Data Analysis for Physics [PHYS-231]

Lecture Notes

Prof. Lenka Zdeborová
2023 TAs: Dr. Vittorio Erba, Giovanni Piccioli, Freya Behrens, Cedric Koller, Odilon Duranthon

First edition updated in 2024

Institute of Physics – IPHYS
École polytechnique fédérale de Lausanne – EPFL

Contents

Why this lecture?

Data are everywhere in modern physics—whether from particle detectors, astronomical observations, simulations, or imaging devices. This course is about learning how to make sense of them. We build a toolbox that starts with linear algebra and numerical methods, then expands to probability and statistics, and finally reaches stochastic models that describe randomness in nature.

Part I (Weeks 1–5): Introduction to data analysis. We begin with the power method and Google's PageRank as an example of how eigenvalues appear in real-world problems. We then study principal component analysis (PCA) for uncovering structure in high-dimensional data, with motivating examples such as genetic variation across Europe. Next, we turn to linear regression and the least-squares method, exploring prediction, error, overfitting, and regularization. Finally, we introduce gradient descent and see how it applies not only to regression but also to logistic regression and simple neural networks. By the end of Part I, students have a computational toolkit for fitting, compressing, and interpreting data.

Part II (Weeks 6–10): Quantifying uncertainty. Physics is never free of noise, so we next study uncertainty and statistics. We learn how to generate random variables, propagate errors, and understand the law of large numbers. This leads to the central limit theorem, maximum likelihood estimation, and hypothesis testing. We revisit regression from a probabilistic point of view, introducing weighted least squares and Bayesian regularization. The emphasis is always on applications: how physicists quantify error bars, assess reliability, and make inferences from data.

Part III (Weeks 11–14): Stochasticity in physical systems. The final part turns to randomness as a fundamental feature of nature. We study Brownian motion, diffusion, and random walks, and connect them to stochastic differential equations. We then develop Monte Carlo methods for sampling probability distributions, including Markov chain Monte Carlo and the Metropolis algorithm, with examples from physics and data analysis. Finally, we explore phase transitions—percolation, random graphs, and sharp thresholds—setting basis for how ideas from data analysis are connected to modelling in physics.

By following this progression, students see how linear algebra, probability, and statistics connect into a coherent framework for modern data analysis. The course emphasizes both conceptual understanding and practical application, preparing students for advanced physics courses, laboratory work, and independent research.

PART I

Introduction to Data Analysis

We build a computational toolkit for extracting structure from data: power method and PageRank, principal component analysis (PCA), least squares/regression (estimation vs. prediction, overfitting, regularization), and gradient descent up to logistic regression and simple neural networks.

1 Lecture 1: The power method and Google's PageRank algorithm

In this lecture we will present an algorithm to compute the dominant eigenvalue and the related eigenvector of a square matrix. We will then consider Google's PageRank algorithm [1] as a very impactful application of the method.

1.1 Eigenvalues and eigenvectors

We start by recalling the definition of eigenvalues and eigenvectors of a matrix.

Definition 1. *Consider a square matrix $A \in \mathbb{C}^{n\times n}$. A vector $\vec{v}$ is an **eigenvector** of the matrix A if $\vec{v} \neq 0$ and if there exists $\lambda \in \mathbb{C}$ such that*

$$A\vec{v} = \lambda\vec{v}\,. \tag{1.1}$$

*λ is an **eigenvalue** of A associated to $\vec{v}$. Finally, the **characteristic polynomial** of A is defined as*

$$p(\lambda) = \det\left(A - \lambda\mathbb{I}_{n\times n}\right)\,, \tag{1.2}$$

where $\mathbb{I}_{n\times n}$ is the $n \times n$ identity matrix.

We recall that the eigenvalues of an $n \times n$ matrix are exactly the roots of its characteristic polynomial, which is a degree n polynomial. Thus, each square matrix of size $n \times n$ has exactly n eigenvalues on the complex field.

Notice that if $\vec{v}$ is an eigenvector with eigenvalue λ, then $c\vec{v}$ is again an eigenvector with eigenvalue λ for any $c \in \mathbb{C} \setminus \{0\}$. Thus, we can always normalize each eigenvector such that their Euclidean norm equals one, i.e.

$$\|\vec{v}\| = \sqrt{v_1^2 + \cdots + v_n^2} = 1\,, \tag{1.3}$$

where v_i is the i-th component of the vector $\vec{v}$. In the following we will often consider normalized eigenvectors.

In this lecture, we will focus on two questions.

1. How can we compute the eigenvalues and eigenvectors of a matrix *numerically*, i.e. using a computer?
2. Why is it interesting to compute the eigenvalues and eigenvectors of a matrix?

We will provide one answer of interest to the 2nd question, there are of course, many more applications in physics and elsewhere.

1.2 Power Method

We first consider Question 1, and introduce an algorithm to compute the dominant eigenvalue and eigenvector of a matrix. By dominant eigenvalue we mean the eigenvalue with largest modulus.

Theorem 1. *Consider a square matrix $A \in \mathbb{C}^{n\times n}$ with eigenvalues $\{\lambda_1, \ldots, \lambda_n\}$ and associated normalized eigenvectors $\{\vec{v}_1, \ldots, \vec{v}_n\}$. Define the iteration*

$$\vec{w}^{(t+1)} = \frac{A\vec{w}^{(t)}}{\left\|A\vec{w}^{(t)}\right\|} \tag{1.4}$$

with a given initial condition $\vec{w}^{(0)} \in \mathbb{C}^n$. Suppose that

1. *A has a well defined dominant eigenvalue (without loss of generality, the first), i.e.*

$$|\lambda_1| > \max\left(|\lambda_2|, \dots, |\lambda_n|\right) \tag{1.5}$$

where the complex modulus is defined as

$$|a + ib| = \sqrt{a^2 + b^2}\,. \tag{1.6}$$

In particular, this implies that $\lambda_1 \neq 0$.

2. *The initial condition $\vec{w}^{(0)}$ has non-zero projection on the dominant eigenvector v_1, i.e.*

$$\vec{w}^{(0)} \cdot \vec{v}_1 \neq 0\,. \tag{1.7}$$

Then, as $t \to +\infty$, we have that $\vec{w}^{(t)}$ converges to a dominant eigenvector of A, and

$$\left\|A\vec{w}^{(t)}\right\| \xrightarrow{t\to+\infty} |\lambda_1|\,. \tag{1.8}$$

Theorem 1 gives us a way to numerically compute the dominant eigenvalue and eigenvector of A: just iterate (1.4) for a sufficiently long amount of steps. The resulting algorithm is called **power method**, or method of iterated powers.

We will illustrate how to prove Theorem 1 under more restrictive hypotheses. We will consider only *diagonalizable matrices* A, i.e. A can be written as

$$A = V\Lambda V^{-1}\,, \tag{1.9}$$

where V is the $n \times n$ matrix whose i-th column is the i-th eigenvector $\vec{v}_i$, and Λ is the diagonal matrix

$$\Lambda = \mathrm{diag}\left(\lambda_1, \lambda_2, \dots, \lambda_n\right)\,. \tag{1.10}$$

If this is the case, recall that the eigenvectors form a basis of $\mathbb{C}^n$. (The more general case of a non-diagonalizable matrix, which we will not consider here, can be proven by using Jordan's normal form.)

A side-remark: Notice that $\vec{v}_1$ is not uniquely defined even after normalization, as $\mu\vec{v}_1$ is again an eigenvector of A with eigenvalue λ_1 for any $\mu \in \mathbb{C}$, and $|\mu| = 1$. Thus, while the theorem guarantees us that $\vec{w}^{(t)}$ converges to a dominant eigenvector of A, the actual sequence of vectors $\{\vec{w}^{(t)}\}_{t=0}^{\infty}$ may not converge component-wise, as it may enter some loop where $\vec{w}^{(t)} = \mu^{(t)}\vec{v}_1$ for some sequence of phases $\mu^{(t)}$. In that case, while each $\vec{w}^{(t)}$ is an eigenvector, the sequence does not converge.

Proof of Theorem 1. We start by remarking that at each time $t \geq 1$, the iterates $\vec{w}^{(t)}$ are normalized, i.e.

$$\left\|\vec{w}^{(t)}\right\| = 1\,. \tag{1.11}$$

Indeed, using the definition (1.4), we have

$$\left\|\vec{w}^{(t)}\right\| = \left\|\frac{A\vec{w}^{(t-1)}}{\left\|A\vec{w}^{(t-1)}\right\|}\right\| = \frac{1}{\left\|A\vec{w}^{(t-1)}\right\|}\left\|A\vec{w}^{(t-1)}\right\| = 1 \tag{1.12}$$

where we used the linearity of the norm.

Now consider the initial condition $\vec{w}^{(0)}$, and decompose it in the basis of the eigenvectors of A, i.e.

$$\vec{w}^{(0)} = c_1\vec{v}_1 + c_2\vec{v}_2 + \cdots + c_n\vec{v}_n\,, \tag{1.13}$$

for some complex coefficients $\{c_1, \dots, c_n\}$. Assumption 2 implies that $c_1 \neq 0$.

Now, using the definition of the iteration (1.4), we can obtain a decomposition on the same basis for the iterate $\vec{w}^{(t)}$ at time t. We first notice that the iterate t has a simple relation with the initial condition

$$\vec{w}^{(t)} = \frac{A\vec{w}^{(t-1)}}{\left\|A\vec{w}^{(t-1)}\right\|} = \frac{A\frac{A\vec{w}^{(t-2)}}{\left\|A\vec{w}^{(t-2)}\right\|}}{\left\|A\frac{A\vec{w}^{(t-2)}}{\left\|A\vec{w}^{(t-2)}\right\|}\right\|} = \frac{\frac{AA\vec{w}^{(t-2)}}{\left\|A\vec{w}^{(t-2)}\right\|}}{\frac{\left\|AA\vec{w}^{(t-2)}\right\|}{\left\|A\vec{w}^{(t-2)}\right\|}} = \frac{A^2\vec{w}^{(t-2)}}{\left\|A^2\vec{w}^{(t-2)}\right\|} = \cdots = \frac{A^t\vec{w}^{(0)}}{\left\|A^t\vec{w}^{(0)}\right\|}\,. \tag{1.14}$$

Then, we consider the numerator in the basis of eigenvectors of A

$$\begin{aligned} A^t\vec{w}^{(0)} &= A^t\left(c_1\vec{v}_1 + c_2\vec{v}_2 + \cdots + c_n\vec{v}_n\right) \\ &= c_1A^t\vec{v}_1 + c_2A^t\vec{v}_2 + \cdots + c_nA^t\vec{v}_n \\ &= c_1\lambda_1^t\vec{v}_1 + c_2\lambda_2^t\vec{v}_2 + \cdots + c_n\lambda_n^t\vec{v}_n\,, \end{aligned} \tag{1.15}$$

where we used the definition of eigenvalues to state that, for all $i = 1, \ldots, n$,

$$A^t\vec{v}_i = A^{t-1}(A\vec{v}_i) = A^{t-1}(\lambda_i\vec{v}_i) = \lambda_iA^{t-1}\vec{v}_i = \cdots = \lambda_i^t\vec{v}_i\,. \tag{1.16}$$

Now we use Assumption 2, telling us that $c_1 \neq 0$, as well as Assumption 1, telling us that $\lambda_1 \neq 0$, to rewrite

$$A^t\vec{w}^{(0)} = c_1\lambda_1^t\vec{v}_1 + c_2\lambda_2^t\vec{v}_2 + \cdots + c_n\lambda_n^t\vec{v}_n = c_1\lambda_1^t\left(\vec{v}_1 + \frac{c_2}{c_1}\left(\frac{\lambda_2}{\lambda_1}\right)^t\vec{v}_2 + \cdots + \frac{c_n}{c_1}\left(\frac{\lambda_n}{\lambda_1}\right)^t\vec{v}_n\right)\,. \tag{1.17}$$

Finally, Assumption 1 tells us that for all $i = 2, \ldots, n$,

$$\left|\frac{\lambda_i}{\lambda_1}\right| < 1 \implies \left(\frac{\lambda_i}{\lambda_1}\right)^t \xrightarrow{t\to+\infty} 0\,. \tag{1.18}$$

This immediately implies that

$$A^t\vec{w}^{(0)} \approx c_1\lambda_1^t\vec{v}_1 \tag{1.19}$$

so that

$$\vec{w}^{(t)} = \frac{A^t\vec{w}^{(0)}}{\left\|A^t\vec{w}^{(0)}\right\|} = \frac{c_1\lambda_1^t}{|c_1\lambda_1^t|}\vec{v}_1 + \vec{r}_t\,, \tag{1.20}$$

where we used that $\vec{v}_1$ is normalized, and where $\vec{r}_t$ is the reminder term whose norm is going to zero as $t \to \infty$. We now see that for large times, when the reminder vanishes, $\vec{w}^{(t)}$ will always be a multiple of $\vec{v}$.

A note on convergence: $\vec{w}^{(t)}$ may not converge to a well-defined vector at infinite time due to the possibly time-evolving phase $(\lambda_1/|\lambda_1|)^t$. Indeed, take the example where $c_1 = 1$ and $\lambda_1 = -1$ (thus fully real, there is no need to invoke complex numbers here). In that case

$$\vec{w}^{(t)} \approx (-1)^t\vec{v}_1\,, \tag{1.21}$$

which is an oscillating sequence that never converges, even though at any time $\vec{w}^{(t)}$ is approximately an eigenvector associated to the dominant eigenvalue. The only, but in practice the most important, case in which the sequence of phases $(\lambda_1/|\lambda_1|)^t$ converges (instead of spinning around the unit circle) is when $\lambda_1/|\lambda_1| = 1$, i.e. when the dominant eigenvalue is real and strictly positive. In that case we have

$$\vec{w}^{(t)} \xrightarrow{t\to+\infty} \vec{v}_1\,. \tag{1.22}$$

On the other hand, the following convergence holds in the usual sense without restrictions on λ_1:

$$\left\|A\vec{w}^{(t)}\right\| \xrightarrow{t\to+\infty} |\lambda_1|\,. \tag{1.23}$$

□

The power method allows in principle to compute also subdominant eigenvalues and eigenvectors. Indeed, one can modify the matrix A to surgically put the dominant eigenvalue to zero, so that the second dominant eigenvalue becomes the dominant one. In the simplest case of A *orthogonally diagonalizable* for which $V^{-1} = V^T$, i.e. A can be written as

$$A = \sum_{i=1}^{n}\vec{v}_i\,\lambda_i\,\vec{v}_i^{\,T}\,, \tag{1.24}$$

the modified matrix is

$$\tilde{A} = A - \vec{v}_1\,\lambda_1\,\vec{v}_1^{\,T}\,, \tag{1.25}$$

which has the same eigenvalues and eigenvectors of A, with the only difference that the eigenvalue λ_1 of A is now equal to zero in $\tilde{A}$. Notice that we have access to this matrix, as λ_1 and $\vec{v}_1$ can be computed using the power iteration algorithm. Then, if $\tilde{A}$ satisfies the assumptions of Theorem 1, then the power iteration method can be applied to it to compute the second eigenvalue and eigenvector, and so on.

The power method is, of course, only one of many numerical methods that can be used to compute the eigenvalues and eigenvectors of a matrix. It is one of the simplest ones if not the simplest one and is particularly suited for large matrices for which multiplication by a vector can be implemented efficiently (e.g. matrices with many zero elements) and particularly for cases where only eigenvectors corresponding to one of several largest eigenvalues are needed.

1.3 The Page Rank algorithm

In the previous section we introduced a method to compute the dominant eigenvalue and eigenvector of a matrix. But why is it interesting to compute these quantities? In this section we consider the Page Rank algorithm [1], the algorithm that led to the creation of Google, and we will see that it is very closely related to the power method, applied to a particular matrix.

1.3.1 The internet and the ranking problem

We first mention that in 1998, when Page Rank was introduced, there already existed a number of search engines for the internet. Given a string of text, they would return the list of websites containing the string. What was problematic was the ranking problem, namely how to sort the websites in order of relevance or importance. Indeed, already at that time a given search result could amount to thousands of websites, making it difficult to decide what to show to the user. Page Rank provides a solution to this ranking problem that worked extremely well in practice and led to the creation of the Google company.

The core component of the internet that is used by Page Rank is the concept of link. Given two websites i and j, we say that i links to j, or that j is linked from i, if the website i contains an hypertextual link to website j. In simpler words: if I am surfing website i, and I can click somewhere in it and get redirected to website j, then i links to j. For example, the website https://people.epfl.ch/lenka.zdeborova/ contains a link to the lab website https://www.epfl.ch/labs/spoc/, which in turn has links pointing towards the most recent publications of the group. Thus, we can imagine the internet as a collection of nodes (the websites) that are linked to each other by arrows (so called *directed edges*). Such a structure is typically called a directed graph in mathematics.

Let us call n the number of websites on the internet. A compact way of encoding the linking structure, or graph structure, of the internet is through the so called *adjacency matrix* $A \in \mathbb{R}^{n\times n}$, defined as

$$A_{ij} = \begin{cases} 1 & \text{if website } j \text{ links to website } i \\ 0 & \text{otherwise} \end{cases} . \tag{1.26}$$

We also define the concept of *out-degree* of a website j as the number of other websites that are linked from j, which can be compactly expressed through the adjacency matrix as

$$d_j = \sum_{i=1}^{n} A_{ij} . \tag{1.27}$$

1.3.2 Definition of the Google matrix and motivation behind it

We are now ready to define the so-called Google matrix $G \in \mathbb{R}^{n\times n}$, which is the key object in Page Rank, as

$$G = (1-\epsilon)S + \epsilon I \tag{1.28}$$

where I is the matrix with all entries equal to $1/n$, $\epsilon \in (0,1)$ is a fixed constant (a hyperparameter with a value that is set so that the method works well in practice), and

$$S_{ij} = \begin{cases} \frac{A_{ij}}{d_j} & \text{if } d_j \geq 1 \\ \frac{1}{n} & \text{if } d_j = 0 \end{cases} . \tag{1.29}$$

An example of internet graph with $n = 5$, along with the relevant quantities, is given in Figure 1.

Before going on and defining the Page Rank algorithm, let us gather a bit of intuition about this Google matrix G. Let us imagine a dynamical process of liquid redistribution on the internet graph. Namely, define a variable $w_i^{(t)} \geq 0$ that counts the number of liters of a certain liquid located at a given node i of the graph at time t. At each time step we redistribute the liquid in the following way:

- if $d_j = 0$, namely if the node has no outgoing links, we distribute its liquid uniformly over all the nodes.
- If $d_j \geq 1$, namely if the node has at least one outgoing link, we take a fraction ϵ of the liquid of node j and distribute it uniformly over all the nodes (as in the previous case), while we distribute the remaining fraction $1-\epsilon$ uniformly over the nodes i that are linked from j.

$$\text{set of links} = \{2 \to 1, 3 \to 1, 4 \to 1, 2 \to 3, 2 \to 4, 5 \to 4, 4 \to 5\}\,, \tag{1.30}$$

$$A = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \qquad S = \begin{bmatrix} 1/5 & 1/3 & 1 & 1/2 & 0 \\ 1/5 & 0 & 0 & 0 & 0 \\ 1/5 & 1/3 & 0 & 0 & 0 \\ 1/5 & 1/3 & 0 & 0 & 1 \\ 1/5 & 0 & 0 & 1/2 & 0 \end{bmatrix}. \tag{1.31}$$

Figure 1: An example of internet with $n = 5$ websites, with link structure given above. A is the corresponding adjacency matrix, and S is the matrix defined by (1.29). The out-degrees are $d_1 = 0$, $d_2 = 3$, $d_3 = 1$, $d_4 = 2$ and $d_5 = 1$.

In mathematical terms, we can write explicitly the relationship between the liquid distribution at time t and that at time $t+1$. Calling $V(i)$ the sets of nodes j that link to i, we have

$$\begin{aligned} w_i^{(t+1)} &= \sum_{j:d_j=0} \frac{w_j^{(t)}}{n} + \epsilon \sum_{j:d_j\geq 1} \frac{w_j^{(t)}}{n} + (1-\epsilon) \sum_{j\in V(i)} \frac{w_j^{(t)}}{d_j} \\ &= \epsilon \sum_{j:d_j=0} \frac{w_j^{(t)}}{n} + (1-\epsilon) \sum_{j:d_j=0} \frac{w_j^{(t)}}{n} + \epsilon \sum_{j:d_j\geq 1} \frac{w_j^{(t)}}{n} + (1-\epsilon) \sum_{j\in V(i)} \frac{w_j^{(t)}}{d_j} \\ &= \epsilon \sum_{j=1}^{n} \frac{w_j^{(t)}}{n} + (1-\epsilon) \sum_{j:d_j=0} \frac{w_j^{(t)}}{n} + (1-\epsilon) \sum_{j:d_j\geq 1} A_{ij} \frac{w_j^{(t)}}{d_j} \\ &= \epsilon \sum_{j=1}^{n} I_{ij} w_j^{(t)} + (1-\epsilon) \sum_{j=1}^{n} S_{ij} w_j^{(t)} \\ &= \sum_{j=1}^{n} G_{ij} w_j^{(t)} \end{aligned} \tag{1.32}$$

where we used the definitions of the matrices I, S and G. We also used that

$$\sum_{j\in V(i)} \cdots = \sum_{j:d_j\geq 1} A_{ij} \ldots . \tag{1.33}$$

Thus, we see that the matrix G is a sort of transition matrix for this liquid redistribution process, telling us how much liters we move from site j to site i. Notice that for this interpretation to hold, we check that the total amount of liquid on the nodes of the graph is conserved. We also verify that if all components of $\vec{w}^{(t)}$ are non-negative (it makes no sense to have negative amount of liquid), then $\vec{w}^{(t+1)}$ will also have non-negative components. This holds, as G is a matrix with non-negative entries.

Another helpful intuition about the Google matrix G can be found by interpreting the redistribution process as the dynamics of a random internet user, a random surfer. Indeed, we can interpret the vector $\vec{w}^{(t)}$ as a probability vector, telling us how probable it is that a random internet user is visiting a certain website at time t. The redistribution rules can then be rephrased in the following way. If the random user is at website j at time t, then at time $t+1$

- if $d_j = 0$, i.e. the website j has no out-going links, she will jump to a random website with uniform probability.
- If $d_j \geq 1$, i.e. the website j has at least one out-going link, with probability ϵ she will jump to a random website with uniform probability, and with probability $1-\epsilon$ she will jump to one of the websites linked from j, again with uniform probability.

This process is again described by (1.32)

$$w_i^{(t+1)} = \sum_{j=1}^{n} G_{ij} w_j^{(t)}\,, \tag{1.34}$$

telling us that G encodes information about which websites a random user will end up visiting more often when randomly wandering on the internet. In order for this interpretation to hold, we must check that if $\vec{w}^{(t)}$ is a

valid probability vector, i.e. all its components are non-negative and its components sum to 1 (at each time, with probability 1 the user is visiting one of the n websites), then $\vec{w}^{(t+1)}$ will also be a valid probability vector. This holds, as both the matrix S and G satisfy

$$\sum_{i=1}^{n} G_{ij} = 1 \quad \text{and} \quad \sum_{i=1}^{n} S_{ij} = 1 \quad \text{for} \quad i = 1, \dots, n\,, \tag{1.35}$$

i.e. they are both *column-wise stochastic matrices*. Indeed, suppose that

$$w_j^{(t)} \geq 0 \quad \text{and} \quad \sum_{j=1}^{n} w_j^{(t)} = 1\,. \tag{1.36}$$

Then $w_i^{(t+1)} \geq 0$ as G has non-negative entries, and

$$\sum_{i=1}^{n} w_i^{(t+1)} = \sum_{i=1}^{n}\sum_{j=1}^{n} G_{ij} w_j^{(t)} = \sum_{j=1}^{n} \left(\sum_{i=1}^{n} G_{ij} \right) w_j^{(t)} = \sum_{j=1}^{n} w_j^{(t)} = 1\,. \tag{1.37}$$

Let us show that S is column-wise stochastic. The proof for the matrix G follows easily. If $d_j = 0$, then

$$\sum_{i=1}^{n} S_{ij} = \sum_{i=1}^{n} \frac{1}{n} = 1\,, \tag{1.38}$$

while if $d_j \geq 1$, then

$$\sum_{i=1}^{n} S_{ij} = \sum_{i=1}^{n} \frac{A_{ij}}{d_j} = \frac{\sum_{i=1}^{n} A_{ij}}{d_j} = 1\,, \tag{1.39}$$

where we used the relationship between of out-degree and adjacency matrix (1.27).

Thus, both interpretation (liquid redistribution and random internet surfing) point to the following intuition: a website is more important if, after a large enough amount of time steps (in physics lexicon, after reaching *equilibrium*), the website has more liters of liquid, or has more probability of being visited by a random internet user. This is the intuition over which Page Rank is built.

To conclude this section, let us comment on the role of ϵ. This parameter acts as a kind of regularization, making it easier (and in fact certain) for the process to equilibrate at large times. Equilibration is a delicate consideration, important in physics, and we will return to it later in the course. In practice, you can think of ϵ as a small fixed regularization constant, e.g. $\epsilon = 0.1 - 0.2$.

1.3.3 Putting together the Google matrix and the power method

It is now natural to ask to which liquid distribution, or to which probability distribution, the iterative process (1.32)

$$w_i^{(t+1)} = \sum_{j=1}^{n} G_{ij} w_j^{(t)} \tag{1.40}$$

converges as $t \to +\infty$. We notice that (1.32) is very similar to (1.4), i.e. to the iteration we used in the methods of iterated powers, the only difference being the lack of normalization in (1.32). Thus, we can repeat the steps of the proof of Theorem 1 to get information on the limit of $\vec{w}^{(t)}$ as $t \to +\infty$. We again obtain that

$$\vec{w}^{(t)} = G^t \vec{w}^{(0)}\,, \tag{1.41}$$

and that if the initial condition has the decomposition on the eigenvector basis of G

$$\vec{w}^{(0)} = c_1 \vec{v}_1 + c_2 \vec{v}_2 + \cdots + c_n \vec{v}_n\,, \tag{1.42}$$

with the initial condition such that $c_1 \neq 0$ and assuming that the dominant eigenvalue of G is non-degenerate (which we note here without proof it indeed generically is for $0 < \epsilon < 1$), then

$$\vec{w}^{(t)} \xrightarrow{t \to +\infty} c_1 \lambda_1^t \vec{v}_1 \tag{1.43}$$

where λ_1 and $\vec{v}_1$ are, respectively, the dominant eigenvalue and eigenvector of the matrix G. We now realize that in the power method the normalization is crucial: indeed, it simplifies away the λ_1^t term, which either vanishes in the limit (if $|\lambda_1| < 1$) or diverges (if $|\lambda_1| > 1$). This is not needed in the case of the Google matrix, as we now prove that $\lambda_1 = 1$. Suppose without loss of generality that the initial condition satisfies

$$\sum_{j=1}^{n} w_j^{(0)} = 1\,. \tag{1.44}$$

Then, in the previous section we proved that an any later time $t \geq 1$

$$\sum_{j=1}^{n} w_j^{(t)} = 1\,, \tag{1.45}$$

as the matrix G is column-wise stochastic. But now we argued that

$$\vec{w}^{(t)} \xrightarrow{t\to+\infty} c_1 \lambda_1^t \vec{v}_1\,, \tag{1.46}$$

implying that

$$1 = \sum_{j=1}^{n} w_j^{(t)} \xrightarrow{t\to+\infty} c_1 \lambda_1^t \sum_{j=1}^{n} v_1\,, \tag{1.47}$$

which is consistent if and only if $\lambda_1 = 1$.

To sum it up, we have that (here we are using that λ_1 is real and strictly positive otherwise the sequence of $\vec{w}^{(t)}$ may not converge, see the proof of Theorem 1 for related discussions)

$$\vec{w}^{(t)} \xrightarrow{t\to+\infty} c_1 \vec{v}_1\,, \tag{1.48}$$

meaning that **the dominant eigenvector of the Google matrix is proportional to the stationary distribution of the *random-user* dynamic processes**. The larger the j-th component of this dominant eigenvector is, the more liquid/probability there will be on the j-th node. The intuition above then suggests us that we should rank websites based on the magnitude of the corresponding component of the dominant eigenvector.

We are finally in the position of writing down the Page Rank algorithm:

1. initialize $w_j^{(0)} = 1/n$ for all $j = 1, \ldots, n$.
2. Repeat $\vec{w}^{(t)} = G^t \vec{w}^{(0)}$ until $\left\|\vec{w}^{(t+1)} - \vec{w}^{(t)}\right\| \leq \delta$ (here $\delta > 0$ is a small convergence threshold giving us a practical stopping criterion).
3. Return $\vec{w}^{(t+1)}$.

The basic idea of Google search then to look for all websites containing a given string, and rank them based on the magnitude of the corresponding component of $\vec{w}^{(t+1)}$. Currently, Google search uses many other strategies to rank the results, but the PageRank algorithm is still an essential part of the whole pipeline.

1.3.4 Algorithmic considerations

Is Page Rank efficiently executable on a computer? *A priori*, we would need to store the matrix G in memory, i.e. we would need to store n^2 floating point numbers. For a small Internet of 10^6 websites, this amounts to order 10^{12} numbers, which is far too large. Moreover, to actually multiply G by itself, or to invert G (operations that we would need to perform to use other algorithms to compute v_1), takes respectively an amount of fundamental operations (think of floating point additions and multiplications) of the order of, respectively, $O(n^2)$ and $O(n^3)$, which is impossibly slow in practice.

There are two aspects that allow a much faster execution and that PageRank takes advantage off:

- the internet link structure is *sparse*, meaning that each website only links to a few other websites. Thus, the adjacency matrix A (the non-trivial ingredient of the matrix G) can be stored as a link list (see Figure 1 for an example), which will have an order of $O(\overline{d}n)$ entries, where $\overline{d}$ is the average out-degree

$$\overline{d} = \frac{1}{n}\sum_{j=1}^{n} d_j\,. \tag{1.49}$$

  Notice that if $\overline{d} \ll n$, then this will be much more efficient than storing the full matrix G, without losing any amount of information.

- We never need to multiply together matrices directly while we iterate (1.32). Indeed, the iteration only involves a matrix-vector product. This operation can be performed in order $O(\bar{d}n)$ operations if one leverages the sparsity of the adjacency matrix, much faster than the $O(n^2)$ or $O(n^3)$ operations discussed above. Notice also that the convergence of the power iteration algorithm is exponentially fast, as the terms $(\lambda_2/\lambda_1)^t$ vanish exponentially fast, meaning that one needs to run order $T = 10 - 100$ iterations in practice to reach convergence.

To sum it up, by using a sparse representation of the adjacency matrix, Page Rank can be run using order $O(\bar{d}n)$ memory space and fundamental operations, allowing it to scale to the size of the full internet. Computational complexity considerations such as this one are at the center of modern computational physics and data analysis and we will be giving more examples later in the course. Often, computational considerations drive the development of current technological innovations in a more important way than mathematical ones.

2 Lecture 2: Principal component analysis

In this lecture, we provide a way to analyze structure in data by reducing its dimensionality, the principal component analysis. We exemplify the concept in a case explaining genomics variations across Europe by geographic coordinates. We introduce the singular value decomposition of a matrix.

2.1 Setting

We will consider a rectangular data matrix $A \in \mathbb{R}^{n\times d}$ given by an experiment. Rectangular matrices are common in data analysis. It is obtained by stacking n samples of dimension d. The vector $A_\mu \in \mathbb{R}^d$ for $\mu = 1 \dots n$ represents one sample or measurement. The Greek index μ should be understood as indexing the samples. We will use a Roman index $i = 1 \dots d$ for each sample's coordinates. The rows and the columns of the data matrix have different interpretations.

The example we will use to motivate the concept of principal component analysis (PCA) is given in an article titled "Genes mirror geography within Europe" from 2008 [2]. In this article around $n = 3000$ sequences of DNA were measured from individuals across Europe. For each person, each sample, a sequence of about $d = 2.10^5$ basis on which genetic variations occur relatively often was extracted. These genetic variations are encoded in a matrix $A \in \mathbb{R}^{n\times d}$, where $A_{\mu i} = 1$ if sample μ has a difference on position i with respect to a reference sequence, 0 otherwise. We expect that for people μ and ν that are genetically related, their genetic variations A_μ and A_ν will be correlated. PCA is then a method allowing visualisation of the data in a lower dimension (in the present case dimension 2). The authors of [2] use PCA to demonstrate that the structure present in the first two principal components corresponds very well to the geographic map of Europe. They compute the two principal components of A (after centering and rescaling), and they show that projections on the principal components correspond well to the coordinates of the geographic repartition of the n individuals. The colors of points in Figure 2 are obtained by querying the geographic origin of ancestors of the individuals and are not used in the PCA analysis.

We are now obviously interested in understanding what principal components are and how and why the method works mathematically.

2.2 Singular value decomposition (SVD)

We are interested in analyzing a data matrix A given by experiments that is not square. Therefore, it does not admit an eigen decomposition. Moreover, eigenvectors are nicely interpretable if they form an orthogonal basis; according to the spectral theorem, this is possible if A is normal (i.e. $AA^* = A^*A$). In our case, A is given by stacking measurements and is not normal, it is not even square. We shall rely on another decomposition, the generalization of eigen decomposition to rectangular matrices, called singular value decomposition.

Definition 2. *We recall that given a complex square matrix $A \in \mathbb{C}^{n\times d}$, we write A^* for the conjugate of its transpose, i.e. $A^* = \bar{A}^T$. We can also use the notation $A^\dagger$.*

We say a square matrix $U \in \mathbb{C}^{n\times n}$ is unitary for

$$UU^* = U^*U = I_n \ , \tag{2.1}$$

where I_n stands for the identity.

Theorem 2. *(singular value decomposition, SVD) Let $A \in \mathbb{C}^{n\times d}$; it can always be decomposed as*

$$A = U\Sigma V^* \tag{2.2}$$

$$A = \begin{pmatrix} u_1 & u_2 & \dots & u_n \end{pmatrix} \begin{pmatrix} \sigma_1 & 0 & \dots & 0 \\ 0 & \sigma_2 & & 0 \\ \vdots & & \ddots & \\ 0 & 0 & & \sigma_{\min(n,d)} \\ 0 & & \dots & 0 \\ \vdots & & & \vdots \\ 0 & & \dots & 0 \end{pmatrix} \begin{pmatrix} & v_1^\dagger & \\ & \vdots & \\ & v_d^\dagger & \end{pmatrix} \tag{2.3}$$

$$A = \sum_{\alpha=1}^{\min(n,d)} u_\alpha \sigma_\alpha v_\alpha^\dagger \ ; \tag{2.4}$$

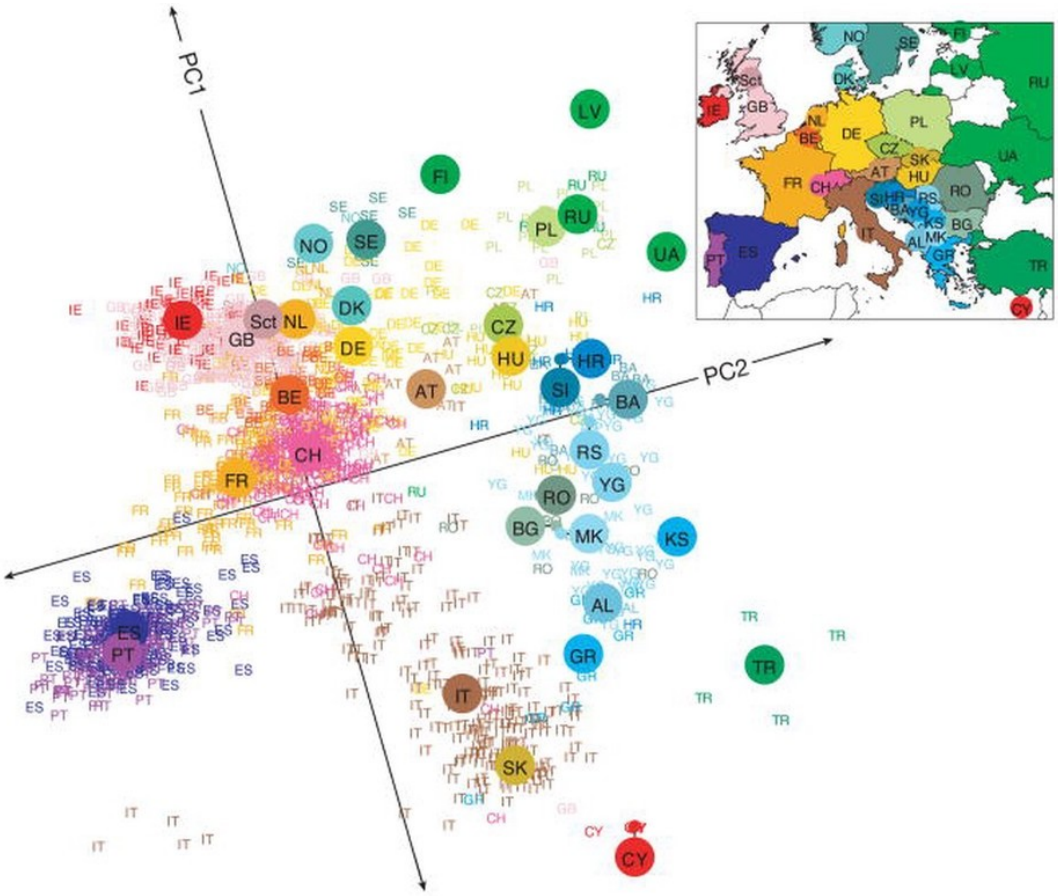

Figure 2: Visualization of each samples μ projected on the two first principal components of A. From [2].

with $U \in \mathbb{C}^{n\times n}$ and $V \in \mathbb{C}^{d\times d}$ unitary, and

$$\Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_{\min(n,d)}) \in \mathbb{R}^{n\times d} \ , \tag{2.5}$$

where $\sigma_i \geq 0$ are real and $\sigma_1 \geq \ldots \geq \sigma_{\min(n,d)}$. The σ_is are unique; they are the singular values of A. The convention is to order them in decreasing order.

We denoted by $u_\alpha \in \mathbb{C}^n$ and $v_\alpha \in \mathbb{C}^d$ the α-th column of the unitary matrix U and V. If $\sigma_1 > \ldots > \sigma_{\min(n,d)}$ then the $\min(n,d)$ first columns of U and V are unique up to multiplicative constants. By "up to multiplicative constants" we mean that if one multiplies a column u_α by a complex phase $e^{i\phi}$ and the corresponding v_α by the same then U and V are still unitary and $U\Sigma V^*$ is unchanged, as can be seen from (2.4).

Theorem 3. *(singular value decomposition, SVD, of a real-valued matrix) Let $A \in \mathbb{R}^{n\times d}$; it can always be decomposed as*

$$A = U\Sigma V^T \tag{2.6}$$

with the same properties as for the complex case, but additionally both matrices U and V having real-valued elements.

We do not provide the proof of these theorems, they can be found in standard linear algebra textbooks. SVD is a decomposition in simple elements in the sense that we write A as the sum of rank-one matrices $u_\alpha v_\alpha^\dagger$ weighted by the singular values σ_α.

Definition 3. *Let $A \in \mathbb{C}^{n\times d}$; we call left and right singular vectors of A vectors $u \in \mathbb{C}^n$ and $v \in \mathbb{C}^d$ such that there exists $\sigma \in \mathbb{C}$ such that*

$$A^* u = \sigma v \quad \text{and} \quad Av = \sigma u \ . \tag{2.7}$$

Proposition 1. *The columns of U and the columns of V are left and right singular vectors of A.*

Indeed, for v_β column of V, we have $Av_\beta = \sum_{\alpha=1}^{\min(n,d)} u_\alpha \sigma_\alpha v_\alpha^\dagger v_\beta = \sigma_\beta u_\beta$ since $v_\alpha^\dagger v_\beta = \delta_{\alpha\beta}$; and same for u_β, having $A^* u_\beta = \sum_{\alpha=1}^{\min(n,d)} v_\alpha \bar{\sigma}_\alpha u_\alpha^\dagger u_\beta = \sigma_\beta v_\beta$.

Proposition 2. *(link with eigenvalues.) Since $AA^*u = A\sigma v = \sigma^2 u$, any left singular vector u is an eigenvector of AA^*. Idem, any right singular vector v is an eigenvector of A^*A.*

This provides a way to compute the SVD of A since we know how to compute the eigenelements of a matrix. This also explains why the singular values σ_i are real, since the matrices A^*A and AA^* are Hermitian.

2.3 Low-rank approximation

We show that the terms of SVD corresponding to the largest singular values give a natural approximation of the matrix. The singular vectors of the highest singular values explain most of the structure in the matrix in the sense we will now explain. We will call the right singular vectors of the (centred and normalized) matrix corresponding to the largest singular values the principal components (PCs) of the matrix. Analyzing the PCs often provides useful information about the experiment we study. In this section, we restrict our attention to real-valued matrices, and the data matrices are real in most cases of interest.

Definition 4. *The rank r of a matrix can be defined in the same manner as:*
– its number of independent rows (or columns);
– the dimension of the space its rows (or columns) span;
– or the number of its singular values that are not null.

Theorem 4. *(Young-Eckart) Let $A \in \mathbb{R}^{n\times d}$ be a real matrix and $A = U\Sigma V^T$ its SVD, where U, Σ and V are real. We call the Frobenius norm of a matrix M the real number $||M||_F^2 = \sum_{\mu,i} M_{\mu i}^2$. Then the best approximation of A of rank k in term of $||.||_F^2$, i.e. that minimizes $||A - A^{(k)}||_F^2$, is*

$$A^{(k)} = \sum_{\alpha=1}^{k} u_\alpha \sigma_\alpha v_\alpha^T = U\Sigma_{\leq k} V^T \tag{2.8}$$

where $\Sigma_{\leq k}$ is the matrix Σ whose values σ_i are set to zero for $i > k$.

We give a sketch of the proof that brings additional insight into the consequences of this theorem:
For a real matrix B whose SVD is $B = \tilde{U}\Gamma\tilde{V}$ with $\Gamma = \mathrm{diag}(\gamma_1, \ldots, \gamma_k)$ we have

$$\begin{aligned} ||B||_F^2 &= \sum_{\mu,i} B_{\mu i} B_{\mu i} = \mathrm{tr}(BB^T) && (2.9)\\ &= \mathrm{tr}(\tilde{U}\Gamma\tilde{V}\tilde{V}^T\Gamma^T\tilde{U}^T) && (2.10)\\ &= \mathrm{tr}(\tilde{U}\Gamma\Gamma^T\tilde{U}^T) && (2.11)\\ &= \mathrm{tr}(\Gamma\Gamma^T) && (2.12)\\ &= \sum_{\alpha=1}^{k} \gamma_\alpha^2 \ . && (2.13) \end{aligned}$$

We use that U and V are unitary (orthogonal) and that Trace is invariant under the change of basis, in other words, that for square matrices A and B $\mathrm{tr}(AB) = \mathrm{tr}(BA)$.

Then

$$\begin{aligned} ||A - A^{(k)}||_F^2 &= \mathrm{tr}(U(\Sigma - \Sigma_{\leq k})V^T V(\Sigma - \Sigma_{\leq k})^T U^T) && (2.14)\\ &= \mathrm{tr}(\Sigma_{>k}\Sigma_{>k}^T) && (2.15)\\ &= \sum_{i=k+1}^{\min(n,d)} \sigma_i^2 \ . && (2.16) \end{aligned}$$

where we denoted by $\Sigma_{>k}$ the diagonal matrix of singular values where the first k terms on the diagonal are set to 0. We see that the norm of the difference between A and its approximation $A^{(k)}$ depends on the smallest singular values we discarded, that is to say, A is mainly explained by its largest singular elements. If the $\sigma_{i>k}$s are small then $A^{(k)}$ is a good approximation of A.

To prove that $A^{(k)}$ is the minimizer we use von Neumann's inequality on traces : for A and B two matrices of shape (n,d) with singular values σ_α and γ_α we have $\mathrm{tr}(AB^T) \leq \sum_\alpha^{\min(n,d)} \sigma_\alpha\gamma_\alpha$, with equality if they share the same singular elements. In particular for B of rank k we have

$$
\begin{aligned}
||A-B||_F^2 &= \mathrm{tr}((A-B)(A-B)^T) && (2.17)\\
&= \mathrm{tr}(AA^T) + \mathrm{tr}(BB^T) - 2\mathrm{tr}(AB^T) && (2.18)\\
&\geq \sum_{\alpha=1}^{\min(n,d)} \sigma_\alpha^2 + \sum_{\alpha=1}^{\min(n,d)} \gamma_\alpha^2 - 2\sum_{\alpha=1}^{\min(n,d)} \sigma_\alpha\gamma_\alpha && (2.19)\\
&= \sum_{\alpha=1}^{\min(n,d)} \sigma_\alpha^2 + \sum_{\alpha=1}^{k} \gamma_\alpha^2 - 2\sum_{\alpha=1}^{k} \sigma_\alpha\gamma_\alpha && (2.20)\\
&= \sum_{\alpha=k+1}^{\min(n,d)} \sigma_\alpha^2 + \sum_{\alpha=1}^{k} (\sigma_\alpha-\gamma_\alpha)^2 && (2.21)\\
&\geq ||A-A^{(k)}||_F^2 && (2.22)
\end{aligned}
$$

and as shown above, the equality is reached for $B = A^{(k)}$, thus $A^{(k)}$ is the low rank approximation minimizing the Frobenius norm.

2.4 Principal component analysis

We will now present how the principal component analysis (PCA) is performed on a data matrix A.

The first step in PCA is centering, and in most applications also normalizing, the data matrix as follows:

- Centered matrix:

$$\hat{A}_{\mu i} = A_{\mu i} - \frac{1}{n}\sum_{\nu=1}^{n} A_{\nu i} \ . \tag{2.23}$$

- Centered and normalized matrix:

$$\tilde{A}_{\mu i} = \frac{A_{\mu i} - \frac{1}{n}\sum_{\nu=1}^{n} A_{\nu i}}{\sqrt{\frac{1}{n}\sum_{\nu}^{n} A_{\nu i}^2 - (\frac{1}{n}\sum_{\nu}^{n} A_{\nu i})^2}} \ . \tag{2.24}$$

Without centring, the principal right singular vector is just the mean of the samples. The normalization is used in cases where the dimensions i have different magnitudes.

Definition 5. *The first k principal components of A are the first right singular vectors $v_1, \ldots, v_k \in \mathbb{R}^d$ of the centred and normalized matrix $\tilde{A}$.*

The SVD of a matrix $\tilde{A}$ is computed via the eigen decomposition of $\tilde{A}\tilde{A}^*$ or $\tilde{A}^*\tilde{A}$, as shown in proposition 2. The singular vectors of $\tilde{A}$ are eigenvectors of these. These vectors explain $\tilde{A}$ the best in the sense that they catch the most significant variations of data. We can use them to represent data with only $k \ll d$ dimensions while preserving the most possible structure in the sense of minimizing the Frobenius norm.

Example 1. *In [2] the authors keep the two first principal components v_1 and v_2 of A. They project the data points in two dimensions, using the SVD: $\tilde{A}v_1 = \sigma_1 u_1$ and $\tilde{A}v_2 = \sigma_2 u_2$; then each sample μ is represented by the coordinates $(u_{1,\mu}, u_{2,\mu})$. This is what Figure 2 depicts. It appears that, up to a rotation, these coordinates are close to the geographic coordinates the samples μ come from. The two first principal elements of A can be interpreted as geographic coordinates. Conversely, we can say that most of the genomic variations among people are correlated to localization.*

The principal components can be used in several ways. Here we mention two of them:

- Dimensionality reduction: We can determine the geographic origin of a new sample: given $A_{\text{new}} \in \mathbb{R}^d$, we compute the coordinates $(\tilde{A}_{\text{new}}^T v_1/\sigma_1, \tilde{A}_{\text{new}}^T v_2/\sigma_2)$. This can also be interpreted as the compression of A_{new} in two numbers.

- Generation of new data points: we can create new typical samples given a geographic localization (x, y) by using the following linear combination of the principal components $\tilde{A}_{\text{new}} = x\sigma_1 v_1 + y\sigma_2 v_2$. $\tilde{A}_{\text{new}}$ can then be un-centred and thresholded then to obtain binary values.

3 Lecture 3: Linear regression and least squares method

In this lecture, we introduce the concept of linear regression using the least squares method. Linear regression allows us to fit observed data, enabling us to make predictions or gain insights into the relationships between the data points.

3.1 Motivating example: estimating the speed of a train

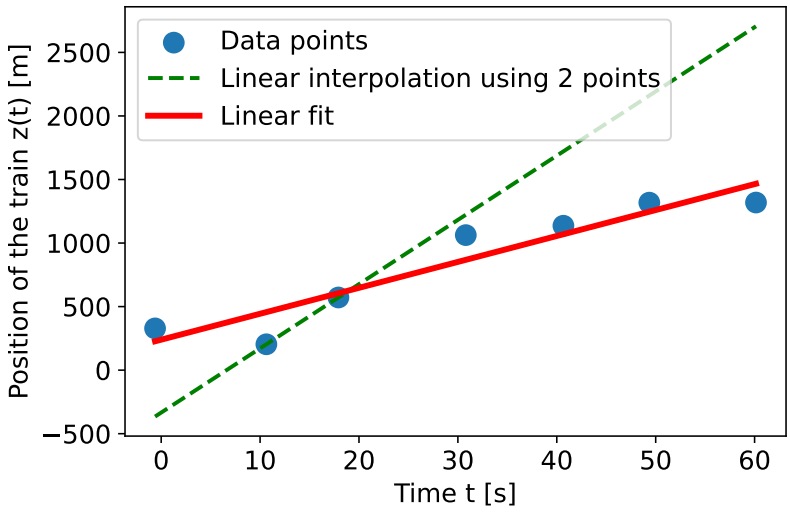

Figure 3: Gathered data from the students.

Consider a train driving at constant velocity in a straight line in the z-direction. 7 physics students want to estimate the speed of the train and predict its position in the future. Each of them goes to a different point of the track and records his position z and the time of passing of the train t. The obtained experimental data is shown in Figure 3, blue dots. Note that real-world data contain errors. In our case, errors may arise from a student being positioned incorrectly, or from small variations in the timing of their measurements.

The students, having followed classical mechanics lectures, know that the train should follow the equation of motion

$$z(t) = vt + z_{\text{init}} \tag{3.1}$$

where v in the velocity and z_{init} the initial position. This equation has two parameters: v and z_{init}. To determine them uniquely, the students only need two points. Indeed, by choosing the points (t_i, z_i) and (t_j, z_j) we have two equations and two unknowns that we can solve:

$$\begin{cases} z_i = vt_i + z_{\text{init}} \\ z_j = vt_j + z_{\text{init}} \end{cases} \Rightarrow z_i - z_j = v(t_i - t_j) \Rightarrow v = \frac{z_i - z_j}{t_i - t_j} \tag{3.2}$$

where in the first step we subtracted the second equation from the first. We can then obtain z_{init} by inserting the found velocity v in one of the two original equations, for example the first one:

$$z_i = \frac{z_i - z_j}{t_i - t_j} t_i + z_{\text{init}} \Rightarrow z_{\text{init}} = z_i - \frac{z_i - z_j}{t_i - t_j} t_i. \tag{3.3}$$

One of these solutions is drawn as a green dashed line in Figure 3. However, this does not solve the problem in a satisfactory way for the students. Indeed, in some cases (as the one chosen in Figure 3) the obtained line is not where the students would intuitively draw the line. They would like to obtain something resembling the red line in Figure 3. There are multiple possible ways to find such a line. A possible idea would be to take an average of multiple estimation of velocities v computed using different pair of points. In the next section, we will present another way to obtain this line: the least squares method.

3.2 Least squares method for linear regression

Consider n data points (t_μ, z_μ) with $\mu = 1, ..., n$ and $t_\mu, z_\mu \in \mathbb{R}$. In later lectures we will add uncertainty to the data points (i.e. error bars). However, we will suppose for now that there is no specified uncertainty on individual data points. Our aim is to find $a, b \in \mathbb{R}$ (previously our velocity v and initial position z_{init}) such that in some sense

$$z_\mu \approx at_\mu + b \ \forall \mu. \tag{3.4}$$

An intuitive way to achieve this is to minimize the distance between each data point (t_μ, z_μ) and the line given by $z = at + b$. The least squares method formalizes this intuition and says that one way to obtain a and b is to minimize the function

$$\mathscr{L}(a, b) = \frac{1}{n}\sum_{\mu=1}^{n}(z_\mu - at_\mu - b)^2. \tag{3.5}$$

over the values of a and b. Here $\mathscr{L}$ is called a *loss function* or alternatively a *cost function*. The particular loss function (3.5) is called the *mean square error* (abbreviated MSE), *quadratic loss*, *square loss* or *L2 loss*.

Note that there are other possible choices of loss functions, for instance by taking the absolute value instead of the square. However, the specific form of the loss function (3.5) is not arbitrary and will be justified probabilistically in later lectures.

The minimization of the loss function (3.5) can be done explicitly. The stationary points are found by determining the values of a and b where the derivatives are 0:

$$\frac{\partial \mathscr{L}(a, b)}{\partial a} = -\frac{2}{n}\sum_{\mu=1}^{n}(z_\mu - (at_\mu + b))t_\mu \stackrel{!}{=} 0 \tag{3.6}$$

$$\frac{\partial \mathscr{L}(a, b)}{\partial b} = -\frac{2}{n}\sum_{\mu=1}^{n}(z_\mu - (at_\mu + b)) \stackrel{!}{=} 0. \tag{3.7}$$

The exclamation points above the equal sign indicate that we want to enforce equality. We can get rid of the constant $-\frac{2}{n}$ in front of both equations by multiplying on both sides by $-\frac{n}{2}$:

$$\sum_{\mu=1}^{n}(z_\mu - (at_\mu + b))t_\mu \stackrel{!}{=} 0 \tag{3.8}$$

$$\sum_{\mu=1}^{n}(z_\mu - (at_\mu + b)) \stackrel{!}{=} 0. \tag{3.9}$$

Then, split the sum in the first equation (3.8), put the terms depending on a and b on the right-hand side and take the constants out of the sums to obtain

$$\sum_{\mu=1}^{n} z_\mu t_\mu \stackrel{!}{=} a\sum_{\mu=1}^{n} t_\mu^2 + b\sum_{\mu=1}^{n} t_\mu. \tag{3.10}$$

For the second equation (3.9), do the same but notice additionally that $\sum_{\mu=1}^{n} b = b\sum_{\mu=1}^{n} 1 = bn$:

$$\sum_{\mu=1}^{n} z_\mu \stackrel{!}{=} a\sum_{\mu=1}^{n} t_\mu + bn. \tag{3.11}$$

Multiplying (3.10) by n and (3.11) by $\sum_{\mu=1}^{n} t_\mu$ and subtracting them we obtain

$$n\sum_{\mu=1}^{n} z_\mu t_\mu - \sum_{\mu=1}^{n} t_\mu \sum_{\mu=1}^{n} z_\mu \stackrel{!}{=} an\sum_{\mu=1}^{n} t_\mu^2 + bn\cancel{\sum_{\mu=1}^{n} t_\mu} - a\sum_{\mu=1}^{n} t_\mu \sum_{\mu=1}^{n} t_\mu - bn\cancel{\sum_{\mu=1}^{n} t_\mu} = a\left[n\sum_{\mu=1}^{n} t_\mu^2 - \left(\sum_{\mu=1}^{n} t_\mu\right)^2\right] \tag{3.12}$$

Thus,

$$a = \frac{n\sum_{\mu=1}^{n} z_\mu t_\mu - \sum_{\mu=1}^{n} t_\mu \sum_{\mu=1}^{n} z_\mu}{n\sum_{\mu=1}^{n} t_\mu^2 - \left(\sum_{\mu=1}^{n} t_\mu\right)^2} = \frac{\frac{1}{n}\sum_{\mu=1}^{n} z_\mu t_\mu - \frac{1}{n^2}\sum_{\mu=1}^{n} t_\mu \sum_{\mu=1}^{n} z_\mu}{\frac{1}{n}\sum_{\mu=1}^{n} t_\mu^2 - \frac{1}{n^2}\left(\sum_{\mu=1}^{n} t_\mu\right)^2} \tag{3.13}$$

where in the last equality we have multiplied the equation by $\frac{1/n^2}{1/n^2} = 1$. Using the notation

$$\bar{t} = \frac{1}{n}\sum_{\mu=1}^{n} t_\mu \tag{3.14}$$

$$\bar{z} = \frac{1}{n}\sum_{\mu=1}^{n} z_\mu \tag{3.15}$$

to denote the arithmetic average, expression (3.13) becomes

$$a = \frac{\overline{zt} - \bar{t}\bar{z}}{\overline{t^2} - \left(\bar{t}\right)^2}. \tag{3.16}$$

To obtain b, isolate it in equation (3.11):

$$b = \frac{1}{n}\sum_{\mu=1}^{n} z_\mu - a\frac{1}{n}\sum_{\mu=1}^{n} t_\mu = \bar{z} - a\bar{t}. \tag{3.17}$$

Notice that to avoid a division by 0 in (3.16), we require that

$$\overline{t^2} \neq \left(\bar{t}\right)^2. \tag{3.18}$$

This condition can be rewritten as

$$\overline{t^2} - \left(\bar{t}\right)^2 = \overline{\left(t - \bar{t}\right)^2} \neq 0. \tag{3.19}$$

Thus, it is sufficient to check the existence of a μ such that $t_\mu \neq \bar{t}$ to guarantee that there is no division by 0.

Finally, recall that the points where the derivatives equal zero indicate stationary points which are not necessarily a minimum. Nevertheless, the nature of the loss function (3.5), which takes the form of a parabola in the variables a and b, ensures that the estimated parameters a and b indeed correspond to the minimum of the loss function.

3.3 General formulation of linear regression

Previously, we considered n data points and used $d = 2$ parameters a and b to fit them. We will now generalize the formulation of linear regression to data in an arbitrary dimension $d \geq 1 \in \mathbb{N}$.

We define

$\vec{y}$	$\in \mathbb{R}^n$	the output values,	$y_\mu \in \mathbb{R}$,	$\mu = 1, ..., n$
X	$\in \mathbb{R}^{n\times d}$	the input data,	$\vec{X}_\mu \in \mathbb{R}^d$,	$X_{\mu i} \in \mathbb{R}$
$\vec{w}$	$\in \mathbb{R}^d$	the parameters,	$w_i \in \mathbb{R}$,	$i = 1, ..., d$.

The goal of linear regression is to find the parameters $\vec{w}$ such that

$$y_\mu \approx \vec{X}_\mu \cdot \vec{w}. \tag{3.20}$$

More precisely, we will write the previous equation

$$y_\mu = \vec{X}_\mu \cdot \vec{w} + \epsilon_\mu \tag{3.21}$$

where $\epsilon_\mu \in \mathbb{R}$ is an error term, which represents the difference between the prediction $\vec{X}_\mu \cdot \vec{w}$ and the output value y_μ. The goal is then to find the parameters $\vec{w}$ such that the error variable ϵ_μ is small for every μ.

3.4 Examples of linear regression

In this section, we present a few examples to apply the formalism of the previous section 3.3 and show the multiple use cases of linear regression.

3.4.1 Finding velocity of a train

The examples we used in section 3.1 is a $d = 2$ dimensional linear regression where the data is fitted with a function of the form $y = at + b$. We identify

$$\vec{y} = \begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{pmatrix}, \quad X = \begin{pmatrix} t_1 & 1 \\ t_2 & 1 \\ \vdots & \vdots \\ t_n & 1 \end{pmatrix}, \quad \vec{w} = \begin{pmatrix} a \\ b \end{pmatrix}$$

so that we indeed have $y_\mu \approx \vec{X}_\mu \cdot \vec{w} = at_\mu + b$.

3.4.2 Polynomial regression

Polynomial regression is a case of linear regression where we want to fit the data $\{t_\mu, z_\mu\}_{\mu=1}^n$ with a polynomial $y = a_0 + a_1 t + a_2 t^2 + ... + a_p t^p,\ p \in \mathbb{N}$. We can identify

$$\vec{y} = \begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{pmatrix}, \quad X = \begin{pmatrix} 1 & t_1 & t_1^2 & \dots & t_1^p \\ 1 & t_2 & t_2^2 & \dots & t_2^p \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & t_n & t_n^2 & \dots & t_n^p \end{pmatrix} = \begin{pmatrix} \vec{x}_1^T \\ \vec{x}_2^T \\ \vdots \\ \vec{x}_n^T \end{pmatrix}, \quad \vec{w} = \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_p \end{pmatrix}.$$

In this case, $d = p + 1$.

A comment on the nomenclature: polynomial regression is a specific case of linear regression. The term linear regression is used to denote that the family of function $f(\vec{x})$ we consider to find $y_\mu \approx f(\vec{x}_\mu)$ is linear in the parameters $\vec{w}$, i.e. f is of the form $f(\vec{x}) = \vec{x} \cdot \vec{w}$. On the other hand, the term polynomial regression indicates the specific form that the vector $\vec{x}$ has to take: $\vec{x} = (1, t, t^2, \dots, t^p)^T$ (modulo some permutations) such that $\vec{x} \cdot \vec{w}$ is indeed a polynomial in t. This also means that the input data X is not necessarily the raw collected data, but can be some transformation of it.

3.4.3 Image classification

Consider n images belonging to several distinct classes. The aim of image classification is to find a function f that takes an image as an input and outputs its corresponding class. For simplicity, let's focus on a binary classification scenario, where we have two classes: 'cat' (labeled as 1) and 'dog' (labeled as -1). Additionally, we will consider black and white images. Each image can be represented as a vector $\vec{X}_\mu \in \mathbb{R}^d,\ \mu = 1, ..., n$, where the value x_i is the grey level of pixel i and d is the total number of pixels of the image (we assume that every image has the same resolution). Our goal is to find $f : \mathbb{R}^d \to \{\pm 1\}$ such that $f(x_\mu) \approx y_\mu$. Here, y_μ takes on the value of 1 if image μ is a cat and -1 if it is a dog.

In the linear regression case we search for a function of the form $f(\vec{x}) = \vec{x} \cdot \vec{w}$. This function will in general not output ± 1, so how can we make a prediction about the class when given a new image $\vec{X}_{\text{new}}$? One approach is to check whether $f(\vec{X}_{\text{new}})$ is closer to 1 or -1 and to assign the class accordingly. This can be done in practice using the sign function: to predict the class of an image $\vec{X}$ use $\tilde{f}(\vec{X}) = \text{sgn}\left(f(\vec{X})\right) = \text{sgn}\left(\vec{X} \cdot \vec{w}\right)$. This example demonstrates that we can use linear regression even when the output values y_μ are discrete, as in classification tasks.

Note that using linear regression to classify images is far from the state-of-the-art solution for image classification problems. This is to be expected, since the class of function of form $f(\vec{x}) = \vec{x} \cdot \vec{w}$ is not generic enough for this task. A better approach would be to use neural networks, which will be discussed briefly in lecture 5.

3.4.4 Image reconstruction from an X-ray scanner (X-ray tomography)

X-ray computed tomography (usually named CT scans) is a technique used to obtain internal images of 3D objects. Let us send an x-ray of initial intensity I_0 trough the object we want to probe. The intensity of the X-ray decreases exponentially following the law

$$I = I_0 e^{-\int C(x,y,z) ds} \tag{3.22}$$

where $\int ...ds$ is a line integral and $C(x, y, z)$ is the absorption coefficient at position (x, y, z). The line integral is taken along the trajectory of the X-ray (the dashed line in Figure 4). We will suppose that the trajectory is known from the placement of the X-ray source. A detector measures the intensity I of the X-ray after it traveled trough the object.

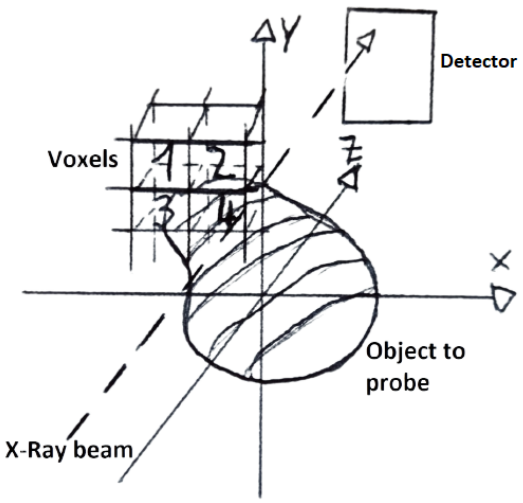

Figure 4: Schema of one measurement for CT scans.

The aim of X-ray tomography is to estimate the value of the absorption coefficient $C(x, y, z)$. This can be very useful in medical imaging, for instance to detect tumors that have a different absorption coefficient than the rest of the tissues. We can rewrite equation (3.22) as

$$-\ln\left(\frac{I}{I_0}\right) = \int C(x, y, z)ds. \tag{3.23}$$

To approximate the integral, we discretise the 3D space in small cubes (called voxels) and index them with $i = 1, ..., d$. The integral corresponding to the beam with index μ can then be estimated by

$$\int C(x, y, z)ds \approx \sum_{i=1}^{d} X_{\mu i} C_i \tag{3.24}$$

where $X_{\mu i} = 1$ if the beam μ passes trough voxel i and $X_{\mu i} = 0$ otherwise, and C_i is the average absorption coefficient of voxel i. As an example in Figure 4 the beam passes trough voxel 4 but not 1,2 and 3.

We now consider we have n beams (different positions of the detector and different angles trough the sample) leading to n measurements on the X-ray detector. Each measurement $\mu = 1, ..., n$ gives an output intensity I_μ and a vector $\vec{X}_\mu$ where $X_{\mu i} = 1$ if the beam μ passes trough voxel i, else $X_{\mu i} = 0$. Defining

$$y_\mu = -\ln\left(\frac{I_\mu}{I_0}\right) \tag{3.25}$$

and identifying $w_i := C_i$, we find an equation with the desired form

$$y_\mu \approx \vec{X}_\mu \cdot \vec{w}. \tag{3.26}$$

3.5 Least squares method for linear regression: general case

We now do the calculation of 3.2 for the general case $d \geq 1$. Let $\vec{y}$, X and $\vec{w}$ be as defined in 3.3. We want to find $\vec{w}$ that minimizes the loss function

$$\mathscr{L}(\vec{w}) = \frac{1}{n}\sum_{\mu=1}^{n}\left(y_\mu - \vec{X}_\mu \cdot \vec{w}\right)^2 \tag{3.27}$$

The stationary points are situated where every partial derivatives is equal to 0:

$$\frac{\partial \mathscr{L}(\vec{w})}{\partial w_k} \stackrel{!}{=} 0 \ \forall k = 1, ..., d \tag{3.28}$$

This stationary point will be a minimum since $\mathscr{L}(\vec{w})$ is a parabola. Recall that $\vec{X}_\mu \cdot \vec{w} = \sum_{i=1}^{d} X_{\mu i} w_i$ to compute the partial derivative

$$\frac{\partial \mathscr{L}(\vec{w})}{\partial w_k} = \frac{-2}{n} \sum_{\mu=1}^{n} \left(y_\mu - \sum_{i=1}^{d} X_{\mu i} w_i \right) X_{\mu k} \stackrel{!}{=} 0. \tag{3.29}$$

We now absorb the constant $\frac{-2}{n}$ in the zero, split the sum and put the negative terms on the right-hand side of the equation to obtain

$$\sum_{\mu=1}^{n} y_\mu X_{\mu k} \stackrel{!}{=} \sum_{\mu=1}^{n} \sum_{i=1}^{d} X_{\mu i} w_i X_{\mu k}. \tag{3.30}$$

Rearranging the two sums in the right hand-side, the previous equation can be expressed as

$$\sum_{\mu=1}^{n} y_\mu X_{\mu k} \stackrel{!}{=} \sum_{i=1}^{d} \left(\sum_{\mu=1}^{n} X_{\mu k} X_{\mu i} \right) w_i. \tag{3.31}$$

Using the fact that $X_{\mu k} = \left[X^T\right]_{k\mu}$ the left-hand side becomes

$$\sum_{\mu=1}^{n} y_\mu X_{\mu k} = \sum_{\mu=1}^{n} \left[X^T\right]_{k\mu} y_\mu = \left[X^T \vec{y}\right]_k \tag{3.32}$$

and the right-hand side

$$\sum_{i=1}^{d} \left(\sum_{\mu=1}^{n} X_{\mu k} X_{\mu i} \right) w_i = \sum_{i=1}^{d} \left(\sum_{\mu=1}^{n} \left[X^T\right]_{k\mu} X_{\mu i} \right) w_i = \sum_{i=1}^{d} \left[X^T X\right]_{ki} w_i = \left[\left[X^T X\right] \vec{w}\right]_k. \tag{3.33}$$

Putting the obtained expressions together, we have

$$\left[X^T \vec{y}\right]_k \stackrel{!}{=} \left[\left[X^T X\right] \vec{w}\right]_k. \tag{3.34}$$

This equation is valid for all k, so we can write it in matrix form

$$X^T \vec{y} \stackrel{!}{=} X^T X \vec{w} \tag{3.35}$$

Thus, if $X^T X$ is invertible we have that

$$\hat{\vec{w}} := \underset{\vec{w} \in \mathbb{R}^d}{\operatorname{argmin}} \left(\mathscr{L}(\vec{w}) \right) = \left(X^T X \right)^{-1} X^T \vec{y}. \tag{3.36}$$

The hat notation is often used to denote the parameters $\vec{w}$ that minimize a given loss function.

The matrix $\left(X^T X\right)^{-1} X^T$ is called the *pseudo-inverse* of X. The pseudo-inverse can be thought as a way to invert rectangular matrices. Notice that if X is invertible, then the pseudo-inverse matrix is equal to X^{-1}.

4 Lecture 4: Linear regression continued

4.1 Estimation vs. Prediction

The last lecture focused on the minimization of the least squares objective, which led to the estimated parameters $\hat{\vec{w}} = (X^T X)^{-1} X^T y$. But why are we interested in obtaining this estimate of $\vec{w}$ in the first place? There are two tasks for which computing $\hat{\vec{w}}$ is needed, and we discuss them in the following.

Estimation The first one is estimation, meaning that we're directly interested in knowing the values of $\vec{w}$. In this case, $\vec{w}$ often has a physical meaning. For example, when we measure the position of a train as a function of time, the parameter a represents the train's speed. Then a is the parameter we want to estimate, to know how fast the train was. Similarly, in X-ray tomography, the parameter $\hat{\vec{w}}$ contains the estimated absorption coefficients of all the voxels, from which we can reconstruct an image of the tissue.

Prediction The second task is prediction. Suppose we get a new data point $\vec{X}_{\text{new}} \in \mathbb{R}^d$, which is different from all other points contained in the dataset X, and for which we don't know the corresponding output y_{new}. One reasonable way to estimate the output is to use

$$\hat{y}_{\text{new}} = \vec{X}_{\text{new}} \cdot \hat{\vec{w}}. \tag{4.1}$$

We call $\hat{y}_{\text{new}}$ the *predicted* output.

- In the example of the train we can ask where the train will be at a future time t_{new}.
- When classifying images, we give a new image $\vec{X}_{\text{new}}$ and ask whether it is a cat or a dog. The prediction is given by computing $\hat{y}_{\text{new}} = \text{sign}\left(\vec{X}_{\text{new}} \cdot \hat{\vec{w}}\right)$.

It depends on the application, whether estimation or prediction is the purpose of finding $\vec{w}$. Notice that in general solving the estimation problem implies that we can also solve the prediction problem, as it's sufficient to plug the estimated parameter into the model to get the prediction. The converse – obtaining a physically meaningful and interpretable estimate of the parameters from a prediction – is generally harder and often an open problem, as the parameters (e.g. of a neural network) can be challenging to interpret.

4.2 Existence and uniqueness of the least squares solution

The formula for $\hat{\vec{w}}$ (3.36) requires inverting the matrix $X^T X$. One can ask: Is it always the case that this operation can be performed? If not, what influence do n (the number of data points) and d (their dimension) have on the invertibility?

In the simple case of linear regression with $d = 2$ (the example with a position of a train as a function of time) the invertibility condition is equivalent to the denominator in (3.16) not being zero. This happens anytime $n \geq 2$ and when at least two different values for t_μ exist. So in this case, and generically for cases where $n \gg d$, the invertibility condition does not pose a problem.

To answer this question in general we need to look at the rank of $X^T X$. Recall that only full rank (i.e., when the rank is equal to the number of columns) matrices can be inverted. To determine the rank of $X^T X$, we start from the rank of X. We have $\text{rank}(X) \leq \min(n, d)$. In Lecture 2 we have derived a relation between the squared singular values of X and the eigenvalues of $X^T X$, Proposition 2. Combining this with the fact that $\text{rank}(X) = \text{rank}(X^T)$, we have that $\text{rank}(X^T X) \leq \min(n, d)$. The first conclusion is that if $n < d$, then $X^T X$ is not invertible. For $n > d$, the matrix is invertible under the condition that at least d datapoints $\vec{X}_\mu$, $\mu = 1, \ldots, n$ are linearly independent. This is often the case in real data.

At this point, we established that for $n < d$, $X^T X$ is never invertible. Let's try to understand better what is happening in this regime. Recall that we aimed to minimize $\mathscr{L}(\vec{w}) = \frac{1}{n} \sum_{\mu=1}^{n} \left(y_\mu - \vec{X}_\mu \cdot \vec{w}\right)^2$. Taking the derivative with respect to $\vec{w}$ and setting it to zero, we found expression (3.34), which can be rewritten as $X^T(X\hat{\vec{w}} - y) = 0$. Then we *assumed* that $X^T X$ was invertible and found the solution (3.36). Now let us move to the non-invertible case, taking $n < d$. Here, we cannot use (3.36), as its assumption is violated. However $X^T(X\hat{\vec{w}} - y) = 0$ still holds. In particular, we can directly solve $X\hat{\vec{w}} - y = 0$, which is a system of n equations in d unknowns. This means that in general[1] there will be an $n - d$ dimensional subspace of parameter vectors that satisfy this equation. In other words

[1] The technical condition is that X has rank n.

all the solutions $\hat{\vec{w}}$ will satisfy $y_\mu = \vec{X}_\mu \cdot \hat{\vec{w}}$, $\forall \mu = 1, \dots, n$. Therefore, all solutions will also have zero loss $\mathscr{L}(\hat{\vec{w}}) = 0$. To summarize, in the $n < d$ regime three things happen:

1. Expression (3.36) for finding $\hat{\vec{w}}$, is not valid anymore as its assumptions are violated.
2. The solution to $\min_{\vec{w}} \mathscr{L}(\vec{w})$, still exists, but it is not unique anymore. Instead, a whole $n - d$ dimensional subspace of solutions appears.
3. All the solutions perfectly fit the points, giving $\mathscr{L}(\hat{\vec{w}}) = 0$. This is somewhat intuitive because we have more fitting parameters (degrees of freedom) than points, hence we can find a predictor that passes through every pair $(\vec{X}_\mu, y_\mu)$.

4.3 The notion of regularization and ridge regression

We saw that whenever $n < d$, the solution to $\min_w \mathscr{L}(w)$ is not unique anymore. So a natural question is: *Among all vectors $\hat{\vec{w}}$ that satisfy $\mathscr{L}(w) = 0$, which one should we pick?* A possible answer is to pick the solution with the smallest norm.[2] To select this solution, we introduce the ridge regression loss, which includes a penalty for a high norm of $\vec{w}$.

$$\mathscr{L}_\lambda(\vec{w}) = \frac{1}{n}\sum_{\mu=1}^{n}\left(y^\mu - \vec{X}_\mu \cdot \vec{w}\right)^2 + \frac{\lambda}{n}\sum_{i=1}^{d} w_i^2, \qquad \lambda > 0 \tag{4.2}$$

The loss is again quadratic in $\vec{w}$, therefore, imposing that $\nabla_{\vec{w}}\mathscr{L}_\lambda(\vec{w}) = 0$, will give us the global minimum. We have

$$\frac{\partial \mathscr{L}_\lambda}{\partial w_k} = -\frac{2}{n}\sum_{\mu=1}^{n}\left(y_\mu - \sum_{i=1}^{d} X_{\mu i} w_i\right) X_{\mu k} + \frac{2\lambda}{n} w_k, \qquad k = 1, \dots, d \tag{4.3}$$

Setting all the derivatives to zero gives the following conditions for the minimizer $\hat{\vec{w}}$:

$$0 = -\sum_{\mu=1}^{n}\left(y_\mu - \sum_{i=1}^{d} X_{\mu i}\hat{w}_i\right) X_{\mu k} + \lambda \hat{w}_k \tag{4.4}$$

$$\overset{(a)}{=} -\sum_{\mu=1}^{n} X_{\mu k} y_\mu + \left[\sum_{i=1}^{d}\left(\sum_{\mu=1}^{n} X_{\mu i} X_{\mu k}\right)\hat{w}_i\right] + \lambda \hat{w}_k \tag{4.5}$$

$$\overset{(b)}{=} -\sum_{\mu=1}^{n} X_{\mu k} y_\mu + \sum_{i=1}^{d}\left[\delta_{ik}\lambda \hat{w}_i + \left(\sum_{\mu=1}^{n} X_{\mu k} X_{\mu i}\right)\hat{w}_i\right] \tag{4.6}$$

$$\overset{(c)}{=} -\sum_{\mu=1}^{n} X_{\mu k} y_\mu + \sum_{i=1}^{d}\left[\left(\sum_{\mu=1}^{n} X_{\mu k} X_{\mu i}\right) + \lambda\delta_{ik}\right]\hat{w}_i, \tag{4.7}$$

To obtain the first expression, we multiplied (4.3) by $n/2$. In (a), we separated the term with y and exchanged the sums over μ and i in the other term. In (b) we brought $(\lambda/n)\hat{w}_k$ inside the sum over i, by introducing the delta function δ_{ij} which is one for $i = j$ and zero otherwise. In (c) we collected $\hat{w}_i$ outside.

Let's rewrite this in matrix notation. We have $\left(\sum_{\mu=1}^{n} X_{\mu k} X_{\mu i}\right) + \lambda\delta_{ik} = \left(X^T X + \lambda \mathbb{I}_d\right)_{ki}$ and $\sum_{\mu=1}^{n} X_{\mu k} y_\mu = (X^T y)_k$. Therefore $\hat{\vec{w}}$ must satisfy $(X^T X + \lambda \mathbb{I}_d)\hat{\vec{w}} = X^T y$. All the eigenvalues of $X^T X + \lambda \mathbb{I}_d$ are larger than λ[3]. Since there are no zero eigenvalues, one can invert the matrix, as $((X^T X + \lambda \mathbb{I})^{-1})_{ij} = \sum_{k=1}^{d} \frac{1}{\lambda_k}(v_k)_i (v_k)_j$, with $\lambda_k \in \mathbb{R}, v_k \in \mathbb{R}^d$ being respectively the $k - th$ eigenvalue and eigenvector of the matrix being inverted. Finally, we arrive at the expression of the ridge regression minimizer

$$\hat{\vec{w}} = \left(X^T X + \lambda \mathbb{I}_d\right)^{-1} X^T y. \tag{4.8}$$

[2] Ok, but why the smallest norm? A possible argument is the following. Take a $\vec{w}$ satisfying $X\vec{w} = \vec{y}$. Decompose $\vec{w}$ as $\vec{w} = \vec{w}_X + \vec{w}_0$, where $\vec{w}_0 \in \ker(X)$ and $\vec{w}_X$ is orthogonal to $\vec{w}_0$. Then we have $X\vec{w} = X(\vec{w}_X + \vec{w}_0) = X\vec{w}_X = 0$. Therefore, we see that $X\vec{w} = \vec{y}$ only constrains w_X, giving no prescription for $\vec{w}_0$. Since there is no sensible way to pick $\vec{w}_0$ based on the data, the best is to put it to zero. The minimal norm prescription does exactly this: $\|\vec{w}\|^2 = \|\vec{w}_X\|^2 + \|\vec{w}_0\|^2$ (thanks to orthogonality), which is minimal when $\vec{w}_0 = 0$.

[3] Let's prove this. Take an arbitrary vector of unit norm $v \in \mathbb{R}^d$. Compute $v^T(X^T X + \lambda \mathbb{I}_d)v = v^T X^T X v + \lambda v^T v = (Xv)^T(Xv) + \lambda = \|Xv\|^2 + \lambda$. Since v is arbitrary this also applies to eigenvectors of $X^T X + \lambda \mathbb{I}_d$. The expression then says that all eigenvalues are positive and larger than λ.

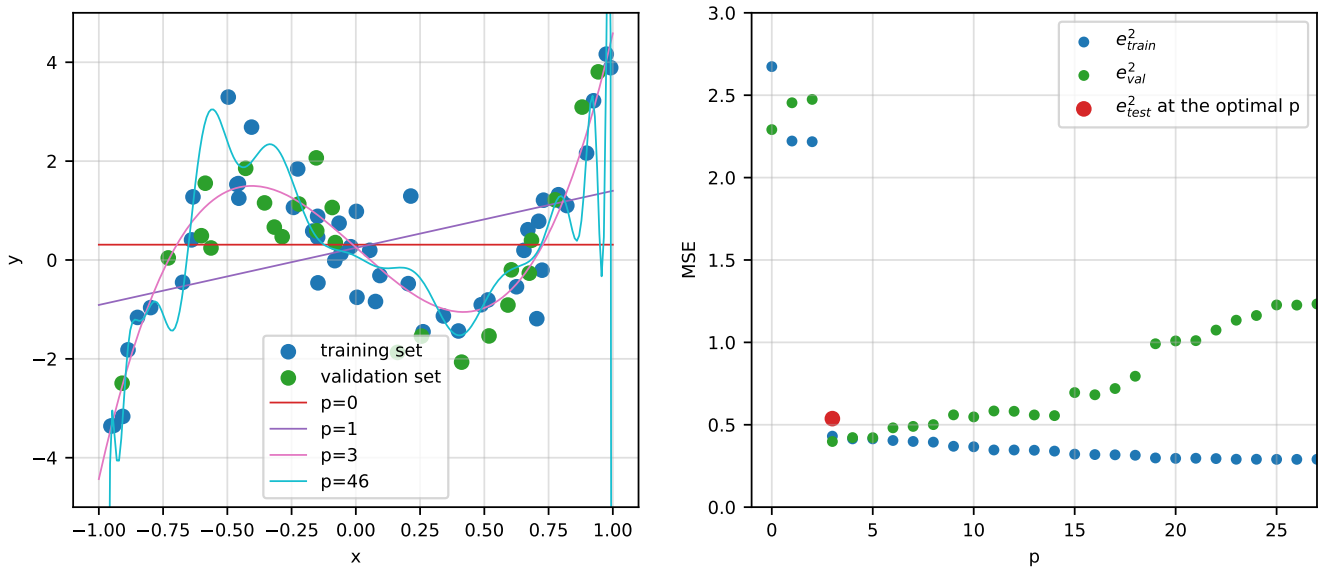

Figure 5: Overfitting in polymonial regression. p is the degree of the polynomial, which was fitted using ridge regression with $\lambda = 10^{-13}$. The left plot shows how both using a too low degree and a too high degree polynomials leads to poor generalization capabilities. In the first case, the polynomial is unable to fit the signal in the data. In the second case, the polynomial also fits the noise, leading to overfitting. The right depicts the train and validation errors as a function of the degree. The train error is monotonously decreasing, since increasing the degree leads to better fitting of the training data. Instead the validation error is U-shaped with a minimum at $p = 3$. We also plot the test error of the predictor with $p = 3$.

From this expression we can also understand why the term $\lambda \sum_{i=1}^{d} w_i^2$ in the loss is called a *regularizer*. Take the case $n < d$ and set $\lambda = 0$. We already saw how (4.8) is not applicable. But as soon as we put $\lambda > 0$, (4.8) works. In this sense, the problem has been regularized (made well posed) by the additional term.

Minimal norm interpolant Before moving on, let's check that in the case $n < d$ and vanishing regularization, one indeed gets the minimal norm solution to $\vec{y} = X\vec{w}$. Pick a small but positive regularization $\lambda = \epsilon$. In this case, the minimization problem becomes

$$\min_{\vec{w}} \mathscr{L}_{\lambda=\epsilon}(\vec{w}) = \frac{1}{n} \min_{\vec{w}} \|\vec{y} - X\vec{w}\|^2 + \epsilon \|\vec{w}\|^2 \approx \frac{\epsilon}{n} \min_{\vec{w}:\vec{y}=X\vec{w}} \|w\|^2. \tag{4.9}$$

To say this in words: Recall that we had a subspace of solutions that perfectly fitted the training data. A small regularization selects the optimal solution with the smallest norm. The reason why the regularization needs to be small in this argument is that otherwise the minimizer of $\mathscr{L}_\lambda$ will not fit perfectly the points anymore.

4.4 Overfitting and its quantification via a hold-out validation set

Overfitting arises when the model has enough parameters to fit noise in the data. In many applications when the data contains noise, we want to fit the signal in the data but ignore the noise. Overfitting is problematic because the model learns the noise in the data instead of the underlying signal, hence it will likely perform poorly on unseen data during prediction, and will not give the correct parameters for estimation.

Take for example the case of polynomial regression, as seen in Section 3.4.2. The left panel of Figure 5 illustrates the phenomenon of overfitting. When one tries to fit the given blue data points with very large degree polynomials, the predictor ends up fitting the noise. This shows in the curly wiggly form that follows the blue dots precisely. Instead, choosing a polynomial with degree only one fails to give a good fit. But how should we determine the optimal degree of the polynomial given the possible choices?

Hyperparameters The previous motivating example introduced us to the important concept of a *hyperparameter*. A hyperparameter is a tunable parameter of the whole data analysis procedure. Examples of hyperparametwrs are the degree of the polynomial p in polynomial regression, and the value of the regularizer λ in ridge regression. A

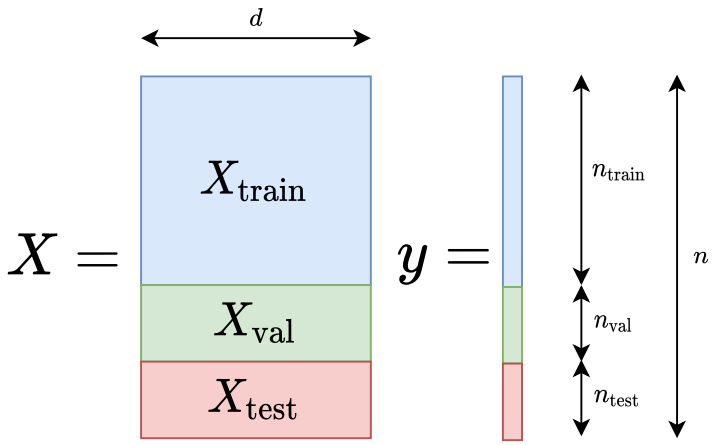

Figure 6: Depiction of the splitting of the dataset into training set, validation set and test set. If the original dataset has a total of n samples we must have $n_{\text{train}} + n_{\text{val}} + n_{\text{test}} = n$. It is important the the 3 sets are sampled independently. Usually, this can be achieved by sampling the rows of the data matrix randomly before performing the split as above.

hyperparameter differs from a parameter (such as $\vec{w}$) because parameters are explicitly set by the fitting algorithm, whereas hyperparameters are not. The value of hyperparameters can have a strong impact on the overall performance, therefore it's important to know how to tune them.

4.4.1 Train, validation, and test sets

Other than using the available data to fit the model we have two more necessities. First, we must find a way to tune the hyperparameters. Second, after we picked the hyperparameters we had to evaluate the model's performance.

In the procedure used most commonly in current machine learning, each of these operations requires a new independent set of hold-out samples. Here, hold-out means that we use some samples only for evaluation of the model, and not the training.

Concretely, we split the whole dataset randomly into three parts, as depicted in Figure 6 after a random permutation of the rows of the data matrix X. This procedure is practical when the number of available samples is large so that fluctuations coming from different random splits of the dataset are negligible. Later in the course, we will discuss other ways of quantifying errors in linear regression using probabilistic tools.

The usage of the tree parts is the following:

- The **training set** ($X_{\text{train}} \in \mathbb{R}^{n_{\text{train}} \times d}, \vec{y}_{\text{train}} \in \mathbb{R}^{n_{\text{train}}}$) contains the samples that are fed to the fitting algorithm. For example the ridge regression predictor will be $\hat{\vec{w}}_\lambda = \left(X_{\text{train}}^T X_{\text{train}} + \lambda \mathbb{I}_d\right)^{-1} X_{\text{train}}^T y_{\text{train}}$. The training error is the mean square error of the predictor on the training set, in formulas $e^2_{\text{train}} = \frac{1}{n_{\text{train}}} \sum_{\mu=1}^{n_{\text{train}}} \left((\vec{y}_{\text{train}})_\mu - (\vec{X_{\text{train}}})_\mu \cdot \hat{\vec{w}}_\lambda \right)^2$. [4]

- The **validation set** $(X_{\text{val}}, \vec{y}_{\text{val}})$ is used to pick the best value for the hyperparameters. In the ridge regression example we compute $\hat{\vec{w}}_\lambda$ for various values of λ and then for each we find the validation error $e^2_{\text{val}}(\hat{\vec{w}}_\lambda) = \frac{1}{n_{\text{val}}} \sum_{\mu=1}^{n_{\text{val}}} \left((\vec{y}_{\text{val}})_\mu - (\vec{X_{\text{val}}})_\mu \cdot \hat{\vec{w}}_\lambda \right)^2$. We then obtain the best hyperparameters by minimizing the validation error. For the regularizer in ridge regression we have $\lambda_\star = \arg\min_\lambda e^2_{\text{val}}(\hat{\vec{w}}_\lambda)$.

- The **test set** $(X_{\text{test}}, \vec{y}_{\text{test}})$ is used only once after the best hyperparameter has been chosen based on the validation set. The purpose of the test set is to give a faithful measure of the performance of the algorithm. The test error is $e^2_{\text{test}} = \frac{1}{n_{\text{test}}} \sum_{\mu=1}^{n_{\text{val}}} \left((\vec{y}_{\text{val}})_\mu - (\vec{X_{\text{test}}})_\mu \cdot \hat{\vec{w}}_{\lambda_\star} \right)^2$.

The same framework applies in the case of polynomial regression, where the hyperparameter is p (or both p and λ). It is interesting to comment on the behavior of the training and validation error as a function of p. This is

[4] Watch out! The training error does not necessarily coincide with the loss function we minimize. In the case of ridge regression the training error never includes the regularizer.

shown in the right panel of Figure 5. As p increases the polynomial gets better at fitting the train data points, hence the training error decreases monotonously. Therefore it is hard to determine the best p by looking at the training error. Looking at the validation error, we see that it first starts decreasing with p and then increases again, with a minimum at $p = 3$. We select this value and compute the corresponding test error, shown as a red dot in the plot.

An important aspect is the relative size of the training, validation, and test set. How many data samples should we use to train, how many to select hyperparameters and how many to test? In many cases with a large number of samples roughly 60-80% of the dataset is used for training, 10-20% for validation and another 10-20% for test. On one hand, one wants to maximize the number of samples in the training set to be able to do a good estimation. On the other hand, having too few data in the validation and test set means that the validation and test error will be more noisy and will not be representative of the true performance of the algorithm.

4.5 More general cases of linear regression and gradient descent

Let us now introduce an important algorithm by considering a more general case of regression starting from a simple example. Suppose we collected data for which we expect the following dependence

$$z_\mu = a t_\mu^b + c, \ \mu = 1, \dots, n$$

for some unknown parameters a, b, c. How to perform the fit?

If $c = 0$, we can still cast this problem as linear regression by taking the log of both sides, which gives $\log z_\mu = \log a + b \log t_\mu$. Hence defining $\vec{X}_\mu = (1, \log t_\mu)$, $y_\mu = \log z_\mu$, and $w = (b, \log a)$, we can express the relation between z and t as $\vec{y}_\mu = \vec{w} \cdot \vec{X}_\mu$.

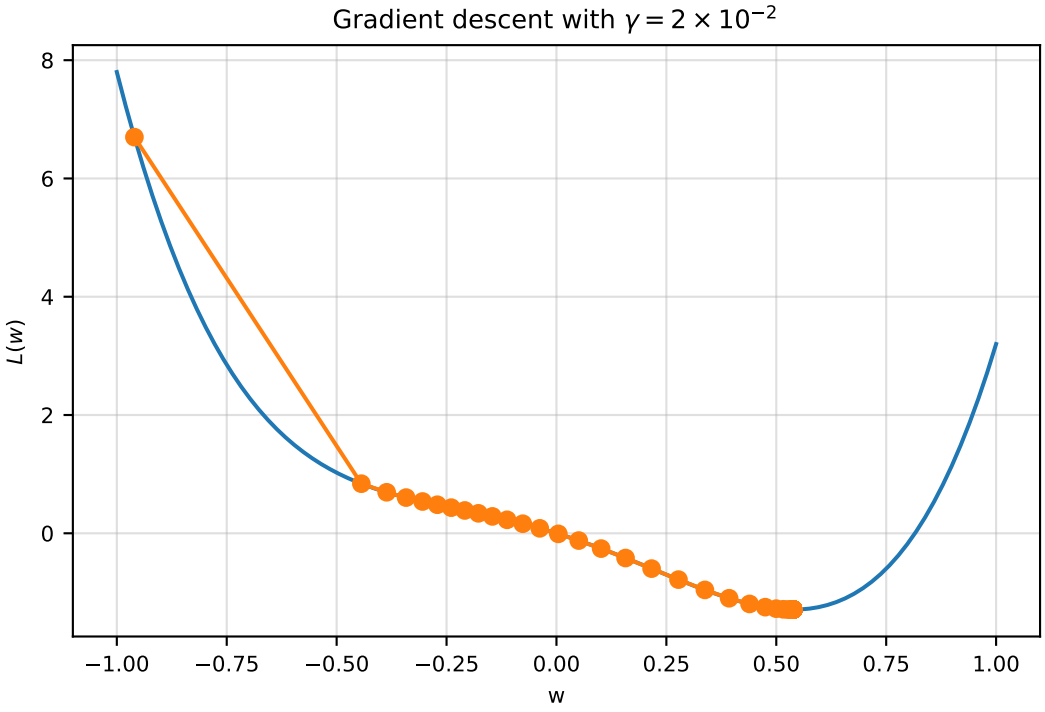

Figure 7: Example of gradient descent (orange points) iterations on a loss function (blue) with a fixed learning rate.

But what about the original examples where $y_\mu = a t_\mu^b + c, \ \mu = 1 \dots, n$ for unknown a, b, c. This time the log trick does not work anymore and therefore we cannot map the problem into a linear regression one. However, it is still sensible to minimize the square loss:

$$\mathscr{L}(a, b, c) = \frac{1}{n} \sum_{\mu=1}^{n} \left(y_\mu - a t_\mu^b - c \right)^2 . \tag{4.10}$$

We hope to find a minimizer by first finding the stationary points. For this we need to compute the gradient and set

it to zero. The gradient of the loss with respect to the parameters a, b, c is

$$\frac{\partial \mathscr{L}}{\partial a}(a,b,c) = -\frac{2}{n}\sum_{\mu=1}^{n}\left(y_\mu - a t_\mu^b - c\right) t_\mu^b, \tag{4.11}$$

$$\frac{\partial \mathscr{L}}{\partial b}(a,b,c) = -\frac{2}{n}\sum_{\mu=1}^{n}\left(y_\mu - a t_\mu^b - c\right) a t_\mu^b \log t_\mu, \tag{4.12}$$

$$\frac{\partial \mathscr{L}}{\partial c}(a,b,c) = -\frac{2}{n}\sum_{\mu=1}^{n}\left(y_\mu - a t_\mu^b - c\right). \tag{4.13}$$

However, this gives a system of three equations in three unknowns, which does not have an analytical solution. The approach we will now describe is to find the minimum numerically. One commonly used way to do so is using the gradient descent algorithm. The idea is to iteratively update the parameters in the direction of the negative gradient from the formulas

$$\vec{w}_i^{t+1} = \vec{w}_i^t - \gamma \frac{\partial \mathscr{L}}{\partial w_i}(\vec{w}^t), \qquad i = 1, \dots, d \tag{4.14}$$

where $\gamma > 0$ is the learning rate, which can be understood as the step size of the algorithm. The algorithm is initialized with some $\vec{w}^0$, usually chosen randomly with a small norm, and then iterated until convergence. The convergence is usually checked by looking at the norm of the gradient, which should be small (a small norm means that there are no big changes locally anymore, so we might as well stop moving). A typical condition for convergence is

$$\sum_i \left(\frac{\partial \mathscr{L}}{\partial w_i}\right)^2 < \epsilon\,, \tag{4.15}$$

for some small ϵ.[5] Using an appropriate stopping criterion gradient descent converges to a *local* minimum of the loss. Therefore there is no guarantee that we'll reach the global minimum of the loss. The learning rate γ is a hyperparameter of the algorithm. If it's too small the algorithm will converge very slowly. If it's too large, the algorithm might not converge but instead jump all over. Figure 7 shows some of these gradient descent iterations. One indeed sees that the first step is much bigger than the rest because the loss at initialization is very steep. Then, as we approach the local minimum, the gradient norm gets even smaller and becomes practically 0 at the minimum.

[5] Another stopping criterion consists of monitoring the validation error as a function of the iteration number t, and stop the algorithm when the validation error starts increasing. This is called *early stopping*.

5 Lecture 5: Gradient Descent for Regression and Neural Networks

5.1 Understanding the Gradient Descent Update Step

In this lecture, we will have a look at the gradient descent algorithm in greater detail. In particular we ask, why is the update step proposed in (4.14) a reasonable choice?

To answer this, consider we want to update the parameters $\vec{w}^t$ to get the parameters $\vec{w}^{t+1}$ which should be closer to a minimum than before. Because the function $\mathscr{L}(\vec{w})$ can be an arbitrary function, the immediate analytical minimization might not be feasible. But what is very much possible is minimizing a local analytic approximation around $\vec{w}^t$ instead.

We will illustrate the reasoning in one dimensional case when w is a scalar. The generalization to vectors is then straightforward. For a local approximation we consider the Taylor series up until the second degree at the point w^t:

$$\tilde{\tilde{\mathscr{L}}}(w) = \mathscr{L}(w^t) + \frac{\partial \mathscr{L}(w)}{\partial w}\Big|_{w=w^t}(w - w^t) + \frac{1}{2}\frac{\partial^2 \mathscr{L}(w)}{\partial w^2}\Big|_{w=w^t}(w - w^t)^2 + o\left(\left(w - w^t\right)^2\right) \tag{5.1}$$

$$= \mathscr{L}(w^t) + \frac{\partial \mathscr{L}(w^t)}{\partial w}(w - w^t) + \frac{1}{2}\frac{\partial^2 \mathscr{L}(w^t)}{\partial w^2}(w - w^t)^2 + o\left(\left(w - w^t\right)^2\right) \tag{5.2}$$

We use the color blue to make clear on which variables where the function $\tilde{\tilde{\mathscr{L}}}(w)$ depends, and which w^t act as constants. The second line is a simplified notation for writing that the derivatives are taken at w^t. The resulting function $\tilde{\tilde{\mathscr{L}}}$ is a local polynomial approximation of $\mathscr{L}(w)$ of degree two.

In large dimension d the second derivative is the $d \times d$ Hessian matrix which is computationally costly to obtain and store. Hence to derive gradient descent we replace the second derivative at the point w^t with a scalar $1/\gamma$. Effectively, changing this parameter allows us to adapt the curvature of the approximating parabola (small $\gamma \to$ small width, large $\gamma \to$ large width). This gives a new local approximation at w^t of the function $\mathscr{L}$, depending on the scalar γ:

$$\tilde{\mathscr{L}}(w) = \mathscr{L}(w^t) + \frac{\partial \mathscr{L}(w^t)}{\partial w}(w - w^t) + \frac{1}{2\gamma}(w - w^t)^2 \,. \tag{5.3}$$

Figure 8 shows an example of function $\mathscr{L}$ with two approximating parabolas with different values of the parameter γ. Now, we can analytically find the minimum of this function, by deriving it w.r.t. w and setting the derivative to zero. This gives

$$\frac{\partial \tilde{\mathscr{L}}(w)}{\partial w} = \frac{\partial \mathscr{L}(w^t)}{\partial w} + \frac{1}{\gamma}(w - w^t) = 0 \tag{5.4}$$

$$\Rightarrow w^{t+1} := w = w^t - \gamma \frac{\partial \mathscr{L}(w^t)}{\partial w} \tag{5.5}$$

Since the approximate function $\tilde{\mathscr{L}}$ is minimized at w^{t+1}, this is a reasonable choice for a next value that might be closer to a minimum of the $\mathscr{L}$. Considering the case of higher dimension, we realize that the above argument generalizes straightforwardly, recovering the update step from (4.14). As a final remark, notice that the curvature parameter γ is often called "step size" or "learning rate".

5.2 Properties of Gradient Descent

Setting the learning rate γ. Let us reflect a moment on how to set γ, the learning rate of the update step, which is the width of the parabola approximating $\mathscr{L}(\vec{w})$ at the point $\vec{w}^t$. In Figure 8 you can see that different values of γ lead in general to different values of the next iterate of the algorithm, $\vec{w}^{t+1}$. Thus, we may ask whether there is an optimal, or smart, way to set the parameter γ. Sadly, there is no prescribed general way of setting γ so that you will be able to arrive at a minimum efficiently. If γ is too small, it will take many steps to converge to a local minimum of $\mathscr{L}$, and thus the algorithm will be very slow. If γ is too large, you might jump around the loss but never reach a minimum at all (i.e. fulfill your convergence criterion (4.15)). There are some adaptive methods in the literature that make γ a function of t, changing the step size throughout the dynamics of gradient descent. This has been shown to be more effective than a fixed step size in practice. But generally, it is difficult to say which strategy, or which numerical values of the step size, is best. Fro example, you may try different values of γ and check which one is performing best on a validation dataset (recall that our motivating example is that of a non-linear regression on a given dataset, where good performance on a validation set is the objective).

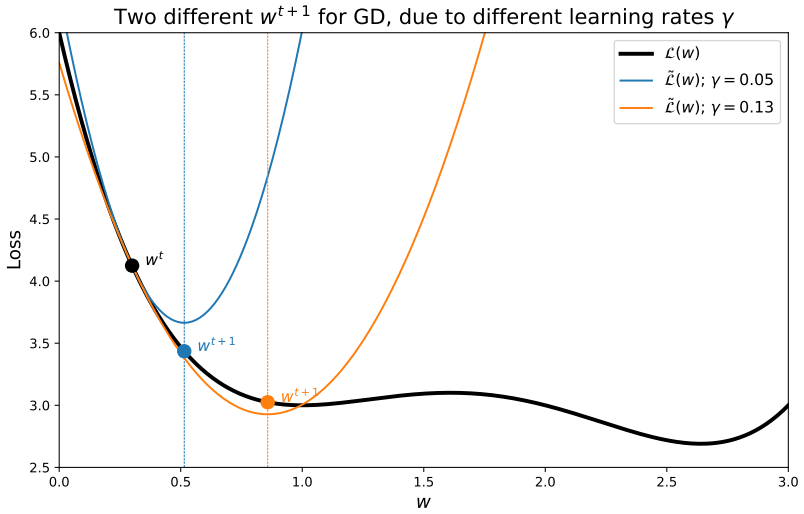

Figure 8: We plot an example of the function $\mathscr{L}(w)$ (in black), together with two approximations $\tilde{\mathscr{L}}$ (in blue and orange) around the point w^t (black dot). The two approximations are obtained setting $\gamma = 0.05, 0.13$ respectively for the blue and orange curves. Notice how changing γ affects the next iterate w^{t+1} of the gradient descent algorithm.

Global minima. Even if you choose an appropriate learning rate, there is absolutely no guarantee that you will reach a global minimum using gradient descent. While a small enough step size guarantees us to get close to a local minimum or a saddle point, nothing assures us that this will be a global minimum. However, many optimization problems have no known computationally efficient algorithm that is guaranteed to find the global minimizer in a reasonable time.

Practical Use. But if we cannot guarantee that we reach the global minimum, why is gradient descent still so popular and ubiquitous? Most often, it justified by satisfactory practical performance. The algorithm is also very easy to implement and relies only on one hyperparameter that needs to be tuned (via validation data), the step size γ. And not only is the algorithm easy to implement, it is also running fast on optimized computer hardware like GPUs. You can run it for functions $\mathscr{L}$ which depend on millions of parameters. Even if the function $\mathscr{L}$ does not admit a derivative in a discrete set of points (e.g. the absolute value function), in practice we can get around this issue by using the so-called pseudo-gradient. In the pseudo-gradient one sets the gradient at a non-differentiable place to the gradient one would obtain by getting close to that point from either side.

Early stopping. We already saw that a reasonable criterion for stopping the algorithm is when we do not move much between two time steps [6], i.e. when

$$\left\|\vec{w}^{t+1} - \vec{w}^t\right\|^2 = \sum_i (w_i^{t+1} - w_i^t)^2 < \epsilon\,. \tag{5.6}$$

Recall that it might be that the loss function $\mathscr{L}$ is defined in terms of the data $\vec{X}_{\text{train}}$. If we stop the GD algorithm when we are at the minimum of the loss, defined in terms of the training data, we might be overfitting (the final parameters will have a good performance for the training data, but not for new data). In Figure 9 you can see how the training and validation errors behave as a function of the total time for which we ran gradient descent. We can see that overfitting is happening, as the validation error is suddenly rising again as a function of time after reaching a mininum. After the minumum, running gradient descent further makes the prediction performance as measured by the validation error deteriorate, even though the performance on the training set could still be improving (as it does in Figure 9). To mitigate this problem, there is a simple solution: we stop gradient descent when the validation error starts increasing again. This paradigm is called early stopping and is widely used in practice.

[6] Notice this is equivalent to condition (4.15). In fact, assuming that (4.15) holds, we have $\left\|\vec{w}^{t+1} - \vec{w}^t\right\|^2 = \gamma^2 \left\|\frac{\partial \mathscr{L}}{\partial \vec{w}}(w^t)\right\|^2 \leq \gamma^2 \epsilon$, therefore (5.6) holds with $\epsilon\gamma^2$ in the right hand side.

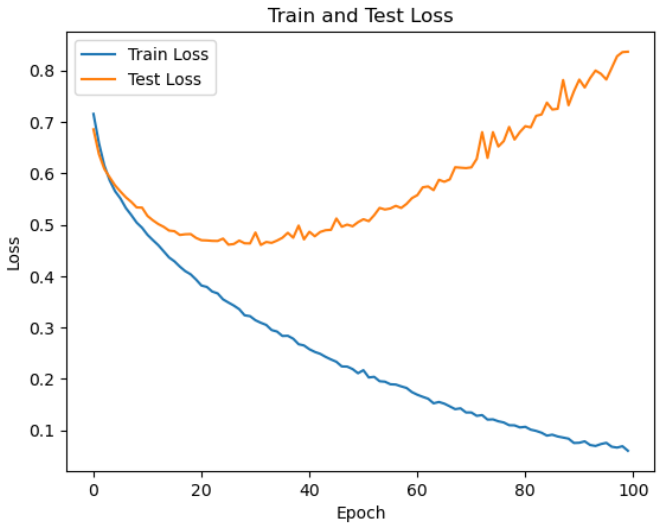

Figure 9: Illustration of the early stopping behavior. The x-axis denotes the iteration time.

Implicit regularization. It is informative to reflect what gradient descent leads to when used on the least squares objective that we discussed in the context of linear regression:

$$\mathscr{L}(\vec{w}) = \frac{1}{n}\sum_{\mu=1}^{n}\left(y_\mu - \sum_{i=1}^{d} X_{\mu i} w_i\right)^2 \tag{5.7}$$

We can illustrate here an important property of gradient descent. In particular, we focus on the high-dimensional regime where $n < d$, so the number of samples is smaller than the number of dimensions, and many equally optimal minimizers $\vec{w}$ exists for the function $\mathscr{L}$. Recall that the explicit solution was that the minimum is at $\hat{\vec{w}} = (X^T X)^{-1} X^T y$ when the matrix $X^T X$ is invertible. In the high-dimensional case the covariance matrix $X^T X$ is not invertible, and we introduced the square regularization with strength λ and instead inverted $X^T X + \lambda \mathbb{I}$, which is always invertible for any $\lambda > 0$.

We now state the property of interest, which is called implicit regularization. Intuitively, we are saying that even though gradient descent minimizes a loss function that is not explicitly regularized, it ends up in a solution that we would get with regularization. SPecifically one can show that for the square loss (5.7) when we initialize gradient descent from $\vec{w}^{t=0} = 0$ and fix the learning rate γ small, the algorithm will eventually arrive at

$$\vec{w}_{\rm GD} = \lim_{\lambda \to 0^+} (X^T X + \lambda \mathbb{I})^{-1} X^T y\,. \tag{5.8}$$

The implicit regularization property can be justified mathematically as follows[7]: We look at what the derivative in the gradient descent update looks like for the least squares,

$$\frac{\partial \mathscr{L}(\vec{w})}{\partial w_j} = -\frac{2}{n}\sum_{\mu=1}^{n}\left(y_\mu - \sum_{i=1}^{d} X_{\mu i} w_i\right) X_{\mu j} = -\frac{2}{n}\sum_{\mu=1}^{n}\left(y_\mu - \vec{X}_\mu \cdot \vec{w}\right) X_{\mu j}\,. \tag{5.9}$$

Since $\vec{X}_\mu \cdot \vec{w}$ is a scalar, we see that the gradient is a linear combination of the data samples $\vec{X}_\mu$. By completing the set of data samples to a (possibly overcomplete) basis, we realize that any vector can be written as linear combination of the data samples plus a vector that is orthogonal on the subspace spanned by the data-samples,

$$\vec{w}^t = \sum_{\mu=1}^{n} \alpha_\mu^t \vec{X}_\mu + \vec{\nu}^t \quad \text{with} \;\; \vec{X}_\mu \cdot \vec{\nu}^t = 0 \;\; \forall \mu = 1, \dots, n\,. \tag{5.10}$$

for some set of coefficients α_μ^t. If we initialize $\vec{w}^{t=0} = 0$, implying $\vec{\nu}^{t=0} = 0$, then $\vec{\nu}^t$ will remain equal to zero during all the iterations of gradient descent, $\vec{\nu}^t = \vec{\nu}^{t+1} = 0$. This is because the gradient and hence the GD updates are linear combinations of the data-samples and thus can only alter the coefficients α_μ^t.

[7] This justification will not be required at the exam, we present it here for completness.

Thus, gradient descent is effectively minimizing $\mathscr{L}$ in a restricted linear subspace of dimension equal to the dimension of the span of the data sampled $\vec{X}^\mu$, reducing the dimension of the domain from d to n. In the restricted subspace, the minimum of the least-square objective is unique. Thus, if the step size of gradient descent is small enough, gradient descent will converge to such unique global minimum, which we call $\vec{w}_{\text{GD}}$.

Then, we notice that all the minima of the original problem can be written as

$$\vec{w}_{\text{minimum}} = \vec{w}_{\text{GD}} + \vec{\nu} \tag{5.11}$$

for some $\vec{\nu}$ in the subspace orthogonal to the data samples, as shifting $\vec{w}_{\text{GD}}$ by an $\vec{\nu}$ orthogonal to the data samples in (5.7) does not alter the loss value. Thus, we see that among all minima of the original loss, gradient descent converges to the one with minimum norm, as by orthogonality

$$\|\vec{w}_{\text{minimum}}\|^2 = \|\vec{w}_{\text{GD}}\|^2 + \|\vec{\nu}\|^2 \geq \|\vec{w}_{\text{GD}}\|^2 . \tag{5.12}$$

Finally, we recall that the regularization λ is biasing the original ill-defined least-square problem towards $\vec{w}$ with small norm, so that for $\lambda \to 0^+$ the regularized solution will converge to the minimal norm solution of the ill-defined least-squares problem, justifying (5.8).

Stochastic gradient descent. Note that so far, at every time step, we use all the data points in $\vec{X}_{\text{train}}$ to evaluate the loss function $\mathscr{L}$. However, in practice (e.g. when the dataset is a large number of images), computing the gradient becomes computationally expensive. A solution is to use a random subset of the data to evaluate the gradient at the current parameter values $\vec{w}^t$. Gradient descent with this random subsampling per update step is called *Stochastic gradient descent* (SGD). Since there are fewer terms in the sum over the data, this is more efficient to compute. At the same time, if we still use enough samples, the approximation to the true gradient evaluated over the complete training set is good enough to get a satisfactory performance on the validation set.

5.3 Use cases for GD

5.3.1 Logistic regression

We consider the classification task where we have an image $\vec{X}_\mu \in \mathbb{R}^d$ and a label $y_\mu \in \{\pm 1\}$. We consider the regularized least squares

$$\mathscr{L}(\vec{w}) = \frac{1}{n}\sum_{\mu=1}^{n}\left(y_\mu - \vec{X}_\mu \cdot \vec{w}\right)^2 + \frac{\lambda}{n}\sum_{i=1}^{d} w_i^2 . \tag{5.13}$$

We are looking for $\hat{\vec{w}} = \arg\min_{\vec{w}} \mathscr{L}(\vec{w})$, as usual. The final prediction will be generated as $y_{\text{new}} = \text{sign}(\vec{X}_{\text{new}} \cdot \hat{\vec{w}})$. This separation between the positive $(+1)$ and negative class (-1) can be interpreted as a hyperplane in the input space, as shown in Figure 10. We want for every datapoint to lie on the side of the hyperplane that corresponds to

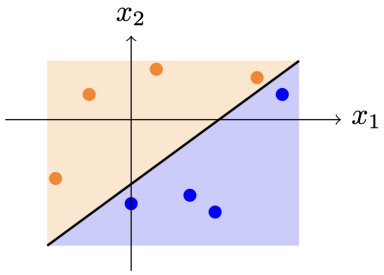

Figure 10: A binary classification problem between datapoints that belong to the blue or orange class. A hyperplane is separating the regions that classify a datapoint to be either orange or blue.

their class. That is, written differently, the classification happens according to the following rule

$$y_\mu(\vec{X}_\mu \cdot \hat{\vec{w}}) > 0 \Rightarrow \text{correct classification} \tag{5.14}$$

$$y_\mu(\vec{X}_\mu \cdot \hat{\vec{w}}) < 0 \Rightarrow \text{wrong classification} \tag{5.15}$$

Now let us inspect the loss function $\mathscr{L}$ we are optimizing more closely. Does it really do what we want it to do, i.e., gives us good values of $\vec{w}$ for classification? We can rewrite the loss $\mathscr{L}$ in an equivalent form (that will also be handy in the next section)

$$\mathscr{L}(\vec{w}) = \frac{1}{n}\sum_{\mu=1}^{n} \ell(y_\mu, z_\mu) + \frac{\lambda}{n}\sum_{i=1}^{d} w_i^2\,, \tag{5.16}$$

$$z_\mu = \sum_{i=1}^{d} X_{\mu i} w_i\,. \tag{5.17}$$

Then, the loss we have been using so far was the square loss $\ell(y,z) = (y-z)^2$. If we have $y \in \{\pm 1\}$ we can reformulate this as $\ell(y \in \{\pm 1\}, z) = (1-yz)^2$. This is the function plotted in Figure 11. Overall, a reasonable loss function should penalize weights that classify a datapoint incorrectly with a high loss value, i.e. (5.15), is fulfilled ($yz < 0$ in Figure 11). At the same time, datapoint that are classified correctly should have a low loss penalty, i.e. (5.14) and $yz > 0$ in Figure 11. Let us see what the squared loss function gives us: It nicely penalizes the side $yz < 0$, but because it has a minimum at $yz = 1$, it actually gives a high penalty also to correctly classified points. The most immediate way to fix this would be to give a constant penalty δ to the wrongly classified datapoints and a value of zero to those that are correct. However, the gradient of this function has a value of zero almost everywhere. This poses a problem for GD, as the second term in the update step would always be zero – and we would not be updating the parameters at all. Therefore, we need an intermediate smooth solution, and what is usually used in the case of binary classification is $\ell(yz) = \log(1+e^{-yz})$. As $yz \to -\infty$ the value becomes infinitely large. As $yz \to +\infty$ is becomes closer to zero. The model that uses this loss for binary classification is called *Logistic Regression*.

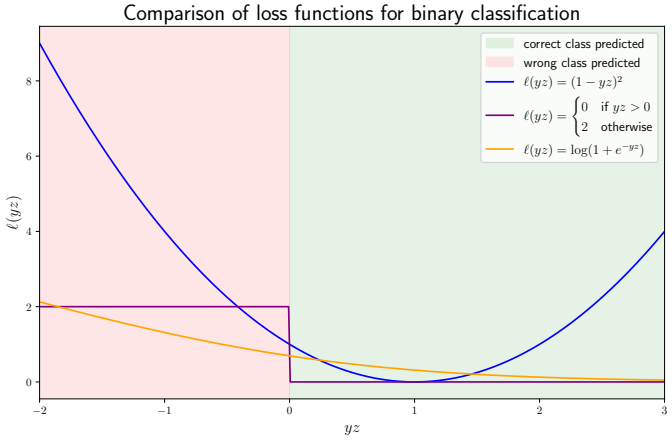

Figure 11: Comparison of different loss functions for binary classification. The yellow loss is most suitable.

5.3.2 Deep/multi-layer Neural Networks

At this point, we have widened our understanding of gradient descent. We introduced a new loss function that accommodates binary classification nicely. We can generalize $\mathscr{L}$ from (5.16) further, to be used with more complicated functions and parameters beyond what we saw so far, but nonetheless we can still optimize the resulting function with gradient descent. In its general form, the loss reads

$$\mathscr{L}(\mathsf{W}) = \frac{1}{n}\sum_{\mu=1}^{n} \ell\left(y_\mu, z(X_\mu, \mathsf{W})\right) \tag{5.18}$$

For the case of linear regression, we have $\mathsf{W} = \{\vec{w}\}$, $\vec{w} \in \mathbb{R}^d$ and $z(\vec{X}_\mu, \mathsf{W}) = \vec{X}_\mu \cdot \vec{w}$.

The most popular more complex version of functions z are *neural networks* (NN), with one or several layers, sometimes called *multilayer perceptron*. A neural network with a single layer is simply linear regression. A two-layer

NN with a hidden layer size of p is defined for a $\mathsf{W} = \{\vec{w}, W^{(1)}\}$ where $\vec{w} \in \mathbb{R}^p, W^{(1)} \in \mathbb{R}^{p\times d}$, so that the function

$$z(X_\mu, \mathsf{W}) = \sum_{a=1}^{p} w_a \psi^{(1)} \left(\sum_{i=1}^{d} W^{(1)}_{ai} X_{i\mu} \right) = \vec{w}\psi^{(1)} \left(W^{(1)} \vec{X}_\mu \right) \tag{5.19}$$

is well defined for a non-linear function $\psi^{(1)}(a)$, also called *activation function*. A common choice in neural networks is the following piece-wise linear function

$$\mathrm{ReLU}(x) = \begin{cases} x, & \text{if } x > 0 \\ 0, & \text{otherwise} \end{cases}, \tag{5.20}$$

but also the sigmoid or tanh functions appear in some contexts among many other choices. Note that when the activation function is applied to a vector we mean that it is applied element-wise, i.e. $[\psi^{(1)}(\vec{a})]_i = \psi^{(1)}(a_i)$.

Let us reflect a moment about what changes about the gradient descent algorithm when we apply it to $\mathscr{L}(\mathsf{W})$ for two layer neural network instead of the linear regression before. First of all, we have more parameters to update, specifically $p + pd$, and we need to calculate the same amount of derivatives in each gradient step. In addition, the function we are deriving is slightly more complicated, but you should be able to see that it is not that difficult – you could even do it by hand (note that for the ReLU you need to use the pseudo-gradient though). Apart from that, the general idea stays the same: We choose an initial W^0, we update every single parameter using a gradient step to obtain a new W^{t+1}, and then keep iterating until we either converge on the training data or the early stopping criterion kicks in.

Now, we can even go to more than two layers in the neural network. For a L-layer NN, we just keep iterating on our previous definition. We take $\mathsf{W} = \{\vec{w}, W^{(L-1)}, W^{(L-2)}, \cdots, W^{(1)}\}$ where we have the dimensions $p_L = 1, p_0 = d$ and we can set the other p_ℓ to arbitrary integers and take the matrices $W^{(\ell)\in R^{p_\ell \times p_{\ell-1}}}$. The function to obtain the classification is then the multilayer perceptron

$$z(X_\mu, \mathsf{W}) = \vec{w}\psi^{(L-1)} \left(W^{(L-1)} \psi^{(L-2)} \left(W^{(L-2)} \cdots \psi^{(1)} \left(W^{(1)} \vec{X}_\mu \right) \cdots \right) \right) . \tag{5.21}$$

Because this function may be difficult to imagine, a common way to visualize the way these matrices are applied one by one is shown in Figure 12.

In the context of machine learning the choice of the function z is usually referred to as the *architecture* of the neural network, and you can imagine that one can be even more creative in defining this function class. While the architectures for modern applications like ChatGPT are more complicated than what you saw here, at their basis they are very similar to the multilayer perceptron we just defined, and they are optimized using flavours of gradient descent.

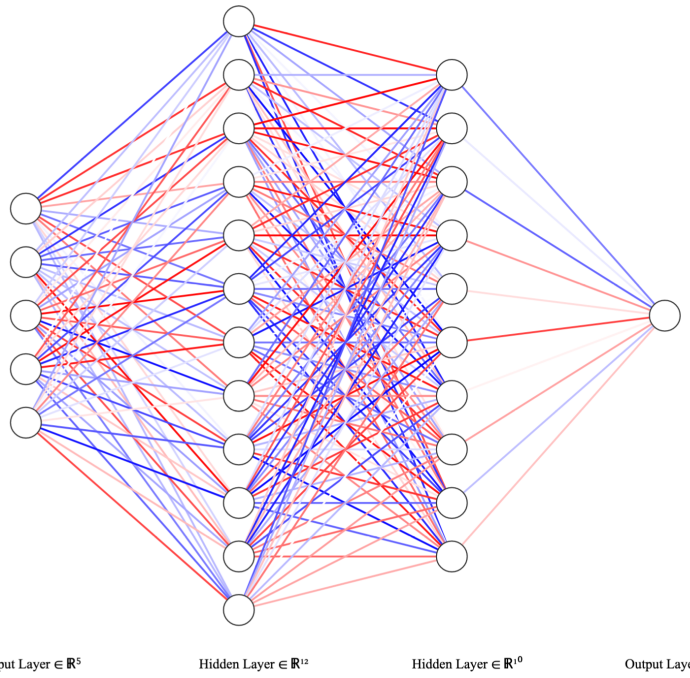

Figure 12: Visual example of a neural net with three layers (2 hidden layers), $L = 3$. The dimensions and widths are $d = 5$, $p_1 = 12$, $p_2 = 10$. Each edge represents a single parameter, the colors are according to a random value at which this network is initialized.

PART II

Quantifying Uncertainty

We introduce probability and statistics as the language for uncertainty: generating random variables, propagation of errors, law of large numbers, central limit theorem, maximum likelihood; regression revisited with weighting and Bayesian regularization for principled error bars and inference in experiments.

6 Lecture 6: Generation of random variables and propagation of uncertainty I

In this lecture, we show in the first part how to draw random variables on a computer, starting from uniform random numbers to more general laws; in the second part we start to study how experimental errors propagate and affect measured values.

6.1 Generating one-dimensional random variables

We recall that the probability density function (p.d.f.) of a continuous random variable X is the function ρ such that

$$\rho(x)\mathrm{d}x = \mathrm{Proba}(X \in [x, x+\mathrm{d}x]) \ . \tag{6.1}$$

$\rho(x)$ is proportional to the probability that the random variable X takes value around x.

Example 2. *Let X be uniformly distributed over $[0,1]$ and ρ_{unif} its p.d.f. We have*

$$\rho_{\mathrm{unif}}(x) = \begin{cases} 0 & x < 0 \\ 1 & 0 \leq x \leq 1 \\ 0 & x > 1 \end{cases} \ . \tag{6.2}$$

It is depicted in fig. 13.

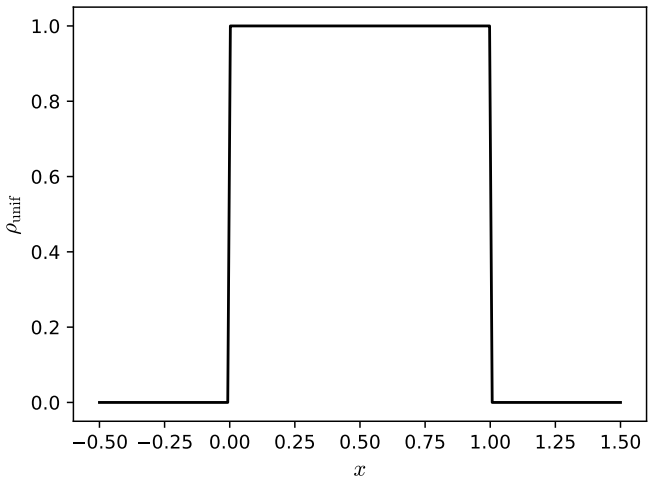

Figure 13: Probability density function ρ_{unif} of the uniform law between 0 and 1.

We recall some properties of p.d.f.s:

- a p.d.f. is normalized : $\int_{\mathbb{R}} \mathrm{d}x\, \rho(x) = 1$. Informally, the probability that the random variable X has any value equals 1.
- A p.d.f. is non-negative : $\rho(x) \geq 0$. Notice that as $\rho(x)$ is only proportional to a probability, it can be greater than 1.

Generating uniform random variables in the unit interval $[0, 1]$**.** We want to generate random numbers uniformly over $[0, 1]$. If we had a balanced coin we could draw 0s and 1s and assemble them to obtain the binary representation of a number e.g. $x = 0.11101\ldots$ More generally if we had an unbiased random process over k values we could obtain a number in base k. How can we simulate tossing a coin on a computer?

Most often in practice, we rely on **pseudo-random number generators**. These are deterministic sequences of variables that seem random.

Example 3. *The linear congruential generator (LCG) is a pseudo-random number generator. Given a seed* i_0 *it computes a sequence of numbers defined by*

$$i_{t+1} = (n\, i_t + k) \bmod m \tag{6.3}$$

and outputs the most significant bits of the i_t*. mod is the modulo and* n*,* k *and* m *are well-chosen integer numbers. An example is* $m = 2^{31}$*,* $n = 1103515245$ *and* $k = 12345$ *(used e.g. in the library glibc). It is periodic of period at most* m*. LCGa are fast and require little memory.*

A good pseudo-random number generator must have a long period and statistics close to the one of truly random sequences. The numbers it outputs must not be correlated. If a program needs good randomness it can be important to check if its results do not depend on the specificities of a random generator, up to the random fluctuations. The LCG is a simple examples of a pseudo-random number generator. Current libraries usually deploy generalizations of the idea behind LCG; for example, in NumPy the Mersenne's twister is used by default. It has a more sophisticated recursive relation based on matrices, and m is a large Mersenne's prime (i.e. there is p such that a prime $m = 2^p - 1$; this allows efficient computation.)

Pseudo-random number generators allow to reproduce the same sequence of numbers from the same seed i_0, this is important for reproducibility of simulations and computations that rely on random numbers. Indeed the entire sequence of numbers is determined by the seed i_0. It is enough and in practice important to save the seed to be able to generate the same numbers.

Sampling from generic distributions We have now access to numbers drawn uniformly on $[0, 1]$. How can we draw numbers that have an arbitrary p.d.f. ρ? We start by recalling that the cumulative function of a random variable X with p.d.f. ρ is the function F such that

$$F(x) = \mathrm{Proba}(X \leq x) = \int_{-\infty}^{x} \mathrm{d}y\, \rho(y)\ . \tag{6.4}$$

You can see an example of the cumulative function in fig. 14.

Example 4. *Let* X *be uniformly distributed over* $[0, 1]$ *and* F_{unif} *its cumulative function. We have*

$$F_{\mathrm{unif}}(x) = \begin{cases} 0 & x < 0 \\ x & 0 \leq x \leq 1 \\ 1 & x > 1 \end{cases}\ . \tag{6.5}$$

We recall some properties of cumulative functions:

- a cumulative function's output is bounded in $[0, 1]$: $F(-\infty) = 0$ and $F(+\infty) = 1$. This follows by the normalization of the associated p.d.f.
- A cumulative function is non-decreasing. This follows from the non-negativity of the associated p.d.f.

Theorem 5. *Let* F *be the cumulative function of the random variable we want to sample. Assume that* F *is invertible (and thus strictly increasing). Let* Y *be a uniform random variable on* $[0, 1]$*. We define the random variable* $X = F^{-1}(Y)$ *and* F_X *its cumulative function. Then* $F_X = F$*.*

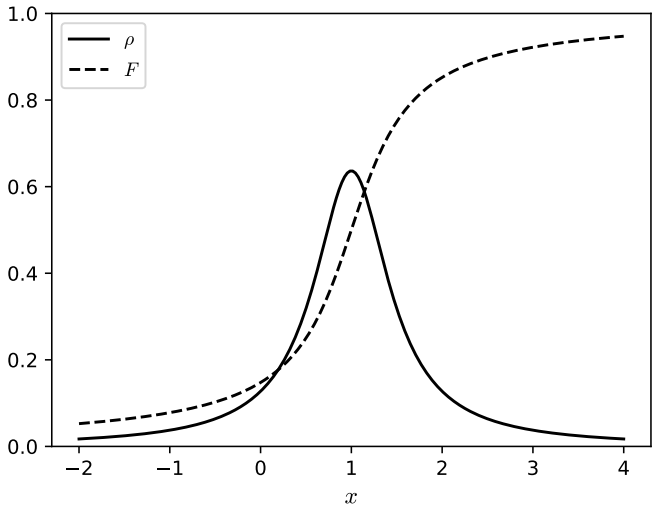

Figure 14: Probability density function ρ and cumulative function F of a Cauchy law $\rho(x) = 2/\pi((2x-2)^2+1)$.

This gives a way to generate random numbers X according to a p.d.f. ρ. One has to compute first the cumulative F, to generate uniform random numbers Y and compute $X = F^{-1}(Y)$.

Proof of the theorem :

$$\begin{aligned} F_X(x) &= \text{Proba}(X \leq x) && \text{by definition of } F_X && (6.6)\\ &= \text{Proba}(F^{-1}(Y) \leq x) && \text{by definition of } X && (6.7)\\ &= \text{Proba}(Y \leq F(x)) && \text{as } F \text{ is strictly increasing} && (6.8)\\ &= F_{\text{unif}}(F(x)) && \text{by definition of } Y && (6.9)\\ &= F(x) && && (6.10) \end{aligned}$$

Notice that F is invertible is equivalent to F is strictly increasing; it does not have plateaux.

Example 5. *Generating random numbers from the exponential distribution. The exponential distribution (useful in physics e.g. for modeling the decay of particles) is defined as*

$$\rho_{\exp}(x) = \begin{cases} 0 & x < 0 \\ e^{-x} & x \geq 0 \end{cases} . \tag{6.11}$$

We compute its c.d.f. and the inverse c.d.f.

$$F_{\exp}(x) = \begin{cases} 0 & x < 0 \\ 1 - e^{-x} & x \geq 0 \end{cases} \quad \text{and} \quad F^{-1}_{\exp}(y) = -\ln(1-y) . \tag{6.12}$$

Then we can draw Y uniform on $[0,1]$ and consider $X = F^{-1}_{\exp}(X)$. It will be exponentially distributed.

Example 6. *Generating random numbers from the normal distribution. The normal (or Gaussian) distribution is defined as*

$$\rho_{\text{normal}}(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x-\bar{x})^2/2\sigma^2} , \tag{6.13}$$

for $\bar{x} \in \mathbb{R}$ and $\sigma \in \mathbb{R}_+$. $\bar{x}$ is the mean, and σ the standard deviation. You can see on fig. 15 the shape of the curve. For $\bar{x} = 0$ and $\sigma = 1$ we have

$$F_{\text{normal}}(x) = \frac{1}{2}\text{erfc}\left(-\frac{x}{\sqrt{2}}\right) \quad \text{and} \quad F^{-1}_{\text{normal}}(y) = -\sqrt{2}\text{erfc}^{-1}(2y) \tag{6.14}$$

where erfc is the complementary error function. It does not admit an expression in term of elementary functions, nor does its inverse.

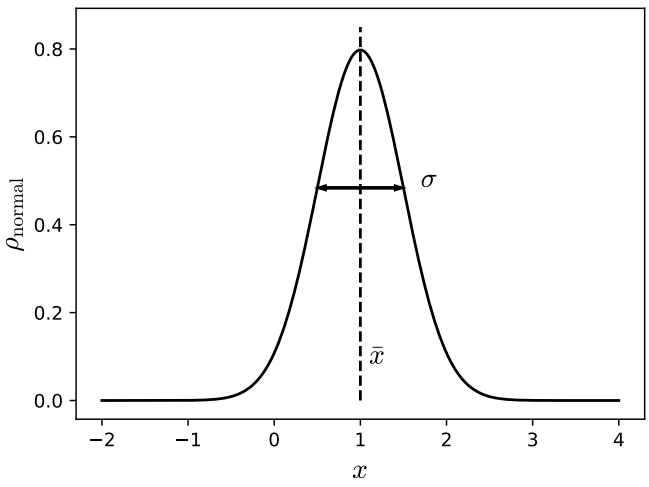

Figure 15: Probability density function ρ_{normal} of a normal law with $\bar{x} = 1$ and $\sigma = 1/2$.

If we want to generate numbers from a normal distribution there is an alternative way used more commonly than using directly the inverse of the cumulative function. In order to derive this method, we start by computing the normalization of the normal p.d.f. using a standard trick. We want to compute

$$I = \int_{-\infty}^{+\infty} \mathrm{d}x\, e^{-x^2/2} = \sqrt{2\pi} \tag{6.15}$$

$$\tag{6.16}$$

Then, we consider and compute I^2:

$$I^2 = \int_{-\infty}^{+\infty} \mathrm{d}x \int_{-\infty}^{+\infty} \mathrm{d}y\, e^{-(x^2+y^2)/2} \tag{6.17}$$

$$= \int_0^{2\pi} \mathrm{d}\theta \int_0^{+\infty} \mathrm{d}r\, r\, e^{-r^2/2} \tag{6.18}$$

$$= \int_0^{2\pi} \mathrm{d}\theta \int_0^{+\infty} \mathrm{d}\gamma\, e^{-\gamma} \tag{6.19}$$

$$= 2\pi \tag{6.20}$$

where we changed the variables according to $x = r\cos\theta$, $y = r\sin\theta$ and $\gamma = r^2/2$. We see here that θ is uniformly distributed over $[0, 2\pi]$ and γ follows an exponential law. We can easily sample them, as we saw before, and then invert the relation to compute $x = \sqrt{2\gamma}\cos\theta$ and $y = \sqrt{2\gamma}\sin\theta$, which will be two independent standard normals.

At the end of this argument we conclude that if we generate two random number u_1 and u_2 independently from the uniform distribution on $[0, 1]$ we can transform them into two independent normally distributed numbers x_1 and x_2 as

$$x_1 = \sqrt{-2\ln u_1}\cos(2\pi u_2), \tag{6.21}$$

$$x_2 = \sqrt{-2\ln u_1}\sin(2\pi u_2). \tag{6.22}$$

$$\tag{6.23}$$

6.2 Propagation of uncertainties

In physics we often model measurements by random variables and evaluate functions of those random variables. How does the uncertainty in the measurements reflect on the uncertainty of their function? We will derive here formulas that are often used in physics and be explicit about the assumptions underlying them.

Recall that the mean of a random variable X with p.d.f. ρ is

$$\mathbb{E}[X] = \int \mathrm{d}x\, \rho(x) x \ . \tag{6.24}$$

The variance of X is

$$\mathrm{Var}[X] = \int \mathrm{d}x\, \rho(x)(x - \mathbb{E}[X])^2 = \mathbb{E}[X^2] - \mathbb{E}[X]^2 \ . \tag{6.25}$$

The standard deviation of X is

$$\sigma_X = \sqrt{\mathrm{Var}[X]} \ . \tag{6.26}$$

An illustration is given on fig. 16.

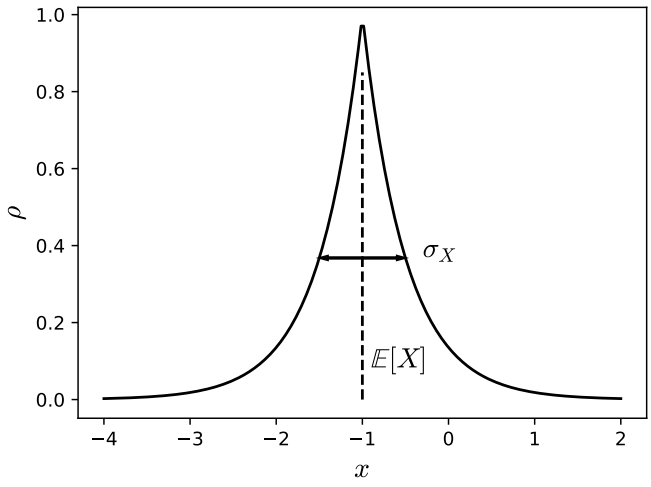

Figure 16: The mean and standard deviation of a Laplace law $\rho(x) = e^{-2|x+1|}$. $\mathbb{E}[X] = -1$ and $\sigma_X = 1/2$

We want to measure a quantity $G(X_1, \ldots, X_k)$ that depends on k scalar measurements; for instance, the surface of a table that depends on its width and its length. The measurements X_i have some uncertainty, some error, modeled by their variances; what is the error on G? Also, what is the average value of G?

For the moment, we imagine that we know the distribution of the measurements $X_1, \ldots, X_k$, i.e. we know their p.d.f., their mean and variances. We will come back later in the course to the problem of estimating the p.d.f., or for example the mean and the variance, from a finite number of samples of each measurement.

6.2.1 Uncertainty of a linear combination of random variables

The distribution of the different measurements can be correlated; so we model the k measurements by their joint p.d.f. $\rho(x_1, \ldots, x_k)$. We want to compute the mean and the variance of $G(X_1, \ldots, X_k)$. We consider first a linear case where $G(X_1, \ldots, X_k) = \sum_{i=1}^k a_i X_i$ where a_i are some real coefficients. The meaning of G could be for instance the perimeter of the table in the example above. We have:

$$\mathbb{E}[G(X_1, \ldots, X_k)] = \mathbb{E}\left[\sum_{i=1}^k a_i X_i\right] = \sum_{i=1}^k a_i \mathbb{E}[X_i] \tag{6.27}$$

where we used the linearity of the mean. We have that the mean is the linear combination of the means. As to the variance we compute first $\mathbb{E}[G^2]$ and $\mathbb{E}[G]^2$:

$$\mathbb{E}[G(X_1, \ldots, X_k)^2] = \mathbb{E}\left[\sum_{i,j} a_i a_j X_i X_j\right] \tag{6.28}$$

$$= \sum_{i,j} a_i a_j \mathbb{E}[X_i X_j] \tag{6.29}$$

$$= \sum_i a_i^2 \mathbb{E}[X_i^2] + 2 \sum_{i<j} a_i a_j \mathbb{E}[X_i X_j] \,, \tag{6.30}$$

where again we used the linearity of the mean, and we decomposed the sum by highlighting the terms in which $i = j$ and $i < j$, and noticing that the terms $i > j$ give a contribution which is equal to that given by the $i < j$ terms.

Then

$$\mathbb{E}[G(X_1,\ldots,X_k)]^2 = \left(\sum_i a_i \mathbb{E}[X_i]\right)^2 \tag{6.31}$$

$$= \sum_{i,j} a_i a_j \mathbb{E}[X_i]\mathbb{E}[X_j] \tag{6.32}$$

$$= \sum_i a_i^2 \mathbb{E}[X_i]^2 + 2\sum_{i<j} a_i a_j \mathbb{E}[X_i]\mathbb{E}[X_j]\,, \tag{6.33}$$

so that for the variance we obtain

$$\mathrm{Var}[G(X_1,\ldots,X_k)] = \sum_i a_i^2 \mathrm{Var}[X_i] + 2\sum_{i<j} a_i a_j \mathrm{Cov}[X_i,X_j] \tag{6.34}$$

where we defined the covariance between two random variables as

$$\mathrm{Cov}[X_i,X_j] = \mathbb{E}[X_i X_j] - \mathbb{E}[X_i]\mathbb{E}[X_j]\,. \tag{6.35}$$

7 Lecture 7: Propagation of uncertainty II

In this lecture we continue the study how experimental errors (or statistical uncertainty) propagates and affects measured values. More precisely, we derive the formulas for error propagation that you use during your laboratory work, and justify them. We will then concentrate on a specific example of a mean of iid random variables and discuss the law of large numbers.

7.1 Reminder of probability density functions and joint probability density functions

Recall that for a random variable X we can define its *probability density function* (p.d.f.) $\rho_X(x)$ as

$$\mathrm{Proba}(a \le X \le b) = \int_a^b \rho_X(x)dx. \tag{7.1}$$

The probability density function is also called *probability distribution* or simply *distribution*. Intuitively, you can say that $\rho_X(x)dx$ is the probability that X takes value in the interval $[x, x+dx]$.

Consider now k real random variables $X_i \in \mathbb{R}$, $i = 1, ..., k$. Similarly as in the case with one variable, we define the *joint probability density function* (joint p.d.f.) $\rho(x_1, x_2, ..., x_k)$ as

$$\mathrm{Proba}(a_1 \le X_1 \le b_1, a_2 \le X_2 \le b_2, ..., a_k \le X_k \le b_k) = \int_{a_1}^{b_1}\int_{a_2}^{b_2} ... \int_{a_k}^{b_k} \rho(x_1, x_2, ..., x_k)dx_1dx_2...dx_k. \tag{7.2}$$

The joint p.d.f. is also called *joint probability distribution* or simply *joint distribution*. The *mean*, also called *average* or *expectation value*, of one variable X_i is given by

$$\mathbb{E}(X_i) = \int ... \int x_i \rho(x_1, x_2, ..., x_k)dx_1dx_2...dx_k. \tag{7.3}$$

Similarly, the mean of the product of two random variables X_i and X_j is given by

$$\mathbb{E}(X_iX_j) = \int ... \int x_ix_j \rho(x_1, x_2, ..., x_k)dx_1dx_2...dx_k. \tag{7.4}$$

In general, the mean of some combination of the random variables $f(X_1, X_2, ..., X_k)$ is

$$\mathbb{E}(f(X_1, X_2, ..., X_k)) = \int ... \int f(x_1, x_2, ..., x_k)\rho(x_1, x_2, ..., x_k)dx_1dx_2...dx_k. \tag{7.5}$$

Recall that we define the *variance* of one random variable X_i as

$$\mathrm{Var}(X_i) = \mathbb{E}(X_i^2) - [\mathbb{E}(X_i)]^2 = \mathbb{E}\left[(X_i - \mathbb{E}(X_i))^2\right]. \tag{7.6}$$

Since the variance is the mean of a quantity squared, it is always non-negative.

For the rest of this lecture we will suppose that the mean and the variance of all the considered distributions are finite. We often also say that the mean and variance *exist*. We will discuss the cases where this does not hold in future lectures.

7.2 Linear combination of k random variables

We recall that at the end of lecture 6 we considered the linear combination of random variables (which is also a random variable)

$$G(X_1, X_2, ..., X_k) = \sum_{i=1}^{k} a_i X_i \tag{7.7}$$

where the a_i are deterministic (i.e. non random) known coefficient. Recall also that we consider the p.d.f. $\rho(x_1, x_2, ..., x_k)$ known. Using the linearity of the expectation value, the mean of G is

$$\mathbb{E}(G(X_1, X_2, ..., X_k)) = \sum_{i=1}^{k} a_i \mathbb{E}(X_i). \tag{7.8}$$

We also saw that the variance of G is given by

$$\mathrm{Var}(G(X_1, X_2, ..., X_k)) = \sum_{i=1}^{k} a_i^2 \left(\mathbb{E}(X_i^2) - [\mathbb{E}(X_i)]^2 \right) + \sum_{i \neq j}^{k} a_i a_j \left(\mathbb{E}(X_i X_j) - \mathbb{E}(X_i)\mathbb{E}(X_j) \right). \tag{7.9}$$

Recall from the properties of the variance that $\mathrm{Var}(G(X_1, X_2, ..., X_k)) \geq 0$.

7.2.1 Independant random variables

In this section we consider the special case where the variables $X_1, X_2, ..., X_k$ are independent. The variables $X_1, X_2, ..., X_k$ are *independent* iff the joint p.d.f. factorizes:

$$\rho(x_1, x_2, ..., x_k) = \rho_1(x_1)\rho_2(x_2)...\rho_k(x_k) \tag{7.10}$$

where $\rho_i(x_i)$ is the p.d.f. of the random variable X_i. Then, we have that the mean of the product of two random variables X_i, X_j simplifies as

$$\mathbb{E}(X_i X_j) = \int ... \int x_i x_j \rho(x_1, x_2, ..., x_k) dx_1 dx_2 ... dx_k = \left(\int x_i \rho_i(x_i) dx_i \right) \left(\int x_j \rho_j(x_j) dx_j \right) = \mathbb{E}(X_i)\mathbb{E}(X_j) \tag{7.11}$$

where we used the fact that $\int \rho_i(x_i) dx_i = 1$. Combining this result with the expression of the variance (7.9), we see that the second sum is 0 so we obtain

$$\mathrm{Var}(G) = \sum_{i=1}^{k} a_i^2 \mathrm{Var}(X_i). \tag{7.12}$$

This means that for independent random variables, the variances are additive.

Recall that we defined the *standard deviation* σ_G as the square root of the variance

$$\sigma_G = \sqrt{\mathrm{Var}(G)}. \tag{7.13}$$

The standard deviation is often used to indicate the uncertainty of measured data, since it has the same units as the measured data. Expression (7.12) expressed in terms of standard deviations gives

$$\sigma_G^2 = \sum_{i=1}^{k} a_i^2 \sigma_{X_i}^2. \tag{7.14}$$

Example 7. *Consider a key example where the variables X_i are independent and have the same p.d.f., i.e. $\rho_1(x_1) = \rho_2(x_2) = ... = \rho_k(x_k)$. In that case we say that the variable are independent and identically distributed, often abbreviated i.i.d.. Suppose also that $a_i = \frac{1}{k} \forall i$. Then the mean of G is given by*

$$\mathbb{E}(G) = \frac{1}{k} \sum_{i=1}^{k} \mathbb{E}(X) = \mathbb{E}(X) \tag{7.15}$$

were we define $\mathbb{E}(X) = \mathbb{E}(X_1) = \mathbb{E}(X_2) = ... = \mathbb{E}(X_k)$. Using equation (7.12) with $a_i = \frac{1}{k}$ we obtain

$$\mathrm{Var}(G) = \frac{1}{k^2} \sum_{i=1}^{k} \mathrm{Var}(X) = \frac{1}{k} \mathrm{Var}(X). \tag{7.16}$$

Writing the previous equation in terms of standard deviation gives

$$\boxed{\sigma_G = \frac{1}{\sqrt{k}}\sigma_X.} \tag{7.17}$$

The factor $\frac{1}{\sqrt{k}}$ is crucial, since it means that the standard deviation of G decreases as k increases. If you think of k as the number of measurements, then the more measurements you make, the smaller σ_G is, the more precise your measurement is. Here σ_G is the error on the average.

7.2.2 Correlated random variables

In this section we consider the the general case where the random variables $X_1, X_2, ..., X_k$ can be correlated. Recall that the *covariance* is defined as

$$\mathrm{Cov}(X_i, X_j) = \mathbb{E}(X_i X_j) - \mathbb{E}(X_i)\mathbb{E}(X_j) = \mathbb{E}\left([X_i - \mathbb{E}(X_i)][X_j - \mathbb{E}(X_j)]\right). \tag{7.18}$$

We define the *linear correlation* as

$$\mathrm{Corr}(X_i, X_j) = \frac{\mathrm{Cov}(X_i, X_j)}{\sqrt{\mathrm{Var}(X_i)\mathrm{Var}(X_j)}}. \tag{7.19}$$

Theorem 6. *Let X_i, X_j be random variables with finite mean and variance. Then $|\mathrm{Corr}(X_i, X_j)| \leq 1$.*

Proof. Let X and Y be two random variables from a probability distribution with finite mean and variance. We first prove the Cauchy-Schwarz inequality in the context of random variables, and then apply it to obtain the desired result.

1. In the context of random variables, the Cauchy-Schwarz inequality reads $|\mathbb{E}(XY)| \leq \sqrt{\mathbb{E}(X^2)\mathbb{E}(Y^2)}$. We will prove the equivalent relation $(\mathbb{E}(XY))^2 \leq \mathbb{E}(X^2)\mathbb{E}(Y^2)$. Let us first consider the quantity $(X - \alpha Y)^2$ with $\alpha \in \mathbb{R}$. This quantity is always non-negative so its mean is also non-negative:

$$\mathbb{E}\left((X - \alpha Y)^2\right) \geq 0. \tag{7.20}$$

Expanding the square we get

$$\mathbb{E}(X^2) - 2\alpha\mathbb{E}(XY) + \alpha^2\mathbb{E}(Y^2) \geq 0. \tag{7.21}$$

Set $\alpha = \frac{\mathbb{E}(XY)}{\mathbb{E}(Y^2)}$, then the previous inequality becomes

$$\mathbb{E}(X^2) - 2\frac{(\mathbb{E}(XY))^2}{\mathbb{E}(Y^2)} + \frac{(\mathbb{E}(XY))^2}{\mathbb{E}(Y^2)} \geq 0 \tag{7.22}$$

which gives the desired inequality

$$(\mathbb{E}(XY))^2 \leq \mathbb{E}(X^2)\mathbb{E}(Y^2) \tag{7.23}$$

2. Apply the Cauchy-Schwarz inequality with $X = X_i - \mathbb{E}(X_i)$ and $Y = X_j - \mathbb{E}(X_j)$:

$$\left(\mathbb{E}\left[(X_i - \mathbb{E}(X_i))(X_j - \mathbb{E}(X_j)\right]\right)^2 \leq \mathbb{E}[(X_i - \mathbb{E}(X_i))^2]\mathbb{E}[(X_i - \mathbb{E}(X_i))^2]. \tag{7.24}$$

In terms of variance and covariance the previous expression is

$$\left(\mathrm{Cov}(X_i, X_j)\right)^2 \leq \mathrm{Var}(X_i)\mathrm{Var}(X_j) \tag{7.25}$$

which gives us the desired result

$$\frac{\left(\mathrm{Cov}(X_i, X_j)\right)^2}{\mathrm{Var}(X_i)\mathrm{Var}(X_j)} \leq 1 \Leftrightarrow \frac{|\mathrm{Cov}(X_i, X_j)|}{\sqrt{\mathrm{Var}(X_i)\mathrm{Var}(X_j)}} \leq 1 \Leftrightarrow |\mathrm{Corr}(X_i, X_j)| \leq 1. \tag{7.26}$$

□

Note that if X_i and X_j are independent then the correlation is 0. However, the opposite is not true: $\text{Corr}(X_i, X_j) = 0$ does not imply that X_i, X_j are independent. This can be seen intuitively if we consider two random variables X, Y representing the two coordinates of points chosen uniformly on a circle centered in 0. The two variables are not independent but the average of X, Y and XY are 0 which gives a linear correlation of 0. This is why we call this quantity the *linear* correlation.

We now consider the case were the random variables $X_1, X_2, ..., X_k$ can be correlated, i.e. the second sum in (7.9) is not necessarily equal to 0. We want to find the relation between the standard deviation σ_G and the standard deviations σ_{X_i}. Let us try to find an upper-bound to the variance of G:

$$\text{Var}(G) = |\text{Var}(G)| \leq \sum_{i=1}^{k} a_i^2 \left(\mathbb{E}(X_i^2) - [\mathbb{E}(X_i)]^2 \right) + \left| \sum_{i \neq j}^{k} a_i a_j \text{Cov}(X_i, X_j) \right| \tag{7.27}$$

where we used the triangular inequality $|\sum_{i=1}^{N} b_i| \leq \sum_{i=1}^{N} |b_i| \, \forall b_i \in \mathbb{R}$, $i = 1, 2, ..., N$. Noticing that all the terms in the first sum are positive and applying the triangular inequality again to the second sum we get

$$\text{Var}(G) \leq \sum_{i=1}^{k} a_i^2 \left(\mathbb{E}(X_i^2) - [\mathbb{E}(X_i)]^2 \right) + \sum_{i \neq j}^{k} |a_i||a_j| \, |\text{Cov}(X_i, X_j)| \, . \tag{7.28}$$

The property that $|\text{Corr}(X_i, X_j)| \leq 1$ implies that

$$|\text{Cov}(X_i, X_j)| \leq \left| \sqrt{\text{Var}(X_i)\text{Var}(X_j)} \right| . \tag{7.29}$$

Thus,

$$\text{Var}(G) \leq \sum_{i=1}^{k} a_i^2 \left(\mathbb{E}(X_i^2) - [\mathbb{E}(X_i)]^2 \right) + \sum_{i \neq j}^{k} |a_i||a_j| \left| \sqrt{\text{Var}(X_i)\text{Var}(X_j)} \right| . \tag{7.30}$$

Recall that by definition $\text{Var}(X_i) = \mathbb{E}(X_i^2) - [\mathbb{E}(X_i)]^2$. We see that the right-hand side of the equation above (7.30) factorizes and we obtain

$$\text{Var}(G) \leq \left(\sum_{i=1}^{k} |a_i| \sqrt{\text{Var}(X_i)} \right)^2 . \tag{7.31}$$

Taking the square root on both side, we obtain an expression in terms of standard deviations:

$$\boxed{\sigma_G \leq \sum_{i=1}^{k} |a_i| \sigma_{X_i}} . \tag{7.32}$$

Example 8. *Let us apply the above result in the average of i.i.d. random variables described in 7.2.1, i.e.* $G = \frac{1}{k} \sum_i X_i$ *with the* X_i *i.i.d.. We obtain*

$$\sigma_G \leq \sigma_X \tag{7.33}$$

In this case we don't have the factor $\frac{1}{\sqrt{k}}$ *from equation (7.17), so we can in general not say that the standard deviation of g decreases as k increases. In other terms, equation (7.33) tells us that in the worst case repeating an experiment k times does not decrease your uncertainty of the result. This can be seen in the particular case where* $a_i = \frac{1}{k} \, \forall i$ *and all the random variables are equal, i.e.* $X_1 = X_2 = ... = X_k$*. In that case the variables are not independent, we only draw randomly one of the* X_i *and then all the other random variables take the same value. This maximises the correlation and saturates the inequality (7.33):*

$$G = \sum_{i=1}^{k} a_i X_i = X_i \Longrightarrow \sigma_G = \sigma_{X_i} \tag{7.34}$$

We thus see that for general correlations the inequality (7.32) cannot be improved.

7.3 Generic function of k correlated random variables

Let us now consider the generic case where $G(X_1, X_2, ..., X_k)$ is not necessarily a linear combination of the X_i. Then we do not have $\mathbb{E}\left[G(X_1, X_2, ..., X_k)\right] = G\left[\mathbb{E}(X_1), \mathbb{E}(X_2), ..., \mathbb{E}(X_k)\right]$ in general. For example define $G(X, Y) = X^Y$ with X and Y independent, then clearly:

$$\mathbb{E}G(X,Y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x^y \rho(x)\rho(y)dxdy \neq \left[\int_{-\infty}^{\infty} x\rho(x)dx\right]^{\int_{-\infty}^{\infty} y\rho(x)dy}. \tag{7.35}$$

Thus, the reasoning of the previous section doesn't hold already for the mean.

To retrieve the relations that are often used in experimental physics we need to suppose that the errors are small in comparison to the means, i.e. that

$$G(X_1, X_2, ..., X_k) = G\left[\mathbb{E}(X_1), \mathbb{E}(X_2), ..., \mathbb{E}(X_k)\right] + \epsilon \text{ with } |\epsilon| \ll \left|G\left[\mathbb{E}(X_1), \mathbb{E}(X_2), ..., \mathbb{E}(X_k)\right]\right|. \tag{7.36}$$

More precisely, this means that we can ignore the second order correction in the Taylor expansion of G around $\mathbb{E}(X_1), ..., \mathbb{E}(X_k)$:

$$G(X_1, ..., X_k) = G\left[\mathbb{E}(X_1), ..., \mathbb{E}(X_k)\right] + \sum_{i=1}^{k} \left.\frac{\partial G(X_1, ..., X_k)}{\partial X_i}\right|_{X_j = \mathbb{E}(X_j)\,\forall j} (X_i - \mathbb{E}(X_i)) + \cancel{\sum_{i=1}^{k} \mathscr{O}\left(|X_i - \mathbb{E}(X_i)|\right)}. \tag{7.37}$$

This brings us back to the case of linear combination of random variables. Indeed, all the expressions depending only on the mean of the random variables X_i are deterministic, and thus are considered constant when we take the variance:

$$\mathrm{Var}(G) = \mathrm{Var}\left(\sum_{i=1}^{k} \left.\frac{\partial G(X_1, ..., X_k)}{\partial X_i}\right|_{X_j = \mathbb{E}(X_j)\,\forall j} X_i\right) \tag{7.38}$$

We can then identify the partial derivative with respect to X_i as the a_i from (7.7).

If the variables are independent were are back in the case of section 7.2.1 and equation (7.14) becomes

$$\boxed{\sigma_G^2 = \sum_{i=1}^{k} \left[\left.\frac{\partial G(X_1, ..., X_k)}{\partial X_i}\right|_{X_j = \mathbb{E}(X_j)\,\forall j}\right]^2 \sigma_{X_i}^2.} \tag{7.39}$$

This is the expression that you use in your laboratory reports if the variables are independent, i.e. if each variable X_i results from independent measurement. The independence of the measurements is usually assumed to be true in physics experiments. If the variables are correlated, then we are in the case described in section 7.2.2, and (7.32) becomes

$$\boxed{\sigma_G \leq \sum_{i=1}^{k} \left|\left.\frac{\partial G(X_1, ..., X_k)}{\partial X_i}\right|_{X_j = \mathbb{E}(X_j)\,\forall j}\right| \sigma_{X_i}.} \tag{7.40}$$

The most pessimistic case is when the equality holds. This is the formula that you use in your laboratory reports if you do not assume that the variables are independent. Using the notation ΔA to indicate the standard deviation of a quantity A, and considering only three random variables $X_1 = x$, $X_2 = y$ and $X_3 = z$, we obtain the formula

$$\Delta G = \left|\frac{\partial G}{\partial x}\right| \Delta x + \left|\frac{\partial G}{\partial y}\right| \Delta y + \left|\frac{\partial G}{\partial z}\right| \Delta z \tag{7.41}$$

that you may be more familiar with.

Keep in mind that these formulas are only correct if the errors are small (i.e. that we can neglect the higher order terms in the Taylor expansion). Also note that these errors only consider statistical fluctuations, and not systematic errors if for example a measuring device is badly calibrated.

7.4 Law of large numbers

Until now we supposed implicitly that realizations of G are close to its mean $\mathbb{E}(G)$, and that the fluctuations around the mean of G are related to the variance. For the specific case of k i.i.d variables X_i and $G = \frac{1}{k}\sum_{i=1}^{k} X_i$, we would like to obtain a relation of the type

$$G - \mathbb{E}(G) \approx \mathscr{O}\left(\frac{1}{\sqrt{k}}\right) \tag{7.42}$$

which would allow us to justify what we usually write in physics as

$$G = \mathbb{E}(G) \pm \frac{\sigma_X}{\sqrt{k}}. \tag{7.43}$$

The law of large number allows us to formalize this intuition.

Let us first introduce and prove the related *Chebyshev inequality*:

Theorem 7 (Chebyshev inequality)**.** *Let $\rho(x)$ be the p.d.f. of a random variable X with finite mean and variance. Then*

$$\text{Proba}\left(|X - \mathbb{E}(X)| \geq l\sigma_X\right) \leq \frac{1}{l^2} \text{ with } l \in \mathbb{R}, l > 0 \tag{7.44}$$

where σ_X is the standard deviation of X.

Proof. The proof of the inequality is done in two steps. First we introduce and prove the Markov inequality and then apply it to obtain the Chebyshev inequality.

1. Let X be a non-negative random variable with a finite mean. Then the Markov inequality states that $\forall a > 0$ we have that $\text{Proba}(X \geq a) \leq \frac{\mathbb{E}(X)}{a}$. Proof:

$$\text{Proba}(X \geq a) = \int_a^\infty \rho(x)dx \leq \int_a^\infty \overbrace{\frac{x}{a}}^{\geq 1} \rho(x)dx \leq \frac{1}{a}\int_0^\infty x\rho(x)dx = \frac{\mathbb{E}(X)}{a}, \tag{7.45}$$

   where we notice that the integration domain is $(0, +\infty)$ as X is non-negative.

2. Choose $X \to |X - \mathbb{E}(X)|^2$ and $a = l^2\text{Var}(X)$, $l \geq 0$ and apply Markov's inequality:

$$\text{Proba}\left(|X - \mathbb{E}(X)|^2 \geq l^2\text{Var}(X)\right) \leq \frac{\cancel{\text{Var}(X)}}{l^2\cancel{\text{Var}(X)}} = \frac{1}{l^2} \tag{7.46}$$

   Since $|X - \mathbb{E}(X)|^2 \geq 0$ and $l^2\text{Var}(X) \geq 0$ we can take the square root on both sides and not change the result of the obtained probability. Thus, we have

$$\text{Proba}\left(|X - \mathbb{E}(X)| \geq l\sigma_X\right) \leq \frac{1}{l^2} \text{ with } l \in \mathbb{R}, l \geq 0. \tag{7.47}$$

□

We now define the *confidence* as the probability that a realization of a random variable X is within l standard deviation of the mean $\mathbb{E}(X)$. Table 1 shows the confidence for various values of l. The case $l = 1$ is not noted since the Chebyshev inequality does not provide any information in that case. You may be more familiar with the table 2 which shows the confidence if we additionally suppose that the p.d.f. $\rho(x)$ is Gaussian, i.e. if $X \sim \mathcal{N}(0, 1)$.

l	2	3	4	...	10
number of σ	2σ	3σ	4σ	...	10σ
Confidence	75%	89%	94%	...	99%

Table 1: Confidence obtained from the Chebyshev inequality for various l. Valid for any probability distribution of finite mean and variance.

l	1	2	3	...	6
number of σ	σ	2σ	3σ	...	6σ
Confidence	68%	95%	99.7%	...	99.9999998%

Table 2: Confidence obtained for the Gaussian distribution.

Let us now apply the Chebyshev inequality with $l = \frac{\epsilon}{\sigma_G}$ to the case of mean of k i.i.d variables $G = \frac{1}{k}\sum_{i=1}^k X_i$:

$$\text{Proba}\left(|G - \mathbb{E}(G)| \geq \epsilon\right) \leq \frac{\sigma_G^2}{\epsilon^2} = \frac{\sigma_X^2}{k\epsilon^2} \xrightarrow{k\to\infty} 0 \tag{7.48}$$

were we used (7.17) and to take the limit we notice that σ_X and ϵ are fixed. This means that the random variable G converges *in probability* to its mean when $k \to \infty$. The expression (7.48) is known as the *law of large numbers*. Note that there also exist more general versions of the law of large numbers that does not require the variance to exist (that will be covered and proven in the Probability and Statistics course). Since in the i.i.d. case $\mathbb{E}(G) = \mathbb{E}(X)$, we can intuitively interpret the law of large number as saying that repeating an experiment many times increases the probability that the empirical average (which is one realisation of G) is close to the true average $\mathbb{E}(X)$.

8 Lecture 8: Central Limit Theorem, Statistics and Maximum Likelihood

8.1 Reminders and the Central Limit Theorem

Recall the statements from the previous lecture: Let's say we have random variables $X_i \sim \rho(x_i) \forall i$, which are i.i.d. and for which the mean $\mathbb{E}X$ and the variance $\mathrm{Var}(X)$ exist. We define the mean as $G_k = \frac{1}{k}\sum_{i=1}^{k} X_i$ which has mean $\mathbb{E}G_k = \mathbb{E}X_i$ and $\mathrm{Var}(G_k) = \frac{1}{k}\mathrm{Var}(X)$, so that the standard deviation is $\sigma_G = \frac{1}{\sqrt{k}}\sigma_X$.

Recall the Tschebychev bound, which we previously stated for a random variable Y with an arbitrary distribution $\rho_Y(y)$ and existent mean and variance σ_Y^2. For $l > 0$ the difference between the random variable Y and its mean is bounded as

$$P\Big(|Y - \mathbb{E}Y| \geq l\sigma_Y\Big) \leq \frac{1}{l^2}\,. \tag{8.1}$$

We want to use this bound to make a statement about how well the empirical average G approximates the mean of X_i? To do so, we set $Y = G$, and $l\sigma_Y = \varepsilon \ll 1$. Using the variance and mean we computed for G in terms of k, This gives

$$P\Big(|G - \mathbb{E}X| \geq \varepsilon\Big) \leq \frac{\sigma_G^2}{\varepsilon^2} = \frac{\sigma_X^2}{k\varepsilon^2}\,. \tag{8.2}$$

What happens if we increase the number of samples k? As $k \to \infty$, the fraction on the right hand side of the bound goes to zero because $\lim_{k\to\infty} \frac{\sigma_X^2}{k\varepsilon^2} = 0$. This behaviour of the empirical mean of i.i.d. random variables is commonly referred to as the **Law of Large Numbers** (LLN). As we increase the number of samples, it becomes extremely unlikely for the empirical average G to deviate from the mean. The absolute distances converge with a rate at least $1/k$.

In physics, it is common to write this result using the $\pm$ symbol

$$G = \mathbb{E}(X) \pm \frac{\sigma_x}{\sqrt{k}}\,. \tag{8.3}$$

Usually, this means that we understand G to be distributed in terms of a new random variable R with a distribution $\rho_R(r)$ such that $G = \mathbb{E}(X) + R\frac{\sigma_x}{\sqrt{k}}$. Rearranging, we get that

$$R = \sqrt{k}\frac{(G - \mathbb{E}(X))}{\sigma_X}\,. \tag{8.4}$$

Surprisingly, the probability distribution of this random variable[8] for large k does not depend on the details of $\rho_X(x)$ as long as its variance is finite. As we increase k, R becomes distributed as a standard Gaussian

$$\rho_R(r) \xrightarrow{k\to\infty} \mathcal{N}(r; 0, 1) = \frac{1}{\sqrt{2\pi}} e^{-\frac{r^2}{2}}\,. \tag{8.5}$$

This property is called the **Central Limit Theorem** (CLT) and it has an important place in physics. The fact that the empirical mean of many samples which are identically and independently distributed (remember this is an important part of our assumption) is behaving like a Gaussian for large enough k, is remarkable and one of the reasons why we encounter the Gaussian so often in physical processes. CLT is a type of a universality result where certain properties of a large ensemble of variables do not depend on microscopic details, such as the precise form of $\rho_X(x)$. Universality is another important concept in many body physics, as you will learn next year.

[8] You can think of it as a normalized version of G: We subtracted the mean and divided by the standard deviation.

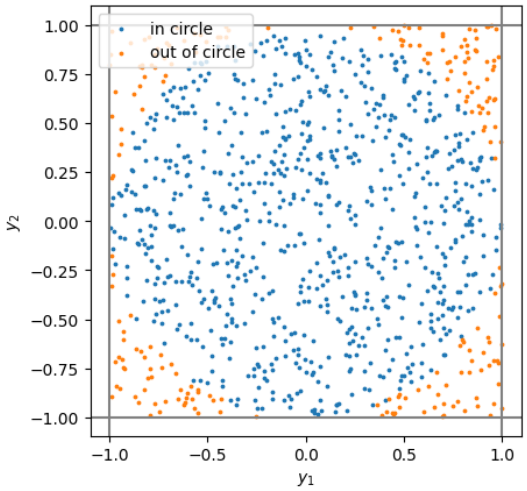

Figure 17: Samples of the random variables Y_1, Y_2. The color is according to whether X_i is one or zero. The samples with $X_i = 1$ fall within the area of the circle with radius 1. All samples fall within the square with the corners $(-1,-1),(1,1)$.

8.2 Example: Estimating π

To make the law of large numbers and the central limit theorem more tangible, we consider an example: Let $Y_1, Y_2 \sim \mathscr{U}(-1,1)$ with Y_1, Y_2 independent. Define a new random variable for $i = 1, ..., k$, as

$$X_i = \begin{cases} 1 & Y_1^2 + Y_2^2 \leq 1 \\ 0 & \text{else} \end{cases} . \tag{8.6}$$

We visualize this distribution over the two dimensions Y_1 and Y_2 in Figure 17. The shown plot helps us understand, that the probability p of X_i being exactly one is

$$P(X_i = 1) = \frac{\text{area of circle}}{\text{area of square}} = \frac{\pi}{4} . \tag{8.7}$$

In consequence we have that $\mathbb{E}X_i = \frac{\pi}{4}$ and $\text{Var}(X_i) = \frac{\pi}{4} - \frac{\pi^2}{4^2} = \frac{\pi}{4}\frac{4-\pi}{\pi}$.

To estimate the value $\mathbb{E}X_i$ (and thereby π) from k i.i.d. samples of X_i, we compute the empirical mean G:

$$G = \frac{1}{k}\sum_{i=1}^{k} X_i \tag{8.8}$$

$$\mathbb{E}(G) = \frac{1}{k}\sum_{i=1}^{k} \mathbb{E}(X_i) = \frac{\pi}{4} \tag{8.9}$$

$$\text{Var}(G) = \frac{1}{k}\text{Var}(X) = \frac{\pi}{4k}\frac{4-\pi}{4} \tag{8.10}$$

$$\sigma_G = \frac{1}{\sqrt{k}}\sqrt{\frac{\pi(4-\pi)}{16}} \tag{8.11}$$

Applying the law of large numbers to this random variable gives the bound

$$P\left(\left|G - \frac{\pi}{4}\right| \geq \frac{c}{\sqrt{k}}\sqrt{\frac{\pi(4-\pi)}{16}}\right) \leq \frac{1}{c^2} . \tag{8.12}$$

In Figure 18 we show that the decay of the error $|G - \frac{\pi}{4}|$ is indeed behaving as in $1/\sqrt{k}$ for different constants c. Likewise, from the central limit theorem, the random variable R behaves as

$$R = \frac{\sqrt{k}(G - \frac{\pi}{4})}{\sqrt{\text{Var}(X)}} . \tag{8.13}$$

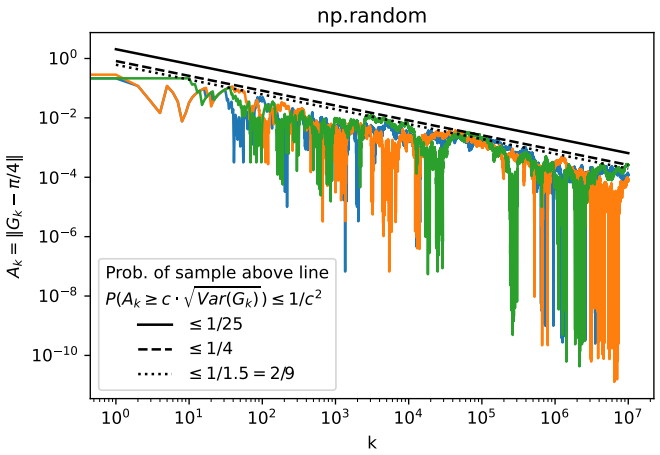

Figure 18: Law of large numbers illustrated using the example for sampling π.

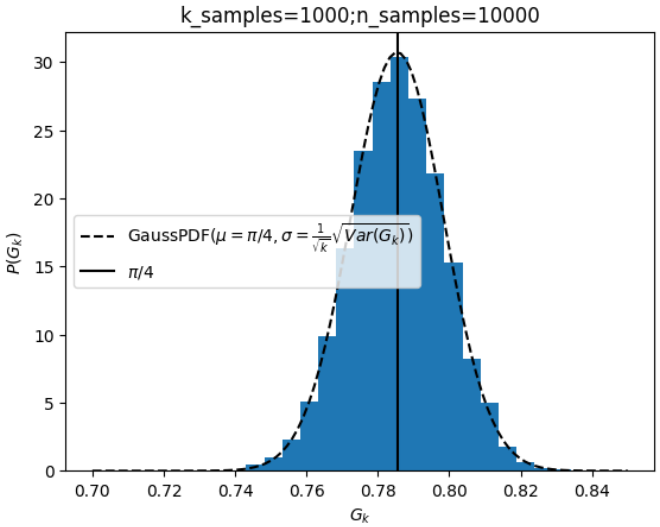

Figure 19: The central limit theorem illustrated for the example of sampling π, with $k = 1000$ and $n = 10,000$.

For a fixed k, we can access this distribution, by sampling n times from X_k and plotting the result as a histogram (so we require overall $2nk$ independent samples from $\mathscr{U}(-1, 1)$). In Figure 19 this behaviour is illustrated for $k = 1000$ and $n = 10,000$.

8.3 Statistics

In probability theory, the setting is that we have a *known* probability distribution $\rho_X(x)$, and we want to derive some of its properties, e.g. properties of samples from $\rho_X(x)$. In statistics, the situation is reversed: We have some data samples given, and we want to know something about the properties of their probability distribution.

We will consider the random variables $X_1, \cdots, X_n$ sampled i.i.d from $\rho_X(x)$ where $\mathbb{E}X = \mu$ and $\mathrm{Var}(X) = \sigma_X^2$. We have access to the samples, but not to ρ_X or the constants μ, σ_X. Our goal is to find estimates of μ, σ_X.

The empirical mean is the estimator[9] $\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} X_i$. The question on how good this estimator is, can then be answered by the law of large numbers as

$$P\left(|\mu - \hat{\mu}| \geq \varepsilon\right) \leq \frac{\sigma_X^2}{n\varepsilon^2}\,. \tag{8.14}$$

This is essentially what we did before, for the example of estimating π.

[9] We put the hat $\hat{\cdot}$ to denote that we are estimating a given property of the original distribution from observed data samples.

But now, can the same strategy apply for an estimate of σ_X? Intuitively, that would mean to set the estimator of the variance, the empirical variance, to

$$\hat{\sigma}_X^2 = \frac{1}{n}\sum_{i=1}^{n}(X_i - \hat{\mu})^2 = \frac{1}{n}\sum_{i=1}^{n} X_i^2 - \hat{\mu}^2 \,. \tag{8.15}$$

Is this a sensible estimator? We see that taking $n = 1$ the empirical variance $\hat{\sigma}_X^2 = 0$ which is not reasonable. A reasonable requirement would be that the expected value of the estimator should be equal to the true property we want to approximate. Recall that we can take an expectation over an estimator, because it is only a function of the random variables that represent the samples X_i. Indeed, we want to have that

$$\mathbb{E}(\hat{\sigma}_X^2) \stackrel{!}{=} \sigma_X^2 \,. \tag{8.16}$$

Is this the case for our intuitive estimator for the variance? We check

$$\mathbb{E}(\hat{\sigma}_X^2) = \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}(X_i^2) - \mathbb{E}[(\frac{1}{n}\sum_{i=1}^{n} X_i)^2] \tag{8.17}$$

$$= \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}(X_i^2) - \frac{1}{n^2}\sum_{i=1}^{n}\mathbb{E}(X_i^2) - \frac{1}{n^2}\sum_{i\neq j}\mathbb{E}(X_i)\mathbb{E}(X_j) \tag{8.18}$$

$$= \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}(X_i^2) - \frac{1}{n^2}\sum_{i=1}^{n}\mathbb{E}(X_i^2) - \frac{n^2-n}{n^2}\mathbb{E}(X_i)^2 \tag{8.19}$$

$$= (\frac{1}{n} - \frac{1}{n^2})\sum_{i=1}^{n}\mathbb{E}(X_i^2) - (1-\frac{1}{n})\mathbb{E}(X_i)^2 \tag{8.20}$$

$$= (1-\frac{1}{n})\mathbb{E}(X_i^2) - (1-\frac{1}{n})\mathbb{E}(X_i)^2 \tag{8.21}$$

$$= \frac{n-1}{n}[\mathbb{E}(X_i^2) - \mathbb{E}(X_i)^2] \tag{8.22}$$

$$= \frac{n-1}{n}\sigma_X^2. \tag{8.23}$$

Therefore, we indeed go to the correct mean as $n \to \infty$. However, for finite n we note that the estimator is biased to a different value than the parameter we aim to estimate (it is off by the factor $\frac{n-1}{n}$). To avoid this, we can instead use a more common unbiased estimator for empirical variance

$$\hat{s}_X^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \hat{\mu})^2 \tag{8.24}$$

so that instead we end up with

$$\mathbb{E}\hat{s}_X^2 = \sigma_X^2 \,. \tag{8.25}$$

In this case, the estimator is unbiased for every value of n, arguably a sensible desiderata.

8.4 Parametric Statistics and Maximum Likelihood

In Statistics, the statistician may assume that the random variables are generated from parameterized density functions. From a physics perspective this is particularly useful: When we have data samples $\mathscr{D}$ coming from a physical process which we understand at least partially. Concretely, we can formulate the probability density of the data conditional on the unknown parameters. Given both the data $\mathscr{D}$ and the knowledge of the physics process behind the data generation, expressed via a conditional probability distribution $\rho(\mathscr{D}|\theta)$, the goal is to estimate the parameters θ using an estimator $\hat{\theta}$ from the observed data $\mathscr{D}$. In this context, we write the density of the data as a *conditional density function* and make the conditional dependence on the parameters clear by using the notation $\rho(\mathscr{D}|\theta)$.

One principle of deriving estimators for the parameter in this setting is called *maximum likelihood estimation*. For this, we want to use the estimator which maximized the likelihood of the data realization,

$$\hat{\theta}_{\mathrm{ML}} = \arg\max_{\theta} \rho(\mathscr{D}|\theta) \,. \tag{8.26}$$

Example 1 (Empirical mean and variance). As in many examples before, we assume $X_1, \cdots, X_n \sim \rho(x_i)$ to be distributed i.i.d. from the same density. However, this time we make an additional assumption on ρ: We assume that we are sampling from a Gaussian distribution with a fixed but unknown mean μ and variance σ. Then we can write the conditional distribution for a single sample X_i as

$$\rho(x_i|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(x_i-\mu)^2} = \mathscr{N}(x_i;\mu,\sigma)\,. \tag{8.27}$$

In the previous notation, $\theta = \{\mu,\sigma\}$ and therefore (using the i.i.d assumption) the maximum likelihood estimator considers the conditional density over the complete dataset $\mathscr{D}$ as

$$\rho(\mathscr{D}|\theta) = \rho(x_1,\cdots,x_n|\mu,\sigma) = \prod_{i=1}^{n} \rho(x_i|\mu,\sigma) = \prod_{i=1}^{n} \left[\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}\right]. \tag{8.28}$$

Then maximization of the above probability distribution is the same as the following minimization

$$\hat{\mu} = \arg\max_{\mu} \rho(X_1,\cdots,X_n|\mu,\sigma) = \arg\min_{\mu} \frac{1}{2}\sum_{i=1}^{n}(X_i-\mu)^2 \tag{8.29}$$

To find the minimum, we set the derivative to zero

$$\left.\frac{\partial\rho(X_1,\cdots,X_n|\mu,\sigma)}{\partial\mu}\right|_{\hat{\mu}} = \sum_{i=1}^{n}(X_i-\hat{\mu}) = 0 \tag{8.30}$$

and therefore

$$\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} X_i\,. \tag{8.31}$$

Here, we found the empirical mean $\hat{\mu}$ is the maximum likelihood estimator for the true mean μ.

The same principle applies for estimating the variance of the Gaussian distribution. Defining, as before $\hat{\sigma}_X^2 = \frac{1}{n}\sum_{i=1}^{2}(X_i-\hat{\mu})^2$

$$\hat{\sigma} = \arg\max_{\sigma} \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_i-\hat{\mu})^2}{2\sigma^2}} = \arg\max_{\sigma} \frac{1}{\sigma^n} e^{-\frac{n}{2\sigma^2}\hat{\sigma}_X^2} \tag{8.32}$$

and taking the derivative w.r.t. σ as

$$0 = \frac{\partial}{\partial\sigma}\left[\frac{1}{\sigma^n} e^{-\frac{n}{2\sigma^2}\hat{\sigma}_X^2}\right]_{\sigma=\hat{\sigma}} = -\frac{n}{\hat{\sigma}}\frac{1}{\hat{\sigma}^n} e^{-\frac{n}{2\sigma^2}\hat{\sigma}_X^2} + 2\frac{n\hat{\sigma}_X^2}{2\hat{\sigma}^3}\frac{1}{\hat{\sigma}^n} e^{-\frac{n}{2\sigma^2}\hat{\sigma}_X^2} \tag{8.33}$$

so that

$$\hat{\sigma}^2 = \hat{\sigma}_X^2\,. \tag{8.34}$$

Note that in this case, we do not recover the $n-1$ term due to the bias as finite n. Maximum likelihood provides reliable estimators rather generally when n is sufficiently large, as we will formalize in the next lecture.

Example 2 (Weighted average). In this example, we are given n measurements of a quantity and the error bars for each of them: $(X_1,\sigma_1^2),(X_2,\sigma_2^2),\cdots,(X_n,\sigma_n^2)$. The $X_1, X_2, \cdots, X_n$ are i.i.d. random variables, and $\sigma_1^2,\sigma_2^2,\cdots,\sigma_n^2$ are considered fixed parameters, in general $\sigma_j \neq \sigma_i$. We additionally assume that every random variable X_i is sampled from a Gaussian distribution with a known variance, but an unknown mean: $X_i \sim \mathscr{N}(\mu,\sigma_i^2)$. Note that the mean μ is assumed to be independent of i.

To estimate the mean μ, we use the maximum likelihood principle again. We write

$$\hat{\mu} = \arg\max_{\mu} \rho(\mathscr{D}|\mu) = \max_{\mu} \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma_i^2}} e^{-\frac{(X_i-\mu)^2}{2\sigma_i^2}} = \arg\min_{\mu} \frac{1}{2n}\sum_{i=1}^{n}\frac{1}{\sigma_i^2}(X_i-\mu)^2 \tag{8.35}$$

and minimize the last term, by setting the derivative to zero

$$\left.\frac{\partial \rho(\mathscr{D}|\mu)}{\partial \mu}\right|_{\hat{\mu}} = \sum_{i=1}^{n} \frac{1}{\sigma_i^2}(X_i - \hat{\mu}) = 0 \tag{8.36}$$

and finally get

$$\hat{\mu} = \frac{\sum_{i=1}^{n} \frac{X_i}{\sigma_i^2}}{\sum_{i=1}^{n} \frac{1}{\sigma_i^2}} \,. \tag{8.37}$$

Note that if $\forall i : \sigma = \sigma_i$, then we get the same estimator we had in the first example. Here we needed the assumption of the random variables being Gaussian to derive the estimator.

9 Lecture 9: Maximum Likelihood, examples and theory

9.1 More examples on maximum likelihood estimators

We start this lecture by continuing Example 2 on the weighted average.

Example 2 continued (Weighted average and its error). During the previous lecture, we saw that, if we are given n measurements $\{X_i\}_{i=1}^n$ distributed as $X_i \sim \mathscr{N}(\mu, \sigma_i^2)$, then the maximum likelihood estimator $\hat{\mu}$ for the mean parameter μ (unknown but assumed shared across all measurements) is

$$\hat{\mu} = \frac{\sum_{i=1}^{n} \frac{X_i}{\sigma_i^2}}{\sum_{i=1}^{n} \frac{1}{\sigma_i^2}} \,. \tag{9.1}$$

Recall that $\hat{\mu}$ is a function of the measurements $\{X_i\}_{i=1}^n$, which are random, implying that $\hat{\mu}$ itself is a random variable. We now want to estimate the error associated with this estimator, i.e. to quantify its variance. We notice that $\hat{\mu}$ is a linear combination of i.i.d. random variables, so to compute its variance we can apply (7.14) with $k = n$ and

$$a_i = \frac{1}{\sigma_i^2 \sum_{j=1}^{n} \sigma_j^{-2}} \,, \tag{9.2}$$

obtaining

$$\mathrm{Var}[\hat{\mu}] = \sum_{i=1}^{n} a_i^2 \mathrm{Var}[X_i] = \left(\sum_{j=1}^{n} \sigma_j^{-2}\right)^{-2} \sum_{i=1}^{n} \sigma_i^{-4} \mathrm{Var}[X_i] = \left(\sum_{j=1}^{n} \sigma_j^{-2}\right)^{-2} \sum_{i=1}^{n} \sigma_i^{-2} = \left(\sum_{j=1}^{n} \sigma_j^{-2}\right)^{-1} \,, \tag{9.3}$$

where we used the assumption that $X_i \sim \mathscr{N}(\mu, \sigma_i^2)$ to say that $\mathrm{Var}[X_i] = \sigma_i^2$. Thus, the estimate of the parameter μ is given by

$$\mu^{\text{measured}} \approx \hat{\mu} \pm \sqrt{\mathrm{Var}[\hat{\mu}]} = \frac{\sum_{i=1}^{n} \frac{x_i}{\sigma_i^2}}{\sum_{i=1}^{n} \frac{1}{\sigma_i^2}} \pm \frac{1}{\sqrt{\sum_{j=1}^{n} \sigma_j^{-2}}} \,. \tag{9.4}$$

As a check, we notice that if $\sigma_i = \sigma$ for all values of i, i.e. all measurements have the same variance, then

$$\mathrm{Var}[\hat{\mu}] = \frac{\sigma^2}{n} \,, \tag{9.5}$$

which recovers the non-weighted case.

We take a moment to highlight the fact that when we derive a maximum-likelihood estimator, or more generally a statistical estimator, we always work under some assumptions. It is important to always assess the validity of such assumptions for the actual data that we are studying. Consider for example Figure 20. We depict three sets of measurements (x axis: number of the measurement, y axis: value of the measurements) for a physical property. For each measurement, we depict the measured value by a dot, and the error on that measure with the error bar. The orange line depicts the weighted average of the measurements.

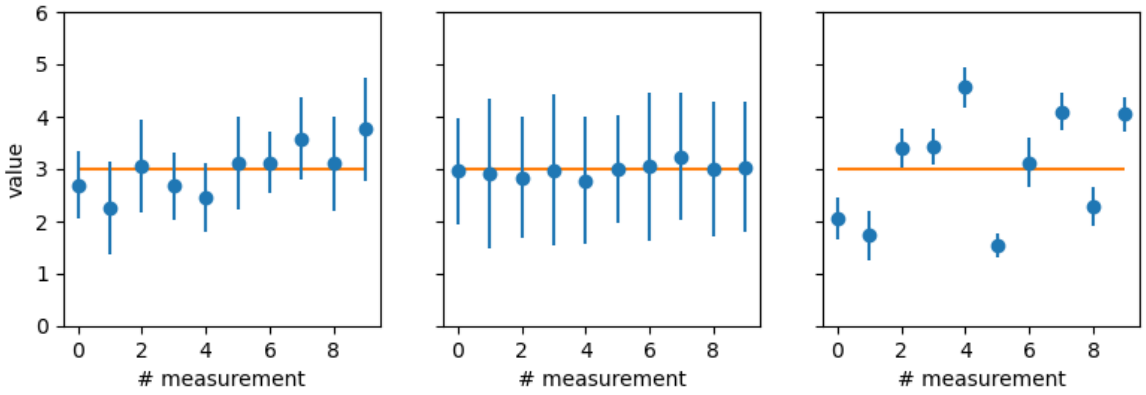

Figure 20: Consistency of the assumptions in Example 2.

In the dataset on the left, we see that the variability in the measurements is comparable to their error bars, indicating that the variance of the weighted average (which depends only on the error bars σ_i) will be a suitable estimator of the error.

In the dataset on the center, we see that the measurements have very little variability, while their error bars are large. This may be, for example, a sign of some error source that is not intrinsically related to the physical quantity we are measuring. We may want to re-think whether the error bars were indeed estimated properly. If yes, the error estimation should be done using the error bars and not the variability in the data. If not, the error of the empirical average would underestimate the error, as it would only take into account the variability of the measurements, disregarding the error bars.

In the dataset on the right, we have a large variability in the measurements, with very little error bars. In this case, it seems not appropriate to model the measurements as Gaussian variables with the same means and different variances. In particular, the error on the weighted average will be small, as it is a sort of geometric mean of the error bars, underestimating the huge variability of the measurements. In this case, it would be more appropriate to disregard the error bars on the measurement, and assume that each measurement is i.i.d.. Under this assumption, the physical quantity would be estimated by performing an empirical mean of the measurements and its error bar.

The take-home message is to always be aware of the assumptions that are implicit in every statistical estimator, and to check their plausibility on the concrete data before using them.

Example 3 (Estimation of the radioactive decay constant). Consider an experimental system in which a radioactive source placed at the origin of the frame of reference emits unstable particles in the direction of the x axis with constant speed.

You will see in the following courses that the probability that the unstable particle decays at a distance X from the source is distributed according to an exponential distribution, with probability density

$$\rho_{\lambda_*}(x) = \begin{cases} \lambda_*^{-1} e^{-x/\lambda_*} & \text{for } x \geq 0 \\ 0 & \text{otherwise} \end{cases}, \tag{9.6}$$

for a given parameter λ_*, called the decay constant. As a reminder (or simple exercise), we have that $\mathbb{E}[X] = \lambda_*$ and $\mathrm{Var}[X] = \lambda_*^2$ for the exponential distribution. Thus, the decay constant is the average position at which the unstable particles decay.

Using a detector, we can measure the position X at which an unstable particle decays. We perform such experiment n times independently, obtaining a dataset of n independent and identically distributed measurements $\mathscr{D} = \{X_i\}_{i=1}^n$, $X_i \sim \rho_{\lambda_*}(x)$. The aim is to estimate the decay constant λ_* from these measurements. To this end, we use the maximum likelihood method. The likelihood (which we recall is a function of a parameter λ, telling us how probable it is that the dataset $\mathscr{D}$ was generated by a process with the given value of λ) is given by

$$\rho(\mathscr{D}|\lambda) = \prod_{i=1}^n \rho_\lambda(x_i) = \lambda^{-n} \exp\left(-\frac{1}{\lambda}\sum_{i=1}^n x_i\right) = \left(\frac{1}{\lambda}\exp\left(-\frac{1}{n\lambda}\sum_{i=1}^n x_i\right)\right)^n \tag{9.7}$$

where in the second passage we used that all the measurements are independent and identically distributed. We recall the definition of the empirical average

$$\hat{x} = \frac{1}{n}\sum_{i=1}^{n} X_i \,. \tag{9.8}$$

By definition, the maximum-likelihood estimator $\hat{\lambda}$ is

$$\hat{\lambda} = \underset{\lambda}{\operatorname{argmax}}\, \rho(\mathscr{D}|\lambda) = \underset{\lambda}{\operatorname{argmax}} \left(\frac{e^{-\hat{x}/\lambda}}{\lambda}\right)^n = \underset{\lambda}{\operatorname{argmax}} \frac{e^{-\hat{x}/\lambda}}{\lambda} \tag{9.9}$$

where in the last passage we used that the function $f(y) = y^n$ is strictly increasing on $\mathbb{R}_+$ to simplify the maximization problem. The stationarity condition (which determines the maximum-likelihood estimator $\hat{\lambda}$) is given by

$$0 = -\frac{1}{\hat{\lambda}}\frac{e^{-\hat{x}/\hat{\lambda}}}{\hat{\lambda}} + \frac{\hat{x}}{\hat{\lambda}^2}\frac{e^{-\hat{x}/\hat{\lambda}}}{\hat{\lambda}} \implies \hat{\lambda} = \hat{x}\,. \tag{9.10}$$

To be precise, we should now check that the stationary point found is indeed the global maximum of the likelihood function. This turns out to be the case, as you can independently check. Thus, as one might have expected, the maximum-likelihood estimate of the decay constant is just the empirical average of the decay positions of the n particles.

What is the error on the estimator? Again, recall that the estimator depends on the measurements, which are random variables and hence is a random variable. We compute the variance explicitly by computing the first and the second moments of the estimator. For the first moment, the mean, we have

$$\mathbb{E}[\hat{\lambda}] = \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}[X_i] = \mathbb{E}[X] = \lambda_*\,, \tag{9.11}$$

which confirms that the estimator is unbiased, i.e. that on average it retrieves the correct value λ_*. For the second moment, we have

$$\begin{aligned}\mathbb{E}[\hat{\lambda}^2] &= \frac{1}{n^2}\sum_{i=1}^{n}\mathbb{E}[X_i^2] + \frac{1}{n^2}\sum_{i\neq j}\mathbb{E}[X_iX_j] = \frac{1}{n}\left(\mathbb{E}[X]^2 + \mathrm{Var}[X]\right) + \frac{1}{n^2}\sum_{i\neq j}\mathbb{E}[X_i]\mathbb{E}[X_j] \\ &= \frac{2}{n}\lambda_*^2 + \frac{n-1}{n}\lambda_*^2 = \left(1+\frac{1}{n}\right)\lambda_*^2\end{aligned} \tag{9.12}$$

where in the first passage we expanded the square on the diagonal terms $i = j$ and the non-diagonal ones $i \neq j$, in the second step we expressed the second moment of X in terms of its mean and variance, i.e.

$$\mathbb{E}[X^2] = \mathbb{E}[X]^2 + \mathrm{Var}[X] \tag{9.13}$$

and finally we used the specific values for the mean and variance of the exponential distribution. We can now compute the variance of the estimator $\hat{\lambda}$:

$$\mathrm{Var}[\hat{\lambda}] = \mathbb{E}[\hat{\lambda}^2] - \mathbb{E}[\hat{\lambda}]^2 = \left(1+\frac{1}{n}\right)\lambda_*^2 - \lambda_*^2 = \frac{1}{n}\lambda_*^2 \approx \frac{1}{n}\hat{x}^2\,, \tag{9.14}$$

where in the last passage we used the estimator for the decay constant to obtain an expression that depends only on the observed measurements, and not on the unknown parameter λ_*. Finally, we obtain the following estimate for the true decay constant

$$\lambda_*^{\text{measured}} \approx \hat{x} \pm \frac{1}{\sqrt{n}}\hat{x}\,. \tag{9.15}$$

Example 4 (Estimation of the radioactive decay constant with a detector of finite length). We continue to study the previous example, but we now consider a somewhat more realistic setting, in which the detector has a finite length, meaning that it can detect particle decays only in the interval $[x_{\min}, x_{\max}]$, with $0 < x_{\min} < x_{\max}$. The probability density of observing a decay at a given position x gets modified to

$$\rho_{\lambda_*}(x) = \begin{cases} Z(\lambda_*)^{-1}\lambda_*^{-1}e^{-x/\lambda_*} & \text{for } x_{\min} < x < x_{\max} \\ 0 & \text{otherwise} \end{cases}\,, \tag{9.16}$$

where we added a normalization factor $Z(\lambda)$ to ensure that the probability density we wrote is normalized, i.e.

$$\int_{x_{\min}}^{x_{\max}} dx\, \rho_{\lambda_*}(x) = 1\,. \tag{9.17}$$

It is a simple exercise to check that

$$Z(\lambda_*) = \lambda_*^{-1} \int_{x_{\min}}^{x_{\max}} dx\, e^{-x/\lambda_*} = e^{-x_{\min}/\lambda_*} - e^{-x_{\max}/\lambda_*}\,. \tag{9.18}$$

Notice that for $x_{\max} = +\infty$ and $x_{\min} = 0$ then $Z(\lambda_*) = 1$ and we fall back to the Example 3. What is the maximum-likelihood estimator in this case? Following the same reasoning as detailed above, we find

$$\hat{\lambda} = \operatorname*{argmax}_{\lambda} \frac{e^{-\hat{x}/\lambda}}{\lambda Z(\lambda)}\,, \tag{9.19}$$

which does not have a closed-form solution (try to write the associated stationarity condition to convince yourself). However, the estimator is well-defined and can be easily computed numerically (for example, by minimising the negative likelihood using gradient descent). At the same time, it is not intuitively clear whether this estimator is unbiased i.e. if on average it gives us the true value λ_*, and computing its mean and variance for generic n would be very cumbersome. As we will see at large n we can deduce very nice and generic properties of the maximum likelihood method.

9.2 Asymptotic properties of maximum likelihood estimators

In Example 4, we saw that it is only in special cases that we can derive an explicit characterization of the maximum-likelihood estimator, as well as for its mean and variance for a generic number of samples n. We would like to know whether the maximum-likelihood estimator is unbiased, i.e. $\mathbb{E}[\hat{\theta}] = \theta_*$ where $\hat{\theta}$ is the maximum-likelihood estimator for an unknown parameter θ_*, and to compute its variance, which is then an estimate of the error of the estimator. We will see in this section that in the limit of many measurements $n \to \infty$ a generic theory of maximum-likelihood estimator is available.

Consider a generic parameter estimation problem, in which we are given n independent and identically distributed measurements $X_i \sim \rho(x_i|\theta_*)$, $i = 1, \dots, n$. Call $\mathscr{D} = \{X_i\}_{i=1}^n$ the collection of measurements. The measurements depend on a parameter θ_* which is unknown and we want to estimate. The likelihood equals

$$\rho(\mathscr{D}|\theta) = \prod_{i=1}^{n} \rho(X_i|\theta) \tag{9.20}$$

where we used the fact that the measurements are i.i.d. Define the log-likelihood as

$$\mathscr{L}(\mathscr{D}, \theta) = \frac{1}{n} \log \rho(\mathscr{D}|\theta) = \frac{1}{n} \sum_{i=1}^{n} \log \rho(X_i|\theta) \tag{9.21}$$

and notice that the maximum likelihood estimator $\hat{\theta}$ satisfies

$$\hat{\theta} = \operatorname*{argmax}_{\theta} \rho(\mathscr{D}|\theta) = \operatorname*{argmax}_{\theta} \mathscr{L}(\mathscr{D}, \theta) \tag{9.22}$$

by increasing monotonicity of the logarithm. We notice that the log-likelihood is a sum of n terms which are random and i.i.d., as each term is a function of a single measurement X_i, each of which is an i.i.d. random variable. This gives us hope that, for large n, the log-likelihood will converge by the law of large numbers to a deterministic quantity that may be amenable to study. Indeed, consider

$$f(\theta) = \mathscr{L}(\mathscr{D}, \theta) - \mathscr{L}(\mathscr{D}, \theta_*)\,. \tag{9.23}$$

This function is just a shifted version of the log-likelihood, as the term we subtracted is constant in θ, so that the maximum-likelihood estimator is also the argmax of f. We can write

$$f(\theta) = \frac{1}{n} \sum_{i=1}^{n} \log \frac{\rho(X_i|\theta)}{\rho(X_i|\theta_*)} \xrightarrow{n\to+\infty} \mathbb{E}_X \log \frac{\rho(X|\theta)}{\rho(X|\theta_*)} = \int dx\, \rho(x|\theta_*) \log \frac{\rho(x|\theta)}{\rho(x|\theta_*)} \tag{9.24}$$

where we took the limit for large n by using the law of large numbers, and thus implicitly assumed that the random variable

$$Y = \log \frac{\rho(X|\theta)}{\rho(X|\theta_*)}\,, \quad X \sim \rho(x|\theta_*) \tag{9.25}$$

has a finite first moment. Remarkably, the left-hand side of (9.24) is a quantity that often appears when studying probabilities:

Definition 6 (Kullback-Leibler divergence)**.** *Let ρ and r be two probability density functions. Their Kullback-Leibler divergence $D_{\mathrm{KL}}(\rho||r)$ is defined as*

$$D_{\mathrm{KL}}(\rho||r) = \int_{-\infty}^{+\infty} dx\, \rho(x) \log \frac{\rho(x)}{r(x)}\,. \tag{9.26}$$

The KL divergence is a sort of "distance" between the two distributions ρ and r, in the following sense (weaker than the usual metric space theory sense):

- $\forall \rho, r$, then $D_{\mathrm{KL}}(\rho||r) \geq 0$.
- $D_{\mathrm{KL}}(\rho||r) = 0$ if and only if $\rho = r$.

Proof.

$$-D_{\mathrm{KL}}(\rho||r) = \int_{-\infty}^{+\infty} dx\, \rho(x) \log \frac{r(x)}{\rho(x)} \leq \int_{-\infty}^{+\infty} dx\, \rho(x) \left[\frac{r(x)}{\rho(x)} - 1\right] = \int_{-\infty}^{+\infty} dx\, \left[r(x) - \rho(x)\right] = 0 \tag{9.27}$$

where we used that

$$\log(x) \leq x - 1 \quad \forall x > 0 \tag{9.28}$$

as the logarithm is concave and $x - 1$ is its tangent line at $x = 1$, and later we used that by normalization

$$\int_{-\infty}^{+\infty} dx\, r(x) = \int_{-\infty}^{+\infty} dx\, \rho(x) = 1\,. \tag{9.29}$$

Thus, $D_{\mathrm{KL}}(\rho||r) \geq 0$. Moreover, we have that the inequality is saturated if and only if the argument of the logarithm equals one, giving

$$D_{\mathrm{KL}}(\rho||r) = 0 \iff \frac{r(x)}{\rho(x)} = 1 \iff r = \rho\,. \tag{9.30}$$

□

Going back to the original task of characterizing the maximum-likelihood estimator, we found that

$$\hat{\theta} = \operatorname*{argmax}_{\theta} f(\theta) = \operatorname*{argmax}_{\theta} -D_{\mathrm{KL}}(\rho(X|\theta)||\rho(X|\theta_*)) = \operatorname*{argmin}_{\theta} D_{\mathrm{KL}}(\rho(X|\theta)||\rho(X|\theta_*)) = \theta_* \tag{9.31}$$

where we used that the minimum value of the KL divergence is achieved when $\rho(X|\theta) = \rho(X|\theta_*)$, i.e. when $\theta = \theta_*$. Thus, we conclude that

Theorem 8. *(Asymptotic consistency of maximum likelihood) Let $X_i \sim \rho(x|\theta_*)$, $i = 1, \ldots, n$ be n i.i.d. random variables, and let*

$$\hat{\theta} = \operatorname*{argmax}_{\theta} \rho(\mathscr{D}|\theta) \tag{9.32}$$

be the maximum-likelihood estimator of the unknown parameter θ_*. Suppose that

$$\int dx\, \rho(x|\theta_*) \log \frac{\rho(x|\theta)}{\rho(x|\theta_*)} \tag{9.33}$$

exists and is finite. Then for $n \to +\infty$ we have $\hat{\theta} \to \theta_*$, i.e. the maximum likelihood estimator is unbiased and has variance vanishing in n.

This is already a very promising result. Given enough independent measurements, we can guarantee that the maximum-likelihood estimator provides a good estimate of the true parameter θ_*. Can we say something about its variance? We just proved a sort of law of large numbers for the maximum-likelihood estimator, thus it would seem natural that a corresponding central limit theorem characterizing its fluctuations around the mean holds. This is indeed the case. We have

Theorem 9. *(Asymptotic normality of maximum likelihood) Let $X_i \sim \rho(x|\theta_*)$, $i = 1, \dots, n$ be n i.i.d. random variables, and let*

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}}\, \rho(\mathscr{D}|\theta) \tag{9.34}$$

be the maximum-likelihood estimator of the unknown parameter θ_*. Suppose that

$$\int dx\, \rho(x|\theta_*) \log \frac{\rho(x|\theta)}{\rho(x|\theta_*)} \tag{9.35}$$

and

$$I(\theta_*) = \int dx\, \rho(x|\theta_*) \left(\frac{\partial \log \rho(x|\theta)}{\partial \theta}\bigg|_{\theta=\theta_*} \right)^2 \tag{9.36}$$

exist and are finite. Then for $n \to +\infty$ we have

$$\sqrt{nI(\theta_*)}(\hat{\theta} - \theta_*) \to \mathscr{N}(0,1)\,, \tag{9.37}$$

i.e. the fluctuations of the maximum-likelihood estimator around its mean value θ_* are Gaussian with mean zero and variance

$$Var[\hat{\theta}] = \frac{1}{nI(\theta_*)}\,. \tag{9.38}$$

$I(\theta_*)$ is called Fisher information.

The proof can be found in standard textbooks on statistics and goes beyond the scope of this lecture.

10 Lecture 10: Back to linear regression. Estimation of the errors.

In this lecture, we propose a probabilistic model for least square regression. It allows us to derive the formula for the least square estimator as a maximum likelihood estimator, to evaluate the error of our estimates and to derive a formula for weighted least-squares when one knows the errors that affect the measurements.

10.1 A probabilistic model for the least-squares

Definition
We consider the problem of linear regression. We are given data $X \in \mathbb{R}^{n\times d}$ and outputs $y \in \mathbb{R}^n$. The least-square estimator $\hat{w}$ is obtained by

$$\hat{w} = \underset{w}{\operatorname{argmin}}\, \mathscr{L}(w) = (X^T X)^{-1} X^T y \tag{10.1}$$

$$\text{with} \quad \mathscr{L}(w) = \frac{1}{n}\sum_{\mu}^{n} (y_\mu - X_\mu^T w)^2 \tag{10.2}$$

We study this problem in the light of likelihood maximization. This particular loss $\mathscr{L}$ can be interpreted in the following way. X is fixed; we assume there is a w^* such that the observed outputs are generated according to

$$y_\mu = X_\mu^T w^* + \epsilon_\mu \quad ; \quad \epsilon_\mu \sim \mathscr{N}(0, \Delta^2)\,. \tag{10.3}$$

The ϵ_μ accounts for the noise or the experimental errors. They are independent normal random variables of variance Δ^2. Δ and w^* are not known.

Proposition 3. *According to this model (10.3), the probability distribution of the output is*

$$\rho(y|w) = \prod_{\mu=1}^{n} \rho(y_\mu|w) = \prod_{\mu=1}^{n} \frac{1}{\sqrt{2\pi\Delta^2}} e^{-\frac{(y_\mu - X_\mu^T w)^2}{2\Delta^2}}\,. \tag{10.4}$$

The corresponding maximum likelihood estimator is

$$\hat{w} = \underset{w}{\operatorname{argmax}}\, \rho(y|w) = \underset{w}{\operatorname{argmin}} \sum_{\mu=1}^{n} (y_\mu - X_\mu^T w)^2 \tag{10.5}$$

$$= \underset{w}{\operatorname{argmin}}\, \mathscr{L}(w) = (X^T X)^{-1} X^T y\,. \tag{10.6}$$

This is exactly the least-square estimator. The probabilistic model (10.3) gives an interpretation of the quadratic loss $\mathscr{L}$ that corresponds to the maximum likelihood estimator for the model.

Estimation of the error

We have an explicit formula for $\hat{w}$. We assume that $n > d$ and that the matrix $X^T X$ is invertible so the estimator is well-defined. We can compute its mean and its covariance with respect to the randomness ϵ.

Proposition 4. *The least-square estimator is consistent: $\mathbb{E}\hat{w} = w^*$. Its covariance is*

$$\mathbb{E}(\hat{w} - w^*)(\hat{w} - w^*)^T = \Delta^2 (X^T X)^{-1} \ . \tag{10.7}$$

In particular, we have

$$\mathrm{Var}(\hat{w}_i) = \Delta^2 \left[(X^T X)^{-1}\right]_{ii} \ . \tag{10.8}$$

Proof of the proposition:

$$\mathbb{E}(\hat{w}) = \mathbb{E}\left((X^T X)^{-1} X^T y\right) \tag{10.9}$$
$$= \mathbb{E}\left((X^T X)^{-1} X^T (X w^* + \epsilon)\right) \tag{10.10}$$
$$= w^* + \mathbb{E}\left((X^T X)^{-1} X^T \epsilon\right) = w^* \tag{10.11}$$

the last line being because X and w^* are not considered to be random variables, and ϵ is a random variable with zero mean.

$$\mathbb{E}(\hat{w} - w^*)(\hat{w} - w^*)^T = \mathbb{E}\left((X^T X)^{-1} X^T \epsilon\epsilon^T X (X^T X)^{-1}\right) \tag{10.12}$$
$$= (X^T X)^{-1} X^T \Delta^2 I X (X^T X)^{-1} \tag{10.13}$$
$$= \Delta^2 (X^T X)^{-1} \tag{10.14}$$

where we used $\mathbb{E}(\epsilon_\mu \epsilon_\nu) = \Delta^2 \delta_{\mu,\nu}$ by independence.

These formulae are exact at any n, not only at large n as for general maximum-likelihood estimators.

Proposition 5. *An estimator of Δ is*

$$\hat{\Delta}^2 = \frac{1}{n} \sum_{\mu}^{n} (y_\mu - X_\mu^T \hat{w})^2 = \mathscr{L}(\hat{w}) \ . \tag{10.15}$$

We can see that if n is large and if $\hat{w} = w^*$ this estimator is correct. It is the empirical estimator of the variance. At large n, $\hat{w} = w^*$ is given by the consistency of the maximum likelihood method.

Theorem 10. *Let $\hat{w}$ be the least-square estimator and the assumptions of the probabilistic model satisfied. The variance of the regressed coefficient can be estimated by*

$$\hat{\mathrm{Var}}(\hat{w}_i) = \hat{\Delta}^2 \left[(X^T X)^{-1}\right]_{ii} \ . \tag{10.16}$$

For this to hold, we assumed: the probabilistic model (10.3), an independent additive Gaussian noise, n large. We do not take into account measurement error in the data X.

Example 9. *We consider the case $d = 2$: we regress $y = at + b$ over the scalar a and b. The more explicit formulae are then*

$$\mathrm{Var}(a) = \frac{1}{n^2} \frac{\sum_\mu^n (y_\mu - a t_\mu - b)^2}{\bar{t^2} - \bar{t}^2} \tag{10.17}$$
$$\mathrm{Var}(b) = \mathrm{Var}(a) \frac{1}{n} \sum_\mu t_\mu^2 \tag{10.18}$$

Derivation:

$$X = \begin{pmatrix} t_1 & 1 \\ \vdots & \vdots \\ t_n & 1 \end{pmatrix} \quad \text{and} \quad w_1 = a, w_2 = b \tag{10.19}$$
$$X^T X = n \begin{pmatrix} \bar{t^2} & \bar{t} \\ \bar{t} & 1 \end{pmatrix} \tag{10.20}$$
$$\text{and so} \quad (X^T X)^{-1} = \frac{1}{n(\bar{t^2} - \bar{t}^2)} \begin{pmatrix} 1 & -\bar{t} \\ -\bar{t} & \bar{t^2} \end{pmatrix} \ . \tag{10.21}$$

These expressions may be encountered in the TPs. There may be a difference, $\frac{1}{n-2}$ instead of the $\frac{1}{n}$. All our previous computations are done neglecting quantities of order $1/n$; using $n-2$, or more generally $n-d$, instead of n is overkill, in particular, because we used several times the consistency of the maximum likelihood estimator that requires $d \ll n$.

10.2 A model for the weighted least-squares

Definition
We consider a model where for each μ one has an estimate of the variance of the noise ϵ_μ that alters each output y_μ; typically, when one has error bars on the measurements. We explain how to perform linear regression in this case taking into account the estimated noise.

We consider X is fixed; we assume there is a w^* such that the observed outputs are generated according to

$$y_\mu = X_\mu^T w^* + \epsilon_\mu \quad ; \quad \epsilon_\mu \sim \mathcal{N}(0, \Delta_\mu^2) \ . \tag{10.22}$$

The ϵ_μ are independent normals of known variances Δ_μ^2. The variances vary from one data point to another; they are the error bars.

Proposition 6. *According to the model (10.22), the probability of the output is*

$$\rho(y|w) = \prod_\mu^n \rho(y_\mu|w) = \prod_\mu^n \frac{1}{\sqrt{2\pi\Delta_\mu^2}} e^{-(y_\mu - X_\mu^T w)^2/2\Delta_\mu^2} \ . \tag{10.23}$$

The corresponding maximum likelihood estimator is

$$\hat{w} = \operatorname*{argmin}_w \mathscr{L}(w) = (X^T \Omega X)^{-1} X^T \Omega y \tag{10.24}$$

$$\text{with} \quad \mathscr{L}(w) = \frac{1}{n} \sum_\mu^n \frac{1}{\Delta_\mu^2} (y_\mu - X_\mu^T w)^2 \tag{10.25}$$

$$\text{and with} \quad \Omega = \begin{pmatrix} 1/\Delta_1^2 & & 0 \\ & \ddots & \\ 0 & & 1/\Delta_n^2 \end{pmatrix} \ . \tag{10.26}$$

$\mathscr{L}$ is sometimes called the χ^2 loss. We emphasize that Ω is known.

Estimation of the error
The error on $\hat{w}$ can be computed.

Proposition 7. *The covariance of the weighted least-square estimator is*

$$\mathbb{E}(\hat{w} - w^*)(\hat{w} - w^*)^T = (X^T \Omega X)^{-1} \ . \tag{10.27}$$

In particular, we have

$$\mathrm{Var}(\hat{w}_i) = \left[(X^T \Omega X)^{-1}\right]_{ii} \ . \tag{10.28}$$

Proof of the proposition:

$$\mathbb{E}(\hat{w} - w^*)(\hat{w} - w^*)^T = \mathbb{E}\left((X^T \Omega X)^{-1} X^T \Omega \epsilon \epsilon^T \Omega X (X^T \Omega X)^{-1}\right) \tag{10.29}$$

$$= (X^T \Omega X)^{-1} \tag{10.30}$$

where we used $\mathbb{E}\epsilon_\mu \epsilon_\nu = \Delta_\mu^2 \delta_{\mu,\nu}$ by definition.

Consistency check

First we notice that

$$\mathbb{E}\mathscr{L}(\hat{w}) = \mathbb{E}\frac{1}{n} \sum_\mu^n \frac{1}{\Delta_\mu^2} (X_\mu^T w^* + \epsilon_\mu - X_\mu^T \hat{w})^2 \tag{10.31}$$

$$= \frac{1}{n} \sum_\mu^n \frac{1}{\Delta_\mu^2} \mathbb{E}\epsilon_\mu^2 \approx 1 \tag{10.32}$$

if indeed $\mathbb{E}\epsilon_\mu^2 \approx \Delta_\mu^2$ i.e. the error on the measurement is well estimated.

We can distinguish three cases. They are depicted on fig. 21.

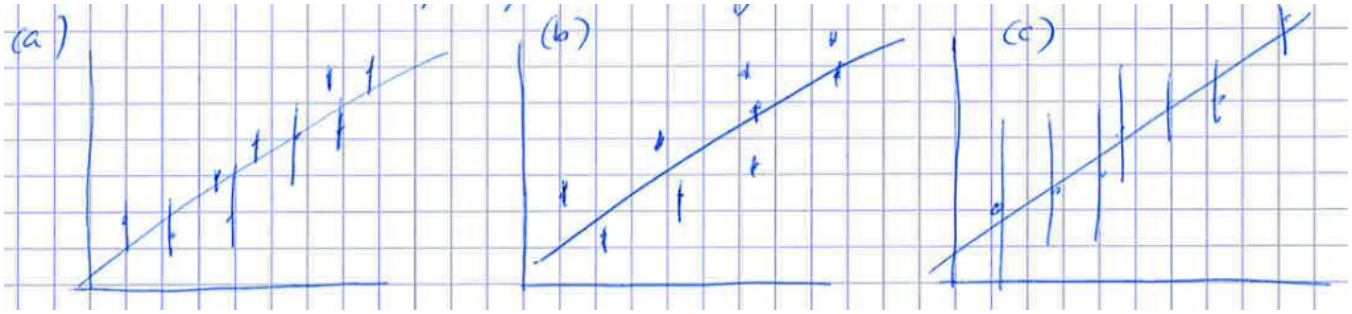

Figure 21: Consistency check between the fit and the error bars.

- Case a: the estimated error bars are consistent with the fit. $\mathscr{L}(\hat{w}) \approx 1$. We can use the weighted least-square estimator (10.24) as it will take into account the error bars in the fit and weight less the points with larger error bars.
- Case b: the estimated error bars are too small for the fit, they are underestimated. $\mathscr{L}(\hat{w}) \gg 1$. Instead we should use the estimate $\hat{\Delta}$ for the error; it will estimate it from the fluctuations of the points. The weighted least-square estimator should not be used; instead, the ordinary one (10.6).
- Case c: the estimated error bars are large compared to the fluctuations of the points, $\mathscr{L}(\hat{w}) \ll 1$. We should use the weighted least suqares (10.24) and its error estimation.

$\mathscr{L}(\hat{w})$ is an interesting quantity to evaluate the consistency in the sense that it is usable for $d > 2$, contrary to the visual check. More precise statements can be made thanks to statistical tests.

10.3 A Bayesian model for regularized least-squares

We present how having a prior on the weights leads to regularization.

The model is similar to the previous ones: we are given data $X \in \mathbb{R}^{n\times d}$ and outputs $y \in \mathbb{R}^n$. We assume there are w^* iid drawn at random such that

$$w_i^* \sim \mathscr{N}(0, \Delta_w^2) \tag{10.33}$$

$$y_\mu = X_\mu^T w^* + \epsilon_\mu \quad ; \quad \epsilon_\mu \sim \mathscr{N}(0, \Delta_\epsilon^2) \ . \tag{10.34}$$

We add a prior on the weights w^* that is a centered normal of variance Δ_w^2.

The resulting probabilities are

$$\rho(w) = \prod_i^d \frac{1}{\sqrt{2\pi\Delta_w^2}} e^{-w_i^2/2\Delta_w^2} \tag{10.35}$$

$$\rho(y|w) = \prod_\mu^n \frac{1}{\sqrt{2\pi\Delta_\epsilon^2}} e^{-(y_\mu - X_\mu^T w)^2/2\Delta_\epsilon^2} \ . \tag{10.36}$$

We recall the Bayes's formula:

$$\underbrace{\rho(w|y)}_{\text{posterior}} = \underbrace{\frac{1}{\rho(y)}}_{\text{normalization}} \underbrace{\rho(w)}_{\text{prior}} \underbrace{\rho(y|w)}_{\text{likelihood}} \ . \tag{10.37}$$

The corresponding estimator for w is the maximum a posterior estimator (MAP) defined by

$$\hat{w}_{\text{MAP}} = \operatorname*{argmax}_w \rho(w)\rho(y|w) \ . \tag{10.38}$$

This is no longer the maximum likelihood estimator: an additional prior multiplies the likelihood.

Maximizing the posterior distribution is equivalent to say that $\hat{w}_{\text{MAP}}$ minimizes the loss $\mathscr{L}^{\text{MAP}}$ defined by

$$\mathscr{L}^{\text{MAP}}(w) = \frac{1}{n}\sum_\mu^n (y_\mu - X_\mu^T w)^2 + \frac{1}{n}\frac{\Delta_\epsilon^2}{\Delta_w^2}\sum_i^d w_i^2 \ . \tag{10.39}$$

This is the loss associated to the regularized least-squares with regularization strength $\lambda = \Delta_\epsilon^2/\Delta_w^2$.

The regularization comes from the Gaussian prior on the weights. Taking a broader sight, we could use other priors to inject physical knowledge on the quantities we regress. For instance, for medical imagery we could precise that the pixels of the image should not vary too much locally.

10.4 Attention with high-dimensionality

The equations we stated are valid for n large compared to all the other parameters, in particular to the dimension d or the number of fitting parameters. This is because we used several times the consistency property of the maximum likelihood estimator. This is particularly important in machine learning where one often fits with number of parameters comparable or even larger than the number of samples. Statistics in high dimensions may behave rather differently even if the assumptions of the models are satisfied. Also, the error estimation is more subtle in this regime. In many practical settings the modeling assumptions leading to the expressions for error estimation are not valid. This is why in lecture 4 we introduced the concept of validation to estimate the prediction error using training, validation and test sets. Using such a data split is proper even in high dimensions and with relatively mild assumptions.

PART III

Stochasticity in Physical Systems

We now turn from using probability to describe measurement errors to using probability to model nature itself: Brownian motion, diffusion, and random walks; sampling with Markov chain Monte Carlo and the Metropolis rule; and phase transitions (percolation, random graphs, sharp thresholds), setting basis for how ideas from data analysis are connected to modelling in physics.

11 Lecture 11: Brownian motion, diffusion, random walks and maximum entropy principle

11.1 Brownian Motion, Diffusion and Random Walks

As a first example of how randomness is used to model physical phenomena, in this lecture, we will look at Einstein's theory of Brownian motion. Brownian motion is the random motion of particles suspended in a fluid. This motion pattern typically consists of random fluctuations in a particle's position inside a fluid. It is named after the botanist Robert Brown, who first described the phenomenon in 1827, while looking through a microscope at pollen immersed in water. To observe Brownian motion yourself you can look at this video https://www.youtube.com/watch?v=cDcprgWiQEY, and for an explanation of the underlying microscopic phenomenon look at this one https://www.youtube.com/watch?v=gPMVaAnij88. In short, Brownian motion is the result of many fluid (water) molecules bumping into the suspended particles. When it was discovered it constituted the first indirect proof of the existence of atoms (if water were a continuous substance even microscopically, there would be no bumping of particles). The experimental confirmation of the existence of atoms and molecules based on Brownian motion was advanced by Jean Perrin, who won the Nobel prize for it in 1926.

11.1.1 Brownian motion as a random walk

To analyze Brownian motion we start from a simple, one-dimensional probabilistic model called a random walk. We will consider a macroscopic particle moving on a one-dimensional line. We will call $x \in \mathbb{R}$ the position of the particle. We consider discrete-time t by letting $t = n\tau$, where τ is our time step and $n \in \mathbb{N}$. At each time step the particle gets displaced by a random amount Δ, giving the discrete-time dynamics

$$x(t+\tau) = x(t) + \Delta. \tag{11.1}$$

Since the bumps are random we assume Δ to be a random variable with probability density function $\varphi(\Delta) : \mathbb{R} \to \mathbb{R}$. By normalization we have $\int_{\mathbb{R}} \varphi(\Delta) d\Delta = 1$. For symmetry reasons (there is no reason why there should be more kicks from the left than from the right or vice versa) we assume $\mathbb{E}[\Delta] = \int_{\mathbb{R}} \varphi(\Delta)\Delta \, d\Delta = 0$. Finally we assume Δ to have a finite variance $\mathrm{Var}(\Delta) = \int_{\mathbb{R}} \Delta^2 \varphi(\Delta) \, d\Delta$. Importantly, we assume that the random variable Δ in one time step is independent of Δ's in all the other steps. Displacement corresponding to different steps are i.i.d. random variables. This can be physically motivated by the fact that there is no dependence between the bumps of the molecules with the particle.

We're now interested in answering the following question: suppose we take n steps ($t = n\tau$) of this dynamics, how much will the particle move?

We use two approaches to answer this question: in the first we will take the continuous limit of this process and derive the Diffusion Equation. In the second one we will apply the Central limit theorem to the dynamics (11.1).

11.1.2 Derivation of the Diffusion equation

Let $\rho(x,t)$ be the probability density of x at time t. Put differently we have $\rho(x,t)dx = \mathrm{Prob}\,(x(t) \in [x, x+dx])$. You can also picture that there are many macroscopic particles and $\rho(x,t)$ is simply their density at position x and

time t. As every p.d.f., also ρ is normalized as $\int_{\mathbb{R}} \rho(x,t)dx = 1$, for all $t \in \mathbb{R}$. If we can compute $\rho(x,t)$ for all x, t then we will know on average how far $x(t)$ is from the origin, thus answering our original question.

We're now ready to write the evolution equation for ρ.

$$\rho(x, t+\tau) = \int_{\mathbb{R}} \rho(x-\Delta, t)\varphi(\Delta)\, d\Delta \tag{11.2}$$

Here we are summing over all possible positions x' of the particle at time t that in one step end up at the position x. As in one step the displacement is Δ we have $x' = x + \Delta$ and we must multiply by the probability that such a Δ occurred. We now take the limit of $\tau \to 0$ and $\mathrm{Var}(\Delta) \to 0$. In this limit the particle is hit by many molecules (if $t = n\tau$ is fixed and $\tau \to 0$, then $n \to \infty$), and in each bump, the particle moves by a very small amount (since the molecules are much smaller than the macroscopic particle). Since Δ is small we can expand $\rho(x-\Delta, t)$ to second order in $x - \Delta$. This gives

$$\rho(x-\Delta, t) = \rho(x,t) - \Delta \frac{\partial \rho}{\partial x}(x,t) + \frac{1}{2}\Delta^2 \frac{\partial^2 \rho}{\partial x^2} \tag{11.3}$$

Plugging it into the right-hand side of (11.2) we get

$$\rho(x, t+\tau) = \int_{\mathbb{R}} d\Delta \varphi(\Delta) \left[\rho(x,t) - \Delta \frac{\partial \rho}{\partial x}(x,t) + \frac{1}{2}\Delta^2 \frac{\partial^2 \rho}{\partial x^2}\right] + o(\Delta^2) = \tag{11.4}$$

$$= \rho(x,t)\, \mathbb{E}_\Delta[1] - \frac{\partial \rho}{\partial x}(x,t)\, \mathbb{E}_\Delta[\Delta] + \frac{1}{2}\frac{\partial^2 \rho}{\partial x^2}(x,t)\, \mathbb{E}_\Delta[\Delta^2] + o(\Delta^2), \tag{11.5}$$

where $\mathbb{E}_\Delta[f(\Delta)] = \int_{\mathbb{R}} f(\Delta)\varphi(\Delta)d\Delta$. Using that $\mathbb{E}_\Delta[1] = 1$, $\mathbb{E}_\Delta[\Delta] = 0, \mathbb{E}_\Delta[\Delta^2] = \mathrm{Var}(\Delta)$ we get

$$\rho(x, t+\tau) = \rho(x,t) + \frac{1}{2}\frac{\partial^2 \rho}{\partial x^2}(x,t)\, \mathrm{Var}(\Delta) + o(\Delta^2). \tag{11.6}$$

To complete our computation we expand $\rho(x, t+\tau)$ to second order in τ. This gives $\rho(x,t+\tau) = \rho(x,t) + \tau \frac{\partial \rho}{\partial t}(x,t) + \frac{1}{2}\tau^2 \frac{\partial^2 \rho}{\partial t^2}(x,t) + o(\tau^2)$. Plugging it back into the left hand side of (11.2) and canceling $\rho(x,t)$ on both sides we obtain

$$\tau \frac{\partial \rho}{\partial t}(x,t) + \frac{1}{2}\tau^2 \frac{\partial^2 \rho}{\partial t^2} + o(\tau^2) = \frac{1}{2}\frac{\partial^2 \rho}{\partial x^2}(x,t)\, \mathrm{Var}(\Delta) + o(\Delta^2). \tag{11.7}$$

We still have to decide how to send $\mathrm{Var}(\Delta), \tau \to 0$. We decide to fix the ratio between $\mathrm{Var}(\Delta)$ and τ to be of order 1: $\mathrm{Var}(\Delta)/\tau = O(1)$. With this choice the term $\frac{\partial \rho}{\partial t}(x,t)$ will be of the same magnitude as $\frac{\partial^2 \rho}{\partial x^2}(x,t)$. We can thus drop the second derivative in t since this is weighted with τ^2 which goes to zero faster and is thus negligible. We then divide everything by τ and define a constant $D \equiv \mathrm{Var}(\Delta)/2\tau$. This way we finally arrive to the diffusion equation, or heat equation, or Fokker-Planck equation. We put it in a box because of its particular importance

$$\boxed{\frac{\partial \rho}{\partial t}(x,t) = D\frac{\partial^2 \rho}{\partial x^2}(x,t), \qquad D = \frac{\mathrm{Var}(\Delta)}{2\tau}.} \tag{11.8}$$

D is called the diffusion coefficient.

11.1.3 Solving the Diffusion Equation

Eq. (11.8) is a partial differential equation (PDE): unlike ordinary differential equations it contains derivatives with respect to multiple variables. This also makes PDEs generally harder to solve. Solving PDEs is beyond the scope of this course, it will be covered in courses next semester. Here we will content ourselves with exhibiting a solution.[10]

Take as initial condition $x(t=0) = 0$. Said differently, we suppose that at time zero we know that the particle is located at $x = 0$, thus $\rho(x,t)$ is extremely peaked at $x = 0$ and zero everywhere else. Under this initial condition the solution to (11.8) is

$$\rho(x,t) = \frac{1}{\sqrt{4\pi D t}} \exp\left[-\frac{x^2}{4Dt}\right] = \mathscr{N}(0, 2Dt) = \mathscr{N}\left(0, t\frac{\mathrm{Var}(\Delta)}{\tau}\right) \tag{11.9}$$

[10] To get familiar with the Diffusion equation you can play with the interactive simulator at https://visualpde.com/sim/?preset=heatEquation1D. In the menu on the top left select "Dirichlet" as boundary condition.

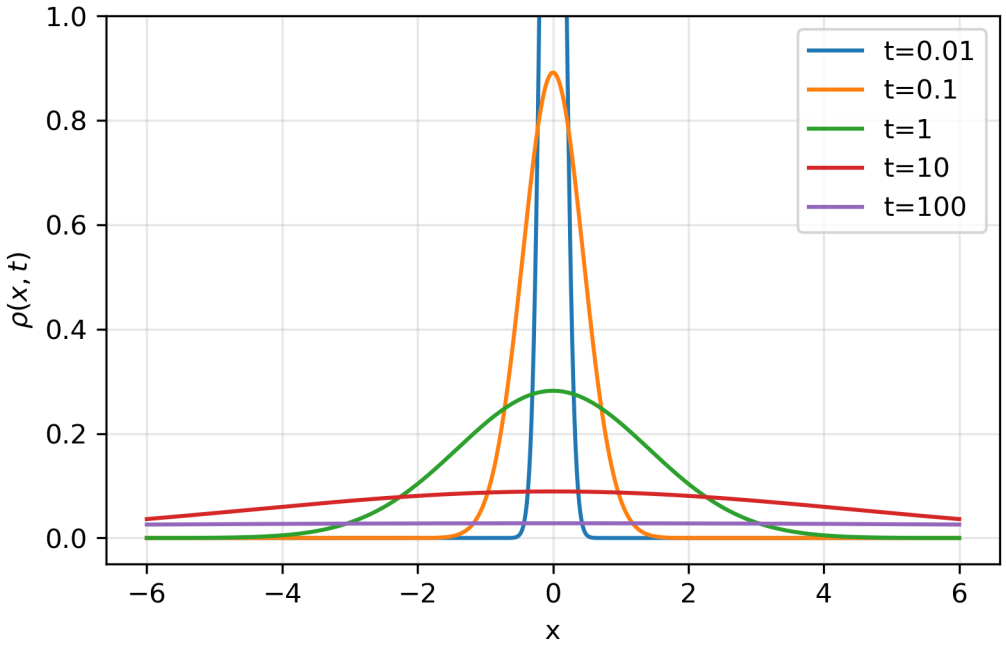

Figure 22: Evolution of the initial condition $\rho(x, t = 0) = \delta(x)$ under the diffusion equation with $D = 1$.

This is just a Gaussian distribution with mean zero and a $\mathrm{Var}(x(t)) = 2Dt$ which grows linearly in time. The evolution of ρ is portrayed in figure 22. This solution also provides some intuition over the nature of the initial condition. We used what is called a Dirac delta initialization, indicated with $\delta(x)$. This can be seen as a function which is infinitely peaked at 0. It also corresponds to the limit of zero variance of a Gaussian $\delta(x) = \lim_{\epsilon\to 0} \mathcal{N}(0, \epsilon)$. Indeed setting $t \to 0$, it's exactly what happens in this example. Mathematically the Dirac delta is a *distribution* (which is not the same distribution as in "probability distribution"). More properly speaking, distributions only make sense when they're integrated against another function. In this sense the Dirac delta is the distribution with the following property $\int_{\mathbb{R}} \delta(x)f(x)dx = f(0)$. $\delta(\cdot)$ evaluates the function at the point where the argument of the delta is zero. In the end we learned that the initial condition we imposed in the beginning can be written as $\rho(x, 0) = \delta(x)$.

The equation we found describes the evolution of the p.d.f. of a single particle in a fluid. How is it linked to diffusion? Diffusion is concerned with the evolution of the concentration of particles in a fluid. In the diffusion case one has N particles moving through the fluid. All particles move independently of each other, and each is a realization of the random process (11.1). We can then look at $\rho_N(x, t) \equiv \frac{1}{N} \times$ (number of particles in $[x, x + dx]) \times \frac{1}{dx}$. By the law of large numbers, when $N \to \infty$, $\rho_N(x, t) \to \mathbb{E}[\rho_N(x, t)] = \rho(x, t)$, since each particle has probability $\rho(x, t)dx$ of being in the interval $[x, x + dx]$. Consequently, equation (11.8) also holds when ρ is the concentration of particles in a fluid.

11.1.4 Probabilistic solution of the Brownian motion

In this section, we will derive the solution (11.9) by thinking in terms of the discrete-time random walk and probability theory. The position $x(t)$ of the particle in time $t = n\tau$ obeys the dynamics $x(t+\tau) = x(t)+\Delta$. Taking $x(t = 0) = 0$, we can write $x(t = n\tau) = \sum_{i=1}^{n} \Delta_i$. Let's switch to the notation $x_n \equiv x(t = n\tau)$, which puts more emphasis on the number of steps. We have $x_n = \sum_{i=1}^{n} \Delta_i$, with $\Delta_i \sim \varphi(\Delta)$, i.i.d.. From the properties of $\varphi(\cdot)$ we immediately see that $\mathbb{E}(x_n) = 0$ and $\mathrm{Var}(x_n) = n\,\mathrm{Var}(\Delta)$.

Consider the case where we take many steps and therefore $n \to \infty$. Then we can apply the Central Limit Theorem (CLT) to x_n. The CLT states that

$$\frac{x_n - \mathbb{E}[x_n]}{\sqrt{\mathrm{Var}(x_n)}} \xrightarrow{d} \mathcal{N}(0, 1), \tag{11.10}$$

where $\xrightarrow{d}$ indicates convergence in distribution. Multiplying by $\sqrt{\mathrm{Var}(x_n)} = \sqrt{n\,\mathrm{Var}(\Delta)}$, we get that the distribution of x_n converges to $\mathcal{N}(0, t\frac{\mathrm{Var}(\Delta)}{\tau})$. This is exactly the same result we obtained in (11.9). It's important that for large n, the distribution of x_n is Gaussian independently of the details of the distribution φ. This is an example of

universality, a recurring theme in physics. In presence of universality the microscopic details (in this case the precise nature of the noise) of a system become irrelevant, and the macroscopic behaviour depends only on a few properties of the microscopic system (in this case, the diffusion coefficient).

In physics the diffusion coefficient D can be related to Boltzmann's constant k_b and to the viscosity of the fluid. You will see this in the TPs. Through the relation $k_b = R/N_A$ (where $R = 8.314\, J\, K^{-1}\, mol^{-1}$ is the ideal gas constant) one can relate the diffusion coefficient to Avogadro's number N_A and therefore estimate N_A.

11.1.5 Note on Stochastic Differential Equations

Let's consider once again the discrete-time dynamics $x(t+\tau) = x(t) + \Delta_t$, where we added the index t in Δ_t to stress that this is an independent random variable at every time step. As in the derivation of the diffusion equation we take the $\tau \to 0, \mathrm{Var}(\Delta) \to 0$ limit with $\tau/\mathrm{Var}(\Delta) = O(1)$. Our goal is to obtain a continuous-time differential equation that governs the single trajectory $x(t)$ when $\tau \to 0$ (instead of the density as we did before). Taking $\tau \to 0$ we aim to work in the regime where $t \gg dt \gg \tau$. dt here is the time differential. Intuitively this means that each dt corresponds to many steps (as measured by τ), giving $x(t+dt) = x(t) + \sum_{i=t/\tau}^{t/\tau+dt/\tau} \Delta_{i\tau}$. Since there are many elements in the sum ($dt/\tau \gg 1$) we can apply the CLT. Hence the noise at scale dt can already be considered Gaussian.

We derived previously that in a time step dt the noise variance is $2Ddt$. To formalize the noise in the continuous-time limit we introduce another differential $dB_t \equiv "\sqrt{dt}\, r"$, with $r \sim \mathcal{N}(0,1)$. The "", signify that square root of a differential is not a properly defined object.

The equation describing the evolution of x becomes

$$dx(t) = \sqrt{2D}\, dB_t \tag{11.11}$$

This is a simple example of a Langevin equation. These equations are written without dividing by dt. This can be done also for ordinary differential equations: instead of writing $dx/dt = f(x)$, one writes $dx = f(x)dt$. The Langevin equation therefore allows to describe a single realization of the process x. In contrast the Fokker-Planck equation (11.8) describes $x(t)$ in terms of its distribution. One can always switch between the two formulations and find the Fokker-Planck equation corresponding to a certain Langevin equation or vice-versa.

11.2 Maximum Entropy Principle

Suppose we have a random variable $x \in \mathbb{R}$ with unknown distribution. The only things we know about x are that $x > 0$ always (meaning $\rho(x) = 0$ for $x \leq 0$) and we know its expected value $\mu = \mathbb{E}[x]$. We want to find $\rho(x) : \mathbb{R} \mapsto \mathbb{R}$, the p.d.f. of x under these two constraints. Of course there are many distributions satisfying the positivity and expected value constraints: which one should we pick? The maximum entropy principle selects the distribution with maximum entropy compatible with the constraints. The continuous version of Shannon's entropy (as seen in the ICC course) is defined as

$$H(\rho) \equiv -\int_{\mathbb{R}} \rho(x) \log \rho(x)\, dx. \tag{11.12}$$

To give an example of the entropy for a continuous probability distribution, the entropy of a Gaussian distribution $H(\mathcal{N}(0,\sigma^2)) = \frac{1}{2}\log\left(2\pi e \sigma^2\right)$. The larger σ, the more spread the Gaussian, the larger the entropy.

In the ICC course, you learned that the entropy is related to the compressibility of the random variable. The larger the entropy, the more bits will be needed to transmit a sequence of random variables on average. Conversely, the lower the entropy the easier it is to predict the next element in the sequence. Maximizing the entropy is thus equivalent to searching for a probability distribution for which it is the hardest to predict the next element in a sequence of random variables. In physics, the maximum entropy principle stands at the basis of statistical mechanics and you will learn about the physical reasons behind it in the statistical physics course.

11.2.1 Lagrangian approach to entropy maximization

To maximize the entropy we use Lagrange multipliers to enforce the normalization constraint and the expected value constraint. The positivity constraint $\rho(x) = 0$ for $x \leq 0$ will be taken into account at the end. The Lagrangian is

$$\mathscr{L}(\rho(\cdot), \lambda, \gamma) = -\int_{\mathbb{R}} \rho(x) \log \rho(x) dx + \gamma \left(\int_{\mathbb{R}} \rho(x) dx - 1 \right) + \lambda \left(\int_{\mathbb{R}} x\rho(x)\, dx - \mu \right) \tag{11.13}$$

We indicate the maximum entropy p.d.f. and the value of λ, γ at optimality respectively as $\rho_{\mathrm{me}}, \lambda_{\mathrm{me}}, \gamma_{\mathrm{me}}$,

$\mathscr{L}$ will be stationary with respect to ρ, λ, γ. The stationarity with respect to λ, γ gives respectively

$$\frac{\partial \mathscr{L}}{\partial \gamma}(\rho_{\text{me}}, \lambda_{\text{me}}, \gamma_{\text{me}}) = 0 \iff \int_{\mathbb{R}} \rho_{\text{me}}(x)dx = 1 \tag{11.14}$$

$$\frac{\partial \mathscr{L}}{\partial \lambda}(\rho_{\text{me}}, \lambda_{\text{me}}, \gamma_{\text{me}}) = 0 \iff \int_{\mathbb{R}} x\rho_{\text{me}}(x)dx = \mu \tag{11.15}$$

In words, the stationarity with respect to the multipliers enforces the normalization and expected value constraints. The stationarity with respect to ρ is more delicate. In fact this Lagrangian is a functional, meaning it takes a function, $\rho(\cdot)$, as an argument. The standard way to deal with this is through functional derivatives. Another option is to use Euler-Langrange equations. To do this rewrite

$$\mathscr{L}(\rho(\cdot), \gamma, \lambda) = \int_{\mathbb{R}} \underbrace{[-\rho(x)\log\rho(x) + \gamma\rho(x) + \lambda x\rho(x)]}_{L(\rho,\lambda,\gamma)} dx - \gamma - \lambda\mu \tag{11.16}$$

Now L defines a functional to which we can apply the Euler-Lagrange equation, and the other two terms do not involve ρ. The Euler-Lagrange equations applied to L read $\frac{\partial L}{\partial \rho}(\rho_{\text{me}}, \lambda_{\text{me}}, \gamma_{\text{me}}) = 0$ (here we simply treat ρ as a scalar variable). Hence we have the equation

$$-\log\rho_{\text{me}}(x) - 1 + \gamma_{\text{me}} + \lambda_{\text{me}}x = 0 \tag{11.17}$$

This is equivalent to

$$\rho_{\text{me}}(x) = e^{\gamma_{\text{me}}-1}e^{\lambda_{\text{me}}x} \tag{11.18}$$

Adjusting the multipliers so that the p.d.f. is normalized on $[0, \infty)$ and has mean μ we obtain

$$\rho_{\text{me}}(x) = \frac{1}{\mu}e^{-x/\mu} \tag{11.19}$$

Therefore the maximum entropy distribution for positive random variable with fixed mean is the exponential distribution.

11.2.2 Maximum entropy for multi-variate case, the Boltzmann distribution

Let $\mathscr{H}(\vec{x}) : \mathbb{R}^d \to \mathbb{R}$ be a function of d random variables. The following argument is particularly important for a function that is called Hamiltonian of a physical system. Variables $\vec{x} \in \mathbb{R}^d$ of the Hamiltonian are considered random with an unknown distribution $\rho(\vec{x})$.

Suppose we know that on average the value of the Hamiltonian is E

$$\int_{\mathbb{R}^d} \mathscr{H}(\vec{x})\rho(\vec{x})d\vec{x} = E \tag{11.20}$$

This E is then interpreted as the average energy E of a physical system.

Repeating the entropy maximization argument one obtains that the maximum entropy distribution is

$$\rho_{\text{me}}(\vec{x}) = \frac{1}{Z}e^{-\tilde{\lambda}\mathscr{H}(\vec{x})}, \tag{11.21}$$

where $\tilde{\lambda} = -\lambda$ (here λ is the multiplier from the previous example, the sign is chosen to comply with the conventional way of writing this), and Z is a normalization constant called the partition function in statistical physics. By setting $\tilde{\lambda} = (k_b T)^{-1}$ we obtain what is a crucial object in statistical physics – the Boltzmann's distribution. Statistical physics then proceeds to derive thermodynamic properties from this distribution as you will learn in the corresponding course. The appearance of the inverse temperature as a Lagrange multiplier of maximizing the entropy under the constraint of fixed energy is a remarkable result.

What is to be retained from this note on Boltzmann distribution is the fact that statistical physics is a probabilistic theory where the key object is a probability distribution over all possible microstates of the system. The fact that a deterministic theory such as thermodynamics can be derived from an intrinsically probabilistic formulation has to do with the fact that the dimensionality d is in physics proportional to the number of particles and the limit of large d becomes deterministic for many quantities, just as we saw in the law of large numbers and in the central limit theorem.

11.2.3 Gaussian as max entropy distributon

Consider now the case in which we know the mean μ and variance σ^2 of a distribution $\rho : \mathbb{R} \mapsto \mathbb{R}$. This amounts to the constraints

$$\int_{\mathbb{R}} x\rho(x)\,dx = \mu, \qquad \int_{\mathbb{R}} (x-\mu)^2\rho(x)\,dx = \sigma^2 \tag{11.22}$$

Once again, we want to find the maximum entropy distribution under these constraints. The Lagrangian reads

$$\mathscr{L}(\rho,\gamma,\lambda_1,\lambda_2) = -\int_{\mathbb{R}} \rho(x)\log\rho(x)\,dx + \gamma\left(\int_{\mathbb{R}}\rho(x)\,dx - 1\right) + \lambda_1\left(\int_{\mathbb{R}} x\rho(x)\,dx - \mu\right) + \lambda_2\left(\int_{\mathbb{R}}(x-\mu)^2\rho(x)\,dx - \sigma^2\right) \tag{11.23}$$

At optimality (indicated by the superscripts $^{\text{me}}$), the Euler-Lagrange equations impose that

$$\frac{\partial}{\partial\rho^{\text{me}}(x)}\left(-\rho^{\text{me}}(x)\log\rho^{\text{me}}(x) + \gamma^{\text{me}}\rho^{\text{me}}(x) + \lambda_1\rho(x)x + \lambda_2^{\text{me}}\rho^{\text{me}}(x)(x-\mu)^2\right) = 0 \tag{11.24}$$

Carrying out the derivative (keep in mind we always treat ρ as a scalar variable) we get

$$-\log\rho^{\text{me}}(x) - 1 + \gamma^{\text{me}} + \lambda_1^{\text{me}}x + \lambda_2^{\text{me}}(x-\mu)^2 = 0. \tag{11.25}$$

Taking the exponential in both sides gives

$$\rho^{\text{me}}(x) = \exp\left[\gamma^{\text{me}} - 1 + \lambda_1^{\text{me}}x + \lambda_2^{\text{me}}(x-\mu)^2\right], \tag{11.26}$$

where the value of the multipliers is determined by enforcing the respective constraints. Looking closer at this expression we see that this is the exponential of a degree two polynomial in x. This together with the normalization constraint tells us that (11.26) is a Gaussian distribution with mean μ and variance σ^2. To conclude we have

$$\rho^{\text{me}}(x) = \frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{1}{2\sigma^2}(x-\mu)^2}. \tag{11.27}$$

In conclusion, the Gaussian distribution is the one that maximizes the Shannon entropy under the constrained mean and variance.

11.2.4 The multidimensional case, note on generative models

We can easily extend this result to the vectorial case. Consider $\vec{x} \in \mathbb{R}^d$, and $\rho : \mathbb{R}^d \mapsto \mathbb{R}$ a p.d.f. We constrain the mean and covariance of ρ.

$$\int_{\mathbb{R}^d} x_i\rho(\vec{x})\,d\vec{x} = \mu_i,\ \forall i \in \{1,2,\dots,d\} \tag{11.28}$$

$$\int_{\mathbb{R}^d} x_ix_j\rho(\vec{x})\,d\vec{x} - \mu_i\mu_j = \Sigma_{ij},\ \forall i,j \in \{1,2,\dots,d\}. \tag{11.29}$$

$\vec{\mu} = (\mu_1,\dots,\mu_d)$ specifies the expected value of each component of $\vec{x}$. $\Sigma \in \mathbb{R}^{d\times d}$ is the covariance matrix. Σ_{ij} is the covariance between the components x_i and x_j if the vector random variable $\vec{x}$. Notice that $\Sigma_{ii} = \text{Var}(x_i)$. The maximum entropy distribution under these constraints is the multivariate Gaussian

$$\rho^{\text{me}}(\vec{x}) = \frac{1}{(2\pi)^{d/2}\sqrt{\det\Sigma}}\exp\left[-\frac{1}{2}(\vec{x}-\vec{\mu})^T\Sigma^{-1}(\vec{x}-\vec{\mu})\right]. \tag{11.30}$$

Suppose we have access to some data $X \in \mathbb{R}^{n\times d}$, where each row of X is a d dimensional point sampled from an unknown distribution $\rho : \mathbb{R}^d \mapsto \mathbb{R}$. Our goal is to draw more samples from ρ, having access only to X. A possible strategy is to compute the empirical mean and covariance from X and then sample from the distribution that has the same mean and variance and has maximum entropy given those constraints. From the above this is then corresponding to the multivariate Gaussian distribution. In formulas, we first find the empirical mean and covariance as

$$\hat{\mu}_i = \frac{1}{n}\sum_{\mu=1}^{n} X_{\mu i},\ i \in \{1,\dots,d\}, \quad \hat{\Sigma}_{ij} = \frac{1}{n}\sum_{\mu=1}^{n}(X_{\mu i}X_{\mu j} - \hat{\mu}_i\hat{\mu}_j),\ i,j \in \{1,\dots,d\}. \tag{11.31}$$

Then we sample from $\mathscr{N}(\hat{\mu},\hat{\Sigma})$.

In machine learning generalizations of this concept are called a Boltzmann machine and constitutes some of the early machine learning tools. The 2024 Physics Nobel prize to Hopfield and Hinton is for closely related concepts. The inverse covariance matrix Σ^{-1} is closely related to the attention matrices in current state-or-the-art generative models based on transformers. The elements of this matrix summarize how one component of the data pays attention to another component.

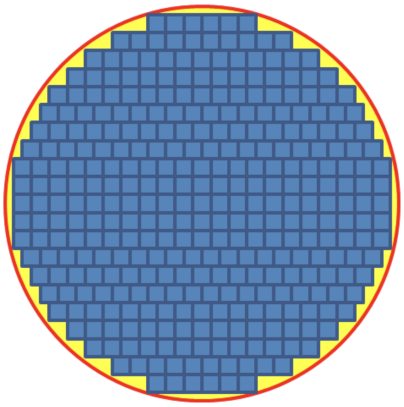

Figure 23: Tiling the circle with squares in order to use direct sampling. The yellow region represents the error we make in approximating the circle with the square tiling.

12 Lecture 12: Sampling with Monte Carlo Markov Chains

12.1 Rejection Sampling: The curse of dimensionality

In section 8.2, we saw how to estimate π via drawing random variables. Our strategy was to sample two independent random variables $y_1, y_2 \sim \text{Unif}([-1,1])$, and then check whether the point (y_1, y_2) lies within the circle of radius one. To do so, we defined a variable $x = \mathbb{I}[y_1^2 + y_2^2 \leq 1]$. The expected value of x is $\mathbb{E}[x] = \pi/4$. By drawing repeated samples$\{(y_1)_i, (y_2)_i\}_{i=1}^n$, giving $\{x_i\}_{i=1}^n$ we can estimate the expected value of x and hence π.

This is an example of using random variables to compute integrals. In this case, the integral we wanted to estimate was simply $\int_{y_1^2+y_2^2\leq 1} 1\, dy_1\, dy_2$. It is also an example of building an algorithm that samples uniformly points inside the unit circle. First, one samples $\{(y_1)_i, (y_2)_i\}_{i=1}^n$ uniformly in the square, then one eliminates (rejects) all the samples with $x_i = 0$. This leaves the subset $S = \{(y_1)_i, (y_2)_i \text{ s.t. } (y_1)_i^2 + (y_2)_i^2 \leq 1, i \in \{1, \ldots, n\}\}$. The points in S are sampled uniformly in the circle. This is an example of the so-called *rejection method*: one first samples from a 'simpler' distribution (uniform on the square in this case) and then only keeps the samples that fall in the desired region.

An alternative simple method to sample uniformly in the circle is a variant of the direct sampling explained in section 6.1. The idea is to divide the circle into many small squares of side $\epsilon < 1$. We tile the circle with squares of side ϵ. This will require about π/ϵ^2 squares (See figure 23). When we want to sample uniformly in the circle, we first pick a square uniformly at random. Let $\tilde{y}$ be the center of this square. Then we generate the position within the square by sampling $\delta \sim \text{Unif}([-\epsilon/2, \epsilon/2]^2)$. Finally, set $y = \tilde{y}^t + \delta$. Of course this point does not exactly come from the uniform distribution on the circle, since it is not possible to tile perfectly the circle with squares. However, the error of this approximation decreases with ϵ.

12.1.1 Sampling in the high dimensional ball

We now aim to extend the sampling method to higher dimensions. The goal is still to draw samples uniformly in the ball. One approach, roughly based on the inversion method, works as follows. Choose a number $\epsilon \leq 1$. We now tessellate the ball with hypercubes of side ϵ

With the inversion of the cumulative distribution method we would need to split the d-dimensional ball in L^d small hypercubes to get a resolution $1/L$, this is exponentially large in d and thus not computationally efficient in large dimension.

Let us hence try the rejection method, i.e., first drawing samples uniformly in the hypercube and then keeping only those that fall inside the unit sphere. Formally, this method works as follows: First generate $Y \in \mathbb{R}^{n\times d}$ where $Y_{\mu i} \sim \text{Unif}([-1,1])$ i.i.d. for all $\mu \in 1, \ldots, n$, for all $i \in \{1, \ldots, d\}$. Y is a matrix with n rows and d columns. Each row is a d dimensional vector uniformly sampled in the d-dimensional hypercube $[-1,1]^d$. For each $\mu \in \{1, \ldots, n\}$ compute $\|Y_\mu\|^2 = \sum_{i=1}^d Y_{\mu i}^2$. If $\|Y_\mu\|^2 \leq 1$ then Y_μ belongs to the d-dimensional unit radius ball (unit ball from now on).[11] We get a set S of samples uniformly distributed in the unit ball by taking $S_n = \{Y_\mu, \text{ s.t. } \mu \in \{1, \ldots, n\}, \|Y_\mu\|^2 \leq 1\}$.

[11] Are a sphere and a ball the same thing? No, the sphere is the surface of the ball. Take a balloon: the air in the balloon is the ball, and the rubber surface is the sphere.

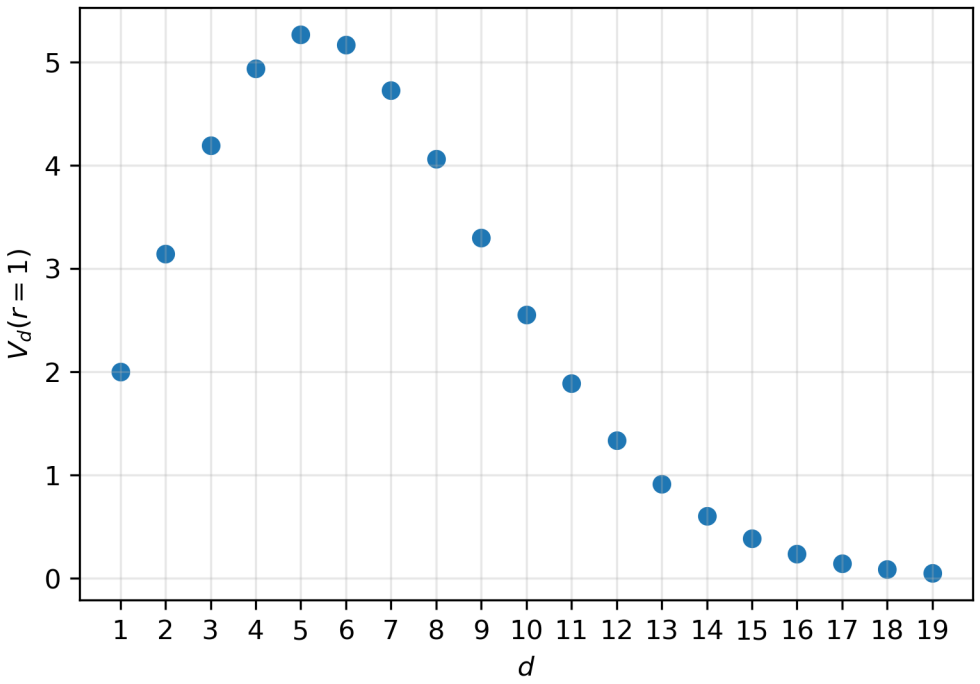

Figure 24: Volume of the unit radius ball in dimension d.

Is this an efficient way of drawing samples from the unit ball? In other words, how many samples will S_n contain? For a single sample, the probability of being in the unit ball is given by the ratio of the volume of the unit ball (here indicated by V_d) to that of the hypercube, which is 2^d (the product of the sides).

$$P(\|Y_\mu\|^2 \leq 1) = \frac{V_d(r=1)}{2^d} \tag{12.1}$$

$V_d(r)$ admits the analytical expression

$$V_d(r) = \frac{\pi^{d/2}}{\Gamma(\frac{d}{2}+1)} r^d, \tag{12.2}$$

where $\Gamma(\cdot)$ is the gamma function. For integer arguments we have $\Gamma(x) = (x-1)! = (x-1)(x-2)\times\cdots\times 1$. To look at the behavior of V_d when $d \to \infty$ we use Stirling's formula to approximate the gamma function. Take d to be even and set $d' = d/2$. In this case $\Gamma(d/2+1) = \Gamma(d'+1) = d'!$. To approximate the factorial we use Stirling's formula: $d'! \approx (d'/e)^{d'}$. In the end, we get $V_d(1) = (e\pi)^{d'}/d'^{d'} \approx d'^{-d'}$, which goes very fast to zero since the denominator grows faster. To see this numerically in Figure 24 we plot the volume of the unit ball as a function of the dimension. Notice the quick decrease when d increases.

From $V_d(r=1)$ to get the probability of being in the unit ball, we further have to divide by 2^d. In conclusion $P(\|Y_\mu\|^2 \leq 1) \leq (d/2)^{-d/2}$ for large d. This implies that to generate one sample inside the unit ball, on average we need to sample more than $(d/2)^{(d/2)}$ times from the hypercube. Clearly, this is extremely inefficient for large d. For example for $d=6$, $P(\|Y_\mu\|^2 \leq 1) \approx 0.2\%$, and for $d=10$, $P(\|Y_\mu\|^2 \leq 1) \approx 0.004\%$.

12.1.2 Watermelons in high dimension

Let's look again at the expression for the volume of the sphere in dimension d. This time, we focus on the dependence on the radius r. Take two concentric balls: one with radius r and another with radius $r-\epsilon$. Their respective volumes are $V_d(r) = c_d r^d$,$V_d(r-\epsilon) = c_d(r-\epsilon)^d$. Here $c_d = \frac{\pi^{d/2}}{\Gamma(\frac{d}{2}+1)}$. The difference between these two balls is the external shell of thickness ϵ.

What proportion of the volume of the ball of radius r is contained in this shell? The volume of the shell is just $V_d(r) - V_d(r-\epsilon)$. In the end we get

$$\text{fraction of volume in the shell} = \frac{V_d(r) - V_d(r-\epsilon)}{V_d(r)} = \frac{c_d(r^d - (r-\epsilon)^d)}{c_d r^d} = 1 - \left(\frac{r-\epsilon}{r}\right)^d \tag{12.3}$$

$((r-\epsilon)/r)^d$ decays exponentially in d, so we see that almost all the volume in high dimensional spheres is contained near the boundary. This is one of many counter-intuitive phenomena of high dimensional geometry. As a consequence watermelons in high dimensions are almost completely made up of the green part. Good thing we live in 3d!

12.2 Markov Chain Monte Carlo (MCMC)

MCMCs are a class of sampling algorithms that allow sampling from a desired probability distribution even in high dimensional spaces. They work by producing a sequence of correlated samples from the desired probability measure. Let's first look at this approach for sampling the circle's interior, i.e., we set $d = 2$. We indicate with $y^t = (y_1^{(t)}, y_2^{(t)}) \in \mathbb{R}^2$ the configuration at time $t \in \mathbb{N}$. At the time t the MCMC works by first proposing a next possible configuration $\tilde{y}^{(t+1)}$. If $\tilde{y}^{(t+1)}$ falls inside the circle one just sets $y^{(t+1)} = \tilde{y}^{(t+1)}$. Otherwise if $\tilde{y}^{(t+1)}$ falls outside the circle, further action is needed. The proposed step $\tilde{y}^{(t+1)}$ is computed by changing the current state according to the following equation

$$\tilde{y}_1^{(t+1)} = y_1^{(t)} + \Delta_1^{(t)} \tag{12.4}$$

$$\tilde{y}_2^{(t+1)} = y_2^{(t)} + \Delta_2^{(t)}, \tag{12.5}$$

where $\Delta_1^{(t)}, \Delta_2^{(t)}$ are random variables. One could choose the randomness $\Delta_1^{(t)}, \Delta_2^{(t)}$ in several ways. Here are some examples:

(a) At each time draw and angle $\theta^{(t)} \sim \text{Unif}([0, 2\pi])$. Then propose a step of size δ is the direction indicated by θ^t. In formulas set $\Delta_1^{(t)} = \delta \cos\big(\theta^{(t)}\big)$, $\Delta_1^{(t)} = \delta \sin\big(\theta^{(t)}\big)$ for some $\delta \in \mathbb{R}$.

(b) Fix positive $\delta \in \mathbb{R}$. Generate $\Delta_1^{(t)} \sim \text{Unif}([-\delta, \delta])$ and $\Delta_2^{(t)} \sim \text{Unif}([-\delta, \delta])$ independently. This amounts to moving uniformly in the square of side 2δ centered at $y^{(t)}$.

(c) Fix $\delta_1 > 0, \delta_2 > 0$ both real numbers. Generate $\Delta_1^{(t)} \sim \text{Unif}([-\delta_1, \delta_1])$ and $\Delta_2^{(t)} \sim \text{Unif}([-\delta_2, \delta_2])$ independently. This amounts to moving uniformly in the rectangle of sides $2\delta_1, 2\delta_2$ centered at $y^{(t)}$.

Now we must specify how to go from the proposed state $\tilde{y}^{(t+1)}$ to the next state $y^{(t+1)}$. If $\tilde{y}^{(t+1)}$ is inside the circle, there is no problem we can just set $y^{(t+1)} = \tilde{y}^{(t+1)}$. But what if $\tilde{y}^{(t+1)}$ goes outside the circle? We can think of a few options also in this case:

1. Just exit the circle and wait for the random walk to come back. At the end of the walk discard the points outside the circle. In this case one applies the steps (12.4), (12.5) and sets $y^{(t+1)} = \tilde{y}^{(t+1)}$ always. This is equivalent to the random walk analyzed in 11.1.4

2. Suppose $\tilde{y}^{(t+1)}$ falls outside the circle. Then we stay put, meaning we set $y^{(t+1)} = y^{(t)}$.

3. Suppose $\tilde{y}^{(t+1)}$ falls outside the circle. Then we resample $\Delta^{(t)}$ and obtain a new proposed $\tilde{y}^{(t+1)}$. We keep resampling the noise until $\tilde{y}^{(t+1)}$ falls inside the circle. Then we set $y^{(t+1)} = \tilde{y}^{(t+1)}$.

Question: out of the options for the steps (a), (b), (c) and 1., 2., 3. for the action to take when one steps outside the circle, which combinations guarantee that the sequence $\{y^{(t)}\}_{t\in\mathbb{N}}$ samples uniformly inside the circle at large times?

Answer: 2a, 2b, 2c all return a series of points sampled uniformly in the circle. 1 does as well, but the return rate is very small and hence computationally impractical. In fact the probability of being inside the circle at time t decreases asymptotically as $1/t$. Option 3 does not return uniform samples. This means that the choice of steps (a, b or c) does not have an impact on the large time probability measure that the MCMC samples. What matters instead is the way we deal with the boundary.

All of this likely confused you, since we did not explain *why* some rules work and others don't. The next section is meant to remedy to this and present theory for Monte Carlo Markov chains.

12.2.1 Theory of MCMC on an example with 9 states

To understand better how MCMC work and what conditions one should satisfy to sample from a target probability measure, we pick a simple example. We want to sample the uniform probability measure on $\{1, 2, 3, 4, 5, 6, 7, 8, 9\}$. To build the MCMC in the following way: we arrange the numbers in the square depicted in figure 25. At each time step one can move either horizontally, or vertically by one step (same as the king in chess), but not diagonally. There

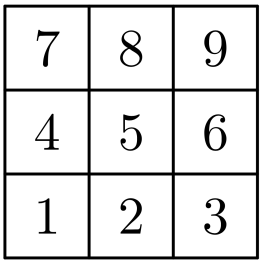

Figure 25: State space of the MCMC. At each time step one can move either horizontally, or vertically by one step, but not diagonally. There are no periodic boundary conditions.

are no periodic boundary conditions. As an example from 6 one can move exclusively to 9,5,3. And from 1 one can go only to 4,2. We define the probability

$$\pi^{(t)}(a) \equiv \text{ Probability of being in } a \text{ at time } t, \qquad a \in \{1, 2, \dots, 9\}. \tag{12.6}$$

and $\pi^{(t)} = (\pi^t(1), \dots, \pi^t(9))$.

Then we introduce the transition probabilities

$$p(b \to a) \equiv \text{ Probability of going from } b \text{ to } a \text{ in one time step.} \tag{12.7}$$

The transition probabilities contain the probabilities of going from one state to another in one time step. These are our degrees of freedom to specify the Markov chain. We organize the transition probabilities into a 9×9 *transition matrix* T with elements $T_{ab} = p(b \to a)$.

Since at every time step the walker must go somewhere, each column of the transition matrix must sum to one: $\sum_{a=1}^{9} p(b \to a) = \sum_{a=1}^{9} T_{ab} = 1$. We say that matrix T is column stochastic.

We're now ready to see how $\pi^{(t)}$ evolves in time. We have

$$\pi^{(t+1)}(a) = \sum_{b=1}^{9} p(b \to a)\pi^{(t)}(b) \tag{12.8}$$

You can see this as a conservation of probability equation: to know how much probability goes into a at time $t+1$, we sum over b the probability that from b goes into a. This is given by the probability of being in b times the probability of moving to a if one is in b. In matrix form, this equation can be written as

$$\pi^{(t+1)} = T\pi^{(t)}. \tag{12.9}$$

This is an iterative equation in $\pi^{(t)}$. We say π^{st} is a stationary probability if π^{st} is a fixed point for equation (12.9). In formulas $\pi^{st} = T\pi^{st}$.

Now let's go back to our original goal, which consisted of devising a Markov chain that sampled from the uniform distribution on $\{1, \dots, 9\}$, under the constraint that one can jump only between adjacent squares in the 3×3 grid. We denote this probability measure we want to sample with $\pi^\star$. How do we design T to reach this goal? We want to design T so that $\pi^\star$ is stationary for it. In other words, we want the property

$$\pi^\star = T\pi^\star. \tag{12.10}$$

In other words, $\pi^\star$ is the eigenvector of matrix T with eigenvalue $\lambda = 1$.

Instead of satisfying (12.10) we ask for a more stringent condition which, however, is easier to verify in practice. The condition is called *detailed balance* and can be stated in the following way

$$\pi^\star(a)p(a \to b) = \pi^\star(b)p(b \to a) \quad \forall a, b \tag{12.11}$$

Proposition 8. *Suppose $\pi^\star(a)p(a \to b) = \pi^\star(b)p(b \to a)$ for all pairs a, b. Then $\pi^\star$ is stationary, i.e., $\pi^\star(a) = \sum_b p(b \to a)\pi^\star(b)$.*

Proof. We prove this by verifying stationarity directly.

$$\sum_b p(b \to a)\pi^\star(b) = \sum_b p(a \to b)\pi^\star(a) = \pi^\star(a)\left(\sum_b p(a \to b)\right) = \pi^\star(a). \tag{12.12}$$

In the first passage, we used the detailed balance by exchanging b with a. Finally, in the last passage, we exploited the fact that $p(a \to b)$ is normalized (when summing over b). □

Notice that while detailed balance is a sufficient condition for stationarity, it is not necessary. There exist cases in which (12.10) is satisfied but (12.11) is not. But in a vast majority of Markov chains we use to simulate physical systems the detailed balance is satisfied.

Going back to our example, with $\pi^\star(a) = 1/9$ for all a, detailed balance imposes that $p(a \to b) = p(b \to a)$ for all pairs a, b. How to pick $p(\cdot \to \cdot)$ satisfying this constraint? The delicate part is to handle the fact that each state has a different number of neighbors: $a = 5$ has four neighbors, $a = 9$ has two, and $a = 2$ has three. One possible strategy is the following. For each state a we set $p(a \to b) = \frac{1}{4}$ for all neighboring $b \neq a$. This guarantees that detailed balance holds since all transition probabilities between adjacent states are $1/4$. But we have another problem: normalization. State $a = 6$ has only three neighbors so $\sum_{b \neq a} p(a \to b) = 3/4$, while we need this sum to be one. To solve this problem we set $p(a \to a) = 1 - \sum_{b \neq a} p(a \to b)$. For example we will have $p(5 \to 5) = 0$, $p(6 \to 6) = 1/4$, $p(1 \to 1) = 1/2$. All the transition probabilities are shown in figure 26.

$$\{p(a \to b)\} = \begin{bmatrix} \boxed{\frac{1}{2}} & \frac{1}{4} & \cdot & \frac{1}{4} & \cdot & \cdot & \cdot & \cdot & \cdot \\ \frac{1}{4} & \boxed{\frac{1}{4}} & \frac{1}{4} & \cdot & \frac{1}{4} & \cdot & \cdot & \cdot & \cdot \\ \cdot & \frac{1}{4} & \boxed{\frac{1}{2}} & \cdot & \cdot & \frac{1}{4} & \cdot & \cdot & \cdot \\ \frac{1}{4} & \cdot & \cdot & \boxed{\frac{1}{4}} & \frac{1}{4} & \cdot & \frac{1}{4} & \cdot & \cdot \\ \cdot & \frac{1}{4} & \cdot & \frac{1}{4} & \boxed{0} & \frac{1}{4} & \cdot & \frac{1}{4} & \cdot \\ \cdot & \cdot & \frac{1}{4} & \cdot & \frac{1}{4} & \boxed{\frac{1}{4}} & \cdot & \cdot & \frac{1}{4} \\ \cdot & \cdot & \cdot & \frac{1}{4} & \cdot & \cdot & \boxed{\frac{1}{2}} & \frac{1}{4} & \\ \cdot & \cdot & \cdot & \cdot & \frac{1}{4} & \cdot & \frac{1}{4} & \boxed{\frac{1}{4}} & \frac{1}{4} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \frac{1}{4} & \cdot & \frac{1}{4} & \boxed{\frac{1}{2}} \end{bmatrix}$$

Figure 26: Transition matrix that has $\pi^\star(a) = 1/9$ as stationary probability. Taken from [3].

13 Lecture 13: Theory and examples of MCMC

13.1 Theory of MCMC

13.1.1 Markov Chains

Let us go back to the concept of Markov chains and realize that the theory we started developing in the previous lecture is valid in a much larger generality than in the stated example. Let us thus start this lecture by summarizing the generic theoretical concepts and further building on them.

We denote with $\mathscr{A}$ the space of possible configurations (or state space). This is the set of all possible states of the system. $K = |\mathscr{A}|$ is the number of possible configurations. For example in physical systems with a large number N of particles, and where each particle can be in two possible states $\{1, -1\}$, we have $\mathscr{A} = \{1, -1\}^N$ and therefore $K = 2^N$. We indicate with $a \in \mathscr{A}$ a particular state of the system. A Markov Chain is a random process that produces a sequence of states $a^{(0)}, a^{(1)}, ...$, there the superscript indicates time. The states are random variables and one can hence consider the probability of being in state a a time t. As in the previous lecture, define

$$\pi^{(t)}(a) \equiv \text{Probability of being in state } a \text{ at time } t, \qquad a \in \mathscr{A}. \tag{13.1}$$

Equivalently $\vec{\pi}^{(t)} \in \mathbb{R}^K$ is the vector of probabilities at time t. The Markov Chains is defined by its transition matrix $T_{ab} = p(b \to a)$ of probabilities to go to state a if one was in state b in the previous time. This matrix is hence column stochastic (i.e., $\sum_{a \in \mathscr{A}} T_{ab} = 1$) and positive ($T_{ab} \geq 0$ for all a, b).

Then, as we saw in the last lecture, $\pi^{(t)}$ evolves as

$$\pi^{(t+1)}(a) = \sum_{b \in \mathscr{A}} p(b \to a)\pi^{(t)}(b) = \sum_{b \in \mathscr{A}} T_{ab}\pi^{(t)}(b), \tag{13.2}$$

which can also be written as $\vec{\pi}^{(t+1)} = T\vec{\pi}^{(t)}$. We already commented on the meaning of this equation in section 12.2.1. We also defined a stationary probability for T to be a probability vector $\vec{\pi}^{st}$ such that $\vec{\pi}^{st} = T\vec{\pi}^{st}$. In words, $\vec{\pi}^{st}$ is invariant in time.

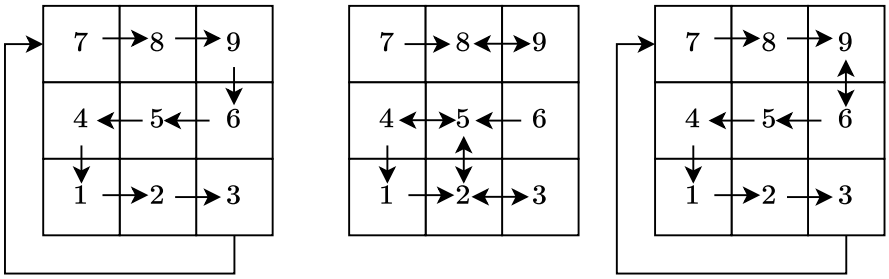

Figure 27: Examples of (from left to right). A periodic, a reducible, and an ergodic Markov chain. The arrows indicate which transition probabilities are nonzero. The middle chain is reducible because from the states $1, 2, 3, 4, 5$ one cannot go to the $6, 7, 8, 9$, and because 7 is unreachable from any other state.

13.1.2 Construction of Monte Carlo Markov Chains

We will call a Markov Chain a Monte Carlo Markov Chain (MCMC) when it is designed to sample from a target probability distribution $\vec{\pi}^\star$. We will summarize properties of the transition matrix T that guarantee that the MCMC will indeed converge towards the target probability distribution $\vec{\pi}^\star$.

Detailed balance A Markov Chain is said to satisfy the detailed balance condition with respect to $\vec{\pi}^\star$ if and only if

$$\pi^\star(a)p(a \to b) = \pi^\star(b)p(b \to a) \quad \forall a, b \tag{13.3}$$

We already saw in Proposition 8 that satisfying detailed balance with respect to $\vec{\pi}^\star$ is a way to ensure that $\vec{\pi}^\star$ is a stationary distribution of the chain, i.e. $\vec{\pi}^{st} = \vec{\pi}^\star$. It is thus desirable to choose the transition matrix T of an MCMC that satisfies the detailed balance with respect to $\vec{\pi}^\star$.

Ergodicity and aperiodicity The detailed balance condition is sufficient for existence of a stationary distribution $\vec{\pi}^{st} = \vec{\pi}^\star$. In order to ensure that the Markov Chain will converge to it it also needs to be ergodic and aperiodic. In this section, we present the concepts of ergodicity and aperiodicity, which together with the detailed balance are sufficient conditions for having $\vec{\pi}^{(t)} \to \vec{\pi}^{st}$.

Definition 7 (Ergodicity)**.** *We say that the Markov chain is ergodic, and the transition matrix T is irreducible if for every pair of states a, b there exists a sequence of states $p_1 = a, p_2, \ldots, p_r = b$, going from a to b and such that $T_{p_{i+1},p_i} > 0$ for all $i \in 1, \ldots, r-1$.*

Said plainly we require that every two states are joined by a directed path that has positive probability. This is done to ensure that the Markov chain cannot get stuck in a state, from which it can go nowhere, or to avoid that the state space is partitioned in two sets with no possible transitions between the two. Some literature refers to the property of the Markov chains as irreducibility as well and calls a Markov chain ergodic only is it is irreducible and also aperiodic (see next).

Definition 8 (Aperiodicity)**.** *A Markov chain is periodic when some of its property repeats itself with a period.*

For instance, consider a random walk that starts at zero and at each time step moves left or right by 1 with equal probability. This Markov chain is periodic because at even times it can only be at even positions and at odd times at odd positions. In particular it can return to the origin only at even times. See Figure 27 for other examples. In short ergodicity (with aperiodicity) grants the uniqueness of the stationary distribution π^{st}.

In addition it is common that T is local, meaning each move changes only slightly the current state. In other words, if the states a and b are in some notion of distance far away from each other, it is common to consider $T_{ab} = 0$.

13.1.3 Justification of the convergence to stationarity and convergence time

We saw that the detailed balance condition implies that $\vec{\pi}^\star$ is a stationary distribution of the Markov Chain, that mean that $T\vec{\pi}^\star = \vec{\pi}^\star$, i.e. the target distribution is an eigenvector of the transition matrix T with eigenvalue of 1. Now we look at the time evolution of a starting probability distribution $\vec{\pi}^{(0)}$. By applying equation (13.2) t times we get

$$\vec{\pi}^{(t)} = T^t\vec{\pi}^{(0)}. \tag{13.4}$$

In order to proceed, it is useful to state the following result about the maximum eigenvalue of a column stochastic matrix.

Theorem 11 (Perron-Frobenius). *Let $T \in \mathbb{R}^{K\times K}$ be a column stochastic matrix with positive entries. Let $\lambda_1, \dots, \lambda_K$ be the eigenvalues of T ordered by decreasing absolute value. Then $\lambda_1 = 1$ and all the components of the associated vector $\vec{v}_1$ are positive.*

The Perron-Frobenius theorem guarantees that all eigenvalues are in absolute value equal to or smaller than 1.

Further we state without a detailed proof that the ergodicity and aperiodicity properties imply that the second eigenvalue of T is $|\lambda_2| < 1$.

Putting these properties together with the power iteration method (introduced in lecture 1, section 1.2) for computing leading eigenvector of matrices we obtain that for $t \to \infty$, $T^t\vec{\pi}^{(0)} \to \vec{\pi}^\star$. This follows from the fact that the largest eigenvalue is 1, and from the existence of eigenvalue gap, i.e. the 2nd largest eigenvalue of the transitions being strictly smaller than one $|\lambda_2| < 1$. It turns out that because of the particular properties of stochastic matrices the condition $\vec{\pi}^{st} \cdot \vec{\pi}^{(0)} \neq 0$, needed for the power method to work, is not even necessary.

We will now discuss the time to convergence. For reasons similar to what we have seen for the power iteration method, the distance from stationarity is given by the t-th power of the second eigenvalue. Consequently, we defined the convergence time $\tau = -1/\log(\lambda_2)$, so that $\lambda_2 = e^{-1/\tau}$. It can then be shown, for details see e.g. https://weitsehsu.com/post/spectral_gap/, that

$$\vec{\pi}^{(t)} = \vec{\pi}^{st} + c_2 e^{-t/\tau}\vec{v}_2 + \dots, \tag{13.5}$$

for some constant c_2 and $\vec{v}_2$ being the 2nd right eigenvector of T. The $'\dots'$ contain all terms that decay faster than $e^{-t/\tau}$ since they correspond to smaller eigenvalues. From (13.5) we see two things:

- Indeed, provided that $|\lambda_2| < 1$, $\vec{\pi}^{(t)}$ converges to $\vec{\pi}^{st}$.
- The convergence time τ tells us the time it takes to go to the stationary distribution. For $t \geq 3\tau$ we have $e^{-t/\tau} = 5 \times 10^{-2}$ and therefore we have basically converged to $\vec{\pi}^{st}$.

As an example, the transition matrix in Fig. 26 has eigenvalues $(1, 3/4, \dots)$, giving $\tau \approx 3.5$. This means that after about 10 steps we can expect to be sampling from the uniform distribution on the 3×3 array considered in the previous lecture.

It remains to be discussed whether computing λ_2 is possible in practice? The answer in no in most examples relevant in physics. Recall that in physics one runs MCMCs for systems with many (N) particles. Say, each particle can occupy two states. This gives a total of $K = 2^N$ states. Calculating an eigenvalue of a $2^N \times 2^N$ matrix is not doable in practice. Therefore other ways are used to establish whether the Markov chain converges to its stationary probability.

13.1.4 The Metropolis rule

We notice that in the construction of the transition matrices that ensure convergence of the MCMC to the target distribution, the non-trivial condition to satisfy is the detailed balance. Satisfying the detailed balance for all pairs of states and at the same time keeping the transition probabilities proper probabilities between 0 and 1 may be a challenging combination.

The so-called Metropolis rule provides a significant help in construction of MCMC that satisfy detailed balance. According to the Metropolis rule we can only choose the transition probability from a state of smaller probability to larger probability. The opposite is always given by the Metropolis rule that states for states a, b such that $\pi^\star(a) \leq \pi^\star(b)$

$$p(b \to a) = \frac{\pi^\star(a)}{\pi^\star(b)} p(a \to b) \tag{13.6}$$

Probability distribution that are of common interest in physics have the Boltzmann form

$$\pi^\star(a) = \frac{1}{Z(\beta)} e^{-\beta E(a)} \tag{13.7}$$

where $E(a)$ is the energy of the state a and $\beta > 0$ is the inverse temperature. In this physics interpretation, the state b that is more probable than a, $\pi^\star(a) \leq \pi^\star(b)$, has a lower energy $E(a) \geq E(b)$.

13.2 Examples

In order to understand better the above concepts, let us discuss some examples of Markov chains.

13.2.1 Example 1: Asymmetric Markov chain for sampling the 3×3 square.

Consider the 3×3 patch with the 9 squares again, and aim to sample again the uniform distributions over its tiles. As before, for the uniform distribution to satisfy the detailed balance condition, the transition matrix must be symmetric.

Assume that we have an MCMC algorithm where the transition probability for every field to an adjacent field on its left or right is $P(\text{move left}) = P(\text{move right}) = 0.4$. Likewise the $P(\text{move down}) = P(\text{move up}) = 0.1$. What is then the probability of staying in place? There is only one tile where all moves are possible. Since the probabilities of all allowed movements on tile 5 sum to 1, the probability of staying is zero. Likewise, in the corners, the probability of staying is 0.5 (one move horizontally and one move vertically is possible). On the tiles 8 and 2, only moving in one vertical direction is forbidden, which gives a probability of staying of 0.1. The remaining tiles 4 and 6 have a staying probability of 0.4.

This MCMC is again ergodic and aperiodic and satisfies the detailed balance and hence converges to the uniform distribution of the patch.

13.2.2 Example 2: Revisiting PageRank

Recall the Pagerank algorithm, that represents a surfer on the web from Section 1. We can now understand the walk of the surfer from webpage to webpage as a Markov chain on the network of webpages. Recall that we had n webpages and the links between the webpages were defined in terms of the matrix A, such that

$$A_{ij} = \begin{cases} 1 & \text{if website } j \text{ links to website } i \\ 0 & \text{otherwise} \end{cases} . \tag{13.8}$$

The walk of the surfer then follows the transition matrix G, which we used in the implementation of this algorithm,

$$G = (1-\epsilon)S + \epsilon I \tag{13.9}$$

where G_{ij} is the probability of the surfer to jump from node j to node i

$$S_{ij} = \begin{cases} \frac{A_{ij}}{d_j} & \text{if } d_j \geq 1 \\ \frac{1}{n} & \text{if } d_j = 0 \end{cases} , d_j = \sum_{i=1}^{n} A_{ij} . \tag{13.10}$$

We can now understand better why we needed some of the components of G in the algorithm: Without the $\frac{1}{n}$ probability to jump to a new point when the node j has no outgoing nodes, we would simply get stuck at the node. Similarly, without the ε the Markov chain might get stuck to a connected component of the graph and then miss out on other connected components[12]. Without these terms, the Markov chain would not be ergodic and would not be guaranteed to converge.

We will, however, note that the pagerank walked is not a Monte Carlo Markoc Chain in the sense that is it not constructed to sample and particular probability distribution $\vec{\pi}^\star$. For this reason we are also not concerned with the detailed balance condition in this example.

13.2.3 Example 3: Sample uniformly a circle.

Recall the goal of estimating π via sampling uniformly points from a square and counting how often they fall inside a unit circle. Up to this point, we have described two ways to do this. Either we uniformly sample from the circle by selecting random points, or we do a random walk using MCMC instead. Recall that the random variable was

$$X_i = \begin{cases} 1 & \text{if point in circle} \\ 0 & \text{else} \end{cases} . \tag{13.11}$$

Then we had the empirical average was $G_T = \frac{1}{T}\sum_{i=0}^{T} X_i \to \frac{\pi}{4}$ as $T \to \infty$. In the case of sampling points from the square, the random variables X_i were independent. For independent random variables we derived $\text{Var}(G_T) = \frac{1}{T}\text{Var}(X_i)$.

[12] There might be several connected components in a graph if there are two sets of nodes A and B, for which it holds that for all pairs $a \in A$ and $b \in B$ there is no directed path connecting the two.

Importantly, in the case of doing the random walk via MCMC obtaining samples X_t during the walk, the samples X_t we obtain this way are not independent random variables. This is because every X_t is dependent on the history of the chain, as new samples are generated based on previous ones. Consequently one cannot use the above expression for variance of a sum of independent random variables. The question is now, how do we estimate the error bars of the estimate of π we obtain from Monte Carlo Markov Chains?

13.2.4 Error estimation for correlated variables.

When using MCMC we must wait for several multiples of the equilibration time τ before the walk actually starts providing samples from the correct distribution. In practice the equilibration time may be difficult to estimate. In worst scenarios, the equilibration time can be as large as exponential in the size of the system. In physics, we usually plot quantities of interest and when they seem to reach a plateau we assume that equilibration has been reached. But one always needs to be aware that this may not be the case.

A basic option to estimate the error bars of quantities measured from MCMC is to start several MCMC chains that are independent at the same time. Then one dismisses the initial τ steps before equilibration is reached for each of the chains. One can then computes the average and variance over the predictions that are obtained from the different independent chains. Finally one obtained the error of the average from the formulas we covered for independent variables. Concretely, defining the estimator for the μ'th chain as $e_\mu = \frac{1}{T-\tau}\sum_{t=\tau}^{T} X_t$, the final empirical mean is then

$$\hat{e} = \frac{1}{n}\sum_{\mu=1}^{n} e_\mu \tag{13.12}$$

With the empirical variance of the random variable e_μ is

$$\hat{\sigma}_e^2 = \frac{1}{n}\sum_{\mu=1}^{n}(e_\mu - \hat{e})^2. \tag{13.13}$$

Since the different chains are independent this leads to the standard error on the mean to be $\sqrt{\hat{\sigma}_e^2}/\sqrt{n}$.

Another way that is used more often in practice is to run a single Monte Carlo chain but estimate the so-called decorrelation time by measuring correlation between an equilibrated configuration at some $t > \tau$ and a configuration at a later time $t + \tilde{t}$. Again such correlation is in general hard to measure, but in practice one uses various kind of observables and seeks for them to plateau as a function of $\tilde{t}$. When they do we set $\tilde{t} = t_{\text{decorrelation}}$. Once we estimated the decorrelation time we can run a single Markov chain for many steps and start taking samples once the convergence time was reached and then take a new sample everytime we pass a time larger than the decorrelation time.

13.3 Sampling with interactions: Hard spheres

Finally, we study a simple case where we sample objects in interaction. Interaction between components of a system (particles) are ubiquitous in physics, this is hence an important situation. We consider the problem of sampling from the distribution of two hard spheres of radius R on a 1-dimensional interval from $[0, L]$. By saying the spheres are hard, we mean that they cannot overlap in the range of this radius. If the first sphere's center is at position x_1, then the second at position x_2 then one must have $|x_1 - x_2| \geq 2R$. Likewise, they cannot overlap with the interval wall i.e. $L - R \geq x_1 \geq R$ and same for the second center x_2. This setup is shown on the left in Figure 28.

The goal is to sample uniformly over all feasible positions (x_1, x_2), i.e. over the green area in Figure 28 right. The joint probability density of the two centers we aim at is thus

$$\rho(x_1, x_2) = \frac{1}{(L-4R)^2} \quad \text{when} \quad |x_1 - x_2| > 2R,\ R \leq x_1, x_2 \leq L - R \tag{13.14}$$

$$\rho(x_1, x_2) = 0 \quad \text{otherwise} \tag{13.15}$$

We consider three propositions for a sampling algorithm:

1. We sample $\sigma_1, \sigma_2 \sim \text{Unif}(0, L-4R)$, we order the two variables so that $\sigma_1 < \sigma_2$ and then set $x_1 = R + \sigma_1$, $x_2 = 3R + \sigma_2$.

2. We sample $x_1 \sim \text{Unif}(R, L-R)$ and $x_2 \sim \text{Unif}\left((R, x_1 - 2R) \cup (x_1 + 2R, L - R)\right)$.

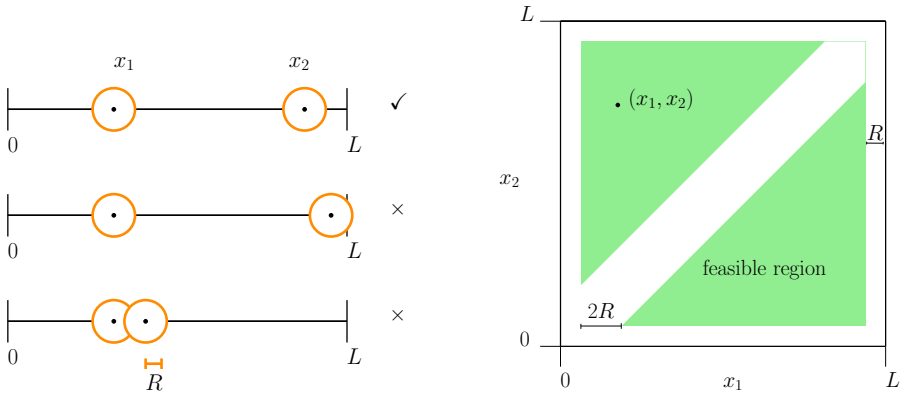

Figure 28: (Left) Hard spheres problem setup. (Right) coordinate system of the two spheres positions. Green highlights the feasible region, which we want to sample from uniformly.

3. We sample $x_1, x_2 \sim \text{Unif}(R, L-R)$, and if we sample values such that they violate the constraint $|x_1 - x_2| < 2R$, we resample both $x_1, x_2 \sim \text{Unif}(R, L-R)$ until the constraint is satisfied.

Which ones of these three options give us a uniform sample of all valid configurations (x_1, x_2)? What we ultimately want, is a uniform sample over the feasible green region on the right of Figure 28. The answer is: 1 and 3 achieve this, but option 2 does not. We can see that options 1 and 3 are in fact equivalent to each other. One way to see that option 2 cannot work is that for box sizes $4R < L < 6R$ there clearly are some compatible positions, but in case the first center falls in the interval $L - 3R < x_1 < 3R$ there is no compatible place for the second ball.

Let us now look at the marginal density $\rho(x)$ of the center of one ball being at position x:

$$\rho(x) = \int_0^L dx_2\, \rho(x, x_2) \tag{13.16}$$

Performing this simple integral we obtain

$$\rho(x) = \frac{L - 6R}{(L-4R)^2} \quad \text{for} \quad 3R \le x \le L - 3R \tag{13.17}$$

$$\rho(x) = \frac{L - 3R - x}{(L-4R)^2} \quad \text{for} \quad R < x < 3R \tag{13.18}$$

$$\rho(x) = \frac{x - 3R}{(L-4R)^2} \quad \text{for} \quad L - 3R < x < L - R \tag{13.19}$$

$$\rho(x) = 0 \quad \text{otherwise} \tag{13.20}$$

In Figure 29 we plot the marginal $\rho(x)$.

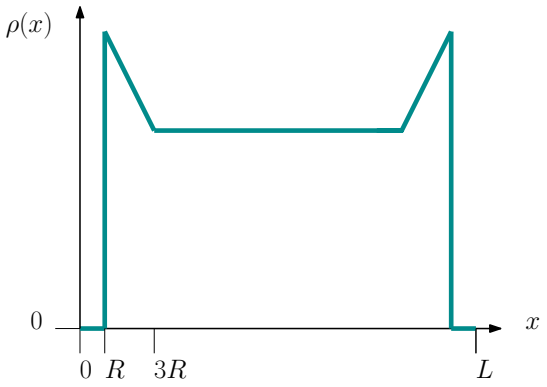

Figure 29: Marginal density for the correct sampling strategies.

We see in Figure 29 that the marginal density is larger towards the borders than it is in the middle. This may seem counter-intuitive at first sight, but it is correct and a consequence of the uniformity in the 2D configuration

space. In fact, this larger density at the boundary is a manifestation of a physical force that emerges from purely entropic considerations, the balls are attracted to the boundary by an entropic force called depletion force that originates from the fact that with one of the spheres close to the boundary, there is more space, more entropy, for the other spheres, thus such a configuration is more likely. Entropic depletion forces are particularly important in the physics of colloids where hard particles tend to attack each other for purely entropic reasons without any other source of force – gravity, electromagnetic, chemical or other.

14 Lecture 14: Phase Transitions

In the final lecture, we will discuss the probabilistic origin of phase transitions. When fluid water turns into frozen water (ice) when the temperature is cold enough, we say that it undergoes a phase transition. Historically it was not evident whether a phase transition is due to the fact that the nature of interaction between molecules changes at the critical temperature or what exactly was the origin of phase transitions. This was clarified with the development of statistical physics, and in this lecture, we will give examples of systems with phase transitions and describe how these sharp changes emerge from the mere fact that elements of the system interact and are numerous, and not that their interactions change.

In the following, we take a look at three examples of systems exhibiting phase transitions. In some of them we will derive explicit theory for where the phase transition appears. We will also see that phase transitions may also appear in systems that are non-physical, but e.g. related to data analysis (as in the 3rd discussed example).

14.1 Example 1: Site Percolation in 2D

We are given a square 2D grid of cells with side length L. Thus it has L^2 cells total. Every cell is occupied i.i.d. with probability $0 \leq p \leq 1$. Otherwise it remains empty.

One can see such a grid as a section through soil in the ground, or a sponge. The question is then: Is there a way to cross from one side of the square to another, by walking only on cells that are occupied (as in the grid examples in the previous lectures, we assume that one can walk only to the direct neighbors and *not* to your diagonal neighbours). If there is such a path, we say that a "percolating cluster" exists. In Fig. 30 we see three examples, in the left grid no such connection from top to bottom exists, however, in the center and on the right there is one.

The answer to the question "Is there a path from top to bottom?" clearly depends on the size of the grid L (how far do we need to cross over?) and the probability that a cell is occupied (more occupied cells should make it easier). For a given grid, the existence of a percolating cluster is a random variable since it will depend on the realization of which cells got occupied and which did not. In Fig. 31 we plot the probability of having a percolating cluster for a grid of size L as a function of the probability p. Curiously, as we increase $L \to \infty$, there is a specific critical value $p_c = 0.59274...$ (called the percolation threshold) which separates two phases. For $p > p_c$ the percolating cluster exists with probability going to 1 as $L \to \infty$, for $p < p_c$ it has probability going to 0 as $L \to \infty$.

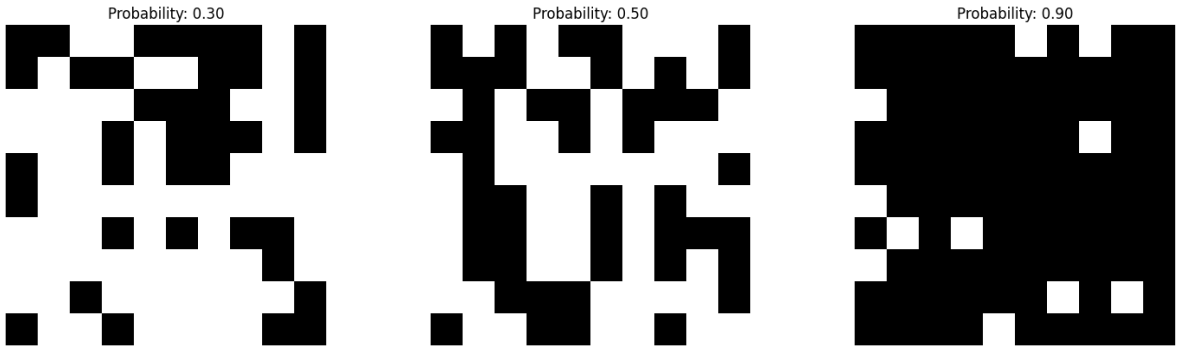

Figure 30: Examples of percolation in a 2D grid with side length $L = 10$. From left to right the probabilities to occupy each cell are $p = 0.3, 0.5, 0.9$. On the left there is no percolating cluster connecting the top to the bottom, but in the middle and on the right there is one.

To understand the system better, we can also look at an order parameter related to the problem. By an order parameter, we mean a property that in some way summarizes the behavior of the many interacting particles. In our example with the water the order parameter could be the density of the water. Similarly here, we can define the density ρ as the number of sites that are occupied in the largest cluster:

$$\rho_{p,L} = \frac{\#\text{ sites in largest cluster}}{L^2}\,. \tag{14.1}$$

The size of the largest cluster $\rho_{p,L}$ is again a random variable as it depends on the realization of the randomness in the grid.

In Fig. 32 on the left we show the probability density of $\rho_{p,L}$ for a given value of p for several different grid sizes L. As L grows, we find that this probability distribution collapses to a delta function on a specific value $\rho(p)$. We can

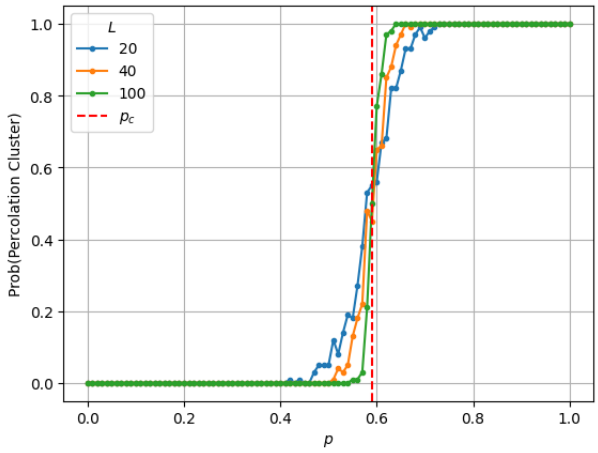

Figure 31: A phase transition – the percolation threshold – in 2D site percolation.

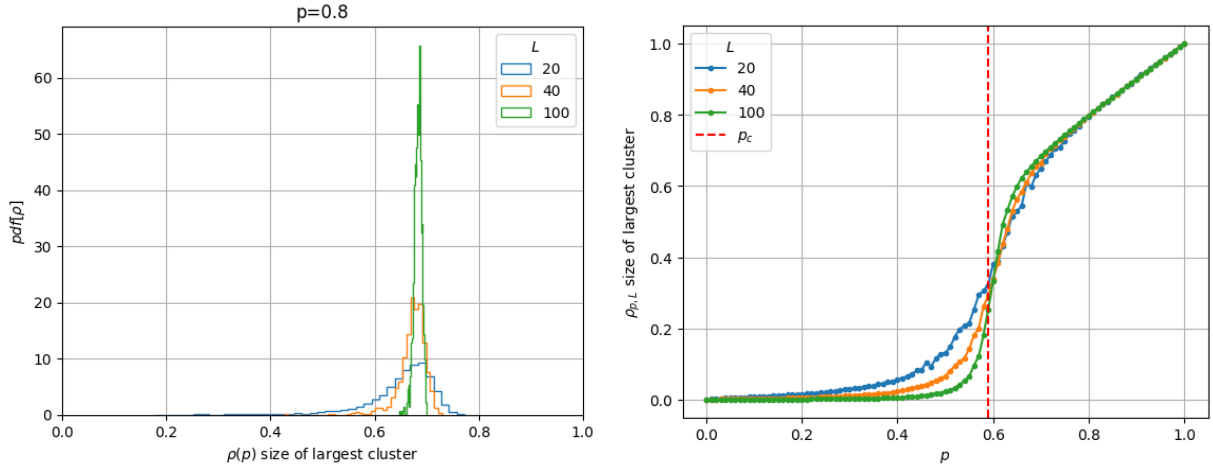

Figure 32: Left: The probability distribution of the size of the largest cluster in 2D site percolation. Right: The average fraction of sites taken by the largest cluster in 2D site percolation as a function of p.

say that the size of the largest cluster converges in probability. i.e. there exists a function $\rho(p)$ s.t.

$$\forall \varepsilon > 0 : \text{Prob}\left(|\rho_{p,L} - \rho(p)| > \epsilon\right) \to 0 \tag{14.2}$$

as $L \to \infty$. In Fig. 32 the average size of the largest cluster is shown is shown. Its limit at $L \to \infty$ is $\rho(p)$.

At the percolation threshold p_c we see that the limiting size of the largest cluster becomes non-zero. This coincides with the point where the probability of the percolating cluster to exist became 1 in the large size limit. For $p > p_c$ the percolation cluster exists with probability going to 1 as $L \to \infty$. The size of that cluster keeps increasing as we increase the site occupation probability p. For $p < p_c$ the percolating cluster has probability 0 to exist and the fraction of sites in the largest cluster goes to 0 as $L \to \infty$.

But what happens at p_c? When we have a value p with $|p - p_c| \ll 1$, we call it "close to criticality". It happens, that for such close-to-critical values the system exhibits non-trivial phenomena, for example self-similarity as observed in fractals, see e.g. the visualization of the largest cluster at the percolation threshold at https://commons.wikimedia.org/wiki/File:Percolation_critique.gif. You will study such critical behaviour in statistical physics.

One might ask, if the value p_c can be found analytically. After all, the system is not too difficult to describe. It is easy to set up and check numerically whether there is a percolation cluster in a given example. However, it turns out that we do not know the answer analytically for the 2D grid site percolation. While it is known accurately from simulations (i.e. taking L very large), we do not have a closed equation for its value. Nonetheless, we will see in example 2 that with some modifications we can evaluate a percolation threshold for a different system analytically.

14.2 Example 2: Giant component in random graphs

In this example we consider a graph with N nodes and the adjacency matrix $A \in \mathbb{R}^{N\times N}$. You can think of the nodes as the occupied sites in example 1. The matrix A is symmetric with $A_{ii} = 0$, $A_{ij} = A_{ji}$. Recall that a $A_{ij} = 1$ means there is an edge between the nodes i and j, and $A_{ij} = 0$ means no edge between i and j. If $A_{ij} = 1$ we say that i and j are *neighbors* in the graph. In analogy to the percolation example, if there is an edge from i to j then it is possible to move from i to j (and vice versa, due to symmetry). We say the nodes i and k are connected, if there exists a sequence of edges so that they can reach each other, i.e. $1 = A_{ij} = A_{jl} = A_{l...} = ... = A_{...k}$. A *connected component* (CC) is defined to be a subset of all nodes $C \subseteq \{1, 2, ..., N\}$, such that all $i, j \in C$ are connected. When we generate the edges in the graph randomly as

$$A_{ij} = \begin{cases} 1 & \text{with probability } p \\ 0 & \text{with probability } 1-p \end{cases} \tag{14.3}$$

the resulting graph is called an Erdős–Rényi graph, denoted as $G(N, p)$.

We now define $\rho_N(G)$ as the fraction of nodes (sites) which are part of the largest CC of the graph G. Note that due to the randomness in G (we create its adjacency matrix randomly) $\rho_N(G)$ is also a random variable. In Fig. 33 the distribution of this random variable is shown (in analogy to Fig. 32 left). Likewise, for large N this distribution converges to a delta function on a specific value $\rho(p)$. But in this case, in contrast to the 2D percolation, it is actually not too difficult to compute the location of this value $\rho(p)$.

In order to derive a closed equation for $\rho(p)$ we assume that the function $\rho(p)$ exists (this can be proven to hold with a bit of work). Each site i is in the largest CC with probability $\rho(p)$, and it is not in the largest CC with probability $1 - \rho(p)$. From this, we can say

$$\text{Prob}[i \text{ is not in the largest CC}] \overset{N\to\infty}{=} [\text{Prob}\,[i \text{ connected to } j\text{; and } j \text{ not in largest CC}] + \text{Prob}\,[i \text{ not connected to } j]]^{N-1} \tag{14.4}$$

$$1 - \rho(p) = [p(1-\rho(p)) + 1 - p]^{N-1} \tag{14.5}$$

$$= [1 - \frac{\lambda}{N-1}\rho(\lambda)]^{N-1} \overset{N\to\infty}{=} e^{-\lambda\rho(\lambda)} \tag{14.6}$$

where we defined the average degree of the graph $\lambda = p(N-1)$ (on average, every node connects to λ other nodes). We have derived the implicit equation

$$e^{-\lambda\rho(\lambda)} = 1 - \rho(\lambda)\,. \tag{14.7}$$

An obvious solution is always $\rho = 0$, regardless of p. Defining $f_\lambda(\rho) = 1 - e^{-\lambda\rho}$, and looking at its intersection with the $\rho = \rho$ line in Fig. 34, we see that this indeed the case. For some λ however, there exists a second solution. To have this second crossing, we need that the derivative at $\rho = 0$ is larger than one (and thus surpasses the diagonal). Formally, we require that

$$\frac{\partial f_\lambda(\rho)}{\partial\rho}\Big|_{\rho=0} \geq 1 \tag{14.8}$$

and doing the derivation this entails that

$$[e^{-\lambda\rho}\lambda]|_{\rho=0} = \lambda \geq 1\,. \tag{14.9}$$

The smallest average degree λ which fulfills this is $\lambda_c = 1$. Therefore, for all $\lambda < \lambda_c = 1$ there is no CC with a finite fraction of nodes, and for $\lambda > \lambda_c = 1$ there exists one with the size given by a solution to the above equation. This theory is compared to numerical simulation in Fig. 35.

14.3 Example 3: Phase transitions in data analysis

The last example of a phase transition connects to inference problems, which we discussed earlier in the course. We consider a statistical inference problem where one aims to find back the vector $\vec{v}^* \in \mathbb{R}^N$ from a specific observation of it. We will find that the two phases in this problem are the phase where we can successfully recover $\vec{v}^*$ (the noise is not problematically large), or we cannot recover it (when the noise is too strong). The observations are

$$Y_{ij} = Z_{ij} + \sqrt{\frac{\lambda}{N}} v_i^* v_j^* \tag{14.10}$$

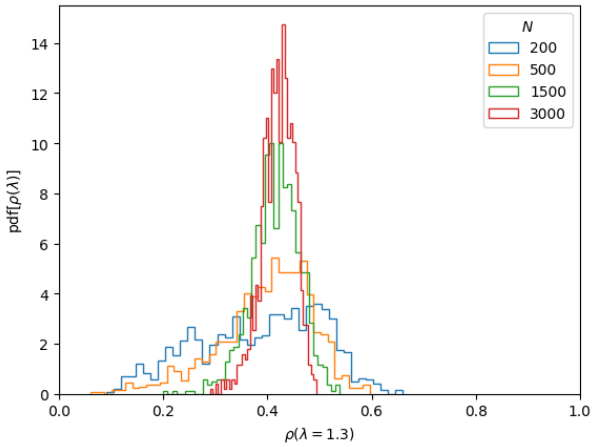

Figure 33: The probability density of the largest connected component in a random graph for average degree $\lambda = 1.3$

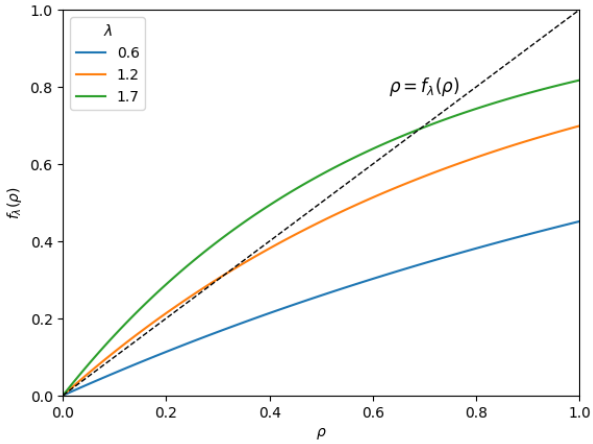

Figure 34: Solving the implicit equation: $f_\lambda(\rho)$ is drawn for different values of λ. For $\lambda > 1$, two distinct solutions exist.

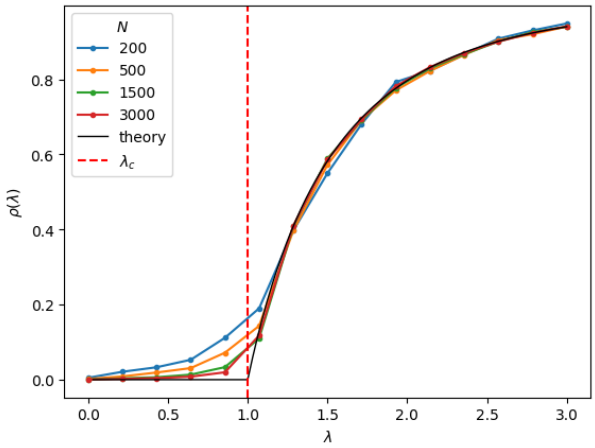

Figure 35: Size of the largest component in random graphs as a function of the average degree.

where Z_{ij} is symmetric noise $Z_{ij} = Z_{ji} \sim \mathcal{N}(0,1)$ and $\lambda > 0$ is the signal to noise ratio. In matrix notation, we observe the symmetric matrix $Y \in R^{N\times N}$:

$$Y = Z + \sqrt{\frac{\lambda}{N}}\vec{v}^*(\vec{v}^*)^T\,. \tag{14.11}$$

Using the maximum likelihood estimator to estimate $\vec{v}^*$, we write the likelihood

$$\rho(Y|\vec{v}) = \prod_{i\leq j\leq N}\left(\frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}(Y_{ij}-\sqrt{\frac{\lambda}{N}}v_iv_j)^2}\right) \tag{14.12}$$

then we have the maximum likelihood estimator

$$\hat{\vec{v}} = \arg\min_{\vec{v}} \sum_{i\leq j\leq N}(Y_{ij} - \sqrt{\frac{\lambda}{N}}v_iv_j)^2\,. \tag{14.13}$$

Recall that the Young-Eckhard theorem tells us that this minimization problem is solved by a vector that is proportional to the leading eigenvector of the matrix Y. Indeed, for an optimal rank-one approximation of Y, we can do the SVD and then use the largest singular vector, which for symmetric matrices is the same as the eigenvector.

We conclude from this that the maximum likelihood estimator is proportional to the leading eigenvector of Y. If we normalize both the vector $\vec{v}^*$ and the eigenvector to have a norm 1 then we can define their overlap $q = \hat{\vec{v}}\cdot\hat{v}^*$. The overlap quantifies how close the estimated vector is the the ground truth one.

Here again the value of q is a random variable that in general depends of the realization of the randomness but in the limit of $N\to\infty$ we again observe convergence in probability toward a value that can be computed analytically. To be concrete, we obtain

$$q = \begin{cases} 0 & \text{if } \lambda < 1 \\ \sqrt{1-\frac{1}{\lambda}} & \text{if } \lambda > 1\,. \end{cases} \tag{14.14}$$

For signal-to-noise ratio $\lambda > 1$ we can find back the signal with a good precision corresponding to the value q and if $\lambda < 1$ we cannot recover the signal and we have that $q = 0$. Therefore the signal to noise ratio with λ separates the two phases, one where recovery is possible, and the other where it is not.

References

[1] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bring order to the web," Technical report, stanford University, Tech. Rep., 1998.

[2] J. Novembre *et al.*, "Genes mirror geography within Europe," *Nature*, vol. 456, 2008.

[3] W. Krauth, *Statistical mechanics: algorithms and computations*. OUP Oxford, 2006, vol. 13.