$s < \frac{1}{2}\left(1+\frac{1}{\mu}\right), \tag{15}$$

then Φ satisfies the s-NSP.

Proof. If $\mu = 0$ then $\ker(\Phi) = \emptyset$ and so it must satisfy the NSP for any s, so we focus on $\mu > 0$.

Let $v \in \ker(\Phi)$ and $k \in [N]$, recall that φ_k is the k-th column of Φ, we have

$$\sum_{l=1}^{N} \varphi_l v_l = 0,$$

and so $\varphi_k v_k = -\sum_{l\neq k} \varphi_l v_l$. Since $\|\varphi_k\| = 1$ we have

$$v_k = \varphi_k^* \left(-\sum_{l\neq k} \varphi_l v_l\right) = \left(-\sum_{l\neq k} \varphi_k^* \varphi_l v_l\right).$$

Thus,

$$|v_k| \leq \left|-\sum_{l\neq k} \varphi_k^* \varphi_l v_l\right| \leq \mu \sum_{l\neq k} |v_l| = \mu(\|v\|_1 - |v_k|).$$

This means that for all $k \in [N]$ we have

$$(1+\mu)|v_k| \leq \mu \|v\|_1.$$

Finally, for $S \subset [N]$ of size s we have

$$\|v_S\|_1 = \sum_{k\in S} |v_k| \leq s\frac{\mu}{1+\mu}\|v\|_1 < \frac{1}{2}\|v\|_1,$$

where the last inequality follows from the hypothesis (15) of the Theorem. Since $\|v\|_1 = \|v_S\|_1 + \|v_{S^c}\|_1$ this completes the proof.

In the next lecture we will study matrices with low worst case coherence.

Remark 10.7. *Different approaches roughly follow the following path: Since due to Theorem 10.5 recovery is formulated in terms of certain vectors not belonging to the nullspace of Φ, if one draws Φ from an ensemble of random matrices the problem reduces to understanding when a random subspace (the nullspace of the random matrix) avoids certain vectors, this is the subject of the celebrated "Gordon's Escape through a Mesh Theorem" (see* [BSS]*), you can see versions of this approach also at* [CRPW12] *or, for an interesting approach based on Integral Geometry* [ALMT14].

11. Low Coherence Frames (01.04.2021)

Motivated by Theorem 10.6 in this section we study the worst-case coherence of frames with the goal of understanding how much savings (in measurements) one can achieve with the technique described last section. We start with a lower bound.

Theorem 11.1 (Welch Bound)**.** *Let $\varphi_1,\dots,\varphi_N \in \mathbb{C}^d$ be N unit norm vectors ($\|\varphi_k\| = 1$). Let μ be their worst case coherence*

$$\mu = \max_{i\neq j} \left|\langle \varphi_i, \varphi_j\rangle\right|.$$

Then

$$\mu \geq \sqrt{\frac{N-d}{d(N-1)}}$$

Proof. Let G be the Gram matrix of the vectors, $G_{ij} = \langle \varphi_i, \varphi_j\rangle = \varphi_i^*\varphi_j$. In other words, $G = \Phi^*\Phi$. It is positive semi-definite and its rank is at most d. Let $\lambda_1,\dots,\lambda_d$ denote the largest eigenvalues of G, in particular this includes all non-zero ones. We have

$$(\operatorname{Tr} G)^2 = \left(\sum_{k=1}^{d}\lambda_k\right)^2 \leq d\sum_{k=1}^{d}\lambda_k^2 = d\sum_{k=1}^{N}\lambda_k^2 = d\|G\|_F^2,$$

where the inequality follows from Cauchy-Schwarz between the vectors with the λ_k's and the all-ones vector.

Note that since the vectors are unit normed, $\operatorname{Tr}(G) = N$, thus

$$\sum_{i,j=1}^{N}|\langle \varphi_i, \varphi_j\rangle|^2 = \|G\|_F^2 \geq \frac{1}{d}(\operatorname{Tr} G)^2 = \frac{N^2}{d}.$$

Also,

$$\sum_{i,j=1}^{N}|\langle \varphi_i, \varphi_j\rangle|^2 = \sum_{i=1}^{N}|\langle \varphi_i, \varphi_i\rangle|^2 + \sum_{i\neq j}^{N}|\langle \varphi_i, \varphi_j\rangle|^2 = N + \sum_{i\neq j}^{N}|\langle \varphi_i, \varphi_j\rangle|^2 \leq N + (N^2-N)\mu^2.$$

Putting everything together gives:

$$\mu \geq \sqrt{\frac{\frac{N^2}{d}-N}{(N^2-N)}} = \sqrt{\frac{N-d}{d(N-1)}}.$$

□

Remark 11.2. *We used $\mathbb{C}$ for the Theorem above since it also follows for a frame in $\mathbb{R}^d$ by simply viewing its vectors as elements of $\mathbb{C}^d$.*

Remark 11.3. *Notice that in the proof above there were two inequalities used, if we track the cases when they are "equality" we can see for which frames the Welch bound is tight. The Cauchy-Scwartz inequality is tight when the vector consisting in the first d eigenvalues of G is a multiple of the all-ones vector, which is the case exactly when Φ is a Tight Frame (recall Definition 8.3). The second inequality is tight when all the terms in the sum $\sum_{i\neq j}^{N} |\langle \varphi_i, \varphi_j\rangle|^2$ are equal. The frames that satisfy these properties are called ETFs – Equiangular Tight Frames.*

Definition 11.4 (Equiangular Tight Frame)**.** *A unit-normed Tight frame is called an Equiangular Tight Frame if there exists μ such that, for all $i \neq j$,*

$$|\langle \varphi_i, \varphi_j\rangle| = \mu.$$

Proposition 11.5. *Let $\varphi_1, \ldots, \varphi_N$ by an equiangular tight frame in $\mathbb{K}^d$ then*

- *If $\mathbb{K} = \mathbb{C}$ then $N \leq d^2$*
- *If $\mathbb{K} = \mathbb{R}$ then $N \leq \frac{d(d+1)}{2}$.*

Proof. Let $\psi_i = \mathrm{vec}\left(\varphi_i \varphi_i^*\right)$, these vectors[8] are unit norm and their inner products are

$$\psi_i^* \psi_j = \langle \mathrm{vec}\left(\varphi_i \varphi_i^*\right), \mathrm{vec}\left(\varphi_j \varphi_j^*\right)\rangle = \mathrm{Tr}\left(\left(\varphi_i \varphi_i^*\right)^* \left(\varphi_j \varphi_j^*\right)\right) = |\langle \varphi_i, \varphi_j\rangle|^2 = \mu^2.$$

This means that their Gram matrix H is given by

$$H = \left(1-\mu^2\right) I + \mu^2 \mathbf{1}\mathbf{1}^T.$$

Since $\mu \neq 1$ we have $\mathrm{rank}(H) = N$. But the rank needs to be smaller or equal than the dimension of the φ_i's which

- For $\mathbb{C}^d$ is at most d^2,
- For $\mathbb{R}^d$, due to symmetry it is at most $\frac{1}{2}d(d+1)$.

Thus $N \leq d^2$ for $\mathbb{K} = \mathbb{C}$, and $N \leq \frac{d(d+1)}{2}$ for $\mathbb{K} = \mathbb{R}$. □

Further Reading 11.1. *Equiangular Tight Frames in $\mathbb{C}^d$ with $N = d^2$ are important objects is Quantum Mechanics, where they are called* `SIC-POVM`*: Symmetric, Informationally Complete, Positive Operator-Valued Measure. It is a central open problem to prove that they exist in all dimensions d, see Open Problem 6.3. in* [Ban16] *(the conjecture that they do exist is known as Zauner's Conjecture).*

[8] vec(M) for a matrix M corresponds to the vectors formed by the entries of M; in this case it is not important how the indexing is done as long as consistent throughout.

While constructions of Equiangular Tight Frames are outside the scope of this course,[9] there are simple families of vectors with worst case coherence $\mu \sim \frac{1}{\sqrt{d}}$: Let $F \in \mathbb{C}^{d\times d}$ denote the Discrete Fourier Transform matrix

$$F_{jk} = \frac{1}{\sqrt{d}} \exp\left[-2\pi i (j-1)(k-1)/d\right].$$

F is an orthonormal basis for $\mathbb{C}^d$. Notably any column of F has an inner product of $\frac{1}{\sqrt{d}}$ with the canonical basis, this means that the $d \times 2d$ matrix

$$\Phi = [I\ F] \tag{16}$$

has worst case coherence $\frac{1}{\sqrt{d}}$.

Theorem 10.6 guarantees that, for Φ given by (16), ℓ_1 achieves exact recovery for sparsity levels

$$s < \frac{1}{2}\left(1+\sqrt{d}\right).$$

Remark 11.6. *There are many constructions of unit norm frames with low coherence, with redundancy quotients (N/d) much larger than* 2*. There is a all field of research involving these constructions, there is a "living" article keeping track of constructions* [FM]*. You can also take a look at the PhD thesis of Dustin Mixon* [Mix12] *which describes part of this field, and discusses connections to Compressed Sensing; Dustin Mixon also has a blog in part devoted to these questions* [Mix]*). We will not discuss these constructions here, but Exploratory Challenge 11.4 will show that even randomly picked vectors do quite well (we will do this at the end of the course, as we need some of the probability tools introduced later on).*

Definition 11.7. *Construction* (16) *suggest the notion of Mutually Unbiased Bases: Two orthonormal bases $v_1,\ldots,v_d$ and $u_1,\ldots,u_d$ of $\mathbb{C}^d$ are called Mutually Unbiased if for all i,j we have $|v_i^* u_j| = \frac{1}{\sqrt{d}}$. A set of k bases are called Mutually Unbiased if every pair is Mutually Unbiased.*

Challenge 11.2. *Show that a matrix formed with two orthonormal bases such as* (16) *cannot have worst case coherence smaller than $\frac{1}{\sqrt{d}}$.*

Further Reading 11.3. *Mutually Unbiased basis are an important object in quantum mechanics, communication, and signal processing, however there is still that is not understood about them. My favourite question about them is: "How many mutually unbiased basis exist in* 6 *dimensions"? The best known upper bound is 7 ($d+1$ is always an upper bound, and is known to be tight for prime powers but not in general), the best known lower bound for 6 dimensions is 3. See Open Problem 6.2. in* [Ban16].

[9] I should point out though that there are fascinating connections with Number Theory, Graph Theory, and other areas.

Exploratory Challenge 11.4. *This Challenge is special, it is meant to be solved later in the course (we will likely put it in the homework). Towards the end of the course, equiped with a few more tools of Probability, you'll be able to show that by simply taking a frame made up of random (independent) vectors in the unit norm sphere, the coherence is comparable to the Welch bound, in particular you will show that N such vectors in d dimensions will have worst case coherence* $\frac{\text{polylog}(N)}{\sqrt{d}}$*, where* $\text{polylog}(N)$ *means a polynomial of the logarithm of N (you will also work out the actual dependency).*

Further Reading 11.5. *Turns out that with matrices consisting of random (independent) columns, one can perform sparse recovery with* ℓ_1 *minimization for much larger levels of sparsity (*$s \lesssim \frac{d}{\log(n)}$ *rather than* $s \lesssim \sqrt{d}$*). Proving this however is outside the scope of this course, as it requires heavier Probability Theory machinery. Interestingly, matching this performance with deterministic constructions seems notoriously difficult, in fact there is only one known construction "breaking the square-root bottleneck". You can read more about this in Open Problem 5.1. in* [Ban16] *(and references therein).*

12. Matrix Completion & Recommendation Systems (01.04.2021)

Recommendation Systems is a central application in Machine Learning. In this Section we will focus on the problem of Matrix Completion and will use the Mathematics we developed in the last few sections to motivate algorithms for this problem; while an analysis of these algorithms is outside the scope of this course, they are essentially analogues of the ones developed for sparse recovery. This section, which concludes the first half of the course, is less technical than the previous noes, and more devoted to describing algorithms and applications (and some "story-telling"). Nonetheless we will include a guarantee for matrix completion as a "delayed" homework, as it will be doable with techniques available at the end of the course.

12.1. **Netflix Prize.** The problem of matrix completion is often referred to as the Netflix problem. This is because a bit over a decade ago Netflix launched a contest to improve their recommendation algorithm; teams could have access to a training data set and propose algorithms, the winning team (based on performance on an unseen test data set) received 1 Million dollars. This lasted a few years and it is quite an incredible story. I couldn't possibly describe it here better than the professional press outlets, so I recommended taking a look at the context website: `https://netflixprize.com/index.html`

You can also access the data here:

`https://www.kaggle.com/netflix-inc/netflix-prize-data`

Exploratory Challenge 12.1. *I recommend, if you find the time, to try simple algorithms on this dataset!*

Here are a few articles with more information, it is quite the story!

`https://www.thrillist.com/entertainment/nation/the-netflix-prize`

`https://www.wired.com/2009/09/how-the-netflix-prize-was-won/`

`https://en.wikipedia.org/wiki/Netflix_Prize`

12.2. **The Problem.** Let's use the Netflix problem as the driving example. Let $M \in \mathbb{R}^{n\times m}$ denote the matrix of user/movie ratings (say in either ± 1 or $1--5$ ratings), the goal is to estimate unseen entries of M from a few observed entries. Much like the sparse recovery problem above, this is impossible without (parsimony) assumptions on M. A classically used assumption is that M is low rank (movies and users can be well described as linear combinations of a few "factors"). The Mathematical probelm then becomes:

Problem 12.1 (Low Rank Matrix Completion)**.** *Recover $M \in \mathbb{R}^{n\times m}$ a low rank matrix (meaning* $\text{rank}(M) \leq r$ *for $r \ll n,m$) from observations $\Omega \subset [n]\times[m]$:*

$$M_{ij} = y_{ij}, \quad \textit{for} \quad (i,j) \in \Omega$$

Central questions include

- How large does Ω need to be?
- What are efficient algorithms?

We will discuss the second question first, you can see [H$^+$15] for a nice discussion relating the two algorithms we discuss here. A particularly simple algorithm is motivated by the first part of the course, where we discussed truncated SVD. The idea is to start by setting all non-observed entries to 0, and then update the estimator $\hat{M}$ by repeatedly alternative through the following two iterations

(1) Perform truncated SVD on $\hat{M}$
(2) "Correct" the observed Ω entries of $\hat{M}$ to coincide with the observations.

Both iteration (1) will in general change the entries in Ω and iteration (2) will in general increase the rank of $\hat{M}$, so they need to be performed until convergence (or a stopping criteria is satisfied).

12.3. **Nuclear Norm Minimization.** Another algorithm appears as a natural generalization of ℓ_1 minimization for sparse recovery. Since M is low rank, the vectors of singular values of M, $\sigma(M)$ is sparse. The tools developed in the sections above suggest minimizing the ℓ_1 norm of $\sigma(M)$, which we have seen is the Schatten-1 norm of M (recall Definition 4.3). This suggests the following optimization problem

$$\begin{aligned} \min \quad & \|M\|_{(S,1)} \\ s.t. \quad & M_{ij} = y_{ij} \text{ for all } (i,j) \in \Omega \end{aligned} \tag{17}$$

This is known in the literature as Nuclear Norm miminization and is known to recover a low rank matrix M, under certain incoherence assumptions, from only $r(m+n)\text{polylog}(n+m)$ observations (note that if r is small this is significantly smaller than $n\times m$). While guarantees for this algorithm are outside the scope of this course, as they require tools from random matrix theory, we point out that the key reasons for the success of the algorithm are analogous to the ones for ℓ_1 minimization in sparse recovery. There is a fascinating line of work establishing guarantees for this algorithm [CT10, CR09, Rec11, Gro11]. We point out that solving (17) is indeed efficient, it is a so called Semidefinite Program [VB04].

12.4. **Guarantees.** Here we discuss a guarantee for a particularly simple, but already mathematically rich, example. Let us consider the case of $r = 1$, thus $M = uv^T$ for some unknown vectors u and v. Let us assume further that u and v have only non-zero entries (this is a particularly strong version of the "incoherence assumptions" mentioned above). We are interested in understanding when is it that M is uniquely determined by the set of observations $\Omega \subset [n] \times [m]$, we will relate it with the connectivity of a a certain graph.

Recall that a bipartite graph is a graph with two sets of nodes such that all edges have an endpoint in each of the sets. Given the set Ω consider the graph G_Ω to be the bipartite graph on $n+m$ nodes (corresponding to rows and columns of M) where node $i \in [n]$ and $j \in [m]$ are connected if and only if $i, j \in \Omega$.

Theorem 12.2. *A matrix $M = uv^T$, with u and v both vectors in no zero entry, is uniquely determined by the entry values in $\Omega \subset [n] \times [m]$ if and only if G_Ω is connected.*

Challenge 12.2. *Prove this Theorem.*

As a homework problem layer in the course, you will show that roughly $(m+n)\log(m+n)$ entries independently picked at random are enough to uniquely determine M.

Exploratory Challenge 12.3. *A bipartite Erdös-Renyi random graph $G(n,m,p)$ is a bipartite graph on $n+m$ nodes where every possible edge between the first n nodes and the last m nodes appears, independently, with probability p. For which values of p is the graph connected with high probability? (Similarly to Exploratory Challenge 11.4, this is meant to be solved later in the course, when posted as homework it will contain more information and "stepping stones").*

Exploratory Challenge 12.4. *This result for random graph is more classical on not necessarily bipartite graphs (which would correspond to a symmetric version of the matrix completion problem). An Erdös-Renyi random graph $G(n,p)$ is a graph on n nodes where every possible edge appears, independently, with probability p. For which values of p is the graph connected with high probability? (Similarly to Exploratory Challenge 11.4, this is meant to be solved later in the course, when posted as homework it will contain more information and "stepping stones").*

Remark 12.3. *Although the connection with random matrix theory won't be discussed here, it likely will not be a complete surprise at this point, given that connectivity of a graph is related to properties of the corresponding adjacency matrix. In the rank 1 case, there is no need for random matrix theory and considerably simpler tools from probability theory will suffice.*

13. CLASSIFICATION THEORY: FINITE CLASSES (15.04.2021)

Theory of Classification is a foundational topic in Statistical Machine Learning. The direction we are discussing in this part of the course was initiated by Vladimir Vapnik and Alexey Chervonenkis in the mid-60s and independently by Leslie Valiant in the mid-80s. The results of Vapnik and Chervonenkis lead to what we now call Vapnik–Chervonenkis Theory. From the early 70s to the present day, their work has an ongoing impact on Machine Learning, Statistics, Empirical Process Theory, Computational Geometry and Combinatorics. In parallel, the work of Valiant looked at a similar problem from a more computational perspective. In particular, Valiant developed the theory of Probably Approximately Correct (PAC) Learning that lead, among other contributions, to his 2010 Turing Award.

In this part of the course, we study the following simple model. We are given a sequence of independent identically distributed observations

$$X_1, \dots, X_n$$

taking their values in $\mathscr{X} \subseteq \mathbb{R}^p$ such that each X_i is distributed according to some unknown distribution P [10]. In our case each X_i might be seen as an image or a feature vector. For example, consider the problem of health diagnostics. In this case these vectors can describe some medical information such as age, weight, blood pressure and so on. An important part of our analysis is that the dimension of the space will not play any role, and classification is possible even in abstract measurable spaces.

In contrast to clustering problems discussed earlier, we also observe labels

$$f^*(X_1), \dots, f^*(X_n),$$

where f^* is an (unknown to us) *target* classifier mapping $\mathbb{R}^p \to \{0,1\}$. Depending on a particular problem, these labels can represent spam/not spam when classifying the emails, disease/no disease when diagnosing the patient and so on. Of course, we put a standard measurability assumption on f^* so that e.g., $\{f^*(x) = 1\}$ is measurable. Using the labeled sample

$$S_n = ((X_1, f^*(X_1)), \dots, (X_n, f^*(X_n))) \tag{18}$$

our aim is to construct a measurable classifier $\widehat{f} : \mathscr{X} \to \{0,1\}$ that can be used to classify any further element $x \in \mathscr{X}$. The *risk* or the *error* of a (measurable) classifier $f : \mathscr{X} \to \{0,1\}$ is defined by

$$R(f) = \Pr(f(X) \neq f^*(X)).$$

[10] We assume that there is a probability space $(\mathscr{X}, F, P)$, where F is a Borel σ-algebra.

With this definition in mind, we want to consider classification rules that have their risk as small as possible. Our second assumption is that f^* belongs to some known class $\mathscr{F}$ of (measurable) classifiers mapping $\mathscr{X}$ to $\{0,1\}$. We remark that in the Computer Science literature these classifiers are usually called *concepts*.

Definition 13.1. *We say that a classifier $\widehat{f}$ is consistent with the sample S_n if for all $i = 1,\dots,n$,*

$$\widehat{f}(X_i) = f^*(X_i).$$

Observe that to find a consistent classifier one should observe the labeled sample S_n, but no other knowledge (except the fact that $f^* \in \mathscr{F}$) on f^* is required.

Before presenting our next result, let us give its informal description. Given the sample S_n, the most natural way is to choose any $\widehat{f} \in \mathscr{F}$ consistent with it. Our hope is that this classifier will likely be close to the true classifier f^*. Since the sample S_n is random, we cannot guarantee this definitely. Instead, we may only say that $\widehat{f}$ is close to f^* with high probability: this would mean intuitively that for a large fraction of all random realizations of the sample S_n, any classifier consistent with a particular realization of the sample has a small risk. There is an alternative and equivalent way of looking at this. Instead of saying that with high probability any classifier that is consistent with the sample has a small risk, one may argue that any classifier that has a large risk cannot be consistent with the sample S_n. The latter idea will be central in our proofs.

Theorem 13.2. *Assume that $f^* \in \mathscr{F}$, and $\mathscr{F}$ is finite. For the confidence parameter $\delta \in (0,1)$ and the precision parameter $\varepsilon \in (0,1)$ fix the sample size*

$$n \geq \left\lceil \frac{\log|\mathscr{F}|}{\varepsilon} + \frac{1}{\varepsilon}\log\frac{1}{\delta} \right\rceil.$$

Let $\widehat{f}$ be any classifier in $\mathscr{F}$ consistent *with the sample S_n. Then,*

$$\Pr_{(X_1,\dots,X_n)}\left(R(\widehat{f}) < \varepsilon\right) \geq 1-\delta. \tag{19}$$

Equivalently, with probability at least $1-\delta$, any classifier f such that its risk is larger than ε (that is, $R(f) > \varepsilon$) cannot be consistent with the sample S_n.

Although our proofs are usually based on the second principle, we present our result in the short form (19).

Proof. We follow the strategy discussed above. Fix any $f \in \mathscr{F}$ such that $R(f) = \Pr(f(X) \neq f^*(X)) \geq \varepsilon$. We have,

$$\begin{aligned}
&\Pr_{(X_1,\ldots,X_n)}(f(X_i) = f^*(X_i) \text{ for } i = 1,\ldots,n) \\
&= \prod_{i=1}^{n} \Pr_{X_i}(f(X_i) = f^*(X_i)) \\
&= \prod_{i=1}^{n} (1 - \Pr_{X_i}(f(X_i) \neq f^*(X_i))) \\
&\leq (1-\varepsilon)^n \\
&\leq \exp(-n\varepsilon),
\end{aligned}$$

where in the last line we used $1 - x \leq \exp(-x)$. Recall the union bound: for a set of events $A_1,\ldots,A_k$ we have $\Pr(A_1 \cup \ldots \cup A_k) \leq \sum_{i=1}^{k} \Pr(A_i)$. Now, using the union bound we have

$$\begin{aligned}
&\Pr_{(X_1,\ldots,X_n)}(\text{there is } f \in \mathscr{F},\ R(f) \geq \varepsilon \text{ such that } f(X_i) = f^*(X_i) \text{ for } i = 1,\ldots,n) \\
&\leq \sum_{f \in \mathscr{F}} \Pr_{(X_1,\ldots,X_n)}(R(f) \geq \varepsilon \text{ and } f(X_i) = f^*(X_i) \text{ for } i = 1,\ldots,n) \\
&\leq |\mathscr{F}| \exp(-n\varepsilon),
\end{aligned}$$

where the last line follows from our bound for a single fixed classifier. We want this probability to be small. We set $|\mathscr{F}|\exp(-n\varepsilon) = \delta$. Therefore, it is sufficient to take the sample size

$$n = \left\lceil \frac{\log|\mathscr{F}|}{\varepsilon} + \frac{1}{\varepsilon}\log\frac{1}{\delta} \right\rceil$$

to guarantee that, with probability at least $1-\delta$, any classifier $f \in \mathscr{F}$ consistent with the sample has its risk smaller than ε. Indeed, if n is chosen this way, we have

$$\Pr_{(X_1,\ldots,X_n)}(\text{there is } f \in \mathscr{F},\ R(f) \geq \varepsilon \text{ such that } f(X_i) = f^*(X_i) \text{ for } i = 1,\ldots,n) \leq \delta.$$

Taking the complementary event proves the claim. □

One may rewrite the result of Theorem 13.2 as a *risk bound*. That is, we first fix the sample size n and want to estimate the precision ε of our classifier. Repeating the lines of the proof of Theorem 13.2, we have for any $\widehat{f}$ consistent with S_n,

$$\Pr_{(X_1,\ldots,X_n)}\left(R(\widehat{f}) \leq \frac{\log|\mathscr{F}|}{n} + \frac{1}{n}\log\frac{1}{\delta}\right) \geq 1-\delta.$$

The result of Theorem 13.2 inspires the following definition. In what follows, PAC stands for *Probably Approximately Correct*. Indeed, we showed that for any finite class $\mathscr{F}$ any sample consistent classifier is approximately correct (risk $\leq \varepsilon$) with high probability.

Definition 13.3. *A (possibly infinite) class $\mathscr{F}$ of classifiers is* PAC-learnable *with the* sample complexity $n(\delta,\varepsilon)$ *if there is a mapping $A:\cup_{m=0}^{\infty}(\mathscr{X}\times\{0,1\})^m\to\{0,1\}^{\mathscr{X}}$ (called the* learning algorithm*; given a sample S of any size it outputs a classifier $A(S)$) that satisfies the following property: for* every *distribution P on $\mathscr{X}$,* every $\delta,\varepsilon\in(0,1)$ *and* every *target classifier $f^*\in\mathscr{F}$, if the sample size n is greater or equal than $n(\delta,\varepsilon)$, then*

$$\Pr_{(X_1,\dots,X_n)}(R(A(S_n))\leq\varepsilon)\geq 1-\delta.$$

Remark 13.4. *When considering finite classes we have little problems with measurability and we may only request that for all $f\in\mathscr{F}$ the set $\{f(x)=1\}$ is measurable. The notion of PAC-learnability allows infinite classes. In this case the question of measurability is more subtle. However, as a rule of thumb, one may argue that measurability issues will almost never appear in the analysis of real-life algorithms. In particular, starting from late-*80*s there is a useful and formal notion of* well-behaved *classes: these are essentially the classes for which these measurability issues do not appear. We add some clarifying references thought the text.*

An immediate outcome of Theorem 13.2 is the following result.

Corollary 13.5. *Any finite class $\mathscr{F}$ is PAC learnable with the sample complexity*

$$n=\left\lceil\frac{\log|\mathscr{F}|}{\varepsilon}+\frac{1}{\varepsilon}\log\frac{1}{\delta}\right\rceil.$$

Moreover, to learn this class, we simply need to output any consistent classifier $\widehat{f}$.

14. PAC-Learning for infinite classes: stability and sample compression (15.04.2021)

An important limitation of Theorem 13.2 is that it only deals with finite classes. Working only with discrete spaces of solutions is also somewhat impractical: many modern machine learning techniques are based on the gradient descent methods that require that the class $\mathscr{F}$ is parametrized in a relatively smooth way by $\mathbb{R}^p$. In order to analyze infinite sets $\mathscr{F}$, we can use the following simple argument that can be again attributed to Vapnik and Chervonenkis. Given $f^*\in\mathscr{F}$, consider a non-random labeled sample of size $n+1$:

$$s_{n+1}=\{(x_1,f^*(x_1)),\dots,(x_{n+1},f^*(x_{n+1}))\}.$$

We use lowercase s to emphasize that this sample is non-random. For $i=1,\dots,n+1$, let

$$s_{n+1}^i=S\setminus\{(x_i,f^*(x_i))\}.$$

That is, we removed the i-th observation from the sample S. Let A be any learning algorithm, and denote $\widehat{f^i} = A(s^i_{n+1})$. The Leave-One-Out error (LOO) is defined by

$$\mathrm{LOO}_A = \mathrm{LOO}_A(s_{n+1}) = \frac{1}{n+1}\sum_{i=1}^{n+1} \mathbf{1}_{\widehat{f^i}(x_i)\neq f^*(x_i)},$$

where as before $\mathbf{1}_E$ is an indicator of E. It is easy to see how LOO works: one by one we remove this instances from the sample, construct a new classifier and test its performance on the removed point. Even though the Leave-One-Out error is deterministic, we may also look at it as a random variable by replacing all x_i by corresponding independent identically distributed X_i-s.

Theorem 14.1. *For any learning algorithm A,*

$$\mathbb{E}_{X_1,\dots,X_n} R(A(S_n)) = \mathbb{E}_{X_1,\dots,X_{n+1}} \mathrm{LOO}_A(S_{n+1}).$$

Proof. Since the random variables $X_1,\dots,X_{n+1}$ are independent and identically distributed, the terms in the sum

$$\mathrm{LOO}_A(S_{n+1}) = \frac{1}{n+1}\sum_{i=1}^{n+1} \mathbf{1}_{\widehat{f^i}(X_i)\neq f^*(X_i)}$$

have the same distribution. Therefore, the expected values of each summand are the same. By the linearity of expectation, we have

$$\frac{1}{n+1}\mathbb{E}_{X_1,\dots,X_{n+1}} \sum_{i=1}^{n+1} \mathbf{1}_{\widehat{f^i}(X_i)\neq f^*(X_i)} = \mathbb{E}_{X_1,\dots,X_{n+1}} \mathbf{1}_{\widehat{f^{n+1}}(X_{n+1})\neq f^*(X_{n+1})}.$$

Using the independence of $X_1,\dots,X_{n+1}$, we first take the expectation only with respect to X_{n+1} (one may formally use either the conditional expectation argument or the Fubini theorem). This implies

$$\mathbb{E}_{X_1,\dots,X_{n+1}} \mathbf{1}_{\widehat{f^{n+1}}(X_{n+1})\neq f^*(X_{n+1})} = \mathbb{E}_{X_1,\dots,X_n} R(\widehat{f^{n+1}}) = \mathbb{E}_{X_1,\dots,X_n} R(A(S_n)).$$

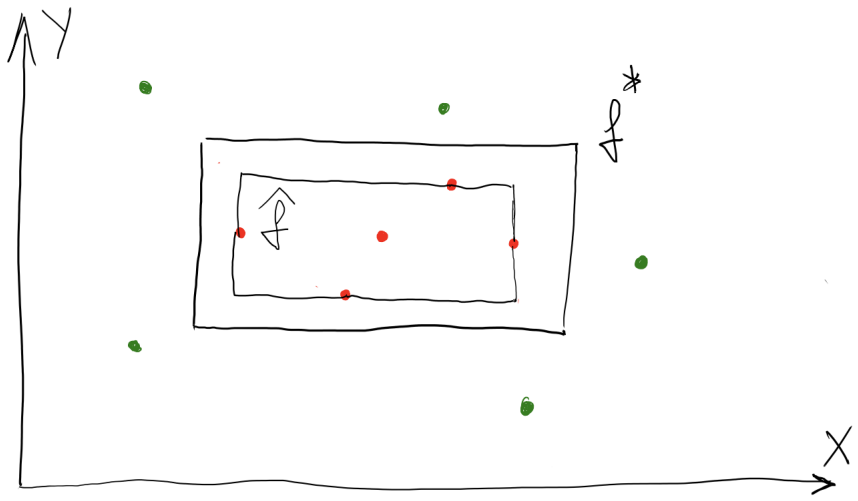
The claim follows. □

Example 14.2. Learning axis-aligned rectangles. *The following simple example can be described in words. Consider the class of classifiers $\mathscr{F}$ induced by axis-aligned rectangles in $\mathbb{R}^2$: that is, for each axis-aligned rectangle we build a classifier that labels its interior and border by* 1 *and its exterior by* 0*. Assume that some unknown target rectangle $f^* \in \mathscr{F}$ is chosen. We need to describe our learning algorithm. Let the classifier $\widehat{f}$ correspond to the smallest axis-aligned rectangle that contains all instances labeled by* 1*. It is easy to check that for any deterministic sample of size n:*

$$\mathrm{LOO}_A(s_{n+1}) \leq \frac{4}{n+1}.$$

Indeed, $\widehat{f^i}$ can be wrong only when removing an instance with the label 1 on the border of the described classifier trained on s_{n+1}. Thus, by Theorem 14.1 we have

$$\mathbb{E}_{X_1,\ldots,X_n} R(\widehat{f}) \leq \frac{4}{n+1}.$$



Challenge 14.1. *Using Example 14.2, prove directly that the class of axis-aligned rectangles is PAC-learnable.*

One of the drawbacks of Theorem 14.1 is that it only controls the expected risk. We show that the ideas used in Example 14.2 can be used in full generality in order to provide high probability sample complexity bounds. Observe that the rectangle $\widehat{f}$ in Example 14.2 is defined by the four instances that belong to its border. In particular, if we remove all instances in the sample except these four, then the same classifier $\widehat{f}$ is constructed. Moreover, this classifier will correctly label the original learning sample. In some sense, we compress the information stored in the sample into a small subsample. This motivates the following definition.

Definition 14.3. *Consider a pair of functions: a* compression function

$$\kappa : \cup_{m=0}^{\infty} (\mathscr{X} \times \{0,1\})^m \to \cup_{m=0}^{\infty} (\mathscr{X} \times \{0,1\})^m$$

(it is a function mapping a sample to a subsample of size ℓ) and a reconstruction function

$$\rho : \cup_{m=0}^{\infty} (\mathscr{X} \times \{0,1\})^m \to \{0,1\}^{\mathscr{X}}$$

(it maps labeled samples to classifiers). The pair (ρ, κ) defines the sample compression scheme of size ℓ for $\mathscr{F}$ if the following holds for any $f^* \in \mathscr{F}$, any integer n and any sample $s_n = ((x_1, f^*(x_1)), \ldots, (x_n, f^*(x_n)))$:

- *it holds that $\kappa(s_n) \subseteq s_n$. That is, compression functions always map to a subset of the original sample;*
- *the size of a compression set satisfies $|\kappa(s_n)| \leq \ell$;*
- *the classifier $\widehat{f} = \rho(\kappa(s_n))$ satisfies for all $i = 1, \ldots, n$, $\widehat{f}(x_i) = f^*(x_i)$. That is, the reconstruction function applied to the compression set recovers the labels of s_n. In other words, $\widehat{f}$ is consistent with the sample s_n.*

Example 14.4. *Check that* Example 14.2 *defines a sample compression scheme of size $\ell = 4$.*

Of course, when considering statistical guarantees we always assume that ρ outputs a measurable classifier.

Theorem 14.5. *Assume that $\mathscr{F}$ admits a sample compression scheme (ρ, κ) of size ℓ and that ρ is permutation invariant (does not depend on the order of input instances). For the confidence parameter $\delta \in (0,1)$ and the precision parameter $\varepsilon \in (0,1)$, fix the sample size*

$$n \geq \ell + \left\lceil \frac{\ell}{\varepsilon} \log \frac{en}{\ell} + \frac{1}{\varepsilon} \log \frac{1}{\delta} \right\rceil.$$

Then, for the classifier $\widehat{f} = \rho(\kappa(S_n))$, it holds that

$$\Pr_{(X_1, \ldots, X_n)} (R(\widehat{f}) \leq \varepsilon) \geq 1 - \delta.$$

Proof. The proof is similar to the proof of Theorem 13.2. Consider without loss of generality the first ℓ elements of S_n. This set is denoted by S_ℓ. This corresponds to the naive compression function κ that always maps to the first ℓ instances of S_n. Using the same arguments as in Theorem 13.2 we have (conditioned on $X_1, \ldots, X_\ell$)

$$\Pr_{X_{\ell+1}, \ldots, X_n} (R(\rho(S_\ell)) \geq \varepsilon \text{ and } \rho(S_\ell)(X_i) = f^*(X_i) \text{ for } i = \ell+1, \ldots, n) \leq \exp(-\varepsilon(n-\ell)).$$

Observe that this probability will only decrease if S_ℓ is replaced by $S_k, k < \ell$. Our idea is to prove the statement no matter which subset is selected by κ. Indeed, by our assumption κ always maps to one of at most (provided that $n \geq \ell$)

$$\sum_{j=0}^{\ell} \binom{n}{j} \leq \left(\frac{en}{\ell}\right)^{\ell}$$

subsets and since ρ is permutation invariant, we ignore the order of the elements in the corresponding set. To verify the last inequality we use

$$\sum_{j=0}^{d}\binom{n}{j} \le \sum_{j=0}^{d}\binom{n}{i}\left(\frac{n}{d}\right)^{d-j} \le \sum_{j=0}^{n}\binom{n}{i}\left(\frac{n}{d}\right)^{d-i} = \left(\frac{n}{d}\right)^{d}\left(1+\frac{d}{n}\right)^{n} \le \left(\frac{en}{d}\right)^{d},$$

where we used the binomial theorem. Therefore, by the union bound, no matter which subset is selected by the compression function κ, we have

$$\begin{aligned}
&\Pr_{X_{\ell+1},\ldots,X_n}(R(\rho(\kappa(S_n))) \ge \varepsilon) \\
&\le \left(\frac{en}{\ell}\right)^{\ell} \mathbb{E}_{X_1,\ldots,X_\ell} \Pr_{X_{\ell+1},\ldots,X_n}(R(\rho(S_\ell)) \ge \varepsilon \text{ and } \rho(S_\ell)(X_i) = f^*(X_i) \text{ for } i = \ell+1,\ldots,n) \\
&\le \left(\frac{en}{\ell}\right)^{\ell} \exp(-\varepsilon(n-\ell)).
\end{aligned}$$

The sample complexity bound follows by a simple computation. □

Challenge 14.2. *Prove an analog of Theorem 14.5 in the case where ρ is not permutation invariant.*

Observe that the existence of a sample compression scheme of finite size implies PAC-learnability. Sample compression is a source of several major open problem in Learning Theory. We discuss some of them later.

Remark 14.6. *One may prove the following identity: for $\alpha,\beta,\gamma > 0, n \ge 1$ if $\alpha\beta e^{\gamma/\alpha} > 2$, then*

$$n > 2\gamma + 2\alpha\ln(\alpha\beta) \quad \textit{implies} \quad n > \gamma + \alpha\log(\beta n).$$

See Corollary 4.1 in [Vid13]. *Therefore, it is easy to verify that our requirement $n > \ell + \frac{\ell}{\varepsilon}\log\frac{en}{\ell} + \frac{1}{\varepsilon}\log\frac{1}{\delta}$ can be replaced by*

$$n > 2\ell + \frac{2\ell}{\varepsilon}\log\frac{e}{\varepsilon} + \frac{2}{\varepsilon}\log\frac{1}{\delta}, \tag{20}$$

The last condition does not contain n in the right-hand side.

Further Reading 14.3. *Theorem 13.2 and the model we are studying appears in the work of Vapnik and Chervonenkis* [VC64a]. *This work also provides a matching lower bound. See also* [Cov65b]. *The work of Valiant* [Val84] *introduces the formal notion of PAC-learnability. Theorem 14.1 appears explicitly in the first textbook of Vapnik and Chervonenkis* [VC74]*, though the argument was known in the mid-60s. See additional historical remarks on this result in* [VPG15, Chapter I]. *Theorem 14.5 is essentially due to Littlestone and Warmuth* [LW86]. *Another recommended reference with many related discussions is* [HR21].

15. PERCEPTRON (22.04.2021)

In this lecture, we analyze the Perceptron algorithm. This algorithm was invented in the late-50s by Frank Rosenblatt. Even though it is one of the first learning algorithms, Perceptron has close relations to multiple foundational methods and principles such as stochastic gradient descent and sample compression schemes. Later in our course, we also make a connection with online learning. Moreover, Perceptron is a building block of artificial neural networks.

For notational convenience, we work with the labels $\{-1,1\}$ instead of $\{0,1\}$ when considering the Perceptron algorithm.

Definition 15.1. *We say that a set of labeled vectors $W \subset \mathbb{R}^p \times \{1,-1\}$ is linearly separable with a margin γ if there is a vector $v \in \mathbb{R}^p$ such that for any $(x,y) \in W$, where $x \in \mathbb{R}^p$ and $y \in \{1,-1\}$, it holds that*

$$\frac{y\langle v,x\rangle}{\|v\|} \geq \gamma.$$

The intuition behind the margin is as follows. From linear algebra we know that the distance from x to the separating hyperplane induced by v is equal to

$$\frac{|\langle v,x\rangle|}{\|v\|}.$$

The margin assumption means that each (x,y) lives in a half-space corresponding to its label y and the distance from x to the separating hyperplane is at least γ. In fact, for any vector $w \in \mathbb{R}^p$ there is a natural way to define a classifier: we consider the mapping $x \mapsto \operatorname{sign}(\langle w,x\rangle)$. That is, the vector w induces a half-space and is orthogonal to the separating hyperplane. In particular, w misclassifies $x \in \mathbb{R}^p$ whenever

$$y\langle w,x\rangle \leq 0.$$

Remark 15.2. *For the sake of simplicity, in this lecture we focus on homogenous half-spaces. That is, we assume that the separating hyperplane passes through the origin.*

Our first aim will be the following: given a linearly separable sample of labeled vectors $s_n = \{(x_1,y_1),\dots,(x_n,y_n)\}$, find a vector $\widehat{w}$ that certifies this linear separability. The following algorithm is aiming to do so.

Perceptron Algorithm.

- Input: $s_n = \{(x_1,y_1),\dots,(x_n,y_n)\}$.
- Set $w_1 = 0$.
- For $i = 1,\dots,n$ do

(1) If $y_i\langle w_i, x_i\rangle \leq 0$
(2) $\quad w_{i+1} = w_i + y_i x_i$,
(3) Else
(4) $\quad w_{i+1} = w_i$,

- Return: w_{n+1}.

The simplest interpretation of this algorithm is as follows. Whenever w_i misclassifies x_i, we update it by using the rule $w_{i+1} = w_i + y_i x_i$. This implies, in particular, that

$$y_i\langle w_{i+1}, x_i\rangle = y_i\langle w_i, x_i\rangle + y_i^2\|x_i\|^2 = y_i\langle w_i, x_i\rangle + \|x_i\|^2.$$

The additional positive term $\|x_i\|^2$ implies that w_{i+1} is less likely to misclassify x_i. The next result is the basic error bounds for the Perceptron algorithm.

Theorem 15.3. *Assume that we are given a finite set of labeled vectors $s_n = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ in $\mathbb{R}^p$ that is linearly separable with a margin γ. Let $\max_{i=1,\ldots,n} \|x_i\| \leq r$, for some $r > 0$. The number of updates (misclassifications) made by the Perceptron algorithm when processing s_n is bounded by r^2/γ^2. This values does not depend on n.*

Proof. Let $J \subseteq \{1, \ldots, n\}$ denote the set of indexes where the Perceptron algorithm updates the weight vector. By $\frac{y\langle v, x\rangle}{\|v\|} \geq \gamma$ and the linearity of the inner product, we have

$$|J|\gamma \leq \sum_{i\in J} \frac{y_i\langle v, x_i\rangle}{\|v\|}.$$

By the Cauchy-Schwarz inequality and the definition of the update rule we obtain

$$\sum_{i\in J} \frac{y_i\langle x_i, v\rangle}{\|v\|} \leq \left\|\sum_{i\in J} y_i x_i\right\| = \left\|\sum_{i\in J} (w_{i+1} - w_i)\right\| = \|w_{\max(J)+1}\|,$$

where in the last line equality we used telescopic sums (with $w_1 = 0$). Further, using telescopic sums again, we have

$$\begin{aligned}
\|w_{\max(J)+1}\| &= \sqrt{\sum_{i\in J} (\|w_{i+1}\|^2 - \|w_i\|^2)} \\
&= \sqrt{\sum_{i\in J} (\|w_i + y_i x_i\|^2 - \|w_i\|^2)} \\
&\leq \sqrt{\sum_{i\in J} (2y_i\langle w_i, x_i\rangle + \|x_i\|^2)} \qquad (\text{since } y_i\langle w_i, x_i\rangle \leq 0) \\
&\leq \sqrt{\sum_{i\in J} \|x_i\|^2} \leq \sqrt{|J| r^2}.
\end{aligned}$$

Therefore, we have

$$|J|\gamma \leq \sqrt{|J|r^2} \implies |J| \leq r^2/\gamma^2.$$

The claim follows. □

Observe that we only restrict the norms of the vectors that correspond to the instances misclassified by the Perceptron algorithm. These instances are usually called the support vectors. We are ready to show that these support vectors define a sample compression scheme.

Proposition 15.4. *Consider the case of linearly separable data with a margin γ and such that the norms of the vectors are restricted to the ball of radius r. Then, the Perceptron algorithm defines a sample compression scheme of size $\ell \leq r^2/\gamma^2$.*

Proof. The exposition of the proof is again simplified if we present it informally. We first need to order the vectors in $\mathbb{R}^p$. For any $x, y \in \mathbb{R}^p$ we write $x \prec y$ if $x_j < y_j$, where $j \in \{1, \ldots, p\}$ is the first index where $x_j \neq y_j$. Given a learning sample s_n, we always first sort the instances in ascending order according to $\prec$. Consider running the Perceptron algorithm through s_n and cycling through this ordered set repeatedly until it makes a full pass through it without making any mistakes. This procedure will surely terminate since the sample s_n is finite, and by Theorem 15.3, we only make a finite number of mistakes on a sample of arbitrary length.

Let $s_n^* \subseteq s_n$ denote the set of mistakes of the Perceptron algorithm executed as above. By Theorem 15.3 we have $|s_n^*| \leq r^2/\gamma^2$. Therefore, the Perceptron algorithm defines a compression function of size $\ell \leq r^2/\gamma^2$. This function maps s_n to s_n^*. We only need to construct a reconstruction function. Fix any $x \in \mathbb{R}^p$ that does not have a labeled copy in s_n^*. We put the unlabeled x among s_n^* and sort them according $\prec$. We run the Perceptron algorithm on the ordered set $s_n^* \cup \{x\}$ until the label of x is determined. That is, the label of x is $\operatorname{sign}(\langle x, w \rangle)$, where w is the current vector when x is processed. For any x that has a copy in s_n^* we return its correct label. Indeed, in this case the label of x is stored in s_n^*. This defines (point-wise) the reconstruction function. By our construction for any $x \in s_n$ its label will be correctly classified by this reconstruction function. Thus, we defined a sample compression scheme of size $\ell \leq r^2/\gamma^2$. □

Challenge 15.1. *Make this proof formal.*

Challenge 15.2. *Prove that alternatively one may use the following reconstruction function: run the Perceptron algorithm through s_n^* and cycling through this ordered set repeatedly until it makes a full pass through it without making any mistakes. Any new point x is now classified by the final output vector $\widehat{w}$. Argue that in this case $\rho(\kappa(S_n))$ matches the output of the Perceptron algorithm trained until it passes through S_n without making any mistakes.*

Remark 15.5. *For the reconstruction function of Proposition 15.4 one may use a compression function making only a single pass through the sample.*

Thus, in our setup the sample complexity bound (20) implies that if the sample size

$$n > \frac{2r^2}{\gamma^2} + \frac{2r^2}{\gamma^2\varepsilon}\log\frac{e}{\varepsilon} + \frac{2}{\varepsilon}\log\frac{1}{\delta},$$

then the output classifier $\widehat{w}$ of the Perceptron-based sample compression scheme of Proposition 15.4 satisfies

$$\Pr_{(X_1,\ldots,X_n)}\left(R\left(\widehat{w}\right) \leq \varepsilon\right) \geq 1-\delta.$$

Challenge 15.3. *Verify that the Leave-One-Out argument (Theorem 14.1) can also be used to provide an in-expectation performance of the Perceptron algorithm.*

The renowned Support-Vector Machine algorithm (SVM) was introduced in the early work of Vapnik and Chervonenkis [VC64b] as a particular modification of the Perceptron algorithm. Due to time restrictions and a more involved theoretical analysis, we omit this algorithm in this course.

16. Basic concentration inequalities (22.04.2021)

Before proceeding with more involved results we need to derive several technical results. First, we need to introduce a moment generating function. Given a random variable define the real-valued function M_X as

$$M_X(\lambda) = \mathbb{E}\exp(\lambda X),$$

whenever this expectation exists. For example, it is standard to verify that if X is distributed according to the normal law with mean 0 and variance σ^2, then

$$\mathbb{E}\exp(\lambda X) = \exp(\lambda^2\sigma^2/2).$$

We show that a similar relation holds for any zero mean bounded random variable. This result is originally due to Wassily Hoeffding. In this proof we use Jensen's inequality: if X is an integrable random variable and φ is a convex function, then $\varphi(\mathbb{E}X) \leq \mathbb{E}\varphi(X)$.

Lemma 16.1. *Let X be a zero mean random variable ($\mathbb{E}X = 0$) such that $X \in [a,b]$ almost surely. Then,*

$$M_X(\lambda) \leq \exp(\lambda^2(b-a)^2/8).$$

Proof. For the sake of presentation, we only provide a proof of a slightly weaker inequality:

$$M_X(\lambda) \leq \exp(\lambda^2(b-a)^2/2).$$

Consider the random variable X' that is an independent copy of X. When taking the expectation with respect to this independent random variable, we write $\mathbb{E}'$ instead of $\mathbb{E}$. Observe that $\mathbb{E}'X' = 0$.

Therefore, by the linearity of expectation and Jensen's inequality we have

$$M_X(\lambda) = \mathbb{E}\exp(\lambda X) = \mathbb{E}\exp(\lambda(X - \mathbb{E}'X')) \leq \mathbb{E}\mathbb{E}'\exp(\lambda(X - X')).$$

We use the idea of symmetrization. Let ε be a random variable independent of X and X' and taking the values ± 1 each with probability $1/2$. This random variable is usually referred to as the Rademacher random variable. By symmetry we have that

$$X - X' \stackrel{d}{=} X' - X \stackrel{d}{=} \varepsilon(X - X'),$$

where the symbol $\stackrel{d}{=}$ denotes that corresponding random variables have the same distribution. Because random variables with the same distribution have the same expectation, we have

$$M_X(\lambda) \leq \mathbb{E}\mathbb{E}'\mathbb{E}_\varepsilon \exp(\lambda\varepsilon(X - X')),$$

and we can change the order of integration (taking the expectation) by the Fubini theorem. Let's take the expectation with respect to ε first. We have

$$\mathbb{E}_\varepsilon \exp(\lambda\varepsilon(X - X')) = \frac{1}{2}\left(\exp(\lambda(X - X')) + \exp(\lambda(X' - X))\right) \leq \exp(\lambda^2(X - X')^2/2),$$

where we used that $(\exp(x) + \exp(-x))/2 \leq \exp(x^2/2)$. Observe that $|X - X'| \leq b - a$ almost surely. Thus, we have

$$M_X(\lambda) \leq \mathbb{E}\mathbb{E}'\exp(\lambda\varepsilon(b-a)^2/2) = \exp(\lambda\varepsilon(b-a)^2/2).$$

The claim follows. □

Let Y be a random variable. Denote its moment generating function by M_Y. For any $\lambda, t > 0$, we have

$$\Pr(Y \geq t) = \Pr(\lambda Y \geq \lambda t) = \Pr(\exp(\lambda Y) \geq \exp(\lambda t)) \leq \exp(-\lambda t) M_Y(\lambda),$$

where the last inequality follows from Markov's inequality: for a random variable Z such that $Z \geq 0$ almost surely and any $t > 0$, it holds that $\Pr(Z \geq t) \leq \mathbb{E}Z/t$. Therefore, we have

$$\Pr(Y \geq t) \leq \inf_{\lambda > 0} \exp(-\lambda t) M_Y(\lambda).$$

This relation is usually called the Chernoff method. We are now ready to prove the basic concentration inequality for bounded random variables.

Theorem 16.2 (Hoeffding's inequality)**.** *Let $X_1, \ldots, X_n$ be independent random variables such that $X_i \in [a_i, b_i]$ almost surely for $i = 1, \ldots, n$. Then, for any $t \geq 0$, it holds that*

$$\Pr\left(\sum_{i=1}^n (X_i - \mathbb{E}X_i) \geq t\right) \leq \exp\left(\frac{-2t^2}{\sum_{i=1}^n (b_i - a_i)^2}\right).$$

Similarly, we have

$$\Pr\left(\sum_{i=1}^{n}(\mathbb{E}X_i - X_i) \geq t\right) \leq \exp\left(\frac{-2t^2}{\sum_{i=1}^{n}(b_i-a_i)^2}\right).$$

Moreover,

$$\Pr\left(\sum_{i=1}^{n}|X_i - \mathbb{E}X_i| \geq t\right) \leq 2\exp\left(\frac{-2t^2}{\sum_{i=1}^{n}(b_i-a_i)^2}\right).$$

Proof. We proceed with the following lines. For any $\lambda > 0$, it holds that

$$\begin{aligned}
\Pr\left(\sum_{i=1}^{n}(X_i - \mathbb{E}X_i) \geq t\right) &\leq \exp(-\lambda t)\,\mathbb{E}\exp\left(\lambda\sum_{i=1}^{n}(X_i - \mathbb{E}X_i)\right) \\
&= \exp(-\lambda t)\prod_{i=1}^{n}\mathbb{E}\exp(\lambda(X_i - \mathbb{E}X_i)) \quad \text{(by independence)} \\
&\leq \exp(-\lambda t)\prod_{i=1}^{n}\exp\left(\lambda^2(b_i-a_i)^2/8\right) \quad \text{(by Hoeffding's lemma)} \\
&= \exp(-\lambda t)\exp\left(\lambda^2\sum_{i=1}^{n}(b_i-a_i)^2/8\right).
\end{aligned}$$

Observe that we used that the length of the interval to which $X_i - \mathbb{E}X_i$ belongs is the same as the corresponding length for X_i. Choosing $\lambda = 4t/\sum_{i=1}^{n}(b_i-a_i)^2$, we prove the first inequality. The proof of the second inequality follows the same lines. Finally, by the union bound

$$\begin{aligned}
\Pr\left(\sum_{i=1}^{n}|X_i - \mathbb{E}X_i| \geq t\right) &\leq \Pr\left(\sum_{i=1}^{n}(X_i - \mathbb{E}X_i) \geq t\right) + \Pr\left(\sum_{i=1}^{n}(\mathbb{E}X_i - X_i) \geq t\right) \\
&\leq 2\exp\left(\frac{-2t^2}{\sum_{i=1}^{n}(b_i-a_i)^2}\right).
\end{aligned}$$

The claim follows. □

Further Reading 16.1. *The Perceptron algorithm appears first in* [Ros57]. *Theorem 15.3 is due to Novikoff* [Nov63]. *The error bound for the Perceptron algorithm based on the leave-one-out argument is due to Vapnik and Chervonenkis* [VC74]. *For more details on Perceptron and SVM, see the textbook* [MRT18]. *The sharpest known high probability bounds for both algorithms in the PAC model are presented in* [BHMZ20, HK21]. *Hoeffding's inequality appears in the foundational work of Hoeffding* [Hoe63]. *Similar techniques were used in 1920-s by Bernstein and Kolmogorov* [Kol29].

17. Uniform Convergence of Frequencies of Events to Their Probabilities (29.04.2021)

Assume that we are given a probability space $(\mathscr{X}, F, P)$. Let $\mathscr{A}$ be a collection of events. These are measurable subsets of $\mathscr{X}$. Given a sample

$$X_1, \dots, X_n$$

of independent random variables each distributed according to P, we define for any $A \in \mathscr{A}$ the frequency ν_A of this event. That is, we set

$$\nu_n(A) = \nu_n(A, X_1, \dots, X_n) = \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}_{X_i \in A},$$

where $\mathbf{1}_E$ is an indicator of E. Observe that $\mathbb{E}\mathbf{1}_{X_i \in A} = P(A)$. By the law of large numbers we know that for any given event A,

$$\nu_n(A) - P(A) \overset{\text{a.s.}}{\to} 0.$$

However, in what follows, we are interested in the analysis of this convergence uniformly over all events in $\mathscr{A}$. Therefore, we consider the random variable

$$\sup_{A \in \mathscr{A}} |\nu_n(A) - P(A)|.$$

We always assume that this is a measurable function. As above, if $\mathscr{A}$ is finite or even countable, then no problems of this sort appear. However, for infinite classes of events we need some relatively mild assumptions to work with $\sup_{A \in \mathscr{A}} |\nu_n(A) - P(A)|$. We discuss them shortly in what follows. When $\mathscr{A}$ is finite, combining the union bound and the Hoeffding inequality, we have for any $t \geq 0$,

$$\begin{aligned} \Pr\left(\max_{A \in \mathscr{A}} |\nu_n(A) - P(A)| \geq t \right) &\leq \sum_{A \in \mathscr{A}} \Pr\left(|\nu_n(A) - P(A)| \geq t \right) \\ &\leq 2|\mathscr{A}| \exp(-2nt^2), \end{aligned}$$

where we used that in our case $b_i = 1, a_i = 1$ for $i = 1, \dots, n$. To present this result in a more convenient form, we set $\delta = 2|\mathscr{A}| \exp(-2nt^2)$. A simple computation shows that, with probability at least $1 - \delta$, it holds that

$$\max_{A \in \mathscr{A}} |\nu_n(A) - P(A)| \leq \sqrt{\frac{1}{2n} \left(\log(2|\mathscr{A}|) + \log \frac{1}{\delta} \right)}.$$

The analysis becomes more complicated when $\mathscr{A}$ is infinite. Indeed, in this case the term $\log(2|\mathscr{A}|)$ is unbounded.

Definition 17.1. *Given a family of events $\mathscr{A}$ the growth (shatter) function $\mathscr{S}_{\mathscr{A}}$ is defined by*

$$\mathscr{S}_{\mathscr{A}}(n) = \sup_{x_1,\dots,x_n} |\{(\mathbf{1}_{x_1\in A},\dots,\mathbf{1}_{x_n\in A}) : A \in \mathscr{A}\}|.$$

That is, the growth function bounds the number of projections of $\mathscr{A}$ on the sample $x_1,\dots,x_n$.

Observe that $\mathscr{S}_{\mathscr{A}}(n) \le 2^n$. Let us give some simple examples:

(1) The growth function of a finite family of events satisfies $\mathscr{S}_{\mathscr{A}}(n) \le |\mathscr{A}|$.
(2) Assume that $\mathscr{X} = \mathbb{R}$ and that $\mathscr{A}$ consists of the sets induced by all rays of the form $x \le t$, $t \in \mathbb{R}$. Then, $\mathscr{S}_{\mathscr{A}}(n) = n+1$.
(3) Assume that $\mathscr{X} = \mathbb{R}$ and $\mathscr{X}$ consists of all open sets in $\mathbb{R}$. Then, $\mathscr{S}_{\mathscr{A}}(n) = 2^n$.

We are ready to formulate the main result of this lecture.

Theorem 17.2 (Vapnik-Chervonenkis Theorem)**.** *Consider a family of events $\mathscr{A}$ with the growth function $\mathscr{S}_{\mathscr{A}}$. For any $t \ge \sqrt{\frac{2}{n}}$, it holds that*

$$\Pr\left(\sup_{A\in\mathscr{A}} |\nu_n(A) - P(A)| \ge t\right) \le 8\mathscr{S}_{\mathscr{A}}(n)\exp(-nt^2/32).$$

In particular, with probability at least $1-\delta$, we have

$$\sup_{A\in\mathscr{A}} |\nu_n(A) - P(A)| \le 4\sqrt{\frac{2}{n}\left(\log(8\mathscr{S}_{\mathscr{A}}(n)) + \log\frac{1}{\delta}\right)}.$$

The main ingredient of the proof is the following lemma. First, recall that

$$|\nu_n(A) - P(A)| = \left|\frac{1}{n}\sum_{i=1}^n (\mathbf{1}_{X_i\in A} - P(A))\right|.$$

Lemma 17.3 (Symmetrization lemma)**.** *Assume that $\varepsilon_1,\dots,\varepsilon_n$ are independent (from each other and from X_i, $i = 1,\dots,n$) random variables taking the values ± 1 each with probability $1/2$. Then, for any $t \ge \sqrt{\frac{2}{n}}$, it holds that*

$$\Pr_{X_1,\dots,X_n}\left(\sup_{A\in\mathscr{A}}\left|\frac{1}{n}\sum_{i=1}^n (\mathbf{1}_{X_i\in A} - P(A))\right| \ge t\right) \le 4\Pr_{\substack{X_1,\dots,X_n,\\ \varepsilon_1,\dots,\varepsilon_n}}\left(\sup_{A\in\mathscr{A}}\left|\frac{1}{n}\sum_{i=1}^n \varepsilon_i \mathbf{1}_{X_i\in A}\right| \ge t/4\right).$$

Proof. Assume that $X_1',\dots,X_n'$ is an independent copy of $X_1,\dots,X_n$ and set

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

Given $X_1,\dots,X_n$ assume that a (random) $A_n \subset \mathscr{A}$ achieves the supremum. We may assume it without loss of generality as otherwise there is A_n' that gives an arbitrary close value and taking

the limits carefully will give the same result. We have

$$\sup_{A\in\mathscr{A}} |\nu_n(A) - P(A)| = |\nu_n(A_n) - P(A_n)|.$$

Further, for $t \geq 0$, we may write the following deterministic relation

$$\begin{aligned}\mathbf{1}_{|\nu_n(A_n)-P(A_n)|\geq t}\mathbf{1}_{|\nu_n'(A_n)-P(A_n)|<t/2} &\leq \mathbf{1}_{|\nu_n(A_n)-P(A_n)|-|\nu_n'(A_n)-P(A_n)|\geq t/2}\\ &\leq \mathbf{1}_{|\nu_n(A_n)-\nu_n'(A_n)|\geq t/2},\end{aligned}$$

where is the last line we used the triangle inequality. Now we take the expectation of both sides of this inequality with respect to $X_1', \ldots, X_n'$. We have,

$$\left(\mathbf{1}_{|\nu_n(A_n)-P(A_n)|\geq t}\right) \cdot \Pr_{X_1',\ldots,X_n'}\left(\left|\nu_n'(A_n) - P(A_n)\right| < t/2\right) \leq \Pr_{X_1',\ldots,X_n'}\left(\left|\nu_n(A_n) - \nu_n'(A_n)\right| \geq t/2\right).$$

Observe that A_n depends on $X_1, \ldots, X_n$ but does not depend on $X_1', \ldots, X_n'$. Thus, by Chebyshev's inequity and independence of $X_1', \ldots, X_n'$ we have

$$\Pr_{X_1',\ldots,X_n'}\left(\left|\nu_n'(A_n) - P(A_n)\right| \geq t/2\right) = \frac{4}{t^2}\operatorname{Var}\left(\frac{1}{n}\sum_{i=1}^{n}(\mathbf{1}_{X_i'\in A_n} - P(A_n))\right) \leq \frac{1}{nt^2},$$

where we used that for a Bernoulli random variable taking its values in $\{0,1\}$ its variance is at most $\frac{1}{4}$. Therefore, considering the complementary event we have

$$\Pr_{X_1',\ldots,X_n'}\left(\left|\nu_n'(A_n) - P(A_n)\right| < t/2\right) \geq \frac{1}{2},$$

whenever $\frac{1}{nt^2} \leq 2$. Thus, if $t \geq \sqrt{\frac{2}{n}}$, we have

$$\left(\mathbf{1}_{|\nu_n(A_n)-P(A_n)|\geq t}\right) \leq 2\Pr_{X_1',\ldots,X_n'}\left(\left|\nu_n(A_n) - \nu_n'(A_n)\right| \geq t/2\right).$$

Taking the expectation with respect to $X_1, \ldots, X_n$, and using the symmetrization argument (that is, $\mathbf{1}_{X_i\in A} - \mathbf{1}_{X_i'\in A}$ has the same distribution as $\varepsilon_i(\mathbf{1}_{X_i\in A} - \mathbf{1}_{X_i'\in A})$) we obtain

$$\begin{aligned}\Pr_{X_1,\ldots,X_n}(|\nu_n(A_n) - P(A_n)| \geq t) &\leq 2\Pr_{\substack{X_1,\ldots,X_n,\\ X_1',\ldots,X_n'}}\left(\left|\nu_n(A_n) - \nu_n'(A_n)\right| \geq t/2\right)\\ &\leq 2\Pr_{\substack{X_1,\ldots,X_n,\\ X_1',\ldots,X_n'}}\left(\left|\nu_n(A_n) - \nu_n'(A_n)\right| \geq t/2\right)\\ &\leq 2\Pr_{\substack{X_1,\ldots,X_n,\\ X_1',\ldots,X_n'}}\left(\sup_{A\in\mathscr{A}}\left|\nu_n(A_n) - \nu_n'(A_n)\right| \geq t/2\right)\\ &= 2\Pr\left(\sup_{A\in\mathscr{A}}\left|\frac{1}{n}\sum_{i=1}^{n}\varepsilon_i(\mathbf{1}_{X_i\in A} - \mathbf{1}_{X_i'\in A})\right| \geq t/2\right),\end{aligned}$$

where the last probability symbol corresponds to the joint distribution of X_i, X_i', ε_i for all $i = 1, \dots, n$. Finally, using the triangle inequality and the union bound, we obtain

$$\begin{aligned}
&2\Pr\left(\sup_{A\in\mathscr{A}}\left|\frac{1}{n}\sum_{i=1}^{n}\varepsilon_i(\mathbf{1}_{X_i\in A}-\mathbf{1}_{X_i'\in A})\right|\geq t/2\right)\\
&\leq 2\Pr\left(\sup_{A\in\mathscr{A}}\left|\frac{1}{n}\sum_{i=1}^{n}\varepsilon_i\mathbf{1}_{X_i\in A}\right|+\sup_{A\in\mathscr{A}}\left|\frac{1}{n}\sum_{i=1}^{n}\varepsilon_i\mathbf{1}_{X_i'\in A}\right|\geq t/4+t/4\right)\\
&\leq 4\Pr_{\substack{X_1,\dots,X_n,\\ \varepsilon_1,\dots,\varepsilon_n}}\left(\sup_{A\in\mathscr{A}}\left|\frac{1}{n}\sum_{i=1}^{n}\varepsilon_i\mathbf{1}_{X_i\in A}\right|\geq t/4\right).
\end{aligned}$$

The claim follows. □

Proof. (of Theorem 17.2) Using the symmetrization lemma, we consider the term

$$4\Pr_{\substack{X_1,\dots,X_n,\\ \varepsilon_1,\dots,\varepsilon_n}}\left(\sup_{A\in\mathscr{A}}\left|\frac{1}{n}\sum_{i=1}^{n}\varepsilon_i\mathbf{1}_{X_i\in A}\right|\geq t/4\right).$$

Our key observation is that even though the set of events $\mathscr{A}$ is infinite, there are at most $\mathscr{S}_{\mathscr{A}}(n)$ realizations of $(\mathbf{1}_{X_1\in A},\dots,\mathbf{1}_{X_n\in A})$ for a given sample $X_1,\dots,X_n$. Thus, using the union bound together with the Hoeffding inequality (observe that $\varepsilon_i\mathbf{1}_{X_i\in A}\in[-1,1]$ and zero mean), we have

$$\begin{aligned}
&4\Pr_{\substack{X_1,\dots,X_n,\\ \varepsilon_1,\dots,\varepsilon_n}}\left(\sup_{A\in\mathscr{A}}\left|\frac{1}{n}\sum_{i=1}^{n}\varepsilon_i\mathbf{1}_{X_i\in A}\right|\geq t/4\right)\\
&\leq 4\mathbb{E}_{X_1,\dots,X_n}\left(\mathscr{S}_{\mathscr{A}}(n)\sup_{A\in\mathscr{A}}\Pr_{\varepsilon_1,\dots,\varepsilon_n}\left(\sup_{A\in\mathscr{A}}\left|\frac{1}{n}\sum_{i=1}^{n}\varepsilon_i\mathbf{1}_{X_i\in A}\right|\geq t/4\right)\right)\\
&\leq 4\mathbb{E}\mathscr{S}_{\mathscr{A}}(n)\cdot\left(2\exp(-2nt^2/(4\cdot 16))\right)\\
&= 4\mathscr{S}_{\mathscr{A}}(n)\cdot\left(2\exp(-2nt^2/(4\cdot 16))\right)\\
&= 8\mathscr{S}_{\mathscr{A}}(n)\exp(-nt^2/32).
\end{aligned}$$

The claim follows. □

Remark 17.4. *The above results requires that the random variable* $\sup_{A\in\mathscr{A}}|\nu_n'(A)-\nu'(A)|$ *is measurable. See Chapter 2 in* [VPG15] *for a more detailed discussion and relevant references. We additionally refer to Appendix A.1 in* [BEHW89].

Challenge 17.1. *Improve the constants in the uniform convergence theorem by directly analyzing* $\Pr\left(\sup_{A\in\mathscr{A}}\left|\frac{1}{n}\sum_{i=1}^{n}\varepsilon_i(\mathbf{1}_{X_i\in A}-\mathbf{1}_{X_i'\in A})\right|\geq t/2\right)$ *instead of introducing Rademacher random signs* ε_i.

Further Reading 17.2. *The uniform convergence theorem and the growth function appear in the foundational work of Vapnik and Chervonenkis* [VC71]. *Symmetrization with random Rademacher*

signs appears in [GZ84]. *A modern presentation of similar results can be found in the textbook* [Ver18].

18. The Vapnik-Chervonenkis dimension (06.05.2021)

In this lecture, we introduce one of the key components of the theory of uniform convergence.

Definition 18.1. *Given a family of events $\mathscr{A}$, the Vapnik Chervonenkis (VC) dimension of $\mathscr{A}$ is the largest integer d such that $\mathscr{S}_{\mathscr{A}}(d) = 2^d$. That is, it is the size of the largest subset of $\mathscr{X}$ such that when $\mathscr{A}$ is restricted on this set, all possible 2^d projections are realized. If no such finite integer exists, we set $d = \infty$.*

We say that a finite set $\{x_1, \dots, x_m\} \subset \mathscr{X}$ is *shattered* by $\mathscr{A}$ if the number of restrictions of $\mathscr{A}$ on $\mathscr{X}'$ (the size of the set $\{(\mathbf{1}_{x_1 \in A}, \dots, \mathbf{1}_{x_m \in A}) : A \in \mathscr{A}\}$) is equal to 2^m. In other words, the VC dimension is the size of the largest shattered subset of $\mathscr{X}$. First, we consider several simple examples.

Example 18.2. *The VC dimension of the family of events induced by closed intervals in $\mathbb{R}$ is equal to 2. This is because a pair of distinct points can be shattered. But there is no interval that contains two points but does not contain a point between them. Thus, the set of three points cannot be shattered.*

Example 18.3. *The VC dimension of the family of events induced by halfspaces in $\mathbb{R}^2$ (not necessarily passing through the origin) is equal to 3. Indeed, a set of three distinct points can be shattered in all possible 2^3 ways. At the same time, for a set of 4 points it is impossible to shatter the set in a way such that two diagonals of the corresponding rectangle are in two different halfspaces.*

The following result relates the VC dimension and the growth function. Quite surprisingly, this theorem was shown by several authors independently around the same time. While Vapnik-Chervonenkis were motivated by uniform convergence, other authors looked at it from a different perspective. Currently there are several known techniques that can be used prove this result.

Theorem 18.4. *(*Sauer-Shelah-Vapnik-Chervonenkis*) Assume that the VC dimension of $\mathscr{A}$ is equal to d. Then the following upper bound holds*

$$\mathscr{S}_{\mathscr{A}}(n) \leq \sum_{i=0}^{d} \binom{n}{i}.$$

Proof. We use the approach based on the *shifting* technique. Fix any set of points $x_1, \dots, x_n$ in $\mathscr{X}$. Set $V = \{(\mathbf{1}_{x_1 \in A}, \dots, \mathbf{1}_{x_n \in A}) : A \in \mathscr{A}\}$ and observe that $V \subseteq \{0,1\}^n$. For $i = 1, \dots, n$ consider the shifting operator $S_{i,V}$ acting on $(v_1, \dots, v_n) \in V$ as follows:

$$S_{i,V}((v_1, \dots, v_n)) = \begin{cases} (v_1, \dots, v_{i-1}, 0, v_{i+1}, \dots, v_n), & \text{if } (v_1, \dots, v_{i-1}, 0, v_{i+1}, \dots, v_n) \notin V; \\ (v_1, \dots, v_n), & \text{otherwise.} \end{cases}$$

That is, $S_{i,V}$ changes the i-th coordinate 1 by 0 if this does not create two copies of the same vector in V. Otherwise, no changes are made. Define $S_i(V) = \{S_{i,V}(v) : v \in V\}$. This means that we apply the shifting operator to all vectors in V. By our construction we have $|S_i(V)| = V$. More importantly, any set $I \subset \{1,\dots,n\}$ shattered by $S_i(V)$ is also shattered by V. To prove this we take any set J shattered by $S_i(V)$. If $i \notin J$, then the claim follows immediately since the shifting operator does not affect this index. Otherwise, without loss of generality assume that $j = 1$ and $I = \{1,\dots,k\}$. Since I is shattered by $S_1(V)$, for any $u \in \{0,1\}^k$ there is $v \in S_1(V)$ such that $v_i = u_i$ for $i = 1,\dots,k$. If $u_1 = 1$, then both v and $v' = (0, v_1, \dots, v_n)$ belong to V since otherwise v would have been shifted. Thus, for any $u \in \{0,1\}^k$ there is $w \in V$ such that $w_i = u_i$ for $i = 1,\dots,k$. This means that I is also shattered by V.

Starting from the set V, we apply shifting repeatedly to all columns i until no shifts are possible. That is, we reach the set V' such that $S_i(V') = V'$ for all $i = 1,\dots,n$. This happens because whenever a nontrivial shift happens, the total number of 1-s in V decreases. Finally, we prove that V' contains no vector with more than d 1-s. Indeed, if there is a vector $v \in V'$ with $k > d$ 1-s, then V' contains a shattered set supported on these k coordinates. It is easy to see that otherwise shifting would have reduced the number of 1-s in v. This implies that the same subset of size $k > d$ is also shattered by V. We obtain a contradiction with the fact that the VC dimension of $\mathscr{A}$ is equal to d. So, we have

$$|V'| \le \sum_{i=0}^{d} \binom{n}{i}.$$

The claim follows since $|V| = |V'|$. □

We may now present a key corollary of this lemma.

Theorem 18.5. *Consider a family of events $\mathscr{A}$ with the VC dimension d. If $n \ge d$, then for any $t \ge \sqrt{\frac{2}{n}}$, it holds that*

$$\Pr\left(\sup_{A\in\mathscr{A}} |\nu_n(A) - P(A)| \ge t\right) \le 8\,(en/d)^d \exp(-nt^2/32).$$

In particular, with probability at least $1-\delta$, we have

$$\sup_{A\in\mathscr{A}} |\nu_n(A) - P(A)| \le 4\sqrt{\frac{2}{n}\left(d\log\left(\frac{8en}{d}\right) + \log\frac{1}{\delta}\right)}.$$

Proof. The proof uses the uniform convergence theorem together with Theorem 18.4. We use an elementary identity we used before

$$\sum_{i=0}^{d} \binom{n}{i} \le \left(\frac{en}{d}\right)^d,$$

whenever $n \geq d$. Further, we have

$$\log(8\mathscr{S}_{\mathscr{A}}(n)) \leq \log(8\,(en/d)^d) \leq d\log(8en/d).$$

The claim follows. □

We are now extending Example 18.3.

Proposition 18.6. *The VC dimension of the family of events induced by non-homogeneous half-spaces in $\mathbb{R}^p$ is equal to $p+1$.*

The proof is based on the following result.

Theorem 18.7. *(*Radon's theorem*) Assume that we are given a set $x_1,\ldots,x_{p+2}$ of vectors in $\mathbb{R}^p$. There is a way to partition this set into two sets whose convex hulls intersect.*

Proof. For a set of real valued variables $a_1,\ldots,a_{p+2}$ consider the following system of equations with respect to these variables

$$\sum_{i=1}^{p+2} a_i x_i = 0 \quad \text{and} \quad \sum_{i=1}^{p+2} a_i = 0.$$

Observe that it is the system of $p+1$ equations with $p+2$ variables having a trivial zero solution. Thus, we know that there is at least one non-zero solution of this system of equations. We denote it by $b_1,\ldots,b_{p+2}$. Since $\sum_{i=1}^{p+2} b_i = 0$, there should be at least one positive and one negative value among $b_1,\ldots,b_{p+2}$. In particular, both sets $I_+ = \{i \in \{1,\ldots,p+2\} : b_i \geq 0\}$ and $I_- = \{i \in \{1,\ldots,p+2\} : b_i < 0\}$ are non-empty and $\sum_{i\in I_+} b_i = -\sum_{i\in I_-} b_i$. We denote $b = \sum_{i\in I_+} b_i$. We may write

$$\sum_{i\in I_+} \frac{b_i}{b} x_i = \sum_{i\in I_-} \frac{-b_i}{b} x_i.$$

Thus, the point $\sum_{i\in I_+} \frac{b_i}{b} x_i$ is a convex combination of points in $\{x_i : i \in I_+\}$ and in $\{x_i : i \in I_-\}$. Therefore, it belongs to both convex hulls. The claim follows. □

Proof. (of Proposition 18.6) First, we show that the VC dimension of this family is at least $p+1$. The idea is to show that any simplex can be shattered by halfpspaces. To do so, we consider the set of vectors $\{0, e_1,\ldots,e_p\}$, where e_i is the i-th basis vector. For any $(v_0, v_1,\ldots,v_p) \in \{0,1\}^{p+1}$ define the d dimensional vector $w = (2v_1 - 1,\ldots,2v_d - 1)$ and consider the half-space

$$\langle x, w\rangle + v_0 - \frac{1}{2} \geq 0.$$

By plugging any vector in $\{0, e_1,\ldots,e_p\}$ into this inequality, we show that the corresponding vector belongs to the half-space if and only if $v_i = 1$. Indeed, $\langle e_i, w\rangle + v_0 - \frac{1}{2} = 2v_i - 3/2 + v_0$

and the sign is only defined by the value of v_i. Similarly, $\langle 0, w\rangle + v_0 - \frac{1}{2} = v_0 - 1/2$ and the sign is defined by the value of v_0.

Finally, Radon's theorem implies that a set of $d+2$ points cannot be shattered. Indeed, since the the convex hulls of two distinct subset intersect, at least one of the binary vectors in the definition of the VC dimension will not be realized. The claim follows. □

One may think that the VC dimension is closely related to the number of parameters. However, there is a classical example of a family of events parametrized by a single parameter such that its VC dimension is infinite.

Example 18.8. *Consider the family of events $\mathscr{A}$ consisting of all sets $\{x \in \mathbb{R} : \sin(xt) \geq 0\}$ for all $t > 0$. One may easily verify that a set of any size can be shattered by this family of sets. Therefore, its VC dimension is infinite.*

For a set of binary valued classifiers $\mathscr{F}$ one may define the growth function and the VC dimension completely analogously. Indeed, the role of the events $\mathscr{A}$ are played by the sets where the corresponding classifier predicts with 1. For example, the class of classifier induced by the halfspaces in $\mathbb{R}^p$ has the VC dimension $d = p+1$.

Theorem 18.9. *Any class $\mathscr{F}$ with the finite VC dimension d is PAC learnable by any algorithm choosing a consistent classifier in $\mathscr{F}$.*

Proof. For a classifier f, let $R_n(f)$ define its empirical risk. That is, we set

$$R_n(f) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}_{f(X_i)\neq f^*(X_i)}.$$

Let $\widehat{f} \in \mathscr{F}$ be any classifier consistent with f^* on the sample S_n. We have $R_n(\widehat{f}) = 0$. Therefore,

$$R(\widehat{f}) = R(\widehat{f}) - R_n(\widehat{f}) \leq \sup_{f\in\mathscr{F}} (R(f) - R_n(f)).$$

Observe that $\sup_{f\in\mathscr{F}} (R(f) - R_n(f))$ can be controlled by the uniform convergence theorem or more precisely by Theorem 18.5. Consider the set of events $\mathscr{A}$ induced by $\{x \in \mathscr{X} : f(x) \neq f^*(x)\}$ for all $f \in \mathscr{F}$. One can easily verify that the VC dimension of this set of events is the same as the VC dimension of $\mathscr{F}$ (corresponding events are of the form $\{x \in \mathscr{X} : f(x) = 1\}$ for all $f \in \mathscr{F}$). Therefore, by Theorem 18.5 it holds that, with probability at least $1-\delta$,

$$R(\widehat{f}) \leq 4\sqrt{\frac{2}{n}\left(d\log\left(\frac{8en}{d}\right) + \log\frac{1}{\delta}\right)}.$$

Hence if the sample size $n(\varepsilon,\delta)$ is such that $4\sqrt{\frac{2}{n}\left(d\log\left(\frac{8en}{d}\right) + \log\frac{1}{\delta}\right)} \leq \varepsilon$, then $R(\widehat{f}) \leq \varepsilon$. □

Example 18.10. *The classes $\mathscr{F}$ of halfspaces in $\mathbb{R}^p$, intervals and rays in $\mathbb{R}$ are PAC learnable.*

So far we observed that either the existence of a finite sample compression scheme or the finiteness of the VC dimension imply the PAC-learnability. However, it is not immediately clear if the VC dimension is related to sample compression. The following question is one of the oldest conjectures in learning theory.

Exploratory Challenge 18.1. *Prove that there is a universal constant $c > 0$ such that for any family of classifiers $\mathscr{F}$ with the VC dimension d there is a sample compression scheme of size at most cd.*

Further Reading 18.2. *The Sauer-Shelah-Vapnik-Chervonenkis lemma appears independently and in different contexts in* [VC71, Sau72, She72]. *Further relations between PAC learning and the VC dimensions were made in* [BEHW89]. *Radon's Theorem appears in* [Rad21]. *The sample compression conjecture is due to Warmuth* [War03].

19. CLASSIFICATION WITH NOISE (20.05.2021)

safs Up to date, we only discussed the classification setup where the label of X_i is defined by some unknown classifier in a deterministic manner. However, the label generating mechanism can have some noise. This happens, for example, when the true label is flipped with constant probability before being revealed to us. Thus, we assume that there is an unknown joint distribution of the objects and the labels (X,Y). As before, X is a random vector in $\mathbb{R}^p$. In the noise-free case we assume that there is a classifier f^* such that $Y = f^*(X)$ almost surely. In the worst regime X and Y are completely independent: the noise of the problem is so high, so that there is no reasonable way to predict label Y given X. Of course, in practice people encounter an intermediate regime where some strong correlation between X and Y is present.

The risk of a classifier $f : \mathscr{X} \to \{0,1\}$ is now given by

$$R(f) = \Pr(f(X) \neq Y),$$

where the probability is taken with respect to the joint distribution of X and Y. A natural question is in finding a classifier that minimizes the risk R among all measurable functions. This classifier is called the *Bayes optimal* classifier.

Theorem 19.1. *Given a joint distribution (X,Y) the Bayes optimal classifier f_B^* defined by*

$$f_B^*(x) = \begin{cases} 1 & \text{if } \Pr(Y=1|X=x) \geq 1/2; \\ 0 & \text{otherwise.} \end{cases}$$

minimizes the risk R. That is, for any (measurable) classifier $f : \mathscr{X} \to \{0,1\}$ it holds that

$$R(f_B^*) \leq R(f).$$

Proof. Fix any x in the support of the distribution of X and any measurable classifier f. Denote $\eta(x) = \Pr(Y=1|X=x)$. We have

$$\begin{aligned} \Pr(f(x) \neq Y|X=x) &= 1 - \Pr(f(x) = Y|X=x) \\ &= 1 - \Pr(Y=1, f(X)=1|X=x) - \Pr(Y=0, f(X)=0|X=x) \\ &= 1 - \mathbf{1}_{f(x)=1} \Pr(Y=1|X=x) - \mathbf{1}_{f(x)=0} \Pr(Y=0,|X=x) \\ &= 1 - \mathbf{1}_{f(x)=1}\eta(x) - \mathbf{1}_{f(x)=0}(1-\eta(x)). \end{aligned}$$

Since the same equality works for f_B^*, we have

$$\begin{aligned}
&\Pr(f(x) \neq Y | X = x) - \Pr(f_B^*(x) \neq Y | X = x) \\
&= \eta(x)(\mathbf{1}_{f_B^*(x)=1} - \mathbf{1}_{f(x)=1}) + (1 - \eta(x))(\mathbf{1}_{f_B^*(x)=0} - \mathbf{1}_{f(x)=0}) \\
&= (2\eta(x) - 1)(\mathbf{1}_{f_B^*(x)=1} - \mathbf{1}_{f(x)=1}) \\
&\geq 0,
\end{aligned}$$

where the last line follows from the definition of f_B^*. Indeed, whenever $\eta(x) \geq 1/2$ we have $f_B^*(x) = 1$, and otherwise $f_B^*(x) = 0$. We conclude by taking the expectation with respect to X:

$$R(f) - R(f_B^*) = \mathbb{E}_X(\Pr(f(x) \neq Y | X) - \Pr(f_B^*(x) \neq Y | X)) \geq 0.$$

□

The proof reveals an even stronger property of the Bayes optimal classifier: in some sense it is the best way to predict Y given any realization $X = x$. The issue is that the Bayes optimal rule depends on the distribution, while in the typical classification setup we only observe

$$(X_1, Y_1), \ldots, (X_n, Y_n).$$

Direct estimation of the distribution $Y|X$ is usually complicated, especially when the dimension of X is high. Instead, we are hoping to construct an estimator $\widehat{f}$ such that

$$R(\widehat{f}) - R(f_B^*)$$

is small with high probability. At appears that even this problem is quite difficult. Indeed, the standard method of classification is based on the assumption that one is choosing a classifier $\widehat{f} \in \mathscr{F}$ consistent with the observations. Thus, given the class $\mathscr{F}$, we may write

$$R(\widehat{f}) - R(f_B^*) = \underbrace{R(\widehat{f}) - \inf_{f \in \mathscr{F}} R(f)}_{\text{estimation error}} + \underbrace{\inf_{f \in \mathscr{F}} R(f) - R(f_B^*)}_{\text{approximation error}}.$$

The following decomposition shows the so-called *estimation-approximation* tradeoff. The larger the class $\mathscr{F}$ is the smaller the approximation error is and vice versa. Indeed, from the upper bounds we have that in the noise-free case we may guess that the estimation error becomes larger as the size of class increases. In the limiting case if $f_B^* \in \mathscr{F}$, then $\inf_{f \in \mathscr{F}} R(f) - R(f_B^*) = 0$ and the approximation error is equal to zero. And the other way around, if $|\mathscr{F}| = 1$, then $R(\widehat{f}) - \inf_{f \in \mathscr{F}} R(f) = 0$ for the learning algorithm that always outputs a classifier in $\mathscr{F}$. In practice, one should choose a class $\mathscr{F}$ such that both the estimation and the approximation errors are optimally balanced to minimize $R(\widehat{f}) - R(f_B^*)$.

The estimation error allows us to define the notion of agnostic PAC learnability.

Definition 19.2. *A (possibly infinite) class $\mathscr{F}$ of classifiers is* agnostic PAC-learnable *with the* sample complexity $n(\delta,\varepsilon)$ *if there is a mapping $A : \cup_{m=0}^{\infty}(\mathscr{X}\times\{0,1\})^m \to \{0,1\}^{\mathscr{X}}$ (called the* learning algorithm*; given a sample S of any size it outputs a classifier $A(S)$) that satisfies the following property: for* every *distribution P on $\mathscr{X}\times\{0,1\}$,* every *$\delta,\varepsilon\in(0,1)$, if the sample size n is greater or equal than $n(\delta,\varepsilon)$, then*

$$\Pr_{(X_1,\dots,X_n)}\left(R(A(S_n))-\inf_{f\in\mathscr{F}}R(f)\le\varepsilon\right)\ge 1-\delta.$$

Of course, in the presence of the noise one cannot guarantee that there is a classifier in $\mathscr{F}$ consistent with the learning sample. Therefore, we introduce the Empirical Risk Minimization (ERM) strategy. Given the sample S_n, define

$$\widehat{f}_{\mathrm{ERM}}=\operatorname{argmin}_{f\in\mathscr{F}}\sum_{i=1}^{n}\mathbf{1}_{f(X_i)\neq Y_i}.$$

Thus, $\widehat{f}_{\mathrm{ERM}}$ minimizes the number of misclassifications on the learning sample S_n. If there are multiple minimizers, we always choose any of them. Of course, ERM generalizes the notion of a consistent classifier. The following result holds.

Theorem 19.3. *Any class $\mathscr{F}$ with the finite VC dimension d is agnostic PAC learnable by any ERM in $\mathscr{F}$. Moreover, the following risk bound holds: with probability at least $1-\delta$,*

$$R(\widehat{f}_{ERM})-\inf_{f\in\mathscr{F}}R(f)\le 8\sqrt{\frac{2}{n}\left(d\log\left(\frac{8en}{d}\right)+\log\frac{1}{\delta}\right)}.$$

Proof. The proof is similar to the proof Theorem 18.9 and we borrow some notation from there. Assume without loss of generality that $\inf_{f\in\mathscr{F}}R(f)$ is achieved at some $f^*\in\mathscr{F}$. We have

$$R_n(\widehat{f}_{\mathrm{ERM}})\le R_n(f^*).$$

Further, we can expand

$$\begin{aligned}R(\widehat{f}_{\mathrm{ERM}})-R(f^*)&=R(\widehat{f}_{\mathrm{ERM}})-R_n(\widehat{f}_{\mathrm{ERM}})+R_n(f^*)-R(f^*)+R_n(\widehat{f}_{\mathrm{ERM}})-R_n(f^*)\\&\le R(\widehat{f}_{\mathrm{ERM}})-R_n(\widehat{f}_{\mathrm{ERM}})+R_n(f^*)-R(f^*)\\&\le 2\sup_{f\in\mathscr{F}}(R(f)-R_n(f)),\end{aligned}$$

where in the last inequality we used that $f^*,\widehat{f}_{\mathrm{ERM}}\in\mathscr{F}$. From now on we used the argument of Theorem 18.9 (check that the VC dimension of the corresponding family of events is also equal

to d) to show that, with probability at least $1-\delta$,

$$R(\widehat{f}_{\mathrm{ERM}}) - R(f^*) \le 8\sqrt{\frac{2}{n}\left(d\log\left(\frac{8en}{d}\right) + \log\frac{1}{\delta}\right)}.$$

This implies that our class is agnostic PAC learnable. □

One may observe that the error bound in the agnostic case scales as $\sqrt{\frac{\log n}{n}}$ if only the dependence on n is considered, whereas in the noise-free classification the rates of convergence are faster. For example, the expected error in the case of axis-aligned rectangles is bounded by $\frac{4}{n+1}$. For large enough n the latter approaches zero much faster. An interesting point is that both bounds are essentially not improvable and agnostic learning is indeed more complicated than the noise free case: it requires more observations to achieve the same estimation error.

20. Online Learning: Follow the Leader and Halving algorithms (20.05.2021)

In this part of the course we discuss another standard setting of statistical learning called the online learning. The Perceptron algorithm studied above is a particular instance of an online algorithm. In particular, at each round we observe a vector x_i and then make a prediction of the label y_i. These observations happen at rounds $t = 1,\dots,T$. Similarly to the PAC classification model, we consider the case where there is an unknown classifier f^* such that $y_i = f^*(x_i)$ for $i = 1,\dots,T$ and f^* belongs to some known set of classifiers $\mathscr{F}$. This case will be called the *realizable* case.

Follow the Leader Algorithm.

- Input: a finite class $\mathscr{F}$.
- Set $\mathscr{F}_1 = \mathscr{F}$.
- For $t = 1,\dots,T$ do
  (1) Observe x_t.
  (2) Choose any $f \in \mathscr{F}_i$.
  (3) Predict $\widehat{y}_t = f(x_t)$.
  (4) Observe the true label $y_t = f^*(x_t)$.
  (5) Update $\mathscr{F}_{i+1} = \{f \in \mathscr{F}_i : f(x_i) = y_i\}$.

The name Follow the Leader is motivated by the fact that at round t we predict according to one of the classifiers having the best performance on the currently observed points.

Proposition 20.1. *In the realizable case, for any number of rounds T and for any sequence of the data points (which can be chosen adversarially), the Follow the Leader algorithm makes at most $|\mathscr{F}|-1$ mistakes.*

Proof. If our classifier makes a mistake, then at least one classifier in the current set $\mathscr{F}_t$ is removed when we update it . Because we are in the realizable case we cannot remove all classifiers as at least one classifier is always consistent with our observations. Thus, the number of mistakes is bounded by $|\mathscr{F}|-1$. □

It appears that following the leader is not the best strategy in this setup. A much better result can be achieved by the following prediction strategy.

Halving algorithm

- Input: a finite class $\mathscr{F}$.
- Set $\mathscr{F}_1 = \mathscr{F}$.
- For $t = 1, \dots, T$ do
  (1) Observe x_t.
  (2) Predict $\widehat{y}_t = \operatorname{argmax}_{y\in\{0,1\}} |\{f \in \mathscr{F}_t : f(x_t) = y\}|$.
  This is exactly the majority vote and the ties are broken arbitrary.
  (3) Observe the true label $y_t = f^*(x_t)$.
  (4) Update $\mathscr{F}_{t+1} = \{f \in \mathscr{F}_t : f(x_t) = y_t\}$.

Proposition 20.2. *In the realizable case, for any number of rounds T and for any sequence of the data points, the Halving algorithm makes at most $\log_2(|\mathscr{F}|)$ mistakes.*

Proof. If the classifier makes a mistake, then at least half of the classifiers are removed from the current set. We cannot remove all classifiers as at least one classifier is always consistent with our observations. Thus, if M is the total number of mistakes we have the following inequality for the number of classifiers remaining after making M mistakes

$$1 \leq |\mathscr{F}| 2^{-M}.$$

The claim follows. □

Further Reading 20.1. *The notion of agnostic PAC learnability was essentially introduced in* [VC74]. *An extension of the online classification setup to the case of infinite classes was done by Littlestone* [Lit88]. *See also the textbook* [SSBD14].

21. EXPONENTIAL WEIGHTS ALGORITHM (27.05.2021)

We considered the online learning setup with zero labelling noise. A natural question is to extend these results to the case where this restrictive assumption is not made. Some nontrivial results are possible in this challenging setup if we introduce the notion of *regret*. It can be seen as an online analog of the excess risk. Let T denote the number of rounds and our prediction at round t is denoted by $\widehat{y}_t$. As before, at each round we observe a vector x_t and then make a prediction of the label y_t, and this happens at rounds $t = 1, \dots, T$. We define the *regret* by

$$R_T = \max_{f \in \mathscr{F}} \sum_{t=1}^{T} (\mathbf{1}_{\widehat{y}_t \neq y_t} - \mathbf{1}_{\widehat{y}_t \neq f(x_t)}) = \sum_{t=1}^{T} \mathbf{1}_{\widehat{y}_t \neq y_t} - \min_{f \in \mathscr{F}} \sum_{t=1}^{T} \mathbf{1}_{\widehat{y}_t \neq f(x_t)},$$

and we want it to be as small as possible. By making the regret small, we try to predict as well as the best classifier in the class (as if the best possible function in $\mathscr{F}$ is known in advance). The regret is trivially bounded by the number of rounds T. However, the interesting regime is when the dependence of the regret on T is *sub-linear* for any number of rounds and for any (adversarially chosen) sequence of the data $(x_1, y_1), \dots, (x_T, y_T)$. Unfortunately, at this level of generality the problem is too complicated as shown by the following simple example.

Example 21.1 (Cover's impossibility result)**.** *For $\mathscr{F}$ consisting of two functions $f_1 = 0$ and $f_2 = 1$ for any prediction strategy there is a sequence $(x_1, y_1), ..., (x_T, y_T)$ such $R_T \geq T/2$. That is, the sub-linear regret is impossible.*

The construction is the following: the adversarially chosen sequence takes our strategy into account and our predictions are always wrong. That is, $\sum_{t=1}^{T} \mathbf{1}_{\widehat{y}_t \neq y_t} = T$. In this case, at least one of the classifiers is wrong on at most half of the data. Therefore, $\min_{f \in \mathscr{F}} \sum_{t=1}^{T} \mathbf{1}_{\widehat{y}_t \neq f(x_t)} \leq T/2$. This proves that the regret is at least $T/2$.

Fortunately, if we give a little bit of extra power to our algorithm, this problem can be avoided. In particular, we want to allow our algorithm to output a label that belongs to the interval $[0, 1]$. That is, $\widehat{y}_t \in [0, 1]$ and our *loss function* is

$$|\widehat{y}_t - y_t|.$$

When both values are binary, we have $|\widehat{y}_t - y_t| = \mathbf{1}_{\widehat{y}_t \neq y_t}$ Any value $\widehat{y}_t \in [0, 1]$ can be interpreted as the randomized prediction where we predict 1 with probability $\widehat{y}_t$. In this case, $|\widehat{y}_t - y_t|$ is exactly equal to an expected binary loss when predicting y_t with a random label $\widehat{y}_t$. As an example, assume that $y_t = 0$. If we output $\widehat{y}_t = 0.6$, then we interpret this value as predicting 1 with probability 0.6 and 0 with probability 0.4. The expected binary loss of such a prediction is

$0.6 \times 1 + 0.4 \times 0 = |\widehat{y}_t - y_t|$. We will frequently use this probabilistic interpretation of $[0,1]$-valued predictors. Our modified notion of regret is

$$R_T = \max_{f \in \mathscr{F}} \sum_{t=1}^{T} (|\widehat{y}_t - y_t| - |y_t - f(x_t)|).$$

For this notion of regret we want to have a small regret with respect to any sequence of the data. Our key algorithm is called the *Exponential-Weights* (or *Weighted Majority*) algorithm and is defined as follows.

Exponential-Weights (Weighted Majority)

- Input: a finite class $\mathscr{F} = \{f_1, \ldots, f_N\}$; the number of rounds T; the rate $\eta > 0$.
- Set the weights of each classifier to be equal to 1. That is, $w^1 = (1, \ldots, 1)$ and $w^1 \in \mathbb{R}^N$.
- For $t = 1, \ldots, T$ do
  (1) Observe x_t.
  (2) Predict $\widehat{y}_t = \frac{\sum_{i=1}^N w_i^t f_i(x_t)}{\sum_{i=1}^N w_i^t}$.
  (3) Observe the true label y_t.
  (4) For $i = 1, \ldots, N$ do
  (5) Update $w_i^{t+1} = w_i^t \exp(-\eta |y_t - f_i(x_t)|)$.

The intuition behind this algorithm is quite simple. We always predict with a linear combination of classifiers in $\mathscr{F}$. After observing a correct label, we update our weights so that each classifier that was wrong has its weight decreased. The following key result shows that the regret of this algorithm is sub-linear and scales as $\sqrt{T}$. This is much smaller than the naive upper bound T.

Theorem 21.2. *If $\eta = \sqrt{8 \log(|\mathscr{F}|/T)}$, then for any sequence of the data the regret*

$$R_T = \max_{f \in \mathscr{F}} \sum_{t=1}^{T} (|\widehat{y}_t - y_t| - |y_t - f(x_t)|).$$

of the Exponential-Weights *algorithm satisfies for* any *sequence of the data*

$$R_T \leq \sqrt{\log(|\mathscr{F}|) T/2}.$$

Remark 21.3. *In this lecture we are focusing on $y_t \in \{0,1\}$ and $\{0,1\}$-valued classifiers. However, the proof goes through without changes even if $y_t \in [0,1]$ and the functions are $[0,1]$-valued.*

Proof. Define $\Phi_t = \log \sum_{i=1}^N w_i^t$. Using our update rule, we have

$$\Phi_{t+1} - \Phi_t = \log \left(\frac{\sum_{i=1}^N w_i^t \exp(-\eta |y_t - f_i(x_t)|)}{\sum_{i=1}^N w_i^t} \right).$$

For a pair (x_t, y_t) we may look at $-|y_t - f_i(x_t)|$ as a random variable (denote it by Z), where the randomness comes when each i is chosen with probability $w_i^t / (\sum_{i=1}^N w_i^t)$. In particular, $\mathbb{E}_i$ means

the expectation with respect to the random choice of the index i. Thus, we may write

$$\begin{aligned}
\log\left(\frac{\sum_{i=1}^N w_i^t \exp(-\eta|y_t - f_i(x_t)|)}{\sum_{i=1}^N w_i^t}\right) &= \log(\mathbb{E}_i \exp(\eta Z)) \\
&= \log(\exp(\eta \mathbb{E}_i Z)\mathbb{E}_i \exp(\eta(Z - \mathbb{E}_i Z))) \\
&\leq -\eta \mathbb{E}_i|y_t - f_i(x_t)| + \frac{\eta^2}{8} \quad \text{(Hoeffding's Lemma)} \\
&\leq -\eta|y_t - \mathbb{E}_i f_i(x_t)| + \frac{\eta^2}{8} \quad \text{(Jensen's inequality)} \\
&= -\eta|y_t - \widehat{y}_t| + \frac{\eta^2}{8}.
\end{aligned}$$

Therefore,

$$\Phi_{T+1} - \Phi_1 = \sum_{t=1}^T (\Phi_{t+1} - \Phi_t) \leq -\sum_{t=1}^T \eta|y_t - \widehat{y}_t| + \frac{\eta^2 T}{8}.$$

On the other hand, since $w^1 = (1, \ldots, 1)$ and, we have (using our update rule)

$$\begin{aligned}
\Phi_{T+1} - \Phi_1 &= \log\left(\sum_{i=1}^N \exp\left(-\eta \sum_{t=1}^T |y_t - f_i(x_t)|\right)\right) - \log(|\mathscr{F}|) \\
&\geq \log\left(\max_i \exp\left(-\eta \sum_{t=1}^T |y_t - f_i(x_t)|\right)\right) - \log(|\mathscr{F}|) \\
&= -\min_i\left(\eta \sum_{t=1}^T |y_t - f_i(x_t)|\right) - \log(|\mathscr{F}|).
\end{aligned}$$

Comparing the expressions for $\Phi_{T+1} - \Phi_1$, we have

$$-\min_i\left(\eta \sum_{t=1}^T |y_t - f_i(x_t)|\right) - \log(|\mathscr{F}|) \leq -\sum_{t=1}^T \eta|y_t - \widehat{y}_t| + \frac{\eta^2 T}{8}.$$

Rearranging the terms, dividing by $\eta > 0$ and choosing $\eta = \sqrt{8\log(|\mathscr{F}|)/T}$, we conclude the proof. □

A natural question is if Exponential Weights perform as good as the halving algorithm in the case where there is no classification noise. Let us assume that there is $f^* \in \mathscr{F}$ such that $y_t = f^*(x_t)$ for $t = 1, \ldots, T$. The following results shows that up to multiplicative constants factors the more general Exponential Weights algorithm has the same performance as the halving algorithm.

Proposition 21.4. *Assume that for some $f^* \in \mathscr{F}$ we have $y_t = f^*(x_t)$ and consider the case where $y_t \in \{0,1\}$ and $f_i(x_t) \in \{0,1\}$ for all i,t. If $\eta = 1$, then for any sequence of the data the regret of*

the Exponential-Weights *algorithm satisfies*

$$R_T = \sum_{t=1}^{T} |\widehat{y}_t - y_t| \leq 2\log(|\mathscr{F}|).$$

Proof. First, we modify several steps in the proof of Theorem 21.2. Our aim is to show the following bound

$$\log\left(\frac{\sum_{i=1}^N w_i^t \exp(-\eta|y_t - f_i(x_t)|)}{\sum_{i=1}^N w_i^t}\right) \leq -\frac{1}{2}\eta|y_t - \widehat{y}_t|. \tag{21}$$

The key difference is that we added a multiplicative factor $\frac{1}{2}$ but as a result removed the term $\frac{\eta^2}{8}$. Consider the case where $y_t = 0$. In this case, denoting $x = \sum_{i:f_i(x_t)=1} w_i^t / (\sum_{i=1}^N w_i^t)$ we have

$$\frac{\sum_{i=1}^N w_i^t \exp(-\eta|y_t - f_i(x_t)|)}{\sum_{i=1}^N w_i^t} = (1-x) + x\exp(-\eta).$$

Observe that $x \in [0,1]$ and by the definition of our prediction strategy we have $\widehat{y}_t = x$. Thus, to verify (21) we want to choose η such that for all $x \in [0,1]$,

$$(1-x) + x\exp(-\eta) \leq \exp(-\eta x/2).$$

Let us prove that $\eta = 1$ is sufficient. The inequality is satisfied when $x = 0$. Taking the derivative one observes that $x \mapsto \exp(-\eta x/2) - (1-x) - x\exp(-\eta)$ is non-decreasing whenever

$$1 \geq \exp(-\eta) + \frac{\eta}{2}.$$

This relation is clearly satisfied when $\eta = 1$. Consider now the case where $y_t = 1$. In this case

$$\frac{\sum_{i=1}^N w_i^t \exp(-\eta|y_t - f_i(x_t)|)}{\sum_{i=1}^N w_i^t} = (1-x) + x\exp(-\eta) = x + (1-x)\exp(-\eta).$$

In this case we need to check that

$$x + (1-x)\exp(-\eta) \leq \exp(-\eta(1-x)/2)$$

The same value of η implies this relation, since we can always reparametrize x by $1-x$. Repeating the lines of the proof of Theorem 21.2, we use $\min_i \left(\sum_{t=1}^T |y_t - f_i(x_t)|\right) = 0$ and write

$$\Phi_{T+1} - \Phi_1 \geq -\log(|\mathscr{F}|).$$

Altogether, we have

$$-\log(|\mathscr{F}|) \leq -\frac{1}{2}\sum_{t=1}^{T} |\widehat{y}_t - y_t|.$$

The claim follows. □

The problem of the exponential weights algorithm in the general case is that the optimal choice $\eta = \sqrt{8\log(|\mathscr{F}|)/T}$ depends on the time horizon T. This can be eliminated by the following approach called *the doubling trick*. For $k = 0, 1, \ldots$ we divide the time periods in a sequence of nonintersecting intervals $I_k = \{2^k, 2^k + 1, \ldots, 2^{k+1} - 1\}$ each consisting of 2^k rounds. For each period we run the algorithm with $T = \sqrt{8\log(|\mathscr{F}|)/2^k}$. The total number of splits is $n = \log_2(T+1)$ (we assume without loss of generality that n is integer). We sum all regrets bounds and get

$$\sum_{t=1}^{T} |\widehat{y}_t - y_t| \leq \sum_{k=0}^{n} \min_{f \in \mathscr{F}} \sum_{t \in I_k} |f(x_t) - y_t| + \sum_{k=0}^{n} \sqrt{\log(|\mathscr{F}|) 2^{k-1}}.$$

Now, we use

$$\sum_{k=0}^{n} \min_{f \in \mathscr{F}} \sum_{t \in I_k} |f(x_t) - y_t| \leq \min_{f \in \mathscr{F}} \sum_{t=1}^{T} |f(x_t) - y_t|.$$

This implies

$$R_T \leq \sum_{k=0}^{n} \sqrt{\log(|\mathscr{F}|) 2^{k-1}}.$$

Finally, we have

$$\sum_{k=0}^{n} 2^{(k-1)/2} = \frac{1}{\sqrt{2}} \frac{2^{(n+1)/2} - 1}{\sqrt{2} - 1} = \frac{1}{\sqrt{2}} \frac{\sqrt{2(T+1)} - 1}{\sqrt{2} - 1}.$$

This implies that our regret bound is

$$R_T \leq \sqrt{\log(|\mathscr{F}|)} \left(\frac{1}{\sqrt{2}} \frac{\sqrt{2(T+1)} - 1}{\sqrt{2} - 1} \right).$$

The difference with the previous results is only in slightly different constant factors.

One of the surprising properties of online algorithms is their sensitivity to the curvature of the loss. For example, imagine that we are predicting a value in $[0, 1]$ but this time our loss function is quadratic. That is, instead of the binary loss $\mathbf{1}_{\widehat{y}_t \neq y_t}$ or the absolute loss $|\widehat{y}_t - y_t|$, we use the quadratic loss $(\widehat{y}_t - y_t)^2$. For the rest of the lecture we also allow $y_t \in [0, 1]$ and $f_i(x_t) \in [0, 1]$. In this case we consider

$$R_T = \max_{f \in \mathscr{F}} \sum_{t=1}^{T} ((\widehat{y}_t - y_t)^2 - (y_t - f(x_t))^2).$$

Exponential-Weights (Squared loss)

- Input: a finite class $\mathscr{F} = \{f_1, \ldots, f_N\}$; the number of rounds T; the rate $\eta > 0$.
- Set the weights of each classifier to be equal to 1. That is, $w^1 = (1, \ldots, 1)$.
- For $t = 1, \ldots, T$ do
  (1) Observe x_t.
  (2) Predict $\widehat{y}_t = \frac{\sum_{i=1}^{N} w_i^t f_i(x_t)}{\sum_{i=1}^{N} w_i^t}$.

(3) Observe the true label y_t.
(4) For $i = 1, \dots, N$ do
(5) Update $w_i^{t+1} = w_i^t \exp(-\eta (y_t - f_i(x_t))^2)$.

Proposition 21.5. *Consider the sequential prediction problem with the squared loss, where $y_t \in [0,1]$ and $f_i(x_t) \in [0,1]$ for all t, i. Fix $\eta = 1/2$, then for any sequence of the data the regret of the* Exponential-Weights *algorithm satisfies*

$$R_T \leq 2 \log(|\mathscr{F}|).$$

Proof. We need minor modifications on the proof of Theorem 21.2. We want to prove a different kind of bound on

$$\log \left(\frac{\sum_{i=1}^N w_i^t \exp(-\eta (y_t - f_i(x_t))^2)}{\sum_{i=1}^N w_i^t} \right)$$

We show the following:

$$\frac{\sum_{i=1}^N w_i^t \exp(-\eta (y_t - f_i(x_t))^2)}{\sum_{i=1}^N w_i^t} \leq \exp(-\eta (\widehat{y}_t - y_t)^2).$$

To do so, we check that for any $y \in [0,1]$ the function $x \mapsto \exp(-\eta (y-x)^2)$ is concave for some specifically chosen $\eta \geq 0$. By the properties of the concave functions (the weights $w_i^t / \sum_{i=1}^N w_i^t$ are non-negative and sum up to 1) this will immediately imply the desired inequality. Proving the concavity is standard: one takes the second derivative of the function $x \mapsto \exp(-\eta (y-x)^2)$. We have the following lines

$$\begin{aligned} \frac{\partial^2}{\partial x^2} \exp(-\eta (y-x)^2) &= \frac{\partial}{\partial x} (-2\eta (x-y) \exp(-\eta (y-x)^2)) \\ &= (4\eta^2 (x-y)^2 - 2\eta) \exp(-\eta (y-x)^2). \end{aligned}$$

Observe that $4\eta^2 (x-y)^2 - 2\eta \leq 0$ whenever $2\eta (x-y)^2 \leq 1$. The function $x \mapsto \exp(-\eta (y-x)^2)$ is concave for all y if we choose $\eta = 1/2$. Thus, repeating the lines of the proof of Theorem 21.2 and choosing $\eta = 1/2$, we have

$$-\min_i \left(\sum_{t=1}^T (y_t - f_i(x_t))^2 \right) /2 - \log(|\mathscr{F}|) \leq -\sum_{t=1}^T (y_t - \widehat{y}_t)^2 /2.$$

The claim follows.

Further Reading 21.1. *Cover's counterexample appears in* [Cov65a]. *The Exponential-Weights algorithm is due to Littlestone and Warmuth* [LW94] *and Vovk* [Vov90]. *A detailed presentation and generalizations of the above results can be found in the textbook* [CBL06].

□

22. Introduction to (Stochastic) Gradient Descent (03.06.2021)

In this lecture, we discuss the Gradient Descent method. It is one of the main techniques standing behind modern applications of machine learning. We already have some experience with a variant of this method. Recall the Perceptron algorithm and our setup of classification in $\mathbb{R}^p$. For $w \in \mathbb{R}^p$ there is a natural way to define a classifier: we consider the mapping $x \mapsto \text{sign}(\langle w, x\rangle)$. That is, the vector w induces a half-space and is orthogonal to the separating hyperplane. In particular, w misclassifies $x \in \mathbb{R}^p$ whenever $y\langle w, x\rangle \leq 0$.

Perceptron Algorithm.

- Input: $s_n = \{(x_1, y_1), \ldots, (x_n, y_n)\}$.
- Set $w_1 = 0$.
- For $i = 1, \ldots, n$ do
  (1) If $y_i\langle w_i, x_i\rangle \leq 0$
  (2) $\quad w_{i+1} = w_i + y_i x_i$,
  (3) Else
  (4) $\quad w_{i+1} = w_i$,
- Return: w_{n+1}.

Speaking informally, if the price for each misclassification is $-y_i\langle w_i, x_i\rangle$ (it is positive if x_i is misclassified), then our update rule changes w_i in the direction opposite to the direction of the gradient ∇_w of $-y_i\langle w_i, x_i\rangle$. This motivates a general analysis of similar update rules.

Let $f : \mathscr{W} \to \mathbb{R}$ be a *differentiable convex* function, where $\mathscr{W}$ is a convex closed subset of $\mathbb{R}^p$ containing 0. Consider the update rule for $t = 1, \ldots, T$,

$$w_{t+1} = w_t - \eta \nabla f(w_t), \tag{22}$$

where $\eta > 0$ is a fixed parameter. We set $w_1 = 0$. This update rule makes a step in the direction opposite to the direction of the greatest rate of increase of f. Assume that w^* minimizes f over $\mathscr{W}$ and define the averaged vector

$$\overline{w} = \sum_{t=1}^{T} w_i.$$

where each w_i is obtained by our update rule (22). The following result holds for Gradient Descent (GD).

Theorem 22.1. *For the function f as above, if for some $L > 0$, it holds that $\sup_{w \in \mathcal{W}} \|\nabla f(w)\| \leq L$ and if $\sup_{w \in \mathcal{W}} \|w\| \leq B$, then for $\eta = \frac{B}{L\sqrt{T}}$,*

$$f(\overline{w}) - f(w^*) \leq \frac{BL}{\sqrt{T}}.$$

Proof. Recall that one of the ways to define the convexity of a differentiable function f is to say that for any x, y in the domain

$$f(y) - f(x) \geq \langle \nabla f(x), y - x \rangle. \tag{23}$$

We have

$$\begin{aligned}
f(\overline{w}) - f(w^*) &= f\left(\frac{1}{T}\sum_{t=1}^{T} w_t\right) - f(w^*) \\
&\leq \frac{1}{T}\sum_{t=1}^{T} \left(f(w_t) - f(w^*)\right) && \text{(by convexity)} \\
&\leq \frac{1}{T}\sum_{t=1}^{T} \langle \nabla f(w_t), w_t - w^* \rangle && \text{(by (23))}
\end{aligned}$$

In what follows, we analyze

$$\sum_{t=1}^{T} \langle v_t, w_t - w^* \rangle$$

for general vectors $v_1, \ldots, v_T$ having $\|v_t\| \leq L$ and the update rule $w_{t+1} = w_t - \eta v_t$. The following lines hold for any $\eta > 0$,

$$\begin{aligned}
\sum_{t=1}^{T} \langle v_t, w_t - w^* \rangle &= \frac{1}{\eta}\sum_{t=1}^{T} \langle \eta v_t, w_t - w^* \rangle \\
&= \frac{1}{2\eta}\sum_{t=1}^{T} \left(\eta^2 \|v_t\|^2 + \|w_t - w^*\|^2 - \|w_t - w^* - \eta v_t\|^2\right) \\
&= \frac{\eta}{2}\sum_{t=1}^{T} \|v_t\|^2 + \frac{1}{2\eta}\sum_{t=1}^{T} \left(\|w_t - w^*\|^2 - \|w_t - w^* - \eta v_t\|^2\right) \\
&= \frac{\eta}{2}\sum_{t=1}^{T} \|v_t\|^2 + \frac{1}{2\eta}\sum_{t=1}^{T} \left(\|w_t - w^*\|^2 - \|w_{t+1} - w^*\|^2\right) \\
&= \frac{\eta}{2}\sum_{t=1}^{T} \|v_t\|^2 + \frac{1}{2\eta}\left(\|w_1 - w^*\|^2 - \|w_{T+1} - w^*\|^2\right) \\
&\leq \frac{\eta}{2}\sum_{t=1}^{T} \|v_t\|^2 + \frac{1}{2\eta}\|w_1 - w^*\|^2
\end{aligned}$$

$$\leq \frac{\eta}{2}\sum_{t=1}^{T}\|v_t\|^2 + \frac{1}{2\eta}\|w^*\|^2.$$

The claim immediately follows if we divide the last expression by T and fix η as in the statement. □

Remark 22.2. *For the sake of presentation we focus on differentiable functions. The general case can be analyzed using the notion of subgradient. In this case, gradient descent is replaced by the subgradient descent algorithm.*

The analysis of gradient descent is very useful in many application. To observe this, we usually have to go beyond the binary risk function.

Example 22.3 (Linear regression and GD)**.** *Consider the squared loss as in the previous lecture. We may define the risk (and its empirical counterpart) of the real-valued $x \mapsto \langle x, w\rangle$,*

$$R(w) = \mathbb{E}(Y - \langle X, w\rangle)^2 \quad \textit{and} \quad R_n(w) = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \langle X_i, w\rangle)^2.$$

Here we also assumed that Y is real-valued and X is a random vector in $\mathbf{R}^p$. We can naturally assume that we observe the gradients of the function $f(\cdot) = R_n(\cdot)$. Indeed,

$$\nabla R_n(w) = \frac{2}{n}\sum_{i=1}^{n} X_i(\langle X_i, w\rangle - Y_i)$$

depends only on a given vector w and the observed sample. Assume that our distribution is such that $\|\nabla R_n(w)\| \leq L$ for $w \in \mathscr{W}$. Then, if we can prove (similar to the uniform convergence result we showed for the families of events)

$$R(w) \approx R_n(w), \quad \textit{for all} \quad w \in \mathscr{W}_1,$$

where $\mathscr{W}_1$ is guaranteed to always contain $\overline{w} = \frac{1}{T}\sum_{t=1}^{T} w_t$, then we can show that the solution obtained by gradient descent method implies that $R(\overline{w})$ is small.

The key problem is that in statistical applications it is hard to assume that we can observe a gradient $\nabla R(w)$ of the true risk. Instead, gradient descent can be helpful if f stands for the empirical risk. Then, gradient descent tells us that after t iterations we are *almost* minimizing the empirical error.

As we mentioned, in this case, the analysis of gradient descent should usually be combined with uniform convergence. An alternative that avoids any uniform convergence results is Stochastic Gradient Descent. It works as follows: instead of observing $\nabla f(w_t)$ at each round, we observe a random vector

$$v_t, \quad \text{such that} \quad \mathbb{E}(v_t|w_t) = \nabla f(w_t) \quad \text{almost surely.}$$

Our randomized updated rule is

$$w_{t+1} = w_t - \eta v_t.$$

We remark that we used the notion of conditional expectation since each w_t is random (based on previous radomized updates). Our assumption is that given w_t, the random vector v_t is a good estimate of the true gradient $\nabla f(w_t)$. We show that an analog of Theorem 22.1 holds in-expectation for Stochastic Gradient Descent (SGD).

Theorem 22.4. *For f as above, if for some $L > 0$, it holds that $\|v_t\| \leq L$ almost surely and if $\sup_{w \in \mathscr{W}} \|w\| \leq B$, then for $\eta = \frac{B}{L\sqrt{T}}$,*

$$\mathbb{E}(f(\overline{w})) - f(w^*) \leq \frac{BL}{\sqrt{T}},$$

where the expectation is taken with respect to the randomness of stochastic gradients.

Proof. A fully formal proof requires an introduction of martingale sequences. We instead make it slightly informal and work directly with conditional probability. First, using the derivations of the proof of Theorem 22.1 and the linearity of expectation, we have

$$\mathbb{E}\left(f(\overline{w}) - f(w^*)\right) \leq \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}\left\langle \nabla f(w_t), w_t - w^* \right\rangle.$$

We want to show that for $t = 1, \ldots, T$,

$$\mathbb{E}\langle \nabla f(w_t), w_t - w^* \rangle \leq \mathbb{E}\langle v_t, w_t - w^* \rangle.$$

By the properties of conditional expectation and since $\mathbb{E}(v_t | w_t) = \nabla f(w_t)$ we have

$$\begin{aligned}
\mathbb{E}\langle v_t, w_t - w^* \rangle &= \mathbb{E}(\mathbb{E}(\langle v_t, w_t - w^* \rangle | w_t)) \quad \text{(the law of total expectation)} \\
&= \mathbb{E}(\langle \mathbb{E}(v_t | w_t), w_t - w^* \rangle) \quad \text{(since } \mathbb{E}(w_t | w_t) = w_t\text{)} \\
&= \mathbb{E}(\langle \nabla f(w_t), w_t - w^* \rangle)
\end{aligned}$$

Therefore,

$$\frac{1}{T}\sum_{t=1}^{T} \mathbb{E}\left\langle \nabla f(w_t), w_t - w^* \right\rangle \leq \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}\left\langle v_t, w_t - w^* \right\rangle = \sum_{t=1}^{T} \mathbb{E}\left\langle v_t, w_t - w^* \right\rangle.$$

From now on, we reproduce the proof of Theorem 22.1 for the term $\sum_{t=1}^{T} \langle v_t, w_t - w^* \rangle$, as it works for any vectors v_t such that $\|v_t\| \leq L$. □

Stochastic Gradient Descent is useful in the following setup.

Example 22.5 (Linear regression and SGD)**.** *Assume that we have a sample $(X_1, Y_1), \ldots, (X_n, Y_n)$ of independent observations. Then, the gradient of the loss is observed at each point: that is, we*

see

$$2X_i(\langle X_i, w_t\rangle - Y_i),$$

where w_t is constructed using an i.i.d. sample $(X_1,Y_1),\dots,(X_n,Y_n)$. Each w_t is constructed using $(X_1,Y_1),\dots,(X_{t-1},Y_{t-1})$. So that whenever the derivative and the expectation are interchangeable (a mild assumption) we have

$$\mathbb{E}(2X_i(\langle X_i, w_t\rangle - Y_i)|w_t) = \nabla\mathbb{E}(Y - \langle X, w\rangle)^2 = \nabla R(w).$$

In this case, we can write the bound of the form

$$\mathbb{E}R(\overline{w}) - R(w^*) \leq \frac{BL}{\sqrt{n}}.$$

Observe that Stochastic Gradient Descent can be preferable from the computational perspective. Indeed, for some large samples it is better to compute the gradient of the loss at point (X_t,Y_t) rather than computing the gradient of the entire empirical risk $R_n(\cdot)$.

Further Reading 22.1. *Gradient descent is studied in optimization where various the relations between properties of the functions as well as are studied. See the monograph* [Bub14] *for related results. Standard convergence results on Stochastic Gradient Descent can be found in the textbook* [SSBD14] *as well as in* [Ora19].

References

[ALMT14] D. Amelunxen, M. Lotz, M. B. McCoy, and J. A. Tropp. Living on the edge: phase transitions in convex programs with random data. *Information and Inference, available online*, 2014.

[Alo86] N. Alon. Eigenvalues and expanders. *Combinatorica*, 6:83–96, 1986.

[AM85] N. Alon and V. Milman. Isoperimetric inequalities for graphs, and superconcentrators. *Journal of Combinatorial Theory*, 38:73–88, 1985.

[Ban16] Afonso S. Bandeira. Ten lectures and forty-two open problems in the mathematics of data science. *Available online at:* `http://www.cims.nyu.edu/~bandeira/TenLecturesFortyTwoProblems.pdf`, 2016.

[BEHW89] Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K Warmuth. Learnability and the vapnik-chervonenkis dimension. *Journal of the ACM (JACM)*, 36(4):929–965, 1989.

[BHMZ20] Olivier Bousquet, Steve Hanneke, Shay Moran, and Nikita Zhivotovskiy. Proper learning, helly number, and an optimal svm bound. In *Conference on Learning Theory*, pages 582–609. PMLR, 2020.

[BSS] A. S. Bandeira, A. Singer, and T. Strohmer. Mathematics of data science. *Book draft available at* `https://people.math.ethz.ch/~abandeira/BandeiraSingerStrohmer-MDS-draft.pdf`.

[Bub14] Sébastien Bubeck. Convex optimization: Algorithms and complexity. *arXiv preprint arXiv:1405.4980*, 2014.

[CBL06] Nicolo Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge university press, 2006.

[Che70] J. Cheeger. A lower bound for the smallest eigenvalue of the Laplacian. *Problems in analysis (Papers dedicated to Salomon Bochner, 1969), pp. 195–199. Princeton Univ. Press*, 1970.

[Chr16] Ole Christensen. *An Introduction to Frames and Riesz Bases*. Birkhäuser, 2016.

[Chu10] F. Chung. Four proofs for the cheeger inequality and graph partition algorithms. *Fourth International Congress of Chinese Mathematicians, pp. 331–349*, 2010.

[Cov65a] T. Cover. Behavior of sequential predictors of binary sequences. 1965.

[Cov65b] Thomas M Cover. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE transactions on electronic computers*, (3):326–334, 1965.

[CR09] E.J. Candès and B. Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9(6):717–772, 2009.

[CRPW12] V. Chandrasekaran, B. Recht, P.A. Parrilo, and A.S. Willsky. The convex geometry of linear inverse problems. *Foundations of Computational Mathematics*, 12(6):805–849, 2012.

[CT10] E. J. Candes and T. Tao. The power of convex relaxation: Near-optimal matrix completion. *Information Theory, IEEE Transactions on*, 56(5):2053–2080, May 2010.

[FM] Matthew Fickus and Dustin G. Mixon. Tables of the existence of equiangular tight frames. *available at arXiv:1504.00253 [math.FA]*.

[FR13] S. Foucart and H. Rauhut. *A Mathematical Introduction to Compressive Sensing*. Birkhauser, 2013.

[Gro11] D. Gross. Recovering low-rank matrices from few coefficients in any basis. *IEEE Transactions on Information Theory*, 57(3):1548–1566, 2011.

[GZ84] Evarist Giné and Joel Zinn. Some limit theorems for empirical processes. *The Annals of Probability*, pages 929–989, 1984.

[H+15] T. Hastie et al. Matrix completion and low-rank SVD via fast alternating least squares. *Journal of Machine Learning Research*, 2015.

[HK21] Steve Hanneke and Aryeh Kontorovich. Stable sample compression schemes: New applications and an optimal svm margin bound. In *Algorithmic Learning Theory*, pages 697–721. PMLR, 2021.

[Hoe63] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

[HR21] Moritz Hardt and Benjamin Recht. Patterns, predictions, and actions: A story about machine learning. *arXiv preprint arXiv:2102.05242*, 2021.

[Kol29] A Kolmogoroff. Über das gesetz des iterierten logarithmus. *Mathematische Annalen*, 101(1):126–135, 1929.

[Lit88] Nick Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine learning*, 2(4):285–318, 1988.

[Llo82] S. Lloyd. Least squares quantization in PCM. *IEEE Trans. Inf. Theor.*, 28(2):129–137, 1982.

[LW86] Nick Littlestone and Manfred Warmuth. Relating data compression and learnability. 1986.

[LW94] Nick Littlestone and Manfred K Warmuth. The weighted majority algorithm. *Information and computation*, 108(2):212–261, 1994.

[Mix] D. G. Mixon. Short, Fat matrices BLOG.

[Mix12] Dustin G. Mixon. Sparse signal processing with frame theory. *PhD Thesis, Princeton University, also available at arXiv:1204.5958[math.FA]*, 2012.

[MRT18] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018.

[Nov63] Albert B Novikoff. On convergence proofs for perceptrons. Technical report, STANFORD RESEARCH INST MENLO PARK CA, 1963.

[Ora19] Francesco Orabona. A modern introduction to online learning. *arXiv preprint arXiv:1912.13213*, 2019.

[Rad21] Johann Radon. Mengen konvexer körper, die einen gemeinsamen punkt enthalten. *Mathematische Annalen*, 83(1):113–115, 1921.

[Rec11] B. Recht. A simpler approach to matrix completion. *Journal of Machine Learning Research*, 12:3413–3430, 2011.

[Ros57] Frank Rosenblatt. *The perceptron, a perceiving and recognizing automaton Project Para*. Cornell Aeronautical Laboratory, 1957.

[Sau72] Norbert Sauer. On the density of families of sets. *Journal of Combinatorial Theory, Series A*, 13(1):145–147, 1972.

[She72] Saharon Shelah. A combinatorial problem; stability and order for models and theories in infinitary languages. *Pacific Journal of Mathematics*, 41(1):247–261, 1972.

[SSBD14] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[Val84] Leslie G Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

[VB04] L. Vanderberghe and S. Boyd. *Convex Optimization*. Cambridge University Press, 2004.

[VC64a] V Vapnik and A Ya Chervonenkis. A class of algorithms for pattern recognition learning. *Avtomat. i Telemekh (in Russian), Available online*, 25(6):937–945, 1964.

[VC64b] Vladimir Vapnik and Alexey Chervonenkis. A note one class of perceptrons. *Automation and remote control (in Russian), available online*, 1964.

[VC71] Vladimir Naumovich Vapnik and Aleksei Yakovlevich Chervonenkis. On uniform convergence of the frequencies of events to their probabilities. *Teoriya Veroyatnostei i ee Primeneniya*, 16(2):264–279, 1971.

[VC74] V. Vapnik and A. Chervonenkis. *Theory of Pattern Recognition [in Russian]*. Nauka, Moscow, 1974. (German Translation: W. Wapnik & A. Tscherwonenkis, *Theorie der Zeichenerkennung*, Akademie–Verlag, Berlin, 1979).

[Ver18] Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.

[Vid13] Mathukumalli Vidyasagar. *Learning and generalisation: with applications to neural networks*. Springer Science & Business Media, 2013.

[Vov90] Volodimir G Vovk. Aggregating strategies. *Proc. of Computational Learning Theory, 1990*, 1990.

[VPG15] Vladimir Vovk, Harris Papadopoulos, and Alexander Gammerman. *Measures of Complexity*. Springer, 2015.

[War03] Manfred K Warmuth. Compressing to vc dimension many points. In *COLT*, volume 3, pages 743–744. Springer, 2003.