Notes and Comments on S. Mallat's Lectures at Collège de France (2022)

Jean-Eric Campagne

Notes and Comments on S. Mallat's Lectures at Collège de France (2022)

Multiscale Models and Convolutional Neural Networks

J.E Campagne *

Janv. 2022; rév. 19 février 2024

*If you have any comments or suggestions, please send them to jeaneric DOT campagne AT gmail DOT com

Table des matières

1. Foreword

***Disclaimer**: What follows are my informal notes in French, translated into rough English, taken on the fly and reformatted with few personal comments ("NDJE" or dedicated sections). It is clear that errors may have crept in, and I apologize in advance for them. You can use the email address provided on the cover page to send me any corrections. I wish you a pleasant read.*

Please note that the Collège de France website has been redesigned. You can find all the course videos, seminars, as well as course notes not only for this year but also for previous years [1].

I would like to thank the entire Collège de France team for producing and editing the videos, without which the preparation of these notes would have been less convenient.

Also, note that S. Mallat [2] provides open access to chapters of his book *"A Wavelet Tour of Signal Processing"*, 3rd edition, as well as other materials on his ENS website.

This year 2022 marks the fifth year of S. Mallat's Data Science chair, with the theme being **Information Theory**.

I have uploaded some notebooks on GitHub [3] to illustrate this course. This initiative is minimalist, so you are invited to provide feedback and suggestions. I have used JAX as the automatic differentiation library because it allows coding directly like Numpy, which simplifies learning.

2. Lecture 19 Jan.

2.1 Introduction

Let's take a look at some significant developments in the field of data science in 2021. For example, the recognition of the performance of very large systems such as

1. https://www.college-de-france.fr/chaire/stephane-mallat-sciences-des-donnees-chaire-statutaire/events
2. https://www.di.ens.fr/~mallat/CoursCollege.html
3. https://github.com/jecampagne/cours_mallat_cdf/cours2022

GPT-3 developed by Open AI [4], which was deployed in mid-2020. This system boasts a whopping 175 billion parameters, making it the largest to date. It is a formal language model that learns from databases drawn from the web, including sources like Common Crawl, WebText2 [5], Google Books, and Wikipedia. It is trained on hundreds of billions of words. The ongoing trend since the inception of neural networks is that the more parameters a model has, the more spectacular its performance becomes. Moreover, GPT-3 is not confined to a specific task or corpus; it becomes somewhat of a generalist as it can generate various types of text (e.g., translation into any language from a single example, arithmetic, any programming language, text generation from examples), as well as engage in dialogues, etc. Humans are increasingly struggling to discern whether articles, even those over 200 words, are of artificial or human origin. Unfortunately, the downside is that this opens the door to disinformation, and to fraudulent messages generated entirely automatically.

Now, the field remains highly experimental, and these performances are poorly understood, despite the "discovery" of *double descent in risk* by Belkin et al. [6], which S. Mallat discussed in his 2020 lecture [7], generating many avenues of study in the field of *over*-parameterization. There is a plethora of publications (e.g., 15,000 papers at the last NISP conference), an acceleration of research. Yet, there is a need to return to fundamentals for a global perspective. While some may think or observe that articles become obsolete within a few months, some endure for centuries. For example, it was around the 1920s that **Ronald A. Fisher** (1890–1962) laid the foundations for *Statistics*, and we are ultimately within the framework he established on January 1, 1922, with his paper "*On the mathematical foundations of theoretical statistics*" [8]. The same can be said of **Claude Shannon**'s (1916-2001) 1948 paper *"A Mathematical Theory of Communication"* [9].

4. https://openai.com/blog/openai-api/, Tom B. Brown et al. *Language Models are Few-Shot Learners*, (July 2020) arXiv:2005.14165v4 https://arxiv.org/abs/2005.14165

5. https://commoncrawl.org/, https://www.eleuther.ai/projects/open-web-text2/

6. Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal, "Reconciling modern machine learning practice and the bias-variance trade-off", arXiv:1812.11118v2

7. J.E.C note, "Notes and comments on S. Mallat's lectures at the Collège de France (2020), Multi-scale Models and Convolutional Neural Networks", February 2020; revised September 17, 2020. https://www.di.ens.fr/ mallat/CoursCollege.html

8. https://doi.org/10.1098/rsta.1922.0009, available on the course website https://www.di.ens.fr/~mallat/CoursCollege.html

9. C. E. Shannon, The Bell System Technical Journal, Vol. 27, pp. 379–423, 623–656, July, October. https://people.math.harvard.edu/~ctm/home/text/others/shannon/entropy/entropy.pdf

In this course, we will therefore study the concept of **Information**. However, before delving into that, we will ask what is meant by building a model (e.g., neural networks) and what type(s) of model(s) can be chosen when dealing with high-dimensional analysis?

2.2 Deterministic vs. Stochastic Models

Implicit in this choice is a certain worldview [10], with, on one side, a Cartesian perspective, mostly French (continental), and on the other, a Bayesian view, primarily English [11]. Let's say that these two perspectives on probability have their respective biases, if we want to keep it brief. While culturally, one might lean one way or the other and consider the two views equivalent, when dealing with high-dimensional problems, we are somewhat limited in our choice.

Take the problem of *supervised classification*, for example. The goal is to estimate a response y from data $x \in \mathbb{R}^d$ ($d \gg 1$), and to do this, we have a training set $\{x_i, y_i\}_{i<n}$. The question that arises is: does it become more challenging to solve this problem as d increases?

From the *deterministic* perspective, the answer is **Yes**, due to the *curse of dimensionality*, which was a topic of discussion in the 2018 Lecture. If we consider an unknown function $y = f(x)$ in 1D, and if we have enough sampling points and the function f is sufficiently regular, then we can interpolate it with good accuracy (Fig 1). However, when we move into high dimensions $x \in \Omega$ (e.g., $\Omega = [0,1]^d$), if we want sufficiently dense data, for example, with a distance ε between adjacent points, we need N points to cover the space Ω. This leads to the following scaling relation

$$N \sim \varepsilon^{-d} \tag{1}$$

Now, if we have a regular function, for example, Lipschitz, then when we are near a training point,

$$\|f(x) - f(x_i)\| \leq C\|x - x_i\| \tag{2}$$

10. See, for example, the 2019 Lecture Section 2.3.2

11. However, we will explore the contribution of Pierre-Simon de Laplace (1749-1827), who rediscovered Bayes' inverse probability law, leading to a *Theory of Probabilities* in 1812. These elements also form the basis of current research.

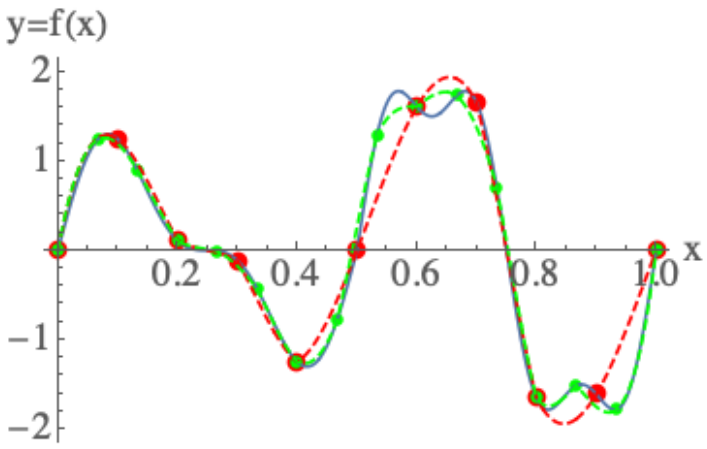

FIGURE 1 – Interpolations of two datasets (x_i, y_i) (non-noisy; red and green points) of an unknown underlying function f (blue curve). The denser the sample population, the better the interpolation.

and the point density helps bound the right-hand side, thereby estimating the generalization error (left-hand side). Thus, the required number of data points N exponentially explodes with dimension d to maintain fitting accuracy. This phenomenon of explosion is what motivates this answer. However, one could always argue about the regularity of the underlying functions of high-dimensional data, thinking that the problem can still be tackled. For example, by expressing invariants/symmetries of the problem (e.g., the theme of the 2020 Lecture) to perform dimensionality reduction. However, in the reasoning above, there is no model on the data. This is where the stochastic approach, in a way, attempts to go further in the analysis.

From the *Bayesian* perspective, the answer is **No**! Indeed, as d increases, we intuitively observe that an image has better resolution (the same applies to a sound clip), so it seems natural that it should be easier to *a priori* recognize a higher resolution object in the image. Thus, the negative answer seems natural, and it creates a dilemma. Let's consider ***level curves***[12]. of the function f (Fig. 2):

$$\Omega_y = \{y \ / \ f(x) = y\} \tag{3}$$

12. Elaborated argument in the 2021 Lecture regarding A. Barron's 1993 theorem (Section 5.2.3)

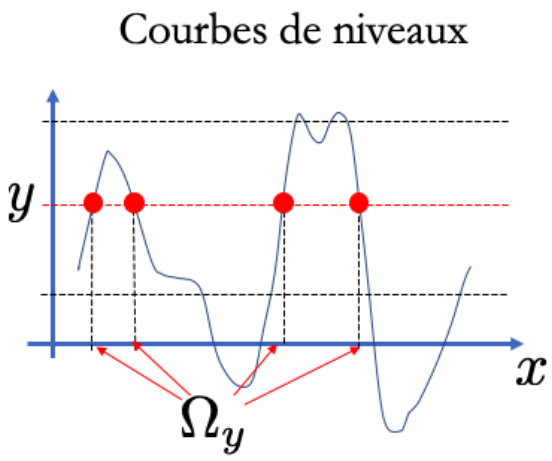

FIGURE 2 – Example of a level curve $f(x) = y$.

What interests us is the ***geometry*** of these curves (in any dimension, these are surfaces). Where do points concentrate in space? There is indeed concentration (Fig. 3), and the space *actually* occupied by, for example, images of dogs, cats, cars, etc., is tiny compared to the total possible space of images of the same dimension. The underlying phenomenon is ***the law of large numbers***. Ultimately, having the view of level curves is, in a way, opting for a perspective similar to Lebesgue's integral that uses *the measure of these sets*. And when we say *measure*, we mean *probability*. Schematically, through a measure, we have the probability density of x given y, denoted as $p(x|y)$:

$$\Omega_y \xrightarrow{\text{measure}} p(x|y) \tag{4}$$

Now, through Bayes' theorem (Thomas Bayes 1701-61), we have

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)} \tag{5}$$

where $p(y)$ and $p(x)$ are the *a priori* probabilities [13], and $p(x|y)$ is called the *likelihood* that x is true given y. The Bayesian classifier defines the best y as the one that ***maximizes***

13. We call it *prior* for $p(y)$ and *marginal likelihood* for $p(x)$ because we can write $p(x) = \int p(x|y)p(y)dy = \int p(x,y)dy$

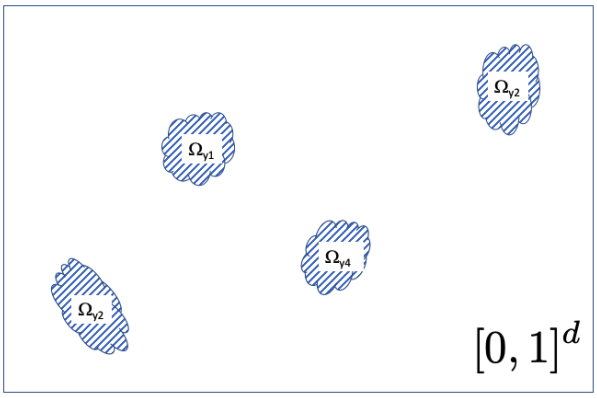

FIGURE 3 – Concentration of object classes.

$p(y|x)$:

$$\hat{y} = \underset{y}{\text{argmax}}\ p(y|x) \tag{6}$$

To realize this scheme, the Bayesian approach poses (or must pose) the question: where do data concentrate? And then, we realize that we don't need to find a solution y for every $x \in \Omega$ but only for the x elements in Ω_y.

Therefore, we need to address ***concentration phenomena*** of the measure and model probabilities $p(x|y)$ (unsupervised problem) or $p(y|x)$ (supervised problem). Typically, we study families of probabilities such as the exponential, and, for example, modeling like

$$p(x|y) = Z_y^{-1} e^{\Theta_y . \Phi(x)} \tag{7}$$

raises the question of modeling $\Phi(x)$, which is **the most appropriate representation of** x that linearizes $\log p(x|y)$.

It is then realized that the fields that study these types of probabilities are ***Statistical Physics*** and ***Information Theory***, which we will explore in this 2022 course.

2.3 Fisher's Perspective

The first question that R. Fisher addresses in the 1922 article is how to define the information in data about the estimation of a parameter θ. This is a problem of *Inference*.

He engages in a thorough reflection on what we are trying to achieve in the field of mathematical statistics and data analysis. According to him, we are attempting a form of ***data compression***, which means representing data with as few parameters as possible while providing a representation of the significant information within the available data. He then develops several key concepts, including:

— the notion of ***consistency*** of an estimator; does the estimator converge when we have an infinite amount of data, and is it biased or not?

— the concept of ***inference*** through maximum likelihood

— the concept of ***information***

— the notion of ***sufficient*** or *exhaustive statistics*, which accounts for the fact that the statistics (a set of operations applied to a dataset) contains all the information about the parameter(s) of the underlying probability distribution.

All these notions form the basis of current statistical mathematics.

To illustrate, if we have a dataset $\chi = \{x_i\}_{i \leq n}$, the problem at hand is to determine the probability distribution underlying the creation of this particular dataset. Thus, R. Fisher defines a *family of probabilities* indexed by θ, $p_\theta(x)$, and the problem boils down to estimating the "right" θ. For example, in 1D, we can think of $\theta = (\mu, \sigma^2)$ such that

$$p_\theta(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{8}$$

Of course, the problem we have in mind with image classification, for example, involves many more variables.

Given an estimator $\hat{\theta}(\chi)$, we would like it to converge as $n = |\chi|$ tends to infinity, meaning

$$\hat{\theta}(\chi) \xrightarrow[|\chi| \to \infty]{} \theta \tag{9}$$

then we qualify $\hat{\theta}(\chi)$ as a ***consistent estimator***. Fisher then finds a way to construct consistent estimators, which is through ***maximum likelihood***:

$$\hat{\theta}(\chi) = \underset{\theta}{\operatorname{argmax}}\ p_\theta(\chi) \tag{10}$$

In this way, we determine *a model for which the observed data is as probable as possible.* Later, we will denote the estimator as $\hat{\theta}$ with observations χ. When we have *identically*

and independently distributed observations (iid hereafter), then

$$p_\theta(\chi) = \prod_{i \leq n} p_\theta(x_i) \tag{11}$$

It is tempting to use the logarithm, so we define the *log-likelihood* (sometimes we will omit "log" to refer only to the *likelihood*)

$$\ell(\theta) := \log p_\theta(\chi) = \sum_{i \leq n} \log p_\theta(x_i) \tag{12}$$

So, the ideal is to find the $\theta(\chi)$ that maximizes the expectation of the likelihood.

$$\hat{\theta}(\chi) = \underset{\theta}{\text{argmax}}\ \mathbb{E}_\chi[\ell(\theta, \chi)] \tag{13}$$

In this context, the ***notion of independence*** (of observations) is ***the form of regularity*** that ultimately overcomes the curse of dimensionality.

Regarding the ***Fisher Information***, it is the idea of calculating the uncertainty about the parameter (and propagating it to the generalization estimation error). Since $\ell(\hat{\theta})$ is maximized, then

$$\left.\frac{\partial \ell}{\partial \theta}\right|_{\theta=\hat{\theta}} = 0 \tag{14}$$

and one can examine whether the maximum is more or less "narrow" by using, for example, higher-order derivatives (notion of *curvature*). Another way to approach the problem, if dealing with an ***unbiased estimator***, meaning [14] $\mathbb{E}(\hat{\theta}) = \theta$, is to look at the variance (given that $\mathbb{E}[\partial \ell(\hat{\theta})/\partial \theta] = 0$)

$$\boxed{I(\theta) = \mathbb{E}\left[\left(\frac{\partial \ell(\hat{\theta})}{\partial \theta}\right)^2\right]} \tag{15}$$

which Fisher calls *information* [15]. The Cramér-Rao result [16] provides an upper bound on

14. Expectation is to be understood in the sense that we are given a law for generating sets of observations χ, which makes $\hat{\theta}(\chi)$ random and allows us to calculate the expectation, variance, etc.

15. NDJE This is indeed the variance involving the first derivative of $\ell(\theta)$, but if the function is twice differentiable, we have the expectation of $-\partial^2 \ell/\partial \theta^2$ with the appropriate sign change.

16. Harald Cramér (1893-1985) and Calyampudi Radhakrishna Rao (1920-).

the estimation error

$$\mathbb{E}[(\hat{\theta} - \theta)^2] \geq \frac{1}{I(\theta)} \tag{16}$$

which gives meaning to the idea that ***the more (useful) information we have about the parameter θ, the better its estimation***.

It must be acknowledged that all of this scheme developed by R. Fisher is essentially what we attempt to do when performing stochastic gradient descent to train a neural network.

2.4 The Case of Neural Networks

Why is Fisher's framework at work in the optimization of neural networks? The problem is not so much about developing the formalism described earlier; the major challenge is to *specify the probability family* $p_\theta(x)$. Neural networks can be seen as a way to specify this family.

For example, in Figure 4, we have schematized different typical stages of a neural network. There is a cascade of linear filters (e.g., convolution), nonlinearities (e.g., ReLU), and finally a linear operation that yields a vector $z_y(x)$ from which, through a "`softmax`" operation [17], we obtain the probability distribution $p_\theta(y|x)$ as follows:

$$p_\theta(y|x) = \frac{e^{z_y(x;\theta)}}{\sum_{y'} e^{z_{y'}(x;\theta)}} \tag{17}$$

where y' ranges over the set of classes to be separated (e.g., digits from 0 to 9). In this context, the parameters θ include all the coefficients of the convolutional filters and the final linearity. The estimator of the network's output, here denoted as $\hat{y}$, maximizes the probability:

$$\hat{y} = \underset{y}{\operatorname{argmax}}\ p_\theta(y|x) \tag{18}$$

To optimize classification, we simultaneously compute $\hat{\theta}$ as the maximum likelihood, which, as we will see, is equivalent to minimizing the Kullback-Leibler "distance" [18], i.e.,

17. See, for example, the 2020 course Section 3.4.
18. See, for example, the 2019 course Section 7.2.3.

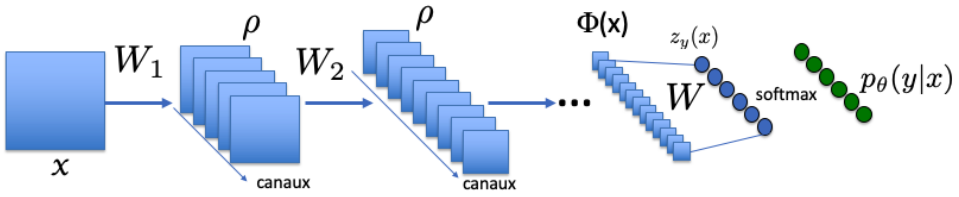

FIGURE 4 – Schematic of a multi-layer neural network (classification).

conditional entropy. If we denote $\mathcal{D} = \{x_i, y_i\}_{i \leq n}$

$$\hat{\theta} = \underset{y}{\operatorname{argmax}} \underset{\{x,y\} \sim \mathcal{D}}{\mathbb{E}} [\log p_\theta(y|x)] \tag{19}$$

Thus, we have a cost function to minimize $(-\ell(\theta))$, and for this, we use gradient descent [19].

The remarkable point is the realization that the probability families that neural networks allow us to access are quite generic, enabling the solution of very broad classes of problems such as image processing, text processing, audio analysis, physics/chemistry, etc. An important focus is then to understand the nature of these probability families and why these neural models are so versatile and complex.

2.5 Another Information: Shannon's Information

The Fisher information developed so far is based on the *a priori* assumption that we have a parameterized model, and we try to infer the best possible parameters based on a given criterion. Another entirely different type of information was introduced by **Claude Shannon** (1916-2001) in the 1940s. It is ***information that no longer depends on the model***. The idea is to ask what *intrinsic information* is contained within the data. Underlying these developments are communication issues between sender-receiver, as one needs to preserve the maximum amount of information in these exchanges. The foundational article dates back to 1948, and its title is similar to that of R. Fisher: *A*

19. See, for example, the 2018 course Section 10 and the 2019 course Section 8.

Mathematical Theory of Communication[20]. Like Fisher, Shannon wrote[21] an article that illuminated an entire field that remains relevant: the "why", the new tools, and the basic theorems.

The framework is the same as before: we have a series of *independent observations* $\chi = (x_i)_{i\leq n}$, generated by the same probability distribution[22], so they are *iid* data. If we denote $p(x_i)$ as the probability of observation x_i, then

$$p(\chi) = \prod_{i=1}^{n} p(x_i) \Longrightarrow \frac{1}{n}\log p(\chi) = \frac{1}{n}\sum_{i} \log p(x_i) \tag{20}$$

The right-hand side represents an average of the log-probabilities of x_i. Now, since the variables are independent, the law of large numbers tells us (if all goes well) that there is convergence as n tends to infinity:

$$\boxed{\lim_{n\to\infty} -\frac{1}{n}\log p(\chi) = \mathbb{E}[-\log p(x)] := \mathbb{H}[p]} \tag{21}$$

where $\mathbb{H}$ is the ***Shannon Entropy***[23]. The key property here, independently of any model, is that ***the probability of a set of observations tends to converge***.

Let's make this property explicit: the fact that it converges in probability means that for any $\varepsilon > 0$, we have

$$\mathbb{P}\left(\left|-\frac{1}{n}\log p(\chi) - \mathbb{H}[p]\right| \leq \varepsilon\right) \xrightarrow{n\to\infty} 1 \tag{22}$$

20. C. E. Shannon, The Bell System Technical Journal, Vol. 27, pp. 379–423, 623–656, July, October. https://people.math.harvard.edu/~ctm/home/text/others/shannon/entropy/entropy.pdf.

21. also at the age of 32, like Fisher.

22. S. Mallat notes that he uses the notion of probability according to the Lebesgue measure, but you can also think of the concept of measure directly.

23. NDJE: A bit of history on the concept of entropy, even though the exercise can't be exhaustive. Since the work of Rudolf Clausius (1822-88), who introduced the concept of entropy in 1865, and then the work of J. Clerk Maxwell (1831-79), who developed the theory of the distribution of velocities in gases, generalized in 1896 by Ludwig Boltzmann (1844-1906), who interpreted the entropy according to the famous formula "S = k log W" engraved on his tombstone, Statistical Mechanics has been based on the works of Josiah Willard Gibbs (1839-1903). In 1901, he wrote a book titled "*Elementary Principles in Statistical Mechanics developped with especial reference to the Rational Foundation of Thermodynamics*" (Yale Univ. published in March 1902), establishing a solid bridge between Statistical Mechanics and Thermodynamics and generalizing the statistical interpretation of a system's entropy, which Claude Shannon adopted in 1948.

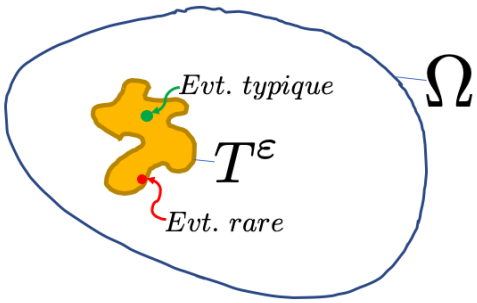

FIGURE 5 – Schematic of a typical set T^ε whose probability of membership tends to 1 as the number of observations tends to infinity. The size of the typical set is essentially proportional to the Shannon entropy $\mathbb{H}$. In green, a typical observation; in red, a rare observation found at the boundary of T^ε.

So, in reality, the observations x do not occupy the entire space Ω but concentrate in a space called the ***typical set*** T^ε, defined by the fact that (Fig. 5)

$$\boxed{T^\varepsilon = \left\{ \{x\} \in \Omega \ / \ \left| -\frac{1}{n} \log p(\{x\}) - \mathbb{H}[p] \right| \leq \varepsilon \right\}} \tag{23}$$

This set is potentially much smaller than the set Ω, and its size is determined by the Shannon entropy $\mathbb{H}$. In a sense, entropy will define *the minimum number of bits* required to encode the observations. This introduces a *notion of information*, the origin of which is more of an *uncertainty concept*, related to the size of the typical set. The remarkable point is that *the probability density within the typical set is uniform*. We ultimately find ourselves dealing with a geometry problem because characterizing the observations is equivalent to characterizing the geometry of the typical set.

The impact of these concepts is profound as it underpins the entire telecommunications industry (coding, channel capacity). Additionally, it resurfaces in Statistical Physics through the concepts of entropy and typical sets. In mathematics, when one wants to look at the probability of rare events, entropy is used again [24]. Rare events are found at the boundary of typical sets.

24. NDJE: S. Mallat refers to the *Theory of Large Deviations* developed in the 1960s in the lineage of C. Shannon, by authors including Harald Cramér (1893-1985), S. R. Srinivasa Varadhan, Jürgen Gärtner, Richard S. Ellis, Ivan Nikolaevich Sanov (1919-1968), and Edwin Thompson Jaynes (1922-98).

So, there is a beautiful theory developed by Shannon and his successors, but one needs to be able to characterize the typical sets. In Fisher's case, there is explicit parameterization, but in Shannon's case, one needs geometry. What is this geometry, then? One case that has been studied in great detail initially because it is simpler is that of ***Gaussian Processes***.

2.6 The Case of Gaussian Processes

In a sense, these Gaussian processes are Shannon's equivalent of the parametrization of the Gaussian family in Fisher's work. Let's denote the joint probability as follows [25]:

$$p_\theta(x) = Z^{-1} \exp\left(-\frac{1}{2} x^T \Theta^{-1} x\right) \tag{24}$$

where Θ is the covariance matrix of the Gaussian process (assumed to have zero mean):

$$\Theta = \mathbb{E}(xx^T) \tag{25}$$

(x is a $d \times 1$ dimensional vector, so Θ is of dimension $d \times d$). In this context, what do the typical sets correspond to? To understand this, we need to study the log-probability, which is very simple here:

$$-\log p_\theta(x) = \log Z + \frac{1}{2} x^T \Theta^{-1} x \tag{26}$$

The immediate idea that comes to mind is to diagonalize the matrix Θ and obtain its eigenvalues and eigenvectors. Then, we can write

$$\frac{1}{2} x^T \Theta^{-1} x = \sum_{k=1}^{d} \frac{x^2(k)}{2\sigma_k^2} \tag{27}$$

where $(x(k))_{k \leq d}$ are the d coordinates of x in a basis that diagonalizes the covariance matrix such that in this basis $\Theta = \mathrm{diag}(\sigma_1^2, \ldots, \sigma_d^2)$. In essence, we perform a Principal Component Analysis (PCA). ***Typical sets are thus characterized by ellipsoids with symmetry axes aligned with the principal axes of the covariance matrix***.

25. NDJE $x^T \Theta^{-1} x$ can be denoted as $\langle x, \Theta^{-1} x \rangle$.

However, even though the central limit theorem tends to support the usefulness of Gaussian Processes, it remains true that real-world problems are rarely Gaussian. Take a picture of our environment, for example; it contains numerous discontinuities that are essential for distinguishing objects from one another. In such cases, Gaussian Processes are incapable of capturing phenomena like turbulence, textures, etc. On the other hand, the remarkable capabilities of neural networks seem to be well-suited for modeling these phenomena. But in this case, characterizing typical sets is much more complex.

The central question, as developed in previous courses, is as follows: how, or through what underlying mechanism, are the probability families induced by neural networks generic? In the sense that the same type of operator cascades (convolution, rectifiers, etc.) can capture the characteristics of widely independent/disconnected problems.

2.7 Complexity and Architectural Structures

This is a topic that S. Mallat addressed in his 2020 course [26] regarding the role of Herbert A. Simon (1916-2001) [27], and his book *The Architecture of Complexity*, published in 1962 [28]. The question raised is: Are there generic families for data processing?

Herbert A. Simon wrote a book that is quite different from those of R. Fisher and C. Shannon, where he takes a step back from the field. In particular, he studies the *generic structures* of the "world" (meaning by observing what happens in biology, language processing, physics, etc.):

- ***Hierarchy*** is almost always the prevailing structure.
- A ***dynamic explanation*** (temporal) of this hierarchical structure is the ***search for stability*** (survival).
- ***Scale separability*** (within the hierarchy) enables overcoming the curse of dimensionality.

26. See the note in the 2020 course, Section 3.2.

27. Winner of the Sveriges Riksbank Prize in Economic Sciences in Memory of Alfred Nobel in 1978, but more importantly, the Turing Award in 1975 for his contributions to Artificial Intelligence, making him one of the pioneers of AI in the USA alongside Allen Newell (1927-92), with whom he shared the Turing Award.

28. Proceedings of the American Philosophical Society, Vol. 106, No. 6. (Dec. 12, 1962), pp. 467-482. https://www2.econ.iastate.edu/tesfatsi/ArchitectureOfComplexity.HSimon1962.pdf.

- The temporal description should be seen as ***aggregative processes*** tending toward global stability, rather than a succession of static states where the order is established from the beginning.

As inspiring and fascinating as reading such articles may be, in the end, one is left somewhat perplexed because there is no mathematical model to grasp onto, and the impact is not at all on the same scale as the articles by Fisher and Shannon. However, what has changed compared to the time when Simon wrote his article is that we now have algorithms that implement hierarchical structures (e.g., the sequence of convolutions followed by subsampling allows changing the analysis scale of an image, for example), but mathematics is not yet able to fully comprehend everything. Nonetheless, in order to try to understand the probability families underpinning neural networks, one must first have a good understanding of the fundamentals of Fisher and Shannon's theories. We will address this in the following sections through ***Coding Theory***, particularly image coding.

2.8 Image Coding

In a certain way, image coding boils down to specifying typical sets. If we use Gaussian Processes to describe an image, we'll see that it's essentially considering structures as smooth functions without any discontinuities, edges, etc. This won't work if we want fine details in the image description. What will genuinely help us is the use of ***sparse representations***, which was the topic of the 2021 course. We will now make the connection with coding.

Intuitively, by drawing the contours of objects in an image (Fig. 6), we already get a good description of it. By doing this, we're looking at the locations of ***singularities***. Can we describe the image in terms of "*transitions*"? If yes, then we realize that pixel information is highly redundant and can be compressed. So, the scheme is as follows:

- Be able to represent transitions/variations,
- Do this at different resolutions.

We can unfold this scheme using an ***orthogonal wavelet basis***[29]. The geometries of typical sets are then elongated along the axes of the basis due to ***sparsity***. Thus, in certain directions, wavelet decomposition coefficients can be large, but most coefficients are close to zero (Fig. 7). In the 2021 course, we saw the equivalence between the ability to perform

29. Also, see elements on this subject in the 2018, 2020, and 2021 courses.

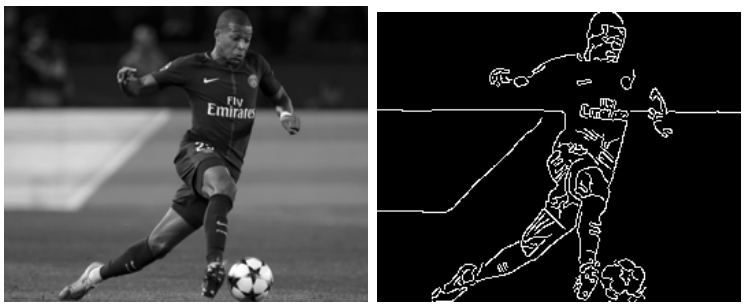

FIGURE 6 – Drawing contours allows us to understand what the image is telling us. And here, the algorithm used provides a lot (perhaps too much) information.

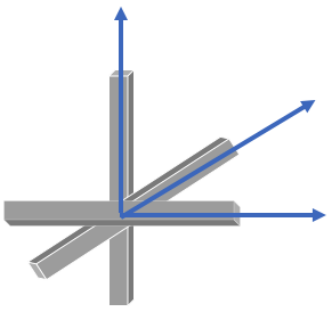

FIGURE 7 – Schematization of typical sets: when using an orthonormal wavelet basis, it concentrates coefficients to be mostly non-zero along the decomposition axes.

approximation, the existence of ***sparse representations***, and the presence of underlying ***regularity***, a triptych then called the RAP triangle. The implementation of the resulting codings is the JPEG-2000 standard, developed from 1997-2000 and officially ratified in 2015 by the three organizations ISO, IEC, and ITU. We will see that it combines wavelets with entropy coding to compress images.

Regarding the course organization and challenges, it's best to watch the recorded course video (>1:15 from the beginning).

3. Lecture 26 Jan.

3.1 Revisiting Determinism vs. Probabilism

S. Mallat revisits the difference in approach between *determinism* and *stochasticity* in high dimensions (Sec. 2.2).

Recall that when we want to relate a variable y to another variable x in the *deterministic* approach, we think of an unknown function f that exists beforehand, such that $y = f(x)$. If we have observations $\{x_i, y_i\}_{i \leq n}$ (Fig. 1), then we know the values of this function at the points $(x_i)_{i \leq n}$. In this context, the mathematical problem is one of *interpolation*, which works well when the function is *regular*. As mentioned in the previous session, interpolation in high dimensions can potentially be very challenging due to the *curse of dimensionality*. Nevertheless, it's worth keeping in mind that ***in low dimensions, interpolation is very effective***, and if we manage to redefine the problem by reducing the dimensionality, we have a powerful tool widely used in physics, where $x \in \mathbb{R}, \mathbb{R}^2, \mathbb{R}^3$, or even in problems involving time. It's also used, for example, in image processing, where interpolation can fill in "dead" areas of a CCD sensor. What happens in low dimensions, which is the key to success, is that the density of sampling points is high (or can be low), while in high dimensions, the problem becomes entirely different, as we've seen. Hence, the idea of considering a *probabilistic* approach.

Now, concerning the types of functions f, in the 2021 course, we delved into the notions of *regularity* and the relationship between *approximation* and *sparsity*. In low dimensions, functional analysis asks questions like: what function space does f belong

to? For instance, in Sobolev spaces, f has derivatives of a certain order, and in Hölder spaces, f has singularities of a certain type, etc. All of this works well in low dimensions. If we shift to high dimensions, which was also the basis for the 2021 course, what matters most is that the x variables of interest are *primarily* indexed by u ($x(u)$), a variable of *low dimension* (e.g., time in an audio sample, pixel position in an image), and *secondly*, these x variables concentrate in relatively small areas compared to the size of the possible space (see the notion of *typical set* in Sec. 2.5 and Fig. 5).

In the *probabilistic* case, what matters to us is not so much f but $p(y|x)$. Therefore, we deal with probability estimation problems, and the fundamental concept that allows us to overcome the curse of dimensionality is ***independence***. This is truly the point that makes statistics effective.

3.2 The Concept of Independence and Separability

Consider observations (e.g., pixels in an image, sound samples in an audio frame, words in a text), denoted as $x = \{x_i\}_{i \leq d}$. If these observations are independent, then we have

$$p(x_1, x_2, \ldots, x_d) = \prod_{i=1}^{d} p(x_i) \tag{28}$$

Why is this crucial in high dimensions? It's thanks to an argument already discussed in the 2020 course[30], namely the ***separability of variables***. Taking the logarithm, we get

$$\log p(x_1, x_2, \ldots, x_d) = \sum_{i=1}^{d} \log p(x_i) \tag{29}$$

This means that from a problem with d variables, we reduce it to d problems with one variable each, returning to the realm of "classic" low dimensions. In the *deterministic* case, we would ask whether it's possible to express the function $f(x)$ as a sum of functions f_k involving subsets of the variables in x, in order to reduce the dimensionality of each f_k.

So, in deterministic terms, we say "let's try to separate the original high-dimensional problem into smaller, hence simpler, subproblems" (akin to René Descartes' approach in

30. See Course 2020 Sec. 4.3

the "Discourse on the Method"). In probabilistic terms, we say "independence of random variables".

The challenge when dealing with real observations is trying to find this "independence" if it exists. For example, consider taking a photo of a tree bark and ask how to generate new images of tree barks. The problem is that the pixel values in the original photo are not independent, or perhaps there's a correlation at one scale and much less at another, and again a correlation at another scale, etc. ***So, either we assume a priori that the observations are independent, and everything works smoothly, or we need to discover the structures/scales that make the variables independent.***

3.3 The Law of Large Numbers: Convergence to the Mean

The law of large numbers tells us that when we have many observations available, frequencies converge to expectations, and we observe average phenomena. Once again, underlying this, we have the notion of independence [31]. The mathematical foundation is built on the works of R. Fisher from 1922 and the concepts of *estimator consistency*, *maximum likelihood*, *information*, and bounds/limits on *approximation*. S. Mallat tells us that compared to the typical course of statistics, he will delve into the realm of high dimensionality to highlight ***the non-obvious nature of these concepts***, and behind them lies the notion of ***optimization***. We can view problems from two perspectives: either that of statistics or optimization. For instance, the ***Hessian*** of likelihood allows us to control convergence, and the error of estimators, which is related to Fisher's information.

Regarding the convergence of a series of n random variables, there is the one introduced by Andrey N. Kolmogorov [32] (1903-87), who defined the ***strong law of large numbers***, which can be summarized by the expression: if we have a random variable $(r.v)$ that depends on n, the number of observations, such that $A_n \xrightarrow[n\to\infty]{} A$

$$\mathbb{P}\left[\lim_{n\to\infty} A_n = A\right] = 1 \tag{30}$$

31. NDJE This notion is crucial for assessing the effectiveness of certain statistical methods, such as Markov chain generation. It defines the efficiency of sampling or the size of the set of independent samples to judge the reliability of the statistics, such as confidence intervals.

32. NDJE Andrey N. Kolmogorov, as early as 1933, following the work of Émile Borel (1871-1956) and Henri Lebesgue (1875-1941), developed *probability theory* and established a link between *measure* and the *probability* of composite events.

but the one we generally use is the ***weak law of large numbers***, which states that

$$\boxed{\left(\forall \varepsilon > 0,\ \lim_{n\to\infty} \mathbb{P}\left[|A_n - A| \leq \varepsilon\right] = 1\right) \Leftrightarrow \left(A_n \xrightarrow[n\to\infty]{prob.} A\right)} \tag{31}$$

In a way, this law tells us that it is *rare* for A_n to deviate from its limit value A.

Here is the theorem in the case where the *r.v* A_n is the average of n *iid* random variables:

Theorem 1 ***(Weak Law of Large Numbers)***
Let $(X_i)_{i\leq n}$ be iid random variables, and $\mathbb{E}[X_i] = \mu < \infty$. Then, if $\bar{X}_n = \frac{1}{n}\sum_{i=1}^{n} X_i$, we have convergence in probability

$$\bar{X}_n \xrightarrow[n\to\infty]{prob.} \mu \tag{32}$$

The proof of this theorem is somewhat technical in the general case, but its main drawback is that it does not inform us about *the rate of convergence*. So, we will prove this theorem in the case where ***the variance*** σ^2 ***exists***, meaning $\mathbb{E}[X_i^2] < \infty$.

Proof 1.
Let $\sigma^2(\bar{X}_n)$ be the variance of the empirical mean. Then, according to the assumption of independence of *r.v*, we have simply

$$\sigma^2(\bar{X}_n) = \frac{\sigma^2}{n} \tag{33}$$

Therefore, it's clear that as n increases, $\bar{X}_n$ will be concentrated around its mean, which is μ, and simultaneously, the tails of the distribution will be weak. The technical point here is the Bienaymé-Tchebychev inequality [33]. It's a result of probability concentration [34], and

33. NDJE This is different from Chebyshev's inequality on sums.
34. The proof relies on the fact that $\forall x \in \mathbb{R}$, $\mathbf{1}[|x| \geq 1] \leq x^2$ ($\mathbf{1}$: indicator function). Applying this to $(X-\mu)/\alpha$ ($\alpha > 0$) and remembering the growth of expectation and that $\mathbb{E}[\mathbf{1}[A]] = \mathbb{P}[A]$, we arrive at the mentioned inequality.

for $Var[X] = \sigma^2$, it states

$$\mathbb{P}\left[|X - \mathbb{E}[X]| \geq \alpha\right] \leq \frac{\sigma^2}{\alpha^2} \tag{34}$$

So, by combining the two results, we have

$$\boxed{\mathbb{P}\left[\left|\bar{X}_n - \mu\right| \geq \varepsilon\right] \leq \frac{\sigma^2}{n\varepsilon^2} \xrightarrow[n\to\infty]{} 0} \tag{35}$$

The significant consequence is that the *convergence* of $\bar{X}_n$ to the expectation μ is *sufficiently rapid* at a rate of $1/n$. ■

3.4 Consistency: Parameter Estimation

Following R. Fisher's idea, we are given n observations, and we want to estimate the underlying probability distribution, using a family of parameterized probabilities $p_\theta(x_i)$. We need to estimate the "best" θ.

Definition 1 *(**Consistency**)*
Let T_n be a statistic as a function of $(X_1, \ldots, X_n)$ (n random variables). We say it is a consistent estimator of θ if

$$T_n(X_1, \ldots, X_n) \xrightarrow[n\to\infty]{prob.} \theta \tag{36}$$

For example, an estimator of the mean μ is the empirical mean $\bar{X}_n$, and concerning the variance, we can think of

$$T_n(X_1, \ldots, X_n) = \frac{1}{n}\sum_{i=1}^{n}(X_i - \bar{X}_n)^2 = \frac{1}{n}\sum_{i=1}^{n} X_i^2 - \bar{X}_n^2 \tag{37}$$

To study the convergence, we need a little theorem.

Theorem 2 *Let a series of random variables A_n converge to A in probability, and let g be a continuous function, then $g(A_n)$ converges in probability to $g(A)$.*

Proof 2. The proof follows from the continuity assumption, which states that

$$\forall \varepsilon > 0 \ \exists \alpha > 0 \text{ s.t. } |a - a'| \leq \alpha \Rightarrow |g(a) - g(a')| \leq \varepsilon \tag{38}$$

So we have

$$1 \geq \mathbb{P}(|g(a) - g(a')| \leq \varepsilon) \geq \mathbb{P}(|a - a'| \leq \alpha) \xrightarrow[n\to\infty]{prob.} 1 \tag{39}$$

hence the result. ■

So, as $\bar{X}_n$ converges in probability to μ, then $\bar{X}_n^2$ also converges in probability to μ^2. Similarly, by setting $Y_i = X_i^2$, $\bar{Y}_n$ converges to the expectation $\mathbb{E}[X_i^2]$, so

$$T_n(X_1, \ldots, X_n) = \frac{1}{n}\sum_{i=1}^{n}(X_i - \bar{X}_n)^2 \xrightarrow[n\to\infty]{prob.} \mathbb{E}[X_i^2] - \mu^2 \tag{40}$$

which gives the convergence in probability to the variance of X_i. What will determine the convergence dynamics is now the variance of X_i^2. Therefore, we will impose that $\mathbf{E}[X_i^4] < \infty$ to apply the law of large numbers.

Now, in the more general case, R. Fisher uses maximum likelihood to obtain consistent estimators.

3.5 Maximum Likelihood

First, let's give a definition of likelihood according to Fisher:

Definition 2 ***(Likelihood)***
Let $X = \{X_i\}_{i\leq n}$ be iid random variables. The likelihood of these observations for a

parameter θ is defined as

$$\mathcal{L}_\theta(X) = p_\theta(X) = \prod_{i=1}^{n} p_\theta(X_i) \tag{41}$$

Fisher's idea is to say that if $\mathcal{L}_{\theta_1}(X) > \mathcal{L}_{\theta_2}(X)$, then generating the observations is more likely if we take $\theta = \theta_1$ than if we take $\theta = \theta_2$. In an abuse of language, we often take a shortcut and say "θ_1 is more likely than θ_2." It would be better to say "θ_1 is a better estimator than θ_2."

Definition 3 *(**MLE**/Maximum Likelihood Estimator)*
The MLE is defined as

$$\hat{\theta}_{MLE}(X) = \underset{\theta}{\operatorname{argmax}}\ \mathcal{L}_\theta(X) \tag{42}$$

The question is whether $\hat{\theta}_{MLE}(X)$ converges to the true[35] value of θ. We need properties of probability to answer this.

Property 1 *(**regularities**)*
We assume the following properties, knowing that $\theta \in \Omega \subset \mathbb{R}^d$:

— *The property of identification*

$$\theta = \theta' \Rightarrow p_\theta = p_{\theta'} \tag{43}$$

— *The supports of p_θ are identical (not necessarily necessary, but practical because it avoids singularities when calculating log-probabilities).*

— *The observations are actually generated by a $\theta^* \in \Omega$.*

Thus, we can formalize the intuition we have about maximum likelihood (see also Sec. 4.7).

35. NDJE: We assume that the data was generated according to a probability of the same family $p_{\theta_{true}}(X)$ that we use for analysis. When conducting numerical simulations, we can control everything, but in real life, what happens if we choose the wrong family?

Theorem 3

Let θ^* be the parameter of the probability underlying the iid observations $X = \{X_i\}_{i\leq n}$, then

$$\forall \theta \neq \theta^* \qquad \mathbb{P}\left(\mathcal{L}_{\theta^*}(X) > \mathcal{L}_\theta(X)\right) \xrightarrow[n\to\infty]{prob.} 1 \tag{44}$$

Proof 3.

We form the ratio of likelihoods and using the log-likelihood $\ell(\theta) = \log \mathcal{L}_\theta$, we have

$$\frac{\ell(\theta)}{\ell(\theta^*)} = \sum_{i=1}^{n} \log \frac{p_\theta(X_i)}{p_{\theta^*}(X_i)} \tag{45}$$

We need to evaluate the probability that $\ell(\theta)/\ell(\theta^*) < 0$, or equivalently

$$\frac{1}{n}\sum_{i=1}^{n} \log \frac{p_\theta(X_i)}{p_{\theta^*}(X_i)} < 0 \tag{46}$$

If we set $Y_i = \log \frac{p_\theta(X_i)}{p_{\theta^*}(X_i)}$, we indeed have independent random variables, so in probability, the left member converges to an expectation. Therefore, we need to evaluate the probability of

$$\mathbb{E}_X\left[\log \frac{p_\theta(X_i)}{p_{\theta^*}(X_i)}\right] < 0 \tag{47}$$

Now, the logarithm is a concave function, and Jensen's inequality gives us

$$\phi \text{ concave function} \Rightarrow \phi(\mathbb{E}(X)) \geq \mathbb{E}(\phi(X)) \tag{48}$$

Strict concavity implies strict inequality. Thus, we know that

$$\mathbb{E}_X\left[\log \frac{p_\theta(X_i)}{p_{\theta^*}(X_i)}\right] < \log\left(\mathbb{E}_X\left[\frac{p_\theta(X_i)}{p_{\theta^*}(X_i)}\right]\right) \tag{49}$$

Now, the observables are drawn according to the $p_{\theta^*}(X)$ distribution, so

$$\mathbb{E}_X\left[\frac{p_\theta(X_i)}{p_{\theta^*}(X_i)}\right] = \int p_{\theta^*}(x)\frac{p_\theta(x)}{p_{\theta^*}(x)}dx = 1 \tag{50}$$

Thus, we have convergence in probability such that

$$\exists \mu \quad \text{s.t.} \quad \bar{Y_n} = \frac{1}{n}\sum_{i=1}^{n} Y_i = \frac{1}{n}\sum_{i=1}^{n} \log \frac{p_\theta(X_i)}{p_{\theta^*}(X_i)} \xrightarrow[n\to\infty]{prob.} \mu \ < 0 \tag{51}$$

So, if we choose $\varepsilon = |\mu|/2$, we ensure that $\bar{Y_n}$ is negative because, according to the law of large numbers (Th. 1)

$$\mathbb{P}(|\bar{Y_n} - \mu| \leq \varepsilon) \xrightarrow[n\to\infty]{prob.} 1 \tag{52}$$

Thus, going back to the question posed (Eq. 46), we guarantee that $\ell(\theta) < \ell(\theta^*)$ with a probability that tends to 1 as n tends to infinity, which gives the theorem. ■

3.6 Some Examples

We will explore through a few examples that the concepts described in the previous sections are not as trivial as they may seem.

3.6.1 Median Estimator vs. Empirical Mean

So, we aim to determine $\hat{\theta}$ that maximizes the likelihood $\mathcal{L}(\theta) = p_\theta(x)$ or rather the log-likelihood denoted as $\ell(\theta)$ (Definition 3). In this context, if we define

$$\hat{\theta} = \underset{\theta}{\text{argmax}}\ \ell(\theta) \Rightarrow \left.\frac{\partial \ell}{\partial \theta}\right|_{\theta=\hat{\theta}} = 0 \tag{53}$$

we define the *score*, which is nothing but the derivative of $\ell(\theta)$ that we will try to set to zero.

Let's take the Laplace distribution [36]

$$p_\theta(x) = \frac{1}{2}\exp\{-|x-\theta|\} \tag{54}$$

36. Here we take the second parameter of the Laplace distribution equal to 1.

and we want to identify the parameter θ, so let's go through the formalism. If we have n observables

$$\ell(\theta) = -n\log 2 - \sum_{i=1}^{n} |x_i - \theta| \Rightarrow \partial_\theta \ell(\theta) = \sum_{i=1}^{n} \text{sign}(x_i - \theta) \tag{55}$$

To set the score to zero, there must be as many positive signs as negative signs, and thus

$$\hat{\theta}_{Laplace} = \text{median}(\{x_i\}_{i\leq n}) \tag{56}$$

If we had taken a *known variance* Gaussian distribution, then

$$p_\theta(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{(x-\theta)^2}{2\sigma^2}\right\} \tag{57}$$

and therefore

$$\ell(\theta) = -\frac{n}{2}\log\left(2\pi\sigma^2\right) - \frac{1}{2\sigma^2}\sum_i (x_i - \theta)^2 \Rightarrow \partial_\theta \ell(\theta) \propto \sum_i (\theta - x_i) = n\theta - \sum_i x_i \tag{58}$$

Thus, the estimator is the empirical mean of x_i

$$\hat{\theta}_{Gaussian} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{59}$$

What is curious is that in both cases, Laplace vs. Gaussian, we need to estimate the mean of the distribution, yet we have two estimators: the first one is the *median*, and the second is the *empirical mean*. The issue with the Laplace distribution lies in the slow decay of the distribution tails, which can generate observations far from the mean (called *outliers*). Now, if we calculate an empirical mean with outliers that occur infrequently, we get large dispersions, whereas the median calculation is much more robust against outliers. This outlier phenomenon is not just a mathematical anecdote because in signal processing, physics, economics, etc., we are confronted with such issues.

3.6.2 Gradient Descent in High Dimensions

Now, let's move on to high dimensions and first, let's study the ***logistic classifier***[37]. It's a classification problem, yet it's often referred to as *logistic regression*. So, we're in

37. Course 2018 Sec. 9.6, Course 2019 Sec. 7.3.3

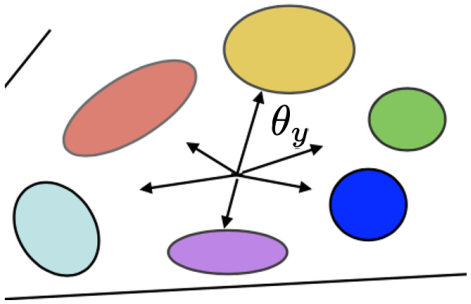

FIGURE 8 – Schematic of the objective in logistic classification, trying to find the preferred directions θ_y pointing towards different clusters of observations with the same label.

a scenario where we want to estimate the conditional probability $p(y|x)$ (y being a label and x an observation) in Fisher's formalism, where the probability family is indexed by θ. Therefore, we aim to identify the best y ($\hat{y}$) for a given x, but before that, we need to determine the best θ from a training set $\{x_i, y_i\}_{i\leq n}$. The probability family is defined as follows:

$$p_\theta(y|x) = \frac{e^{\langle x,\theta_y\rangle}}{\sum_{y'} e^{\langle x,\theta_{y'}\rangle}} = \text{softmax}(\langle x, \theta_y\rangle) \tag{60}$$

where $x \in \mathbb{R}^d$, and the goal is to find the preferred directions $\hat{\theta}_y$ pointing towards areas where observations with the same labels aggregate (Fig. 8). This is because we will choose, as the class estimator for a new observation x^{new}, the label such that:

$$\hat{y} = \underset{y}{\text{argmax}}\ p_{\hat{\theta}}(y|x^{new}) \tag{61}$$

We will connect this problem with the ***identification of exponential distributions***, which is central to Statistical Physics.

Let's define ***hot-vectors*** (dimension $K \times 1$), which are widely used in machine learning:

$$y_i = (0, \ldots, 0, 1, 0, \ldots, 0)^T \tag{62}$$

where the 1 is positioned at the i-th place to identify the class $c(x_i) = c_i$ of observation x_i among K possible classes. Simultaneously, Θ is a $d \times K$ matrix defined by the column

vectors θ_y mentioned earlier, such that:

$$\Theta = (\theta_1, \theta_2, \dots, \theta_K) \tag{63}$$

With these notations:

$$\langle x, \theta_{c_i} \rangle = \langle x, \Theta\ y_i \rangle = x^T \Theta\ y_i \tag{64}$$

Therefore, we can place ourselves in the following family of probabilities:

$$p_\theta(y|x) = Z_\Theta^{-1}(x) \exp\Big\{x^T \Theta y\Big\} \tag{65}$$

The crucial point is that ***the argument of the exponential is linear in the parameters***, which simplifies our task. The log-likelihood for n observations [38] $(x_i, y_i)_{i \leq n}$ becomes:

$$\ell(\Theta) = \sum_{i=1}^{n} x_i^T \Theta\ y_i - \sum_{i=1}^{n} \log \left(\sum_{k=1}^{K} \exp\Big\{x_i^T \Theta\ y_k\Big\} \right) = -\tilde{\ell}(\Theta) \tag{66}$$

Notice that the terms $y_k x_i^T$ form a matrix representing the correlation between observations and classes:

$$x^T \Theta\ y = \sum_{kk'} x_{1,k} \Theta_{k,k'} y_{k',1} = \sum_{kk'} \Theta_{k,k'} (y x^T)_{k',k} := \Theta \bullet (y x^T) \tag{67}$$

The notation $\bullet$ means that we flatten the matrix coefficients to form a vector of dimension Kd and expose an inner product.

How do we get the coefficients of Θ (optimization problem)? In the case at hand, we will perform *gradient descent* [39] (GD) using the cost function $\tilde{\ell}(\Theta) = -\ell(\Theta)$, and we know that this method *converges*. Implicitly, for this to converge, there is a ***convexity property***. The GD algorithm proceeds with an initialization of parameters Θ_0, and step

38. NDJE here, i indexes an observation, and y_i is the hot-vector associated with that observation, encoding its class.

39. See Course 2018 Sec. 10 and 2019 Sec. 8, for example.

by step, it updates as follows (t can be seen as discrete time):

$$\Theta_t = \Theta_{t-1} - \eta \nabla_\Theta \tilde{\ell}(\Theta_{t-1}) \tag{68}$$

with $\eta > 0$. Let H be the Hessian matrix:

$$H[\tilde{\ell}][\Theta] = \left(\frac{\partial^2 \tilde{\ell}}{\partial \Theta_i \partial \Theta_j} \right) \tag{69}$$

As H is a symmetric matrix, it can be diagonalized, and if all eigenvalues are positive, we say that H is positive, denoted as $H \geq 0$. In 1D, this corresponds, for example, to the curvature of the function x^2. If we perform an expansion around Θ_0, then ($g = \nabla_\Theta \tilde{\ell}(\Theta_0)$):

$$\begin{aligned} \tilde{\ell}(\Theta) &= \tilde{\ell}(\Theta_0) + (\Theta - \Theta_0)^T \nabla_\Theta \tilde{\ell}(\Theta_0) + \frac{1}{2}(\Theta - \Theta_0)^T H[\tilde{\ell}](\Theta_0)(\Theta - \Theta_0) \\ &= \tilde{\ell}(\Theta_0) - \eta \|g\|^2 + \frac{\eta^2}{2} g^T H[\tilde{\ell}](\Theta_0) g \end{aligned} \tag{70}$$

The GD method indeed reduces $\tilde{\ell}(\Theta)$ to the first order, and the ***existence of a minimum*** condition implies:

$$\|g\| = 0, \qquad g^T H[\tilde{\ell}](\Theta_0) g \geq 0 \tag{71}$$

The optimal step is then ($\nabla_\Theta \tilde{\ell}(\Theta_1) = 0$):

$$\Theta_1 - \Theta_0 = -\left(H[\tilde{\ell}](\Theta_0) \right)^{-1} \nabla_\Theta \tilde{\ell}(\Theta_0) \tag{72}$$

However, for this to work well, the Hessian must be invertible (**the smallest eigenvalue of the Hessian must be non-zero**). This scheme is ***of the 2nd order***, where we can access second derivatives. The problem is that in high dimensions, this calculation is very costly, if not impossible, so we resort to a **1st order** scheme (Eq. 68) with various scheduling strategies to evolve the parameter (learning rate) η over time (t). In particular, the η factor is bounded because we don't want to take steps larger than what the method with the Hessian allows. So let λ_{max} be the largest eigenvalue of the Hessian; we have the

following bound:

$$\eta < \frac{1}{\lambda_{max}} \tag{73}$$

However, if the gap between the smallest eigenvalue of the Hessian, denoted as λ_{min}, and the largest, λ_{max}, is too significant, then by forcing too small steps to constrain the direction associated with λ_{max}, we will get stuck and stagnate in the direction associated with λ_{min}. This is reflected in the concept of Hessian's condition:

Theorem 4 *(Convergence of GD)*
Let $\lambda_{min} > 0$ and λ_{max} be the minimum and maximum eigenvalues of the ***Hessian****, gradient descent converges if $\eta \leq \frac{1}{\lambda_{max}}$, and the difference between Θ_t and the optimal value Θ^* is given by:*

$$\|\Theta_t - \Theta^*\| \leq \left(1 - \frac{\lambda_{min}}{\lambda_{max}}\right)^t \|\Theta_0 - \Theta^*\| \leq \|\Theta_0 - \Theta^*\| \exp\left\{-\frac{\lambda_{min}}{\lambda_{max}} t\right\} \tag{74}$$

The conditioning rate is given by $\tau = \frac{\lambda_{min}}{\lambda_{max}}$ (τ is the inverse of the Hessian's conditioning).

The theorem tells us that ***the gradient descent method converges especially in the case of exponential families (linear in the parameters), but convergence can be very slow if the Hessian is ill-conditioned***. This is very important, and all the issues related to Hessian's conditioning come from ***Fisher's information***, which defines the statistical properties of the estimator. Through this, we see the connection between the field of *optimization* and *statistics* at the heart of Machine Learning. We cannot think of one without the other.

4. Lecture 2 Feb.

4.1 A brief recap of the previous session

We delved into the connection between *optimization* and *statistics*, two inseparable domains of current Machine Learning. We started studying the development of Fisher's

theory of maximum likelihood, focusing on the *exponential family* of probabilities, where the log-probability depends linearly on the parameters. This case encompasses nearly all of Statistical Physics. We will continue this study because, although the mathematics are somewhat simpler, algorithms converge to a unique minimum, it allows us to tackle the issue of high dimensionality. We will explore *convergence* and the *consistency* of maximum likelihood estimators, which will lead us to the concept of **Fisher Information**. This, through the Hessian, regulates the *convergence conditions* of algorithms, defines the *geometry of the optimization space*, and the *parameter estimation errors* (Cramér-Rao bound).

Referring to section 3.6.2, the *gradient descent* algorithm on $\tilde{\ell}(\Theta)$ can also be seen as *gradient ascent* on the log-probability $\ell(\Theta)$. So, for reference [40] (Eqs. 68, 69), at step t of the algorithm, the parameter update Θ is done through the following relationship:

$$\Theta_t - \Theta_{t-1} = -\eta \nabla_\Theta \tilde{\ell}(\Theta_{t-1}) = \eta \nabla_\Theta \ell(\Theta_{t-1}) \tag{75}$$

with $\eta > 0$; and the Hessian matrix

$$H[\ell][\Theta] = -\left(\frac{\partial^2 \ell}{\partial \Theta_i \partial \Theta_j}\right) = -H[\tilde{\ell}][\Theta] \tag{76}$$

must be *positive* for the minimization to be *convex*. If we require *strict positivity*, meaning that the smallest eigenvalue of H is nonzero [41], then convergence is guaranteed, but the convergence speed can be very slow. We need to consider *Hessian conditioning* (κ) or the *conditioning tau* τ:

$$\tau = \frac{\lambda_{min}}{\lambda_{max}} = \kappa^{-1} \tag{77}$$

which led us to Theorem 4, stating that

$$\|\Theta_t - \Theta^*\| \leq \|\Theta_0 - \Theta^*\| e^{-t/\kappa} \tag{78}$$

The conditioning is better when $\kappa \approx 1$. If, on the other hand, $\kappa \gg 1$, convergence is very slow. This occurs, for example, in the case shown in Figure 9 where one of the

40. NDJE: I'm maintaining consistency with my notations from the previous session. In the video, S. Mallat uses the notation θ for parameters. I hope this isn't too confusing.

41. Reminder: all eigenvalues of the Hessian are positive or zero.

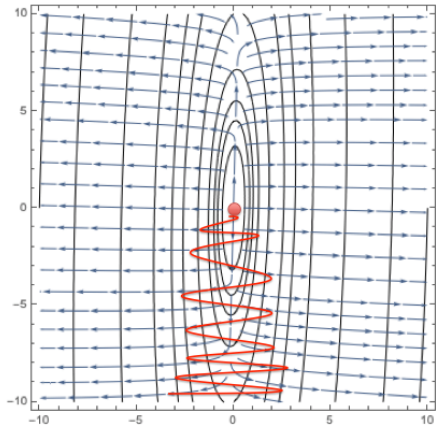

FIGURE 9 – A convex case where the landscape is unfavorable because one direction has very low curvature: the algorithm "oscillates" in the direction of high curvature with very slow progress in the low curvature direction.

two directions in the parameter plane has very low curvature. Indeed, the step size η is controlled by the largest curvature to prevent taking overly large steps. The remedy for this is to make the step η depend on the direction (second-order method where $\eta = H^{-1}(\Theta_t)$) to adjust better and increase the convergence speed. However, *in high dimensions, we can never use a second-order method* because the Hessian is a huge matrix that is impossible to estimate and, consequently, invert. We need other methods to train large neural networks. Nevertheless, we can attempt to precondition the Hessian.

4.2 Case of Exponential Distributions

These distributions, which encompass Statistical Physics, also cover the case of logistic classification discussed in the previous session. Let's consider the probability family expression as follows:

$$p_\theta(x) = Z_\theta^{-1} e^{-\theta \bullet U(x)} \tag{79}$$

where the symbol $\bullet$ has been exaggerated intentionally to emphasize that it represents a potentially high-dimensional dot product (later, it will be reduced to a . and then disappear). Here, $U(x)$, which in Physics is the potential, is a family of functions $\{U_k(x)\}_{k \leq p}$

representing, for example, different types of interactions. To be complete:

$$\theta \cdot U(x) = \sum_k \theta_k U_k(x) \tag{80}$$

Regarding the constant Z_θ (*partition function* in Stat. Physics), it is such that:

$$\int p_\theta(x)dx = 1 \Rightarrow Z_\theta = \int e^{-\theta \cdot U(x)}\ dx \tag{81}$$

We assume that the conditions are met for this integral to make sense: typically in Physics, interaction potentials are either of finite range or vanish at infinity. This probability family makes it simple to calculate the log-likelihood:

$$-\ell(\theta) = -\log p_\theta(x) = \log Z_\theta + \theta \cdot U(x) \tag{82}$$

Note that even if the $U_k(x)$ are potentially nonlinear in x, what matters in the optimization method is the gradient of $-\ell(\theta)$ with respect to θ and not with respect to x.

In the case of a neural network, the function $U(x)$ is denoted as $\Phi(x)$ in Figure 4, and the parameters θ are used to construct an estimator of the log-probability. The $U(x)$ is the result of the sequence of linear and nonlinear operators through which the input x passes. However, it should be noted that in the case of neural networks, $U(x)$ itself depends on parameters. But, it is assumed that there is enough *a priori* information [42] (e.g., system symmetries) so that $U(x)$ does not need to be learned.

Now, let's calculate all the important quantities that will allow us to explore general optimization concepts.

Theorem 5

Consider the partition function:

$$Z_\theta = \int e^{-\theta \cdot U(x)}\ dx \tag{83}$$

42. The theme of the 2020 course, particularly see Sec. 9.5 *Scattering Operators*.

It allows us to calculate all the "average" quantities[a]

$$-\nabla_\theta \log Z_\theta = \mathop{\mathbb{E}}_{x\sim p_\theta(x)}[U] \tag{84}$$

Regarding minimization (gradient descent), we have

$$-\nabla \ell(\theta) = U(x) - \mathbb{E}_\theta[U] \tag{85}$$

In the case of a realization of observables x, the set of parameters to which the minimization leads satisfies $U(x) = \mathbb{E}_\theta[U]$. Finally, the Hessian that governs the convergence speed is given by the covariance of the potential U, i.e.:

$$-H[\ell](\theta) = Cov_\theta(U) \tag{86}$$

a. To simplify notation, we will denote the right-hand side expectation as $\mathbb{E}_\theta$.

Proof 5.

To prove the first two results, it suffices to calculate the gradients, which becomes elementary for the exponential family considered here. Note in passing that if we consider the dependence on a particular parameter θ_k, we have:

$$-\nabla_{\theta_k} \ell(\theta) = U_k(x) - Z_\theta^{-1} \int U_k(x) e^{-\theta \cdot U(x)}\, dx = U_k(x) - \mathbb{E}_\theta[U_k] \tag{87}$$

which can be vectorized easily. Concerning the Hessian, as the potentials $U_k(x)$ do not depend on the parameters θ, we get:

$$\begin{aligned} -\nabla_{\theta_q} \nabla_{\theta_k} \ell(\theta) &= -Z_\theta^{-1} \int U_q(x) e^{-\theta \cdot U(x)}\, dx \times Z_\theta^{-1} \int U_k(x) e^{-\theta \cdot U(x)}\, dx \\ &\quad + Z_\theta^{-1} \int U_q(x) U_k(x) e^{-\theta \cdot U(x)}\, dx \\ &= -\mathbb{E}_\theta[U_q]\mathbb{E}_\theta[U_k] + \mathbb{E}_\theta[U_q U_k] = cov_\theta(U_q, U_k) \end{aligned} \tag{88}$$

which can also be put in matrix form if we consider U as a vector of dimension $p \times 1$ and UU^T of dimension $p \times p$:

$$-H[\ell](\theta) = \mathbb{E}_\theta[UU^T] - \mathbb{E}_\theta[U]\mathbb{E}_\theta[U^T] \tag{89}$$

■

It is noteworthy that $H[\ell](\theta)$ does not depend on x; it only depends on expectations, which are probability averages integrated over x.

4.3 Consistency (BatchNorm)

We would like the Hessian to be as close to the identity as possible to ensure optimal conditioning. Suppose $\mathbb{E}_\theta[U] = 0$. The diagonal terms of the Hessian are then the variances $\sigma_k^2 = \mathbb{E}_\theta[U_k^2]$. We can perform a rescaling:

$$U_k' = \frac{U_k}{\sigma_k} \tag{90}$$

This forces the diagonal terms of the new Hessian to be equal to 1, thereby improving conditioning and accelerating optimization. The operation that achieves this in neural networks is ***BatchNorm***[43].

Is it sufficient to impose that the diagonal elements be equal to 1? Let's consider a counterexample using the (discrete) second derivative operator:

$$-f''(x) \approx \frac{-f(x-h) + 2f(x) - f(x+h)}{h^2} \tag{91}$$

Consider, for example, the following banded matrix:

$$O = \begin{pmatrix} 2 & -1 & 0 & \dots & 0 & -1 \\ -1 & 2 & -1 & 0 & \dots & 0 \\ & & & \vdots & & \\ -1 & 0 & \dots & 0 & -1 & 2 \end{pmatrix} \tag{92}$$

The fundamental observation to make reflexively is that O is a convolution operator, thus

43. See Lecture 2019 Sec. 8.2.3.

diagonalizable on a Fourier basis [44]. Either one analyzes it discretely and realizes that the extreme eigenvalues of the matrix are close to 0 and 4, or one analyzes it continuously. In the continuous case, the eigenvectors of the second derivative operator (up to sign) are of the form $e^{i\omega t}$, and the eigenvalues are ω^2, resulting in equally poor conditioning. All of this to say that "conditioning" the Hessian's diagonal isn't sufficient. ***We need/would need to work in a representation where the natural basis is the Fourier basis***, because in such a representation, BatchNorm ensures that the diagonal terms are equal to 1 without band terms. In fact, under these conditions, using BatchNorm is akin to using a second-order technique without explicitly saying so. The challenge is that we don't *a priori* know the basis that diagonalizes the representation $U(x)$, especially when there are nonlinearities. Nevertheless, we would like to get closer to it, and that's at the heart of ***constructing neural network architectures***.

4.4 Connection with Information Geometry

NDJE: S. Mallat mentions the 2 seminars dedicated to this topic, the first after this session, and the other associated with the next session.

The idea is that the $p_\theta(x)$ are mappings from $\mathbb{R}^p$ to $\mathbb{R}$; they form manifolds, and gradient descent takes us along these manifolds until we reach the point $p_{\theta^*}(x)$. To these manifolds, we attach measures (Riemannian) that, at each point θ, involve the tangent plane, whose principal axes are precisely defined by the Hessian. Therefore, moving efficiently on these manifolds is equivalent to using a second-order method by utilizing the inverse of the Hessian. We can understand this concept with the Kullback-Leibler pseudo-distance, which will be discussed later in this year's course.

In the following, we will explore examples, starting with Gaussian distributions.

44. See, for example, Lecture 2021 Sec. 3.4 *Fourier Analysis*, Lecture 2020 Sec. 6.2, Lecture 2018 Sec. 5.2 for a development of Discrete Fourier Analysis. Also, refer to S. Mallat's book chapters.

4.5 Gaussian Distributions

Let's take the following parameterization of the probability density (zero mean):

$$p(x) = Z^{-1} \exp\left\{-\frac{1}{2} x^T C^{-1} x\right\} \tag{93}$$

with $Z = (2\pi)^{p/2}|C|^{1/2}$. In this case, the parameter vector θ is composed of the covariance matrix C^{-1}. To set the notations, x is a $p \times 1$ vector, and C^{-1} is a positive definite symmetric matrix of dimension $p \times p$, $[C^{-1}]_{kk'} = c_{kk'}$, and the Gram matrix $[xx^T]_{kk'} = x_k x_{k'}$:

$$x^T C^{-1} x = \sum_{k,k'} x_k c_{kk'} x_{k'} := C^{-1} \bullet (xx^T) \tag{94}$$

where we group the elements of the covariance matrix and the Gram matrix into two vectors of dimension p^2 to compute a dot product. Thus, under these conditions, we can rewrite the probability density as follows:

$$p_\theta(x) = Z_\theta^{-1} \exp\{-\theta \bullet U(x)\} \qquad U(x) = \frac{1}{2} xx^T \tag{95}$$

Recall that the covariance matrix C satisfies, for a realization of x ($x \sim p_\theta(x)$):

$$C_{kk'} = [Cov_\theta(U)]_{kk'} = (\mathrm{E}_\theta(x_k, x_{k'}))_{kk'} \tag{96}$$

The point we outlined in the previous section is that we need to identify the basis in which the covariance is diagonal. If we consider a *stationary case*[45] (in an image, this would be the case if we consider translation invariance), then

$$\mathrm{E}_\theta(x_k, x_{k'}) = F(k - k') \tag{97}$$

Translation invariance (Toeplitz matrix) indicates that the diagonalization basis is the Fourier basis. An eigenvalue σ_k^2 of the covariance matrix in this basis is called *spectral power* here indexed by ω ($\sigma_\omega^2 = P(\omega)$), and in the case of a classic image, the power spectrum behaves as a power law $\approx 1/|\omega|^2$ as shown in Figure 10. Therefore, the typical difference between the smallest and largest eigenvalues is very large.

45. See Course 2021 Sec. 4.4

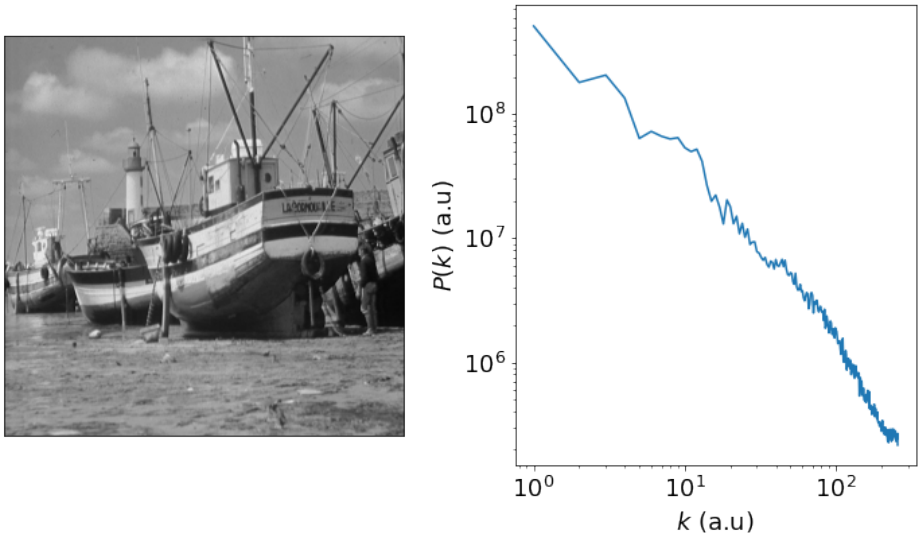

FIGURE 10 – Power spectrum of a conventional image, obtained as the radial mean of the squared norm of the 2D Fourier transform of the image. Here k is the wavenumber in arbitrary units. There is a $1/k^2$ dependency at large k values, i.e. the small scales of the image (at small k we have rather a $1/k$ dependency).

The geometry of realizations x is that of an ellipsoid in dimension p. If we take a small iso-probability volume:

$$dV(\alpha) = \{x, \quad 0 \leq \alpha \leq p_\theta(x) \leq (\alpha + d\alpha) \leq 1\} \tag{98}$$

For example, imagine we are in dimension 2 and in the diagonal basis $C^{-1} = \text{diag}(\sigma_{min}^{-2}, \sigma_{max}^{-2})$, then we indeed have layers of ellipses:

$$-2\log(Z(\alpha + d\alpha)) \leq \frac{x_1^2}{\sigma_{min}^2} + \frac{x_2^2}{\sigma_{max}^2} \leq -2\log(Z\alpha) \tag{99}$$

which become smaller (or larger) as α approaches 1 (or 0). It can be seen that what matters is the product of the probability value and the iso-probability volume. The ellipsoids are *typical sets* (Eq. 23) introduced by C. Shannon.

4.6 Beyond Gaussian Fields

S. Mallat provides some examples from Fluid Mechanics (e.g., turbulence) and Cosmology (e.g., interstellar gas). In Figure 11, we have an example on the left of an image of turbulent fluid [46], in the center its power spectrum, and on the right, a realization of a Gaussian field generated from this power spectrum. To do this, all we need to do is measure the two-point correlation function (Fourier transform of the power spectrum), which is estimating the covariance matrix. At first glance, what we notice is that ***the Gaussian field lacks structures as the turbulent field does***. However, Gaussian turbulence models are not as naive as they might seem. A. Kolmogorov established the foundation for them in the early 1940s [47]. What's remarkable about neural networks is that they can reproduce fields as structured as real ones. However, $U(x)$ is much more complex. Nevertheless, as S. Mallat states, physicists did not wait for neural networks to go beyond Kolmogorov's theory.

The system that has been extensively studied in Statistical Physics is the Ising model of spin networks [48]. Without going into details, what can be said is that the interaction

46. Image source: https://phys.org/news/2015-10-key-features-transition-liquid-smooth.html.

47. He wrote four very short articles that were as enlightening for the field as those by Fisher and Shannon.

48. The problem that Lars Onsager (1903-76) exactly solved in 1944 is the famous 2D Ising model:

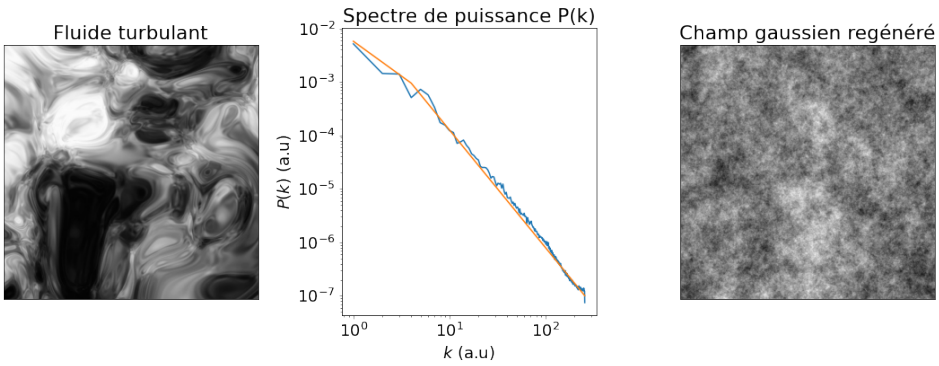

FIGURE 11 – On the left, an image of turbulent fluid in a pipe (Credit: Piotr Siedlecki/public domain); in the center, the power spectrum derived from the image ($\propto k^{-2.2}$); on the right, a Gaussian field generated from this power spectrum.

of all spins on one particular spin can be represented by a potential, the shape of which can be modeled as a so-called "Mexican hat". This gives rise to the $\lambda\phi^4$ theory, which is also used in Particle Physics to explain the generation of the masses of the $W^{\pm}$ and Z^0 bosons through the Higgs mechanism [49]. So, $U(x)$ can be decomposed in such a way that

$$\theta \bullet U(x) = \frac{1}{2} x^T C^{-1} x + V(x) \tag{100}$$

with a Gaussian term and a potential $V(x)$ whose shape is given, for example, in Figure 12 as

$$V(x) = x^4 + (1 + 2b)x^2 \tag{101}$$

this interacting spin model was introduced by Wilhelm Lenz (1888-1957) in 1920, and his student Ernest Ising (1900-98) had solved it in 1D only and could not find a phase transition. Onsager's exact solution allowed understanding its significance and the study of critical exponents and the development of the Renormalization Group Equation (RGE) in Statistical Mechanics. This theory was initiated in Particle Physics Field Theory in 1954 by Murray Gell-Mann (1929-2019) and Francis E. Low (1921-2007) as part of Quantum Electrodynamics (QED), and it was then generalized by Curtis Callan and Kurt Symanzik (1923-83) by establishing what are called the Callan–Symanzik equations. Developments in Statistical Mechanics date back to Kenneth G. Wilson's (1936-2013) Ph.D., obtained under Gell-Mann's supervision in 1961. Wilson bridged the developments in Field Theory with those in Statistical Mechanics, developing the theory of critical exponents in connection with phase transitions, which became a key theme in the field in the 1970s, as seen in the famous "Les Houches Session XXVIII (1975): Methods in Field Theory" with remarkable contributions.

49. This involves additional contributors and becomes the Brout-Englert-Higgs-Hagen-Guralnik-Kibble mechanism.

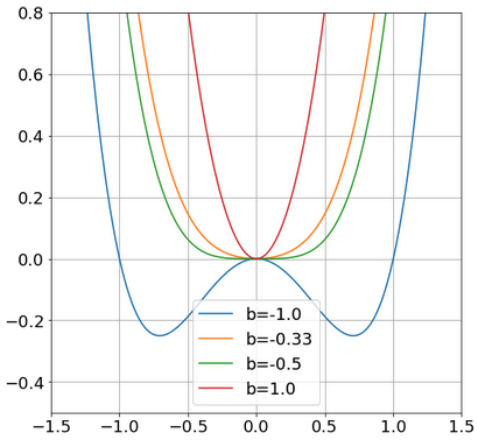

FIGURE 12 – Shape of the potential $V(x)$ experienced by 1 spin in the case of the Ising model in ϕ^4 (or it could be the intensity of a pixel in an image) for different values of the shape parameter b, where the value can yield 2 minima for $b \leq b_c = -1/2$. The minima are located at $\pm\sqrt{-(1+2b)/2}$, and the value of the wells is given by $-(1+2b)^2/4$.

What is the role of this potential $V(x)$? It is there to constrain the values of x to take values "trapped" in the (negative) potential wells and thereby increase the probability $p_\theta(x)$.

This Ising model helped understand ***phenomena of phase transitions*** that manifest themselves in the thermodynamic limit by spontaneous symmetry breaking. To briefly outline the phenomenon, consider a collection of N spins, which should, in principle, have energy invariant under the reversal of all spins. Moreover, the higher the temperature T of the system, the more random the orientation of the spins, and the average residual magnetization is zero. Now, if we subject the system to an external field h, it tends to align the spins in a preferred direction: there is a balance between this tendency toward *order* via h and a tendency toward *disorder* via temperature. For a given N, if we let h tend to 0, we end up in the previous case, with no spontaneous magnetization on average. But when we let the number of spins N tend to infinity (thermodynamic limit) and then let h tend to 0, it turns out that depending on the value of T (which could govern the value of b in the model (101)), especially if it becomes lower than a critical temperature T_c ($b < b_c$), then ***spontaneous magnetization is not zero, indicating the phase transition, a symmetry breaking that shows a collective alignment effect with***

long-range correlations (the correlation length diverging at $T = Tc$***)***. Around $T = T_c$, the system can be considered unstable, transitioning from one phase to another, with each transition involving a symmetry discontinuity.

In neural networks, the parameters θ can be governed by a kind of temperature, and we can also observe collective effects. These kinds of phase transitions with system instability at their edges are ***signs of Hessian instability***, which changes behavior with conditioning breakdowns. So, we have phenomena that are at the heart of optimization and, therefore, touch many domains.

Now, we can ask whether we can go beyond Ising-type models? The answer is yes, and this is where neural networks have changed the game. By using Generative Models or Variational Autoencoders, we can reproduce complex textures such as clouds, piles of rocks, bubbles, etc [50] (Fig. 13). The problem is that these networks have millions of parameters, and we are "quite far from understanding" (sic): why does this work? How do we relate the parameters to the underlying physics interactions?

These are open problems that S. Mallat and his team, for example, are working on, and the fundamental point that emerges is that ***we need to understand the interactions between scales*** (See Course 2020 Sec. 9.). By dividing the image into patches of different sizes, with small scales, we examine highly localized high-frequency interactions, and at larger scales, we examine less localized low-frequency interactions. But what allows the creation of complex structures is ***how different scales interact with each other***.

So, ultimately, with the "linear" θ model of families of probability densities, we can represent very complex and infinitely rich phenomena. The crucial point here is the modeling of $U(x)$. The counterpart in Machine Learning is ***kernel models*** [51], for which the kernel $K(x, x')$ is none other than $\{U(x)U^T(x')\}$, i.e., the covariance matrix. Once the kernel is chosen, linear regression (*Kernel Ridge Regression*) works well, but ***the problem is having the (right) kernel***. And ultimately, there are limitations because if it doesn't fit, what do we do? The field was somewhat stuck for a while until neural networks opened up a new perspective. Indeed, we can see them as a way to learn the right kernel $U(x)$. However, after realizing the effectiveness of neural networks, we end up wondering what's behind these learned $U(x)$?

50. See Course 2019 Sec. 2.7. Also, see the paper S. Zhang and S. Mallat (2021) https://arxiv.org/pdf/1911.10017.pdf.

51. See, for example, Course 2018 Secs. 7.3, 9.5

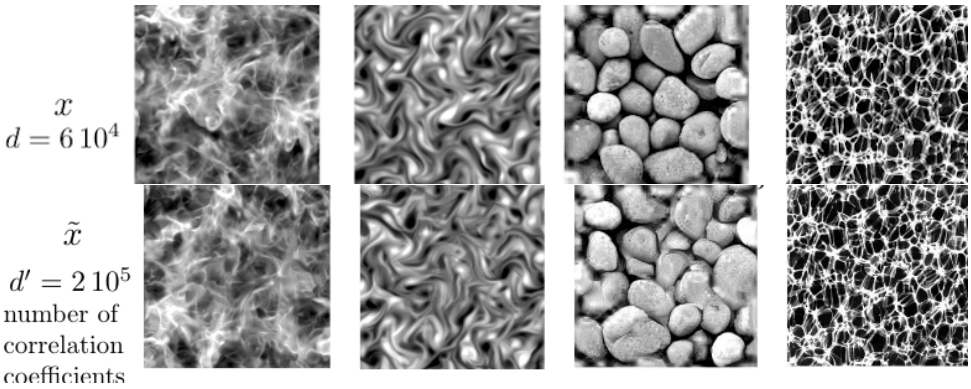

FIGURE 13 – (Top) Example of different textures: a turbulent interstellar cloud, another type of fluid, a pile of stones, and bubbles. (Bottom) Generation of new textures.

4.7 Ensuring Consistency

In the previous sections, we saw how to obtain an optimal estimator, the maximum likelihood estimator (MLE). However, we would like to know the conditions that ensure the consistency of the estimator. That is, what guarantees that when the number of observations tends to infinity, we converge with probability 1 to the correct estimator that maximizes the likelihood on average? This complements the properties discussed in Section 3.5. Let's examine the properties of the maximum likelihood estimator (MLE) defined as

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}}\ \ell(\theta) \tag{102}$$

Theorem 6 *(Change of Variable)*
Let $\eta = g(\theta)$ be a change of variable, then $g(\hat{\theta}) = \hat{\eta}$ is a maximum likelihood estimator (MLE) if $\hat{\theta}$ is an MLE.

Although the proof is simpler if g is invertible, it is not required. The more important result concerns consistency.

Theorem 7 ***(Consistency of MLE)***
Consider the likelihood

$$\ell(\theta, x) = \log p_\theta(x) \tag{103}$$

with $x = (x_1, \ldots, x_n)$ iid. We assume that the observations are described by a certain θ, denoted θ^*, which defines the true probability density. Furthermore, we consider the following regularity assumptions:

R0) If $\theta \neq \theta'$, then $p_\theta(x) \neq p_{\theta'}(x)$;

R1) The supports of the p_θ functions are the same;

R2) θ^* is inside Ω, the parameter space.

R2b) Moreover, we assume that p_θ is differentiable at θ.

For an MLE, we have

$$\frac{\partial \ell(\hat{\theta}_n, x)}{\partial \theta} = 0 \tag{104}$$

This equation potentially has multiple solutions, but there exists a particular solution for which we have convergence in probability, i.e.,

$$\exists\ \hat{\theta}_n \quad s.t. \quad \hat{\theta}_n \xrightarrow[n\to\infty]{prob.} \theta^* \tag{105}$$

This is a different theorem from the one we examined in Section 3.5 (Th. 3, and we will examine the proof in the next session before discussing Fisher Information and Cramér-Rao bounds.

5. Lecture 9 Feb.

5.1 A Brief Prelude

Before delving into the proof of Theorem 7, let's provide a brief preamble. We recall that the data distribution comes from a parameterized family, meaning that $p_{true} \in \{p_\theta\}_\theta = \mathcal{F}_\theta$, and we are trying to determine the correct θ. However, what if $p_{true} \notin \mathcal{F}_\theta$? What happens if we persist in using this family of distributions? We can represent the

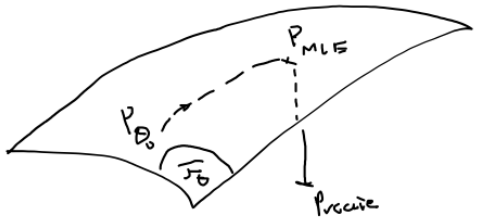

FIGURE 14 – Schematic of the probability family $\mathcal{F}_\theta$ and the potential error if the true probability underlying the observations (p_{true}) does not belong to this family.

family as a manifold, and finding the MLE involves evolving p_θ on this manifold. The search for the MLE is, in fact, associated with the Kullback-Leibler divergence [52]:

$$D_{KL}(p||q) := \int p(x) \log \frac{p(x)}{q(x)} dx \tag{106}$$

It does not have all the properties of a distance, especially $D_{KL}(p||q) \neq D_{KL}(q||p)$. We will see in the second part of the course that $\log(p)$ is the optimal code for encoding elements from the $p(x)$ distribution, and thus, the divergence $D_{KL}(p||q)$ measures an inefficiency in coding that would occur if we took $\log(q)$, which is optimal for coding elements from the $q(x)$ distribution. Thus, if we want to assess the inefficiency of finding p_{true} by using p_θ, it gives

$$\begin{aligned} D_{KL}(p_{true}||p_\theta) &= \int p_{true}(x) \log p_{true}(x) dx - \int p(x) \log p_\theta(x) dx \\ &= \mathbb{E}_{p_{true}}[\log p_{true}] - \mathbb{E}_{p_{true}}[\log p_\theta] \end{aligned} \tag{107}$$

So, ***maximizing the likelihood minimizes the Kullback-Leibler divergence***. But if $p_{true} \notin \mathcal{F}_\theta$, we cannot reach zero; we make an error related to the projection information of p_{true} onto $\mathcal{F}_\theta$ (Fig. 14).

5.2 Consistency of the MLE

Let us examine the MLE consistency theorem.

52. See, for example, the 2019 Course Section 7.2.3.

Proof 7. Recalling from Theorem 3, for $x = (x_i)_{i \leq n}$ *iid*, we have

$$\forall \theta \neq \theta^* \qquad \mathbb{P}\left(\ell(\theta^*, x) > \ell(\theta, x)\right) \xrightarrow[n \to \infty]{prob.} 1 \tag{108}$$

(NDJE here we consider the log-likelihood). And we want to show that there exists a unique sequence of MLEs $\hat{\theta}_n$ that converges in probability to θ^*. If at each step n there are multiple solutions $\hat{\theta}_n$, we can extract a sequence that converges to θ^*. We are reasoning in 1D, but this is generalizable.

Let $a > 0$ be defined such that $[\theta^* - a, \theta^* + a] \in \Omega$, which is possible according to assumption (R2). Let S_n be the set of observations x defined as

$$S_n = \{x \ / \ \ell(\theta^*, x) > \max\left(\ell(\theta^* - a, x), \ell(\theta^* + a, x)\right)\} \tag{109}$$

What we know from Theorem 3 is that in probability

$$\mathbb{P}(S_n) \xrightarrow[n \to \infty]{prob.} 1 \tag{110}$$

In other words, almost all observations will belong to S_n.

Within the interval $[\theta^* - a, \theta^* + a]$, as $\ell(\theta)$ is differentiable and therefore continuous, by the Rolle's theorem, we find a value of θ that nullifies $\partial_\theta \ell(\theta)$, denoted as $\hat{\theta}_n$. Thus, let's define the set $\tilde{S}_n$ of observations as

$$\tilde{S}_n = \left\{x \ / \ \exists \hat{\theta}_n, \text{ s.t. } \partial_\theta \ell(\hat{\theta}_n, x) = 0 \text{ and } \|\theta^* - \hat{\theta}_n\| < a\right\} \tag{111}$$

What we know is that $S_n \subset \tilde{S}_n$ because we cannot a priori determine for $x \in \tilde{S}_n$ if $\ell(\hat{\theta}_n, x) < \ell(\theta^*, x)$. Therefore, $\mathbb{P}(S_n) \leq \mathbb{P}(\tilde{S}_n)$. Thus, by taking the limit in probability, we have

$$\mathbb{P}(\tilde{S}_n) \xrightarrow[n \to \infty]{prob.} 1 \tag{112}$$

So, for any $a > 0$, in probability, we will find a $\hat{\theta}_n$ close to θ^*. Thus, we have our theorem, with the caveat that at each step n, we take one value $\hat{\theta}_n$ if there are multiple solutions to form the sequence that converges to θ^*. ■

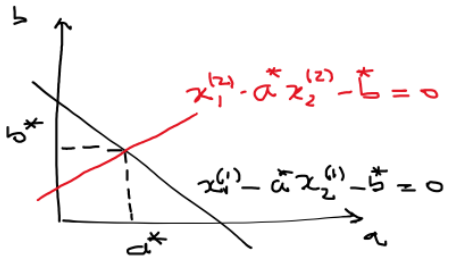

FIGURE 15 – Schematic representation of the constraint of two observations in parameter space.

5.3 Fisher Information

The question that arises after showing that the MLE is a consistent estimator is whether we can do better to estimate θ^*? We then ask about the ***efficiency of the MLE estimator***. To answer this question, we'll take an arbitrary estimator and obtain estimation bounds. This is done here using the concept of Fisher Information.

The underlying idea is to quantify the amount of information that observations will provide about the parameter θ.

(NDJE) Here's how we can think about the idea of information using a simple example. Consider observations $(x_1^i, x_2^i)_{i\leq n}$ and imagine that underneath, $x_2^i = a^* x_1^i + b^*$. We think that by combining 2 observations, we have a system of 2 equations with 2 unknowns, and if our 2 observations are arbitrary, then Cramer gives us the values of (a^*, b^*). But actually, let's ask in the parameter space (a, b) what does it mean to observe (x_1, x_2)? It's a linear constraint as illustrated in Figure 15. This constraint, defining a geometric locus in the (a, b) space, is the information given by the observation. And the data from 2 observations is indeed sufficient to determine the model parameters (a, b). If we consider noisy observables, then the constraint from one observation is not restricted to a line but defines a "tube" as a region of constraints, and the intersection of n tubes from all observations then constrains the determination of parameters (a, b) in a small ellipsoidal region centered on (a^*, b^*).

We'll assume an additional regularity assumption (R3) in addition to those in Theorem 7, namely that p_θ *is twice differentiable in* θ. To facilitate the proof, we also add (R4)

the following assumption:

$$\left(\int p_\theta(x)dx\right)'' = \int p''_\theta(x)dx \tag{113}$$

which means that typically, the second derivative of the probability must be dominated, which is the case in practice. Let's consider ***the score***

$$\boxed{s(\theta, x) = \frac{\partial \log p_\theta(x)}{\partial \theta}} \tag{114}$$

which only equals zero when calculated for $\theta = \theta_{MLE}$. For $\theta = \theta^*$ which gives the true probability distribution of observables ($p_{true} = p_{\theta^*}$), then

$$\boxed{\mathbb{E}_{x\sim p_{\theta^*}}[s(\theta^*, x)] = 0} \tag{115}$$

Indeed,

$$\mathbb{E}_{x\sim p_{\theta^*}}[s(\theta^*, x)] = \int \cancel{p_{\theta^*}(x)} \frac{\partial_\theta p_\theta(x)|_{\theta=\theta^*}}{\cancel{p_{\theta^*}(x)}} dx = \partial_\theta \underbrace{\int p_{\theta^*}(x)dx}_{=1} |_{\theta=\theta^*} = 0 \tag{116}$$

Now, what is the variance of the score? This gives us a definition of the Fisher Information.

Definition 4 *The* ***Fisher Information*** *is the variance of the score*

$$s(\theta, x) = \frac{\partial \log p_\theta(x)}{\partial \theta} \tag{117}$$

calculated at θ^* (i.e., the true θ). That is,

$$\boxed{I(\theta^*) = \mathbb{E}_{x\sim p_{\theta^*}}\left[\left(\frac{\partial \log p_\theta(x)}{\partial \theta}\bigg|_{\theta=\theta^*}\right)^2\right] = Var_{x\sim p_{\theta^*}}[s(\theta^*, x)]} \tag{118}$$

The underlying idea is that if this information is significant, then we are very sensitive to variations in the estimation of the maximum likelihood when we take data samples. ***Being very sensitive means we are better able to determine*** θ^*. To express this intuition, we

state the following theorem:

Theorem 8 *(**Fisher Information and Second Derivative**)*
The Fisher Information is related to the curvature of the log-likelihood calculated at θ^*, i.e.,

$$\boxed{I(\theta^*) = -\mathbb{E}_{x\sim p_{\theta^*}}\left[\left.\frac{\partial^2 \log p_\theta(x)}{\partial\theta^2}\right|_{\theta=\theta^*}\right]} \tag{119}$$

Proof 8. The proof proceeds as follows,

$$\begin{aligned}
\mathbb{E}_{x\sim p_{\theta^*}}\left[\left.\frac{\partial^2 \log p_\theta(x)}{\partial\theta^2}\right|_{\theta=\theta^*}\right] &= \int p_{\theta^*} \times \left.\frac{\partial^2 \log p_\theta(x)}{\partial\theta^2}\right|_{\theta=\theta^*} dx \\
&= \int \left[-\frac{1}{p_{\theta^*}}(p'_\theta(\theta^*))^2 + p''_\theta(\theta^*)\right] dx \\
&= -\underbrace{\int p_{\theta^*}\left(\left.\frac{\partial \log p_\theta(x)}{\partial\theta}\right|_{\theta=\theta^*}\right)^2 dx}_{I(\theta^*)} + \underbrace{\int p''_\theta(\theta^*)dx}_{\left(\int p_\theta(x)dx\right)''_{\theta=\theta^*}\,=\,0}
\end{aligned} \tag{120}$$

■

Now let's consider the additivity of Fisher information. The observations $(x_i)_{i\leq n}$ are *iid*, so we can write

$$\frac{\partial \log p_\theta(x_1,\ldots,x_n)}{\partial\theta} = \sum_{i=1}^{n}\frac{\partial \log p_\theta(x_i)}{\partial\theta} \tag{121}$$

The variables $(\partial \log p_\theta(x_i)/\partial\theta)$ are *iid*, so the variance adds up. Thus [53],

$$\boxed{I(\theta^*; x_1,\ldots,x_n) = n\ I(\theta^*; x_i) = n\ I_1(\theta^*)} \tag{122}$$

53. NDJE: I have introduced the notation I_1 referring to 1 observation.

5.4 Cramér-Rao Bound

We will use the Fisher Information to establish a bound on the precision of estimating the parameter θ^*. In the context of a learning problem, θ represents the set of parameters of a network, and we want to estimate how accurately we can determine them.

Theorem 9 *(**Cramér-Rao**)*
Let $(x_1, \ldots, x_n)$ be n iid observations, all distributed[a] according to $p_\theta(x)$. Consider an estimator Y of θ that is a statistic[b] of the variables $(x_i)_{i \leq n}$:

$$Y = T(x_1, \ldots, x_n) \tag{123}$$

The expectation of this estimator is denoted as follows:

$$\mathbb{E}_{x \sim p_\theta}[Y] := \tau(\theta) \tag{124}$$

The variance of Y is then bounded by:

$$\boxed{Var(Y) \geq \frac{|\tau'(\theta)|^2}{nI_1(\theta)}} \tag{125}$$

where $I_1(\theta)$ represents the Fisher Information for 1 observation.

We qualify ***an estimator as unbiased*** *if $\tau(\theta) = \theta$, and in this case:*

$$Var(Y) \geq \frac{1}{nI_1(\theta)} \tag{126}$$

The Fisher Information gives us the minimal bound on the ability to estimate the parameter θ from observations. Note that the independence of observations gives the factor $1/n$.

a. NDJE: Please note, depending on the context, x is either a specific observation or the set of observations. Additionally, here, for brevity, θ represents the true parameter.
b. Traditionally, the term "statistic" means "function".

Proof 9. The strategy of the proof involves the direct calculation of $\tau'(\theta)$. Using the *iid*

nature of the observations and the results on the score (Eqs. 115, 118), we get:

$$\begin{aligned}
\tau(\theta) &= \int p_\theta(x)T(x)dx = \int T(x_1,\dots,x_n)\prod_{i=1}^{n} p_\theta(x_i)\prod_{k=1}^{n} dx_k \\
\Rightarrow \tau'(\theta) &= \int T(x_1,\dots,x_n)\sum_{i=1}^{n}\underbrace{p'_\theta(x_i)\frac{1}{p_\theta(x_i)}}_{(\log p_\theta(x_i))'}\prod_{j=1}^{n} p_\theta(x_j)\prod_{k=1}^{n} dx_k \\
&= \int T(x_1,\dots,x_n)\frac{\partial \log p_\theta(x_1,\dots,x_n)}{\partial\theta}\prod_{j=1}^{n} p_\theta(x_j)\prod_{k=1}^{n} dx_k \\
&= \int T(x)\frac{\partial \log p_\theta(x)}{\partial\theta}p_\theta(x)dx \\
&= \mathbb{E}_{x\sim p_\theta}[Y\times s(\theta,x)] = Cov[Y\times s(\theta,x)] + \mathbb{E}[Y]\times\underbrace{\mathbb{E}[s(\theta,x)]}_{=0}
\end{aligned} \tag{127}$$

So, the Cauchy-Schwarz inequality tells us that

$$|\tau'(\theta)|^2 = |Cov_{x\sim p_\theta}[Y\times s(\theta,x)]|^2 \leq Var[Y]\times Var[s(\theta,x)] = Var[Y]\times I(\theta) \tag{128}$$

Thus, we have the result of the theorem, knowing that $I(\theta) = nI_1(\theta)$. ■

This Cramér-Rao result is both very important for understanding how to perform parameter inference and quite unique because it is rare to have an explicit bound on the accuracy of an estimator. The primary focus of researchers in the field of inference is to ***find estimators of the underlying model parameters that have the highest Fisher Information***. For example, from astrophysical observations across the electromagnetic spectrum, how can we design observables and statistics to estimate cosmological parameters with the highest possible efficiency (i.e., the highest Fisher Information)? It's worth noting that while the basic standard model of Cosmology (ΛCDM [54]) has 6 parameters [55], inferences typically involve about a hundred parameters, including nuisance parameters related to poorly understood astrophysical effects and instrumental effects, for instance. Therefore, it's an understatement to say that the task is not simple.

The previous formalism easily generalizes to the multidimensional case where $\theta \in \mathbb{R}^d$.

54. Cold Dark Matter + Cosmological Constant
55. Planck 2018 https://arxiv.org/abs/1807.06209.

We have already experienced that the derivative with respect to θ becomes the ***gradient***, thus:

$$\nabla_\theta \ell(\theta_{MLE}, x) = 0 \qquad \mathbb{E}_x[\nabla_\theta \ell(\theta^*, x)] = \mathbb{E}_x[s(\theta^*, x)] = 0 \tag{129}$$

and the Fisher Information becomes:

$$I(\theta^*) = \mathbb{E}_x[\|\nabla_\theta \ell(\theta^*, x)\|^2) = -\mathbb{E}_x[H[\ell](\theta^*, x)] \tag{130}$$

where the ***Hessian*** appears. Thus, ***the Fisher Information governs the convergence rate of the gradient descent algorithm*** (Th. 4) [56].

5.5 Optimality of MLE

Theorem 9 provides us with a bound, but the question is whether this bound can be achieved? To answer this, we will introduce ***the efficiency of an estimator***. Let's consider the case of an *unbiased estimator*, which means that $\mathbb{E}(Y) = \theta^*$ [57]. Throughout, we will denote the estimator as $\hat{\theta}_n$ out of habit from the sections on MLE. The first property we desire is ***consistency*** (Definition 1), meaning convergence in probability of the sequence $(\hat{\theta}_n)_n$ to θ^*. However, we also want to add a property concerning the estimation error: we would like it to reach the Cramér-Rao bound. Consider the variance of $\hat{\theta}_n$ and its relation to the Fisher Information to define the estimator's efficiency:

$$\text{eff.} := \frac{[nI_1(\theta^*)]^{-1}}{Var[\hat{\theta}_n]} \leq 1 \tag{131}$$

The question then arises: can we achieve 100% efficiency? To answer this, we will prove a result about the MLE that tells us ***its distribution converges to a Gaussian distribution with the variance precisely being the Fisher Information***. We need to define what we

56. NDJE: Note, however, that the estimation of parameters and their confidence intervals in an *intermediate* dimension (e.g., astro-cosmo) uses a different method. From ref. 55: *"[The] nuisance parameters are sampled, along with cosmological parameters, during Markov chain Monte Carlo (MCMC) exploration of the likelihood."* This requires the incorporation of *priors*.

57. Note that here, we reuse the notation θ^* as the parameter of the true distribution.

mean by ***convergence in distribution***[58]:

Definition 5 *(**Convergence in Distribution**)*
Consider a collection of random variables $(x_1, \ldots, x_n)$, the question is whether the distribution of these random variables converges to the distribution of a variable x? Denote the cumulative distribution function (or simply distribution function) of the probability $p_n(x)$ as

$$F_n(a) = \int_{-\infty}^{a} p_n(x)dx \tag{132}$$

(similarly, F is the distribution function of $p(x)$). Thus, convergence in distribution, denoted as

$$p(x_n) \xrightarrow[n\to\infty]{dist.} p(x) \tag{133}$$

means that

$$\forall a \text{ such that } F(a) \text{ is continuous}, \quad \lim_{n\to\infty} F_n(a) = F(a) \tag{134}$$

This definition is related to the Central Limit Theorem established in 1809 by Pierre-Simon de Laplace (1749-1827), generalizing the earlier work of Abraham de Moivre (1667-1754) on the Bernoulli distribution:

Theorem 10 *(**Central Limit Theorem**)*
Let $(x_1, \ldots, x_n)$ be iid with $\mathbb{E}(x_i) = \mu$ and $0 < Var(x_i) = \sigma^2 < \infty$. Define

$$\bar{X}_n := \frac{1}{n}\sum_{i=1}^{n} x_i \tag{135}$$

58. NDJE After discussions with S. Mallat, we realized a difference between the notion of convergence in distribution in the Anglo-Saxon sense as presented here and the French notion of "convergence en loi", which states that the sequence of random variables $(X_n)_{n>0}$ in $\mathbb{R}^d$ converges in law: $X_n \xrightarrow[n\to\infty]{loi} X$ if for every bounded continuous function f from $\mathbb{R}^d$ to $\mathbb{R}$: $\mathbb{E}[f(X_n)] \xrightarrow[n\to\infty]{} \mathbb{E}[f(X)]$. The point is that for $d = 1$, convergence in law is equivalent to convergence of distribution functions, thus to convergence in distribution, and furthermore, convergence in law is equivalent to convergence of generating functions.

We know that (see the Law of Large Numbers, Theorem 1 and its proof)

$$\bar{X}_n \xrightarrow[n\to\infty]{prob.} \mu,\ Var[\bar{X}_n] = \frac{\sigma^2}{n} \tag{136}$$

So, if we consider the random variable

$$Z_n := n^{1/2}\left(\frac{\bar{X}_n - \mu}{\sigma}\right) \tag{137}$$

its mean is 0, its variance is 1, and furthermore, we have

$$p(Z_n) \xrightarrow[n\to\infty]{dist.} \mathcal{N}(0,1) \tag{138}$$

S. Mallat indicates that the proof is done using characteristic functions and leaves it to the readers.

Now, consider the following theorem about the normal distribution convergence of the MLE:

Theorem 11 *(**Normal Distribution Convergence of MLE**)*
We revisit the regularity assumptions of the MLE consistency theorem 7, along with those on the second derivative (Eq. 113), to which we add a new assumption (R5)

$$|(\log p_\theta(x))'''_\theta| < M(x) \tag{139}$$

such that $\mathbb{E}[M(x)] < \infty$. This allows us to apply the dominated convergence theorem to the error terms. Given these assumptions, for any sequence of MLEs $\hat{\theta}_n$ that converges in probability to θ^* (we know at least one such sequence exists), then

$$\boxed{p(\sqrt{n}(\hat{\theta}_n - \theta^*)) \xrightarrow[n\to\infty]{dist.} \mathcal{N}(0, I^{-1}(\theta^*) = [nI_1(\theta^*)]^{-1})} \tag{140}$$

*(note: in dimension n, we take the matrix inverse of the Hessian). This theorem tells us that **the MLE is an asymptotically optimal estimator** because it reaches the Cramér-Rao bound.*

Proof 11. We will outline the steps of the proof. If we take an MLE $\hat{\theta}_n$, we know that the derivative of the log-likelihood is zero at this point. So, by performing a Taylor expansion around $\hat{\theta}_n$ and evaluating it at θ^*, we get

$$\ell'(\theta^*) = \cancel{\ell'(\hat{\theta}_n)} + (\theta^* - \hat{\theta}_n)\ell''(\hat{\theta}_n) + \dots \tag{141}$$

In $\ell'(\theta^*)$, we recognize the score of n *iid* random variables whose mean is zero (Eq. 115) and whose variance equals the Fisher Information (Eq. 118) ($I(\theta^*) = nI_1(\theta^*)$). By the central limit theorem, we obtain

$$n^{1/2}\left(\frac{\ell'(\theta^*)}{I^{1/2}(\theta^*)}\right) \xrightarrow[n\to\infty]{dist.} \mathcal{N}(0,1) \Rightarrow n^{1/2}\ell'(\theta^*) \xrightarrow[n\to\infty]{dist.} \mathcal{N}(0, I(\theta^*)) \tag{142}$$

Now, we need to consider

$$n^{1/2}(\hat{\theta}_n - \theta^*) = -\frac{n^{1/2}\ell'(\theta^*)}{\ell''(\hat{\theta}_n)} \tag{143}$$

and deal with the denominator, which is the trickiest part. Be aware that the Fisher Information can be expressed in terms of second derivatives (Theorem 8) ($-\ell''(\theta^*) = I(\theta^*)$), but these derivatives are calculated at the point θ^* and not $\hat{\theta}_n$. If we perform a Taylor expansion around θ^*, we obtain

$$\ell''(\hat{\theta}_n) = \ell''(\theta^*) + (\hat{\theta}_n - \theta^*)\ell'''(\theta^*) + \dots \tag{144}$$

The condition that the term involving $\ell'''(\theta^*)$ tends to zero relies on assumption R5. For now, let's imagine we neglect this term [59]

$$n^{1/2}(\hat{\theta}_n - \theta^*) \underset{n\to\infty}{\approx} \frac{n^{1/2}\ell'(\theta^*)}{I(\theta^*)} \xrightarrow[n\to\infty]{dist.} \mathcal{N}(0, I^{-1}(\theta^*) = [nI_1(\theta^*)]^{-1}) \tag{145}$$

This gives us the theorem. So, the technical challenge lies in controlling the terms involving $\ell'''(\theta^*)$. The theorem generalizes to dimension d thanks to the extension of the dominated convergence theorem. ■

Once we have this result, we realize that we have access to ***confidence intervals*** (2D

59. if $u \sim \mathcal{N}(\mu, \sigma^2)$, then $u/a \sim \mathcal{N}(\mu, \sigma^2/a^2)$.

contours). This is crucial in physics, where the primary task is not only to produce results on certain parameters but also to provide the probability of finding a parameter within certain confidence bounds. Furthermore, the scientific community becomes concerned when there are *tensions* at the n-sigma level between different experiments giving results on the same parameters. However, this issue of estimating confidence intervals tends to emerge in machine learning as well. For example, in logistic regression, we know that there's uniqueness of θ^* (due to convexity), and thus, theoretically, as n tends to infinity, $\hat{\theta}_n$ is consistent (converges in probability), and we have a confidence interval thanks to convergence in distribution. Unfortunately, this technique doesn't work for neural networks for many reasons. Here are two:

— There's no uniqueness of θ^* at all because there's *no convexity*.

— The point is that the formalism implicitly assumes that d is fixed, and $n \gg d$ (*classical statistics regime*). However, the number (huge) of parameters far exceeds the number of samples, and this *over-parametrization* $d \gtrsim n$ is highly effective (see Section 2.1). For example, in the case of neural networks, the number of parameters often greatly exceeds the number of available samples. Therefore, we cannot expect to consistently estimate the entire PCA basis. In such cases, what remains consistent in this partial estimation? In the best cases, we have access to the largest eigenvalues (and associated eigenvectors). Thus, we need to step out of the classical statistical framework, opening up new avenues for exploration in machine learning.

In the second part of the course, we will take a different approach, following in the footsteps of C. Shannon. This approach involves providing ***parameter-independent information***. C. Shannon's problem was not to discover the parameters of a physical phenomenon but to transmit data as efficiently as possible. Therefore, the question was how to minimize the number of bits needed to transmit information. Even though we seem far from Fisher Information at first glance, we will realize that there is convergence between these concepts, highlighting a well-known concept in statistical physics: ***entropy***.

6. Lecture 16 Feb.

6.1 Introduction

In this section, we will explore Claude Shannon's perspective on the concept of information (1948). As previously mentioned (Sec. 2.5), the goal is to uncover ***intrinsic information*** within observations, without reference to any underlying model. This intrinsic information is linked to the ***minimum number of bits*** required to encode or transmit it through channels, a process that can introduce errors. What is remarkable is that Shannon's work has opened up connections to Statistical Physics through the notion of ***Entropy***, which quantifies the number of configurations of a system. In the 1960s, Andrey Kolmogorov revisited Shannon's question from the perspective of the minimum amount of information needed to reproduce observations. However, Kolmogorov used a ***Turing Machine*** to define complexity (or Kolmogorov's information) as the size of the minimum program required to replicate a sequence. There is a correspondence between these two notions when considering stationary ergodic processes [60], as there is equivalence between Shannon's entropy and the quantity of Kolmogorov's information (up to a constant).

Why focus on Shannon's information rather than Kolmogorov's? This can be justified by the fact that Kolmogorov's information, except in a few cases, is very challenging to compute, while Shannon's entropy is not only intuitive but also estimable from observations. We will delve into this notion of entropy, demonstrate *its additivity*, which allows it to be linked to the concept of information. We will also explore the ***concentration phenomena***, which form the basis of Shannon's theory. According to him, the reason entropy effectively quantifies the minimum size of a code that would reproduce the observations is that these observations, geometrically, concentrate within ***typical sets*** (Eq. 23, Fig. 5) whose ***size is specified by entropy***. Therefore, by counting the number of elements in these

60. A brief note on vocabulary: 1) a process is *ergodic* if its statistical properties can be studied from a *single realization* that is sufficiently long (e.g., ergodicity regarding the mean where temporal averages converge to ensemble averages); 2) a process is *stationary* if its statistical properties characterized by mathematical expectations are *independent of time*. These two notions are not identical. If $X(t) = x_0 + n$ with x_0 as a constant and n as a *random variable*, then $\mathbb{E}[x(t)] = x_0 + \mathbb{E}[n]$ is independent of time, making it a stationary process. However, it is not ergodic. For example, if $x_i(t)$ is a realization that fixes the value of n to n_i, then $\frac{1}{2T}\int_{-T}^{T} x_i(t)dt \to x_0 + n_i$. This result depends on the realization, indicating that the process is not ergodic.

sets, we can determine the number of bits required to encode them. Ultimately, using entropy, we will be able to define models, leading us to the concept of Fisher information, through the ***maximum entropy models***. This principle was developed in 1957 by Edwin Thompson Jaynes (1922-98) and establishes a connection with maximum likelihood. One of the applications of this coding theory is ***signal compression*** and the concepts of ***distortion*** versus compression. Through this, we implicitly raise the question of where the signal structures and their ***representation*** lie in order to grasp the geometry of typical sets.

6.2 Shannon's Entropy

We consider a scenario where we have a finite *alphabet*, denoted as $\mathcal{A} = \{a_k\}_{1\leq k\leq K}$, where the symbols a_k represent the values taken by a random variable X with probabilities $p(a_k)$ assigned to each of them. Can we determine the uncertainty about the value of x, a realization of X? Let's imagine that the probabilities $p(a_k)$ are all identical (e.g., $1/K$), then we have a sort of *maximum uncertainty* about the value of x, meaning that no particular symbol is favored. Conversely, if $p(a_{k_0} = 1)$, then we know with certainty what the outcome of x will be. To some extent, variance would provide us with information about the error on x, but it is primarily related to the concept of ***coding***. Indeed, envision a distribution concentrated on only a few symbols; it's tempting to want to express these favored symbols with ***few bits*** since they are often used, while allowing the use of a ***maximum number of bits for rarely used symbols***. This is particularly effective when encoding natural language sentences, where the "symbols" are words from a vocabulary corpus.

Definition 6 *(**Shannon's Entropy**)*
Shannon's entropy is given by

$$\boxed{\mathbb{H}(X) := -\mathbb{E}_{x\sim p}[\log p(X)] \geq 0} \tag{146}$$

which, for a random variable X taking values in an alphabet $\mathcal{A} = \{a_k\}_{1\leq k\leq K}$, is

expressed as [a]

$$\mathbb{H}(X) = -\sum_{k=1}^{K} p(X = a_k) \log p(X = a_k) \tag{147}$$

a. The "log" is used here without specifying the base.

NDJE: By the way, I denote entropy with the letter $\mathbb{H}$ *for two reasons: 1) Claude Shannon used the letter* H*, and 2) it needed to be distinguished from the "H" of the Hessian.*

If we revisit the two extreme examples mentioned earlier, for a uniform distribution, $\mathbb{H}[\mathcal{U}] = \log K = \log |\mathcal{A}|$, and for a distribution concentrated on only one symbol, $\mathbb{H}[\delta] = 0$. We will see that $\log |\mathcal{A}|$ is the upper bound for $\mathbb{H}(X)$ (Eq. 4). Thus, we understand that Shannon's entropy effectively measures an error regarding the value of a realization x.

Considering two random variables, we define joint entropy and conditional entropy as follows:

Definition 7 *The joint entropy of two random variables* X *and* Y *with values in* $\mathcal{A}$ *is defined as*

$$\mathbb{H}(X, Y) := -\mathbb{E}_{(x,y)\sim p}[\log p(X, Y)] = -\sum_{k,k'} p(X = a_k, Y = a_{k'}) \log p(X = a_k, Y = a_{k'}) \tag{148}$$

Definition 8 *The conditional entropy of two random variables* X *and* Y *with values in* $\mathcal{A}$ *is defined as*

$$\begin{aligned} \mathbb{H}(Y|X) &:= -\sum_{k} p(X = a_k) \mathbb{H}(Y|X = a_k) \\ &= -\sum_{k,k'} p(X = a_k, Y = a_{k'}) \log p(Y = a_{k'}|X = a_k) \\ &= -\mathbb{E}_{(x,y)\sim p}[\log(Y|X)] \end{aligned} \tag{149}$$

Later on, we can simplify the notation, either by using $p(a_k) = p(X = a_k)$ and $p(a_k, a_{k'}) = p(X = a_k, Y = a_{k'})$, or by referring to $p(x, y)$, $p(x)$, $p(y|x)$, which are less prone to confusion. The two aforementioned entropies are related as follows:

Property 2

$$\boxed{\mathbb{H}(X,Y) = \mathbb{H}(X) + \mathbb{H}(Y|X)} \tag{150}$$

Indeed,

$$\begin{aligned}
\mathbb{H}(X,Y) - \mathbb{H}(X) &= -\sum_{x,y} p(x,y)\log p(x,y) + \sum_{x} p(x)\log p(x) \\
&= -\sum_{x,y} p(x,y)\log p(x,y) + \sum_{x}\left(\sum_{y} p(x,y)\right)\log p(x) \\
&= -\sum_{x,y} p(x,y)\log\frac{p(x,y)}{p(x)} = -\sum_{x,y} p(x,y)\log p(y|x) \\
&= \mathbb{H}(Y|X)
\end{aligned} \tag{151}$$

In passing, we note that if entropy measures uncertainty, the above relationship is quite intuitive.

Let's explore the additivity of the entropy defined in this manner. To do this, we will use the concept of mutual information through Kullback-Leibler divergence (See note 52), or ***relative entropy***. It is a very useful tool in probability.

6.3 Relative Entropy and Mutual Information

Let's recall the definition of Kullback-Leibler divergence:

Definition 9 *(**Kullback-Leibler**)*
If the support[a] *of* q *includes the support of* p*, then*

$$\boxed{D(p\|q) := \sum_{x} p(x)\log\frac{p(x)}{q(x)} < \infty} \tag{152}$$

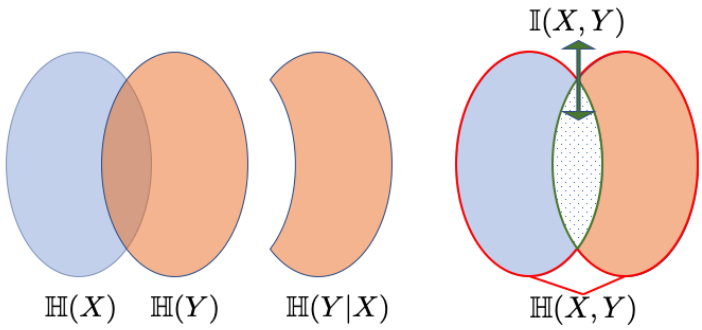

FIGURE 16 – Schematic representation of entropy $\mathbb{H}(X)$ and $\mathbb{H}(Y)$, of $\mathbb{H}(Y|X)$, as well as mutual information (Eq. 153) and joint entropy (Eq. 150).

The sum can be transformed into an integral if necessary.

a. By convention, we set $0 \log 0 = 0$ and $0 \log(0/0) = 0$.

Now, let's define mutual information, which will provide us with a ***measure of independence*** based on Kullback-Leibler divergence:

Definition 10 ***(Mutual Information)***
Consider two random variables, X and Y, with a joint probability distribution $p(x, y)$ and marginal distributions $p(x)$ and $p(y)$. Mutual information is defined as follows:

$$\boxed{\mathbb{I}(X,Y) := D(p(x,y) \| p(x)p(y))} \tag{153}$$

The connection with entropy is expressed through the following property:

Property 3

$$\begin{aligned} \mathbb{I}(X,Y) &= \mathbb{H}(X) + \mathbb{H}(Y) - \mathbb{H}(X,Y) \\ &= \mathbb{H}(X) - \mathbb{H}(X|Y) \\ &= \mathbb{H}(Y) - \mathbb{H}(Y|X) \end{aligned} \tag{154}$$

Indeed, it can be easily shown that:

$$\begin{aligned}\mathbb{I}(X,Y)+\mathbb{H}(X,Y) &= -\sum_{x,y} p(x,y)\log[p(x)p(y)] \\ &= -\sum_{x}\underbrace{\sum_{y} p(x,y)}_{p(x)}\log p(x) - \sum_{y}\underbrace{\sum_{x} p(x,y)}_{p(y)}\log p(y) \\ &= \mathbb{H}(X)+\mathbb{H}(Y) \end{aligned} \tag{155}$$

Then, we use Eq. 150. In some way, ***mutual information is measured by the impact of knowledge about*** x ***on the reduction of uncertainty about the value of*** y (and vice versa when exchanging the roles of x and y). If the two variables are independent, the reduction in uncertainty is zero. These different concepts can be schematically represented as shown in Figure 16.

To prove certain results, we need Jensen's inequality in the context of probabilities [61]:

Theorem 12 *(**Jensen's Inequality**)*
*Let f be a **convex function** in one dimension (with a second derivative that is positive or non-negative). Then, for any random variable X:*

$$\mathbb{E}[f(X)] \geq f(\mathbb{E}[X]) \tag{156}$$

*If f is **strictly convex** (with a second derivative strictly positive), we have equality if and only if the only value taken by X is $\mathbb{E}[X]$.*

We will use this theorem to show that Kullback-Leibler divergence is positive.

Theorem 13 *(**Positivity of Kullback-Leibler Divergence**)*

$$\boxed{D(p\|q)\geq 0} \qquad \boxed{D(p\|q)=0 \quad \textit{if and only if} \quad p(x)=q(x) \quad \forall x} \tag{157}$$

61. See also Eq. 48.

Proof 13. The function log is strictly concave, so $-\log$ is strictly convex:

$$\begin{aligned} D(p\|q) &= \sum_x p(x) \log \frac{p(x)}{q(x)} = -\sum_x p(x) \log \frac{q(x)}{p(x)} \\ &= \mathbb{E}_p \left[-\log \frac{q(x)}{p(x)} \right] \geq -\log \mathbb{E}_p \left[\frac{q(x)}{p(x)} \right] = -\log(1) = 0 \end{aligned} \tag{158}$$

The strict concavity of the log function implies that the inequality above becomes an equality *if and only if* $p(x)/q(x)$ takes a unique value. Let c be this value, as $\sum_x p(x) = \sum_x q(x) = 1$, then $c = 1$, and we have the second result of the theorem.

■

Therefore, Kullback-Leibler divergence indeed provides a kind of "distance" between the two probabilities p and q in the sense that it indicates similarity when close to 0. However, it is not a distance because $D(p\|q) \neq D(q\|p)$.

One consequence of the positivity of $D(p\|q)$ concerns mutual information since it is directly related to it by definition. Thus,

$$\boxed{\mathbb{I}(X,Y) \geq 0} \tag{159}$$

and we have equality *if and only if* in the case of independence:

$$\boxed{\mathbb{I}(X,Y) = 0 \quad \text{if and only if} \quad X, Y \text{ are independent}} \tag{160}$$

Another consequence concerns the entropy of a random variable taking its values in an alphabet (see the two examples in Sec. 6.2):

Property 4 *For a random variable X with values in a finite set $\mathcal{A}$, we have:*

$$\boxed{\mathbb{H}(X) \leq \log |\mathcal{A}|} \tag{161}$$

Indeed, let $\mathcal{A} = \{a_k\}_{1 \leq k \leq K}$ and consider the uniform distribution on this alphabet, $q(a_k) = 1/K$ for all k. Then, for any random variable X with probability $p(x)$:

$$0 \leq D(p\|q) = \sum_k p(a_k) \log p(a_k) + \sum_k p(a_k) \log K = -\mathbb{H}(X) + \log K \tag{162}$$

Another intuitive property, considering entropy as a measure of uncertainty, is that if we add information by conditioning on another random variable, then:

Property 5

$$\boxed{\mathbb{H}(X|Y) \leq \mathbb{H}(X)} \tag{163}$$

This property is evident when we remember the relationships between mutual information and entropy (Eqs. 3) and the fact that mutual information is always positive. One way to visualize this relationship is given in Figure 16, where $\mathbb{H}(Y|X)$ is the smaller orange crescent compared to $\mathbb{H}(Y)$.

So, in conclusion, Shannon's entropy aligns with our intuition about what error in a random process should be. The crucial point we do not prove here is that conversely, if we have the relationships mentioned above and we ask what form $\mathbb{H}(X)$ should take, then we arrive at Shannon's entropy.

(NDJE) In his 1948 paper, Shannon gives 3 conditions for the function $\mathbb{H}(p_1, p_2, \ldots, p_K)$ *where* $(p_k)_k$ *are the known probabilities of* K *events:*

1) $\mathbb{H}$ *should be a continuous function of all its variables* p_k*;*
2) *In the case of equiprobability* $p_k = 1/K$ $(\forall k)$*, then* $\mathbb{H}$ *should be a monotone function of* K*, reflecting that with more choices, there is more uncertainty;*
3) *In the case where the original problem is subdivided into subproblems, then the original function* $\mathbb{H}$ *should be a weighted sum of the functions* $\mathbb{H}$ *of the subproblems.*

Under these conditions, he demonstrates that

$$\mathbb{H} = -C \sum_k p_k \log p_k \tag{164}$$

with C being a positive constant that he takes equal to 1, which is a sort of unit choice (he also doesn't specify which type of logarithm is used).

So, Shannon's entropy aligns with our expectations regarding uncertainty in processes, but where it becomes very powerful is when we link it to ***phenomena of concentration***.

6.4 Typical Sets

When we aim to model observations, we try to visualize ***the geometry of the space*** in which they evolve. As we have already discussed (Sec. 2.2), in high dimensions, where the probabilistic perspective is often more powerful than the deterministic one, this is primarily due to concentration phenomena (Fig. 3). In machine learning, we often talk about "manifolds", but let's not be mistaken: these sets are not necessarily differentiable. Therefore, we would like to characterize the geometry of these sets and calculate their size, which we believe should be much smaller than the size of the set in which they exist. In this context, we will see that ***entropy characterizes the volume of typical sets*** (or the number of elements in the discrete case) (Eq. 23, Fig. 5).

Let's consider a case where we have *iid* realizations $(x_1, x_2, \ldots, x_n)$ of a process X. The probability of these realizations is, of course,

$$p(x_1, x_2, \ldots, x_n) = \prod_{i=1}^{n} p(x_i) \tag{165}$$

and as in the case of the likelihood concept, we want to take the logarithm of this expression. If we weight it by $1/n$, we have

$$\frac{1}{n} \log p(x_1, x_2, \ldots, x_n) = \frac{1}{n} \sum_{i=1}^{n} \log p(x_i) \tag{166}$$

In other words, we have an average of *iid* random variables. So, according to the Law of

Large Numbers (Th 1) for probability convergence, we get

$$\frac{1}{n}\log p(x_1, x_2, \ldots, x_n) \xrightarrow[n\to\infty]{prob.} \mathbb{E}_{x\sim p(x)}[\log p(x)] = -\mathbb{H}(x) \tag{167}$$

Therefore[62],

$$\forall \varepsilon > 0 \quad \lim_{n\to\infty} \mathbb{P}\left(\left|-\frac{1}{n}\log p(x_1, x_2, \ldots, x_n) - \mathbb{H}(x)\right| \leq \varepsilon\right) = 1 \tag{168}$$

Thus, we can focus on the observations that will actually (for a given ε) have an expectation that is within ε of the entropy. By denoting $\{x_i\}_{1\leq i\leq n} = \{x\}$

$$\boxed{T_n^\varepsilon = \left\{\{x\} \in \mathcal{A}^n, \left|-\frac{1}{n}\log p(\{x\}) - \mathbb{H}(x)\right| \leq \varepsilon\right\}} \tag{169}$$

For any $\varepsilon > 0$ and sufficiently large n, we have

$$\mathbb{P}[\{x\} \in T_n^\varepsilon] \geq 1 - \varepsilon \tag{170}$$

In other words, ***almost all realizations will belong to*** T_n^ε***, hence the name "typical set"***. The next question is: what is their size?

If we rewrite the constraint defining T_n^ε and use ***the base-***2 ***logarithm*** (which sets the constant c mentioned above), then for $\forall \varepsilon > 0$

$$\left|-\frac{1}{n}\log p(\{x\}) - \mathbb{H}(x)\right| \leq \varepsilon \Rightarrow \boxed{2^{-n(\mathbb{H}(x)+\varepsilon)} \leq p(\{x\}) \leq 2^{-n(\mathbb{H}(x)-\varepsilon)}} \tag{171}$$

Note that due to the additivity of entropy, $n\mathbb{H}(x) = \mathbb{H}(\{x\})$, which is ***a constant independent of any particular realization*** $\{x\}$. What's remarkable here is that, up to ε, ***the probability is almost constant*** on these typical sets. Thus, ***observations concentrate within these sets while being distributed anywhere inside them at the same time***, which is a consequence of independence. This is referred to as ***asymptotic equipartition***[63]. We

62. NDJE: For the definition of typical sets Eq. 23, the notation was slightly different: $(x_1, \ldots, x_n) = \{x\}$, and $\mathbb{H}[p]$ is actually $\mathbb{H}(x)$. Here, I have used notation related to the definition of entropy in this session.

63. NDJE: This should be considered in the context of the ***fundamental principle of equiprobability*** of microstates in a thermodynamic system of energy within the interval $[E, E + dE]$.

can state the following two properties of typical sets:

Property 6 *In addition to*

$$\boxed{\mathbb{P}[\{x\} \in T_n^\varepsilon] \geq 1 - \varepsilon} \tag{172}$$

the cardinality of the set T_n^ε *satisfies*

$$\boxed{(1-\varepsilon)2^{n(\mathbb{H}(x)-\varepsilon)} \leq |T_n^\varepsilon| \leq 2^{n(\mathbb{H}(x)+\varepsilon)}} \tag{173}$$

which implies that the number of elements $\{x\}$ *in the set is approximately* $2^{n\mathbb{H}(x)}$ *(also* $2^{\mathbb{H}(\{x\})}$*).*

The second property can be demonstrated as follows. For $\{x\} \in \mathcal{A}^n$, using the relation 171, we have

$$1 = \sum_{\{x\}\in\mathcal{A}^n} p(\{x\}) \geq \sum_{\{x\}\in T_n^\varepsilon} p(\{x\}) \geq \sum_{\{x\}\in T_n^\varepsilon} 2^{-n(\mathbb{H}(x)+\varepsilon)} = |T_n^\varepsilon| \times 2^{-n(\mathbb{H}(x)+\varepsilon)} \tag{174}$$

which yields one of the two inequalities. Considering the first property, we have

$$\sum_{\{x\}\in T_n^\varepsilon} p(\{x\}) \geq 1 - \varepsilon \tag{175}$$

thus, using Eq. 171, we get

$$1 - \varepsilon \leq \sum_{\{x\}\in T_n^\varepsilon} 2^{-n(\mathbb{H}(x)-\varepsilon)} = |T_n^\varepsilon| \times 2^{-n(\mathbb{H}(x)-\varepsilon)} \tag{176}$$

which gives the other inequality. These properties will be useful for coding.

6.5 Typical Code

Why can we perform coding? Let's imagine I have[64] $X = (x_1, x_2, \ldots, x_n)$ with n coordinates, knowing that these coordinates are *iid* random variables governed by a

64. NDJE: Here, X represents what was previously denoted as $\{x\}$.

probability distribution $p(x)$ where the values of x are either taken from a finite alphabet of size K or from an interval in $\mathbb{R}$, in any case for which I can calculate $\mathbb{H}(x)$. We associate a binary word $w(X)$ with X, which has a certain length $\ell(X)$ (or $\ell(w(X))$), and we look at the average length (or number of bits) per symbol:

$$R = \frac{1}{n} \sum_X \ell(X) p(X) \tag{177}$$

The word $w(X)$ can vary from one observation X to another, so we want to know the minimum size on average.

Now, knowing the existence of the typical set associated with $p(x)$, we can say that either $X \notin T_n^\varepsilon$, but this will be the case with very low probability ($\leq \varepsilon$), or $X \in T_n^\varepsilon$ with a high probability, which is almost uniform over the set. The idea is, therefore, to use shorter codes when the probability is high and longer codes when the probability is low (recall the coding of natural language, Sec. 6.2). Now, for $X \in T_n^\varepsilon$, since the elements are equiprobable, it is natural to use a code of the same length for these elements. The code length should be sufficient to distinguish all the elements within this typical set, so it is approximately $\log_2 |T_n^\varepsilon|$. Thus, we define ***the typical code or ε-typical code*** as follows:

Definition 11 *(ε-**Typical Code**)*

- *If $X \in T_n^\varepsilon$, $\ell(X) = \lceil n(\mathbb{H}(x) + \varepsilon) \rceil = \lfloor n(\mathbb{H}(x) + \varepsilon) \rfloor + 1$*
- *If $X \notin T_n^\varepsilon$, recalling the size of this set, $\ell(X) = \lfloor n \log_2 K \rfloor + 1$*
- *We add 1 bit to each $\ell(X)$ to indicate whether X is in the typical set or not.*

We can then bound R as follows:

Theorem 14 *(**Shannon Bound**)*
$\exists C$ such that for all $\varepsilon > 0$, for sufficiently large n and an ε-typical code, the average number of bits per symbol satisfies

$$\boxed{R \leq \mathbb{H}(x) + C\varepsilon} \tag{178}$$

Proof 14. Let's express R as follows:

$$\begin{aligned} R &= \frac{1}{n}\sum_{X\in T_n^\varepsilon} \ell(X)p(X) + \frac{1}{n}\sum_{X\notin T_n^\varepsilon} \ell(X)p(X) \\ &= \frac{1}{n}(\lfloor n(\mathbb{H}(x)+\varepsilon)\rfloor + 2)\left(\sum_{X\in T_n^\varepsilon} p(X)\right) + \frac{1}{n}(\lfloor n\log_2 K\rfloor + 2)\left(\sum_{X\notin T_n^\varepsilon} p(X)\right) \end{aligned} \tag{179}$$

Now, $\sum_{X\in T_n^\varepsilon} p(X) \leq 1$ and $\sum_{X\notin T_n^\varepsilon} p(X) \leq \varepsilon$, and since $\lfloor x\rfloor \leq x$, we have

$$R \leq \frac{1}{n}(n(\mathbb{H}(x)+\varepsilon)+2) + \frac{1}{n}(n\log_2 K + 2)\varepsilon \leq \mathbb{H}(x) + \varepsilon\left(\frac{3}{n} + \log_2 K\right) + \frac{2}{n} \tag{180}$$

This allows the upper bound and identifies C by justifying that n is sufficiently large. ■

So, with the ε-typical code, the number of bits (per symbol) is roughly bounded by the entropy of the symbol probability. Can we do better? This would be the case if we could show that observations concentrate "even more" in subsets of typical sets... However, a priori, we have shown that the probability within typical sets is nearly uniform, so it seems quite challenging. This is what we will explore next.

6.6 Typical Sets are "Optimal"

In a sense, we will show that typical sets are the right objects, not only do observations concentrate in them, but also we cannot hope for anything better. Let B_δ^n be the *smallest set* such that

$$\mathbb{P}(X \in B_\delta^n) \geq 1-\delta \tag{181}$$

Can it be smaller in size than the typical set? The answer is no, and this is due to the following theorem:

Theorem 15 ***(Optimality of Typical Sets)***
With $X = (x_1, \ldots, x_n)$ where the x_i are iid random variables with distribution $p(x)$
$\forall \delta, \delta' > 0$,

$$\mathbb{P}(X \in B_\delta^n) \geq 1-\delta \Rightarrow \frac{1}{n}\log_2 |B_\delta^n| \geq \mathbb{H}(x) - \delta' \tag{182}$$

Proof 15. The proof focuses on the intersection between B_δ^n and T_n^ε.

$$\mathbb{P}(T_n^\varepsilon \cap B_\delta^n) = \mathbb{P}(T_n^\varepsilon) + \mathbb{P}(B_\delta^n) - \mathbb{P}(T_n^\varepsilon \cup B_\delta^n) \geq (1-\varepsilon) + (1-\delta) - 1 = 1 - \varepsilon - \delta \quad (183)$$

Now, every element in the intersection is an element of T_n^ε, and we can use the inequalities 171. Thus,

$$\mathbb{P}(T_n^\varepsilon \cap B_\delta^n) = \sum_{X \in T_n^\varepsilon \cap B_\delta^n} p(X) \leq |T_n^\varepsilon \cap B_\delta^n| \times 2^{-n(\mathbb{H}(x)-\varepsilon)} \leq |B_\delta^n| \times 2^{-n(\mathbb{H}(x)-\varepsilon)} \quad (184)$$

Which can be rewritten as

$$|B_\delta^n| \times 2^{-n(\mathbb{H}(x)-\varepsilon)} \geq 1 - \varepsilon - \delta \Rightarrow \frac{1}{n}\log_2 |B_\delta^n| \geq \mathbb{H}(x) - \varepsilon + \frac{1}{n}\log_2(1-\varepsilon-\delta) \quad (185)$$

We can then, for any δ and δ', find ε and n large enough [65] so that

$$\frac{1}{n}\log_2 |B_\delta^n| \geq \mathbb{H}(x) - \delta' \quad (186)$$

■

So, the minimum-size set that concentrates the observations is indeed the set whose size is given by the entropy, i.e., the typical set. Therefore, we cannot find a code that surpasses the typical code, and ***the bound given by Theorem 14 is optimal***.

While geometry gives us insight into typical sets and their connection to entropy, the typical code is not practical at all. The reason is simple to understand: in the typical code, it is necessary to set a bit if a sequence belongs to the typical set, but one would need to be able to test if this is the case directly! However, this is not feasible. Therefore, we need to find ways to implement this notion of typical coding in ***efficient algorithms that achieve the Shannon bound***. These are called ***instantaneous entropy codes***.

To address the topic that will be developed next time, let's take a sequence $X = (x_1, x_2, \ldots, x_n)$ where each x_i takes its value from an alphabet $\mathcal{A} = \{a_k]\}_{1 \leq k \leq K}$. So, each a_k value is encoded by a binary word $w(k)$ of length $\ell(k)$, and defining a code means

65. For example, $\varepsilon \leq \delta'/2$ and $n \geq \log_2(1-\delta-\delta'/2)/(\delta'/2)$.

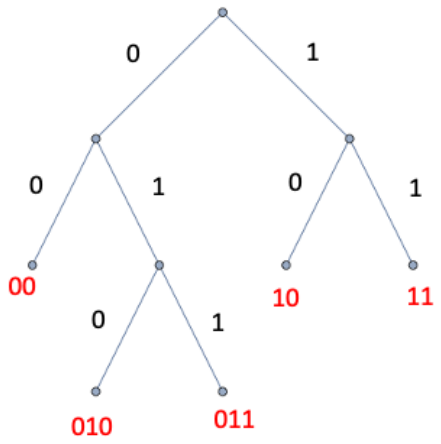

FIGURE 17 – Representation of a binary tree used to create a coding that satisfies the prefix constraint by taking the tree's leaves.

giving each symbol a_k the word $w(k)$ in a way that minimizes the length. We don't code the entire sequence X, but each of its elements individually. Thus, we want to minimize the quantity:

$$R = \sum_k \ell(k) p(a_k) \tag{187}$$

One possibility would be, for example, to arrange the symbols in decreasing order of occurrence, and in the case of 4 symbols, define the variable-length code as follows: $w_1 = 0, w_2 = 10, w_3 = 101, w_4 = 111$, with the background idea of using shorter (longer) binary words for frequent (rare) symbols. Two remarks come immediately: *first*, we need to know the probabilities $p(a_k)$ *a priori*, and *second*, the code must be decodable. Suppose we have an idea of the occurrence frequencies of the symbols through preliminary analysis, the second problem is more serious. Indeed, if I send (a_2a_2) via the code 1010, the receiver can interpret not only (a_2a_2) but also (a_3a_1), which is particularly inconvenient. The code is not satisfactory because it is not uniquely decodable. Note that the receiver who receives 1111 due to a simple transmission error would start decoding a_4 and then would not know what to do with the remaining 1; they would likely wait for additional bits...

Let's assume that the communication channel is ideal. The source of ambiguity between a_2a_2 and a_3a_1 lies in the fact that the word w_2 is the beginning of the word w_3.

Adding separators would not help because we would need to code the separator, which would increase the amount of information to transmit. Instead, we need a constraint known as the *prefix* constraint, which states that *no binary word is the beginning of another word*. With this type of constraint, in a sense, we have a separator without the additional cost of increasing the number of bits to transmit. However, this prefixing is a constraint, so it would be useful to construct codes easily. Here the elegant observation to make is *the correspondence with a binary tree*, and *the prefixing constraint is satisfied if and only if we take the leaves of the tree* (Fig. 17). Indeed, if we take only the leaves of the tree, we cannot have two words where one is the beginning of the other, and conversely, suppose I have a code satisfying the prefixing condition, then I can construct the tree and cut it at the level of the code words, which are, in fact, the leaves of the tree.

The remaining point to see, not to mention problems with noisy and/or faulty channels, is the optimization of R. However, $\ell(k)$ corresponds to the depth of $w(k)$ in the binary tree. So, R ***represents the average depth of the leaves of the binary tree representing the code***. Thus, ***the problem, given the*** $p(a_k)$***, is to construct a binary tree whose leaves have, on average, the smallest possible depth***. Incidentally, this also validates using short words for the most probable sequences. The answer gives the optimal code.

7. Lecture 23 Feb.

For reference, in the last session, we saw that if we want to encode a series of values $X = (x_1, x_2, \ldots, x_n)$ where the x_i are elements from a finite alphabet $\mathcal{A} = \{a_k\}_{1 \leq k \leq K}$, then the average number of bits per symbol satisfies [66]

$$\mathbb{H}(x) \leq R \leq \mathbb{H}(x) + C\varepsilon \tag{188}$$

66. The upper bound is from Theorem 14, which was obtained for typical sets, while the lower bound has not been proven, in fact. We can outline a possible proof: the relationship between the size of the typical set and entropy is roughly $\mathbb{H}(x) \approx 1/n \log_2 |T_n^\varepsilon|$, which, in other words, is the average number of bits needed to encode elements of the set. Taking into account the result in Sec. 6.6 (after Theorem 14) which states that there is no set that concentrates information better than the typical set, the value of R cannot be smaller than entropy $\mathbb{H}(x)$. However, the precise proof within the framework of typical sets remains to be established. I present a proof established using the Kraft lemma in this session.

for an ε-***typical*** coding (Definition 11), and that we cannot do better. From a geometric perspective, the coding of the n symbols, which is an element of $\mathcal{A}^n$, is actually an element of the typical set T_n^ε, which has a size of approximately $2^{n\mathbb{H}(x)}$, and ***so the number of bits is of the order of the logarithm of this quantity, which is*** $\mathbb{H}$ ***per symbol***. Ultimately, there exists no set smaller than T_n^ε.

7.1 Instantaneous Coding (One Symbol at a Time)

In practice, implementing a typical code is generally not feasible because one must determine if a sequence is an element of the typical set or not in order to toggle a bit. We have seen that simpler codes (called instantaneous codes) that operate symbol by symbol, by assigning one binary word w_k to each symbol a_k satisfying a ***prefix constraint***, can be considered. This can be achieved by constructing a ***binary tree***, where the code words are the ***leaves*** (Fig. 17). Each word w_k has a length ℓ_k, and thus the problem becomes: given the probabilities of occurrence $p(a_k) = p_k$ for each symbol, find the w_k such that ***first***, decoding is possible (satisfying the prefix constraint), and ***second***, the average length of a coded symbol (average number of bits)

$$R = \sum_k \ell_k p_k \tag{189}$$

is minimized. Note that the length ℓ_k exactly corresponds to the depth of the word w_k in the binary tree.

Intuitively, to construct the tree, we consider that the most frequent words should be encoded with short words, thus with leaves of the tree close to the root. Conversely, less frequent words are encoded with words corresponding to leaves farther from the root. If this is in the background for constructing the tree, we need to understand the relationship between ℓ_k and p_k. For this purpose, let's see a first theorem by C. Shannon.

Theorem 16 ***(Shannon's Code)***
Given a source X of symbols a_k with known probabilities denoted as p_k, then for a

prefix code, we have

$$R \geq \mathbb{H} = -\sum_k p_k \log_2 p_k \tag{190}$$

and there exists a code called Shannon's code such that

$$R \leq \mathbb{H} + 1 \tag{191}$$

In other words, the inefficiency is at most 1 bit compared to entropy.

The proof is based on a very important lemma in Information Theory, which is as follows [67]:

Lemma 1 ***(Kraft's Lemma)***
Any prefix code with K binary words w_k of length $\ell(w_k) = \ell_k$ satisfies the following inequality:

$$\sum_{k=1}^{K} 2^{-\ell_k} \leq 1 \qquad (I1) \tag{192}$$

Conversely, if the collection of ℓ_k satisfies inequality $(I1)$, then there exists a prefix code $\{w_k\}_{1 \leq k \leq K}$ such that the lengths of the binary words satisfy $\ell(w_k) = \ell_k$.

Proof 1. Let's consider the necessary condition. Suppose we have a prefix code (binary tree with leaves as code words), and let m be the maximum depth of the tree:

$$m := \max_k \ell_k \tag{193}$$

For each leaf of the original binary tree, we make it the root of a new binary tree that is extended to reach depth m. For an original leaf with depth ℓ_k, the number of leaves in its tree at depth m is $2^{m-\ell_k}$ (Fig. 18). At this maximum depth, all words are disjoint, and their number is less than the total number of possible words at this depth (due to the

67. by Leon Gordon Kraft, a lemma he published in his thesis in 1949.

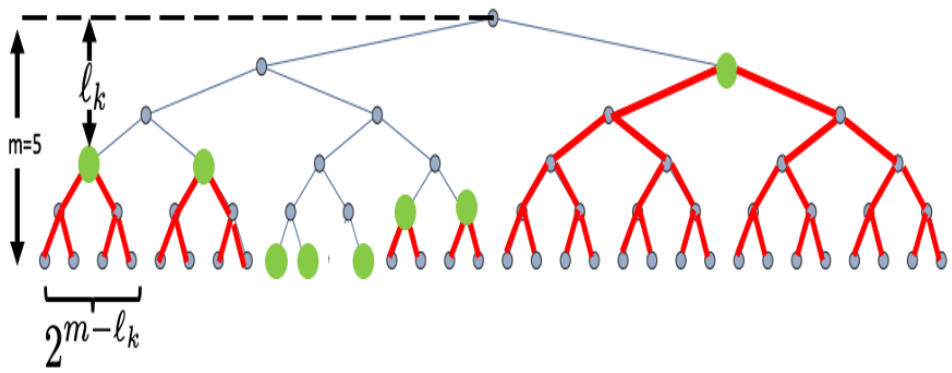

FIGURE 18 – Example of a binary tree with maximum depth m, and its leaves (green dots) extended by binary subtrees to the maximum depth. Kraft's lemma imposes a constraint on the total number of leaves existing at the maximum depth.

presence of the original leaves at this depth):

$$\sum_{k=1}^{K} 2^{m-\ell_k} \leq 2^m \tag{194}$$

This gives us relation $(I1)$ by dividing both sides by 2^m.

Conversely, we have a set of word lengths ℓ_k. Let's start by ordering them: $\ell_1 \leq \ell_2 \leq \cdots \leq \ell_K$ (with a redefinition of indices if needed). For example, in Figure 18, the depths of the leaves (green dots) from left to right are $\{3, 3, 5, 5, 5, 4, 4, 1\}$, which can be rearranged as $\{1, 3, 3, 4, 4, 5, 5, 5\}$. Next, we find the maximum depth, denoted as $m = 5$. We then construct a complete tree up to this depth and attach K subtrees of sizes $2^{m-\ell_k}$, starting from the left, for example. We know that we can include them thanks to inequality $(I1)$ (Fig. 19). The roots of the subtrees are identified as the leaves of the binary coding tree, and we observe that they form a prefix code with lengths $\ell(w_k)$ equal to the original ℓ_k. ■

Now, let's return to the proof of Theorem 16.

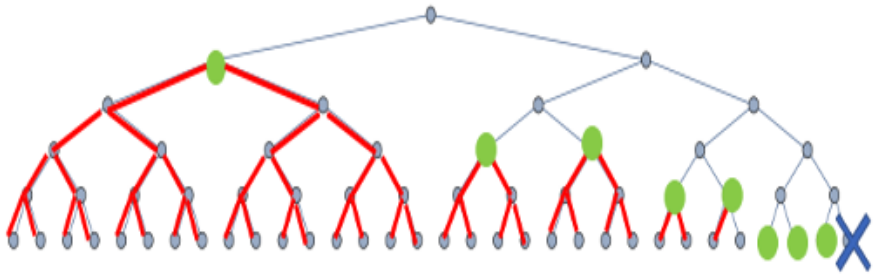

FIGURE 19 – Reconstruction of a binary tree based on word lengths (Kraft's lemma). The cross signifies that an unused leaf is ultimately removed.

Proof 16. To the problem of finding the words w_k such that their lengths ℓ_k minimize

$$R = \sum_{k=1}^{K} \ell_k p_k \tag{195}$$

we add the condition

$$\sum_{k=1}^{K} 2^{-\ell_k} \leq 1 \tag{196}$$

to satisfy the prefix condition. We then have a linear minimization problem with a convex constraint. The solution is unique and given by the Lagrange multipliers [68]. We define the Lagrangian as

$$\mathscr{L}(\{\ell_k\}, \lambda) = \sum_{k=1}^{K} \ell_k p_k + \lambda \left(\sum_{k=1}^{K} 2^{-\ell_k} - 1 \right) \tag{197}$$

The saddle point (or col) satisfies $\forall i$

$$\frac{\partial \mathscr{L}}{\partial \ell_i} = p_i - \lambda \, 2^{-\ell_i} \, \log_e 2 = 0 \tag{198}$$

Summing over all i, the sum of probabilities is 1, and the constraint becoming an equality

68. See Course 2018 Sec. 8.3.

provides

$$\lambda^* \log_e 2 = 1 \qquad \boxed{\ell_i^* = -\log_2 p_i} \tag{199}$$

This establishes the link between the word length ℓ_k and the probability p_k of symbol a_k occurring.

The set of minimum values $\{\ell_k^*\}_k$ gives the corresponding value of R:

$$R_{min} = -\sum_k p_k \log_2 p_k = \mathbb{H} \tag{200}$$

NDJE: With Kraft's lemma, we can prove that for any prefix code, $R \geq \mathbb{H}$. Indeed, for a prefix code based on the probabilities $(p_k)_k$ and the set of lengths ℓ'_k, Kraft's lemma requires that

$$C' = \sum_{k=1}^{K} 2^{-\ell'_k} \leq 1$$

Now, define the probabilities $(q_k)_k$ as

$$q_k := \frac{2^{-\ell'_k}}{C'} \Rightarrow -\log_2 q_k = \ell'_k + \log_2 C'$$

Now,

$$D(p\|q) = \sum_k p_k \log p_k - \sum_k p_k \log q_k \geq 0$$

which gives

$$R' \geq H(x) - \log C' \geq H(x)$$

So, for any prefix binary code, R is greater than or equal to $\mathbb{H}$, and **the result of the theorem then tells us that we could possibly reach the bound by taking the lengths $\{\ell_k^*\}_k$.**

But why, in practice, is this not the case, hence the presence of Shannon's second inequality? The reason is simple: the ℓ_i^* must represent depths in a tree, so they are integers, but **the p_k are not necessarily powers of 2**. In practice, we have an approximation

of this optimal code (called entropy coding). Shannon gives the following example:

$$\boxed{\tilde{\ell}_k = \lceil -\log_2 p_k \rceil} \qquad (Shannon) \tag{201}$$

which satisfies Kraft's inequality because $\lceil x \rceil \geq x$. Regarding the value of R, we have $\lceil x \rceil \leq x + 1$

$$R \leq \sum_k p_k(-\log_2 p_k + 1) = \mathbb{H} + 1 \tag{202}$$

which gives Shannon's second inequality. ■

This theorem is remarkable because it is constructive and provides lower and upper bounds on the average number of bits per symbol. However, it is not optimal; it merely establishes a connection between a theoretical code and a feasible one. *Is it a problem that this code is not optimal?* Well, it depends on the problem, but if we consider images, for example, although the pixel value is coded with 8 bits (0: black, 255: white), on average, we have only about 1/4 of a bit per pixel. In this case, ***adding 1 bit per pixel due to code inefficiency is very penalizing***. So, it's worth the effort to reduce the bound so that typically

$$R \leq \mathbb{H} + O(\varepsilon) \tag{203}$$

The inefficiency arises because the lengths of code words are constrained to be integers. Therefore, we will consider not just 1 symbol at a time but ***blocks of*** n ***symbols***, especially since for large n, we know that we will concentrate the probabilities. We expect to lose 1 bit per block of size n, so per symbol, we will lose only $1/n$ bit.

7.2 Block Entropic Coding

So, let's consider $X = (x_1, \ldots, x_n) \in \mathcal{A}^n$, where each x_i is still an element of an alphabet with K symbols, so the set of all X has a size of $|\mathcal{A}^n| = K^n$. If we apply the theorem from the previous section, we then obtain that the average number of bits

required to code X using Shannon's code satisfies

$$\mathbb{H}(X) \leq R_X \leq \mathbb{H}(X) + 1 \tag{204}$$

If we assume that the $(x_i)_i$ are *iid* (independent and identically distributed), then naturally

$$\mathbb{H}(X) = n\mathbb{H}(x) \tag{205}$$

By the way, when considering *iid* variables, we are dealing with the worst-case scenario. Indeed, if this is not the case, the actual entropy is smaller than the *iid* entropy [69], and hence the code is less efficient than if we had considered the correlation between the symbols. We will revisit this later. But let's stay in the *iid* case for now; then the number of bits per symbol satisfies

$$\mathbb{H}(x) \leq R \leq \mathbb{H}(x) + \frac{1}{n} \tag{206}$$

So, we have an algorithm at hand that, in principle, becomes optimal as we consider blocks of symbols of increasing size (n approaching infinity). However, Shannon's solution is not optimal because, for a fixed n, it does not guarantee that we have the coding that achieves the minimum R.

7.3 Optimal Huffman Code

The idea behind the optimal code stems from the following observation: if we consider the tree associated with the optimal prefix code, then a leaf deeper in the tree always has a lower probability than a shallower leaf. In other words, the deeper we go into the tree, the less probable the symbols become. Indeed, consider the situation depicted in Figure 20. If, all else being equal, $p_{k'} \leq p_k$ for two leaves where $\ell_{k'} \leq \ell_k$, then [70]

$$p_{k'}\ell_k + p_k\ell_{k'} \leq p_k\ell_k + p_{k'}\ell_{k'} \tag{207}$$

So, by swapping symbols k and k', we obtain a coding with a better R value. David

69. NDJE: to put it in the context of statistical mechanics: "there is less disorder due to correlations", and therefore, the code is less efficient than if we had taken the correlation between the symbols into account. We will come back to this. But for now, let's stick to the *iid* case.

70. NDJE: simply realize that $(p_k - p_{k'})(\ell_k - \ell_{k'}) \geq 0$ and expand this expression.

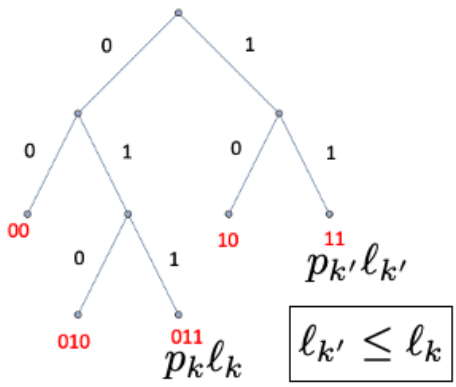

FIGURE 20 – A situation where a deeper leaf has a higher probability, $p_k \geq p_{k'}$: swapping the symbols involved reduces the value of R.

Albert Huffman (1925-99) at MIT in 1951 used this idea to solve a problem posed by his professor Robert Fano [71] (1917-2016). Robert Fano himself had developed the tree starting from the largest probabilities (top-bottom), while Huffman proceeded from the smallest probabilities (bottom-up).

Definition 12 *(**Huffman Code**)*
Suppose we arrange the K symbols in increasing order of probabilities: $p(a_k) \leq p(a_{k+1})$. We will relate the problem of K symbols to that of $K-1$ symbols. To do this, we start with the 2 least frequent symbols (a_1, a_2) to create a symbol $a_{1,2}$ (a_1 or a_2) with a probability equal to the sum $p_1 + p_2 = p_{1,2}$. By eliminating (a_1, a_2) in favor of $a_{1,2}$, we reduce the problem by 1 unit, going from K to $K-1$ symbols. Thus, if we have an optimal code for the $K-1$ symbols $\{a_k\}_{k>2} \cup \{a_{1,2}\}$, then we have an optimal code for the K symbols by dividing the leaf $a_{1,2}$ into two sub-leaves.

The proof is based on the above observation regarding the relative position of probabilities in the tree and the reflection that in a complete prefix tree, there is never a single leaf whose codeword has the maximum length. An example of the implementation of the Huffman code is provided in a Python notebook associated with this course [72]. ***The Huffman code***

71. His older brother Ugo Fano is well-known among nuclear physicists.
72. https://github.com/jecampagne/cours_mallat_cdf/cours2022, `Simple_huffman_code.ipynb`.

is truly optimal in a practical sense, meaning it cannot be improved upon (unless we consider block coding), and its inefficiency compared to the entropy code is significantly less than 1 bit. In the exercise proposed in the notebook, you will observe that $R \approx 2.24$, while the entropy is 2.18, resulting in an inefficiency of 0.06 bits. However, the Huffman code has some drawbacks, such as the need to transmit the coding tree and the rigidity of the code, which must be recalculated for each transmitted text.

7.4 Differential Entropy

All the algorithmic developments we've discussed so far work perfectly for a finite alphabet. The practical problem arises when we have measurements, which are ***real numbers***. Therefore, we need to define the counterparts of information, entropy, and associated algorithms, knowing that a real number can possibly be represented by an infinite number of bits. However, in the real world, so to speak, *"floats"* are represented by 32, 64-bits, or even more sometimes, so we transition into the finite domain at the cost of quantization that introduces an error. Before considering that, let's see how we extend the theory, entropy, and typical sets to real values, thus bridging the gap with the first part of the course, which is Fisher's information.

Definition 13 ***(Differential Entropy)***
Let X be a random variable with probability density with respect to the Lebesgue measure dx denoted as $p(x)$ ($x \in \mathbb{R}$ or $\mathbb{R}^n$). The differential entropy is then defined as

$$\boxed{\mathbb{H}_d = -\int p(x) \log p(x)\ dx} \tag{208}$$

Unlike its discrete counterpart (Def. 6), differential entropy is not necessarily positive. Here's an example:

$$X \sim \mathcal{U}([0, a]) \Rightarrow \mathbb{H}_d = \log a \tag{209}$$

So, if $a < 1$, the differential entropy is negative. We need to view this entropy as relative to a reference measure, in this case, the Lebesgue measure (dx). Concerning the Gaussian

distribution, we have

$$X \sim \mathcal{N}(\mu, \sigma^2) \Rightarrow \mathbb{H}_d = \frac{1}{2} + \frac{1}{2}\log\Big(2\pi\sigma^2\Big) = \frac{1}{2}\log(2\pi e) + \log\sigma \tag{210}$$

In fact, a or σ can be seen as scale factors, and in general ($\forall\alpha > 0$)

$$\boxed{\mathbb{H}_d(\alpha X) = \mathbb{H}_d(X) + \log\alpha} \tag{211}$$

This comes from the fact that $p(\alpha x)(\alpha dx) = p(x)dx$, i.e., $p(\alpha x) = \frac{1}{\alpha}p(x)$. Therefore, the measurement scale affects the relative character of entropy.

From this definition of entropy, we can extend all the concepts we've discussed, including typical sets. What we want to verify is that if we take n *iid* random variables, the joint probability satisfies

$$p(x_1, \ldots, x_n) = \prod_i p(x_i) \approx 2^{-n\mathbb{H}_d(x)} \tag{212}$$

This is the counterpart of Equation 171 in Section 6.4 about typical sets in a finite alphabet. In fact, we have the following property

$$-\frac{1}{n}\log p(x_1, x_2, \ldots, x_n) \xrightarrow[n\to\infty]{prob.} -\mathbb{E}_{x\sim p(x)}[\log p(x)] = \mathbb{H}_d(x) \tag{213}$$

So, the log-probability of a block of n *iid* random variables concentrates, to an epsilon, around its mean, i.e., the differential entropy. This leads us to define sets that contain almost all realizations of these n blocks, which are the associated typical sets. The definition follows the one given by Equation 169 using differential entropy [73]:

$$\boxed{T_n^\varepsilon = \left\{\{x\} \in \mathbb{R}^n, \left|-\frac{1}{n}\log p(\{x\}) - \mathbb{H}_d(x)\right| \leq \varepsilon\right\}} \tag{214}$$

And the convergence in probability, for sufficiently large n, guarantees that

$$\boxed{\mathbb{P}(\{x\} \in T_n^\varepsilon) \geq 1 - \varepsilon} \tag{215}$$

73. NDJE: in Equation 169 and the following ones, $\{x\} \in \mathcal{A}^n$ as opposed to what may have been written in previous versions. I apologize for any confusion.

If the parallel is striking, what is the difference between the two versions (*discrete* vs. *continuous*) of the theory? In the *discrete* case, entropy (always positive) measures, to within ε, the number of bits needed to encode an element of the typical set (See typical code Def. 11), or in other words, it gives the number of elements in the typical set (Prop. 6). In the *continuous* case, not only is differential entropy not guaranteed to be positive, but ***the number of elements in a typical set is infinite***. The connection is given by the following theorem

Theorem 17 *(**Typical Volume**)*
Let the volume of a set Ω be relative to the Lebesgue measure:

$$V(\Omega) := \int_{\Omega} dx$$

For sufficiently large n,

$$\boxed{(1-\varepsilon)2^{n(\mathbb{H}_d(x)-\varepsilon)} \leq V(T_n^{\varepsilon}) \leq 2^{n(\mathbb{H}_d(x)+\varepsilon)}} \tag{216}$$

Proof 17. We provide only a part of the proof, which follows its discrete counterpart. Denoting $\{x\} = X$, membership in X in T_n^{ε} means that

$$2^{-n(\mathbb{H}_d(x)+\varepsilon)} \leq p(X) \leq 2^{-n(\mathbb{H}_d(x)-\varepsilon)} \tag{217}$$

Now,

$$1 = \int p(X)dX \geq \int_{T_n^{\varepsilon}} p(X)dX \geq 2^{-n(\mathbb{H}_d(x)+\varepsilon)} \int_{T_n^{\varepsilon}} dX \tag{218}$$

which gives one side of the double inequality. ■

Thus, typical volumes have volumes such that

$$V(T_n^{\varepsilon}) \approx 2^{n\mathbb{H}_d(x)} \approx \frac{1}{p(X)} \tag{219}$$

So, the probability is nearly constant and is given by the inverse of the volume. Differential entropy can then be seen as the base-2 logarithm of the length of one side of an equivalent

volume (in n dimensions).

To use discrete codes like Huffman's, we need to establish a kind of tiling of the typical sets T_n^ε. Each ball in this tiling defines a symbol, and the set of symbols can be used to describe any element of T_n^ε within a (small) error. The challenge is to find "optimal" tilings. We will see that there are simpler ways to approach the problem, as in the discrete case. Before that, we will draw the link between this notion of entropy and inference (Fisher's information).

7.5 Maximum Entropy Principle

NDJE: In order to reformulate Statistical Mechanics, especially with the aim of addressing problems out of equilibrium, a principle[74] *was formulated in 1957 by Edwin Thompson Jaynes (1922-98): it's called the* ***Maximum Entropy Principle****. It turns out that from this principle, one can reconstruct all of Statistical Mechanics by considering it as a deductive theory (i.e., a theory of inference), as it naturally leads to the Gibbs partition function. Thus, according to E. Jaynes, Shannon's entropy should be regarded as the primary concept from which other observables are derived.*

Jaynes' idea is about how to best utilize ***partial information*** or ***constraints*** that we have about a system. For instance, consider a gas with a fixed temperature. According to Boltzmann's theory, the system will "optimize" its configuration in such a way that the probability of the configuration is given by

$$P \approx Z^{-1} \exp\left\{-\frac{\mathcal{H}}{k_B T}\right\} \tag{220}$$

where $\mathcal{H}$ is the system's Hamiltonian (equal to the constant total energy) governing the motion of each gas particle. Here, we see a connection with the idea of typical sets, where the system's configuration is one point in the set, and the probability is nearly constant. ***Typical sets are the largest sets that correspond to the fixed temperature constraint***. Jaynes extends this idea beyond Statistical Mechanics.

Jaynes points out that in many problems, ***the observables we have are mean values***.

74. In the original sense, it is a guide for ordering the world, some may see it as an axiom/postulate.

So, for an observable $U_k(x)$ ($k \leq K$), which we have access to:

$$\int p(x)U_k(x)dx = \mathbb{E}_{x\sim p}[U_k(x)] = \mu_k \tag{221}$$

But we only know μ_k while $p(x)$ is unknown. The functions $U_k(x)$ can be more or less complicated. Given the $(\mu_k)_k$, the question is: ***What probability density*** $p(x)$ ***underlying the studied processes will naturally satisfy these observation constraints?*** However, just giving the $(\mu_k)_k$ is not sufficient; we need a guiding principle. The idea is to constrain $p(x)$ to be ***as uniform as possible in a space of maximum volume***. This is where the connection with ***typical sets*** comes in. Maximizing volume means ***maximizing differential entropy***. Jaynes seeks ***a priori probability distribution as least informative as possible***[75]. In the end, we have an optimization problem with convex constraints, which yields the following Boltzmann/Gibbs theorem in Statistical Physics:

Theorem 18 ***(Boltzmann/Gibbs)***
Given the problem of finding the probability $p^*(x)$ such that the function

$$H(p) = -\int p(x)\log p(x)dx$$

and K functions c_k from $\mathbb{R}^n \to \mathbb{R}$ satisfy

$$p^* = \underset{p}{\operatorname{argmax}}\ H(p); \qquad \text{and} \quad \forall k, c_k(p) = 0 \tag{222}$$

If the solution p^* exists, it is unique and can be written as

$$\boxed{p^*(x;\theta) = Z^{-1}\exp\left\{-\sum_k \theta_k U_k(x)\right\}} \tag{223}$$

with $\theta = (\theta_k)_k$ as the Lagrange multipliers. We recover the Fisher-parametrized probability density.

Moreover, we know that $\mathbb{H}(p^*) \geq \mathbb{H}(p_{true})$, but if they are equal then $p^* = p_{true}$.

75. NDJE: It can be noted that E. Jaynes follows the subjectivist tradition of probabilities, following Harold Jeffreys (1891-1989), in studying non-informative priors.

This result is related to the inverse problem of finding the $(U_k)_k$ to approximate the true probability, which is related to issues in neural network architectures.

Proof 18. Let's consider the first part of the theorem. The solution achieves the extremum of the Lagrangian

$$\mathscr{L}(p,\theta) = H(p) + \sum_{k=1}^{K} \theta_k c_k(p) + \theta_0 \left(\int p(x)dx - 1 \right) \tag{224}$$

where $c_k(x) = \mu_k - \int p(x)U_k(x)dx$. Here, the variables $(\theta_k)_{k\leq K}$ are the Lagrange multipliers. Thus, p^* satisfies (in the sense of Gâteaux differentiation)

$$\frac{\partial \mathscr{L}}{\partial p(x)} = -\log(p(x)) - 1 - \sum_{k=1}^{K} \theta_k U_k(x) + \theta_0 = 0 \tag{225}$$

Therefore,

$$p^*(x) = \exp\left\{ \theta_0 - 1 - \sum_{k=1}^{K} \theta_k U_k(x) \right\} \tag{226}$$

and the normalization condition yields the value of θ_0, which is represented by the partition function Z as follows:

$$p^*(x;\theta) = Z^{-1} \exp\left\{ -\sum_{k=1}^{K} \theta_k U_k(x) \right\} \tag{227}$$

$$Z(\theta) = \int \exp\left\{ -\sum_{k=1}^{K} \theta_k U_k(x) \right\} dx \tag{228}$$

The θ_k values are determined by the constraints on the means:

$$\int p^*(x;\theta) U_k(x) dx = \mu_k \tag{229}$$

■

Therefore, if p^* exists, we have its expression. However, this is not always the case. But first, let's consider a classic example where we have constraints on the mean and variance

or the covariance matrix in arbitrary dimension $\mathbb{R}^n$. So, let's define the constraints as:

$$\mathbb{E}[X] = \mu \qquad\qquad \mathbb{E}[(X-\mu)(X-\mu)^T] = \Sigma \tag{230}$$

In the case of $d = 1$, where $X \in \mathbb{R}$ and we know the mean μ and variance σ^2, these constraints translate to $\mathbb{E}[X] = \mu$ and $\mathbb{E}[X^2] = \sigma^2 + \mu^2$. Hence, we have $U_1(x) = x$ and $U_2(x) = x^2$. As a result, we find that $1/\theta_2 = 2\sigma^2$, $\theta_1 = -\mu/\sigma^2$, and $Z = \sqrt{-\pi/\theta_2}e^{-\theta_1^2/(4\theta_2)}$. Finally, the distribution $p^*(x)$ takes the form:

$$p^*(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\} = \mathcal{N}(\mu, \sigma^2) \tag{231}$$

This generalizes to dimension n as follows:

$$p^*(x) = \frac{1}{\sqrt{(2\pi)^n \ \det\Sigma}} \exp\left\{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right\} \tag{232}$$

with entropy given by:

$$H_d = \frac{1}{2}\left(n + \log((2\pi)^n \ \det\Sigma)\right) \tag{233}$$

Note that if we had imposed constraints on $\mathbb{E}(X^k)$ with $k = 1, 2, 3$ in dimension 1, we would not have found a solution due to the divergence of the integral caused by the presence of the term $e^{-\theta_3 x^3}$. Therefore, we must set $\theta_3 = 0$, which restricts the number of constraints. The solution found (the Gaussian) will have higher entropy than the original problem with 3 constraints. Hence, we have an upper bound on the entropy given by the entropy of the Gaussian, but there is no physical distribution that can reach the entropy corresponding to the problem with 3 constraints. However, we can perform a perturbative expansion to iteratively approach the solution.

7.6 Link with Inference

Starting from the theorem by imposing constraints and the Maximum Entropy Principle, we arrive at an exponentially parametrized probability distribution. We can view

the problem differently by considering Maximum Likelihood Estimation (MLE). Thus,

$$\ell(\theta) = \log p_\theta(x) \tag{234}$$

which, in the case of an exponential family (Th. 5), yields $\forall k$

$$-\nabla_{\theta_k} \ell(\theta) = U_k(x) - \mathbb{E}_{x\sim p_\theta}[U_k(x)] \tag{235}$$

And if we calculate for $\theta = \theta^*$ that achieves the MLE, then (Eq. 115)

$$\mathbb{E}_{x\sim p_{\theta^*_k}}[\nabla_\theta \ell(\theta)] = 0 \tag{236}$$

So, we deduce that

$$\mathbb{E}_{x\sim p_{\theta^*}}[U_k(x)] = \mathbb{E}_{x\sim p_\theta}[U_k(x)] = \mu_k \tag{237}$$

which is true in particular for the set of θ values that give the true probability [76], which determines the values of μ_k, hence its presence in the above expression. Therefore, the MLE satisfies the constraints on the means and has an exponential form. Thus,

> **Theorem 19** *The maximum entropy solution is the Maximum Likelihood Estimator (MLE).*

MLE and Maximum Entropy Principle are two equivalent concepts. ***In other words, aiming to determine a distribution that maximizes entropy as uniformly as possible (the least informative), which is the Maximum Entropy Principle, is equivalent to adopting an exponentially parametrized model and maximizing likelihood.***

In particular, to determine the Lagrange parameters, one can proceed with gradient descent, as we have seen in Sec. 3.6.2. But again, we encounter problems of instability and the conditioning of the Hessian, which, for the record, is nothing but the Fisher Information $I(\theta^*)$. Note, in passing, that the GD calculation step is written as:

$$\theta_k^{t+1} = \theta_k^t + \varepsilon(\mathbb{E}_{\theta_t}(U_k) - \mu_k) \tag{238}$$

76. NDJE: Recall that in the context of Fisher, the true probability belongs to the same family as the p_θ.

However, the term $\mathbb{E}_{\theta_t}(U_k)$ is difficult to calculate because it requires estimating this integral:

$$\mathbb{E}_{\theta_t}(U_k) = \int U_k(x) p_{\theta_t}(x) dx \tag{239}$$

This is done using Monte Carlo sampling methods (Importance Sampling, Metropolis-Hastings, Gibbs Sampling, Markov Chain, etc.), which demand a lot of resources at each iteration of GD.

Let's return to the second property of Theorem 18. The observation is that if the entropy of the found solution is indeed equal to that of the underlying distribution, then it's the correct solution (recall the case where we didn't reach the entropy of the problem). This is interesting when we flip the problem around. In Statistical Mechanics, we are given observables and asked to calculate the state of the system, but in machine learning, the problem doesn't really take this form. In ML, we generally have observables distributed according to an unknown distribution $\bar{p}$, from which we define descriptors (the $U_k(x)$) whose average values (variances and other moments) are calculated. These descriptors give us a model of $p(x)$ that we hope to approach as closely as possible to $\bar{p}(x)$. In fact, we are looking for the "right" $U_k(x)$, which can, for example, be the result of the cascade layers of a neural network. What we know is the entropy of the studied system $\mathbb{H}(\bar{p})$, and we know that the entropy of the model is always greater ($\mathbb{H}(p) \geq \mathbb{H}(\bar{p})$). So, ***what we're looking for is to minimize the maximum entropy*** (minimax) [77]. In general, we say that we will define ***an approximation by fixing the mean and covariance***, which would correspond in Physics to the term of kinetic energy. However, we now know that we end up with a Gaussian model, which is not suitable for many problems (Sec. 4.6). We need other constraints, but how do we obtain them? One method is to create ***sparse representations***. However, in this case, the distribution of coefficients in the basis (e.g., wavelets) of an image (e.g., Fig. 58 Course 2021) is not at all Gaussian but rather Laplacian because most coefficients are zero and only a few coefficients are important. Thus, the Gaussian model will not be suitable, especially for compression of the image or the field in 1D, 2D, 3D, etc. ***We then impose that the moments of the descriptors reflect this sparsity of coefficients.*** These are the new constraints we are looking for.

77. NDJE: In Statistical Mechanics, it's the problem of the free energy of the model compared to the free energy of the system (e.g., mean field theory).

8. Lecture 2 Mar.

8.1 Towards Compression by Orthogonal Transformation

We will approach Shannon's theory from a more practical perspective by considering the problem of ***signal compression***, i.e., reducing the number of bits required to represent, for example, images, videos, audio, etc. Here, we have two aspects to consider: the perspective of ***Information Theory*** and the perspective of ***Representation***, a theme addressed in 2021. In short, whether in the framework of Fisher or Shannon, there is a fundamental assumption of ***independence of observations***. However, in practice, considering, for example, images or audio, there is a lot of ***redundancy***, and at the same time, there is fundamental ***structure*** since it is what allows us to recognize a face, a voice, etc. And ultimately, ***we are not dealing with independent measurements***. So, the problem is to understand ***how to use the redundancy and structure of observations to minimize the number of bits needed to represent/transmit them and somehow approach problems with independent samples***.

Historically, ***speech*** coding has been particularly enlightening because, fundamentally, we have a physical/physiological model that allows us to establish a ***parameterized model***, which enables addressing information coding/reduction based on these parameters. Thus, we find ourselves more naturally in a Fisher framework for model construction. But there is a broader view of the problem if we approach it from the ***audio*** side, which is to capture any type of sound. And in this case, we don't really have *a priori* models to rely on because, on the one hand, the source is of any nature, and on the other hand, the signal propagation makes modeling even more ***challenging*** and leads to the design of another methodology, especially one involving representations in ***orthogonal bases*** aimed at ***decorrelating the signal coefficients in these representations***. Thus, the concept of ***compression by orthogonal transformation*** was implemented. This perspective has the advantage of working for any type of signal, including images with JPEG/JPEG2000 standards. These standards essentially differ in the choice of the representation basis: JPEG uses the Discrete Cosine Transform (DCT), while JPEG2000 uses a Wavelet basis [78].

For example, considering ***voice*** in the context of telephony with sufficient quality to

78. See Courses from 2018, 2020, and 2021.

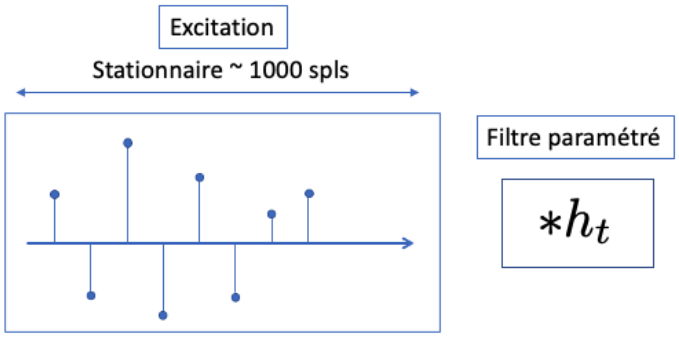

FIGURE 21 – Typical processing for voice in the telephony context, where parameterized filters h_t are considered to be stationary over a scale of 1,000 samples.

recognize and understand the interlocutor, the Fourier spectrum considered is typically limited to the range $[200, 3400]$ Hz, while the human ear (young) can hear in the range $[20, 20k]$ Hz. So, very low frequencies are restricted, and in the high frequencies, only the first 2 harmonics are considered, allowing the recognition of all vowels. Let's assume that the maximum frequency is 4 kHz, and the signal is sampled at 8,000 samples per second (Nyquist-Shannon Theorem [79]). If each sample is encoded as an 8-bit word, then we would have a flow of 64 kb/s (kbits/sec). This is the bitrate that would be required if this kind of brute-force coding were applied. However, for economic reasons, we want to reduce this bitrate while maintaining transmission quality. Currently, for Voice over IP, a significant reduction is achieved, requiring only a flow of about 2.5 kb/s. To do this, models of excitation and response to different physical processes are used, considered as stationary filters over scales of 1,000 samples (Fig. 21) to implement coding.

If we consider ***audio***, which is a broader context than Voice over IP, such as the transmission of pieces of music, we want signals of much higher quality (high fidelity). The CD-Audio, which set the standards in the 1980s, covered a range of $[0, 20k]$ Hz, i.e., the entire range of human hearing sensitivity. Therefore, the sampling rate was 44.1 kHz, and each sample was encoded as 16 bits, resulting in a bitrate of 706 kb/s. However, to stream music in real-time, we need to compress the information without degrading listening quality. The idea this time is to use orthogonal bases that restore high fidelity

79. Course 2021 Sec. 6.4

with bitrates of around 100 kb/s. If you want to reduce the bitrate even further, it will affect the quality of playback.

Regarding ***static imaging***, typically with a size of 1024x1024 pixels, each of which is encoded, for example, as 8 bits (1 byte), this results in roughly 1 MB of data. However, to save space on storage media, JPEG compression reduces this to 0.5 bit per pixel, resulting in a gain between 10 to 20 without significant degradation. The gain of switching to JPEG2000 becomes noticeable when you want to achieve high compression ratios.

Finally, for ***video***, one could think of the flow as a sequence of 2D images, creating a simple 3D extension by combining all the images into a single block. This way of looking at it is quite inefficient because typically during a video, there is a static (invariant) scene in which a few elements move. So, there is a significant difference between the time variable and the 2 spatial variables. Taking into account these specifics was the basis of MPEG coding (1988). In essence, we attempt to calculate the velocity field (optical flow) of pixel movements from one image to the next, and then we encode this field. Thus, starting from one image, we can predict what the scene will be like, and then we perform subtraction from the real image to obtain an error image, which is encoded as in the case of static images, often in JPEG. The part that requires the most bits is encoding the error image because the velocity field is relatively lightweight, as typically very little changes.

8.2 Distortion and High-Resolution Assumption

At the core of capturing sounds, images, etc., there is a ***digitization*** process, which takes us from real values to an infinite amount of information to integers on 8, 16... bits. Therefore, it is necessary to address the distortion that this digitization implies. Let's begin with a random variable X because this will allow us to bridge the gap between entropy over finite-sized alphabets and differential entropy, which applies to real random variables. So, let $p(x)$ be the probability density of X, and digitization involves defining a ***quantizer*** Q, which segments the real axis into possibly variable-sized bins (Fig. 22):

$$Q(x) = a_k \quad \text{if } x \in]y_{k-1}, y_k] \tag{240}$$

Of course, this (non-linear) operator introduces an error, and the distortion is defined

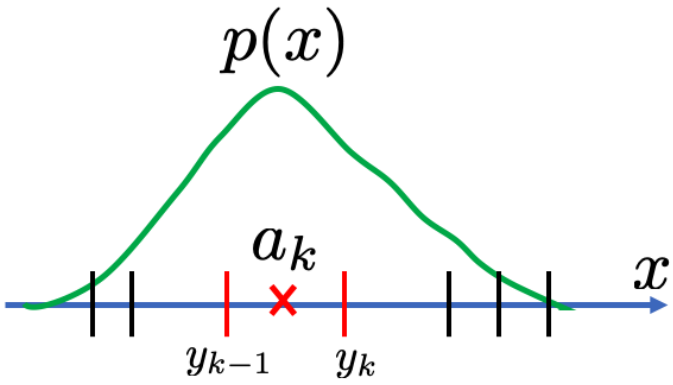

FIGURE 22 – Schematic representation of the quantization operator (Eq. 240).

as [80] (nb. norme quadratique) according to the expression:

$$\boxed{D = \mathbb{E}(\|X - Q(X)\|^2) = \int (x - Q(x))^2 p(x) dx} \tag{241}$$

Certainly, we want to minimize this distortion, which means we need to consider how to choose the bins $(y_k)_k$. To do this, we will make an assumption of simplifying regularity initially before discussing its limits. It is called ***high-resolution quantization*** and consists of the following:

$$\forall x \in]y_{k-1}, y_k], \quad p(x) \approx p(a_k) \qquad \text{(High Resolution Assumption)} \tag{242}$$

Thus,

$$p_k = \mathbb{P}\{Q(X) = a_k\} = \mathbb{P}\{x \in]y_{k-1}, y_k]\} = \int_{y_{k-1}}^{y_k} p(x)dx \approx p(a_k)(y_k - y_{k-1}) = p(a_k)\Delta_k \tag{243}$$

80. In itself, the mean squared norm is not a bad measure, but in certain cases, there are better ones. See the discussion in Sec. 8.6.

Theorem 20

Under the high-resolution quantization assumption, $a_k = (y_k + y_{k-1})/2$, and the distortion can be expressed as:

$$\boxed{D = \sum_k p_k \frac{\Delta_k^2}{12}} \tag{244}$$

The proof proceeds by transforming the expression for D under the given assumption:

$$\begin{aligned} D &= \int (x - Q(x))^2 p(x) dx \\ &= \sum_k \int_{y_{k-1}}^{y_k} (x - a_k)^2 p(a_k) dx \overset{min}{=} \sum_k p(a_k) \frac{2}{3} \frac{\Delta_k^3}{2^3} = \sum_k p_k \frac{\Delta_k^2}{12} \end{aligned} \tag{245}$$

(Nb. it is not surprising that we recover the contribution of the variances of the uniform distributions over each bin of width Δ_k.) In the case of uniform quantization where all Δ_k are constant and equal to Δ, then:

$$\boxed{D = \frac{\Delta^2}{12} \qquad \text{(Constant Quantization)}} \tag{246}$$

The error is independent of the underlying distribution $p(x)$ but *provided that we can make the high-resolution assumption*. We will revisit this.

8.3 Optimal Quantizer

Now, the problem of compression can be stated as follows: ***how to choose the right quantizer, either to minimize the number of bits for a fixed error or to minimize the error given a fixed number of bits***. Let's consider the second version of the problem. One might think that we should reduce the size of the bins where the probability is high and vice versa where the probability is low; we would tend to want to increase the size of the bin. Is this the correct answer? *Be cautious; we are fixing the number of bits!* However,

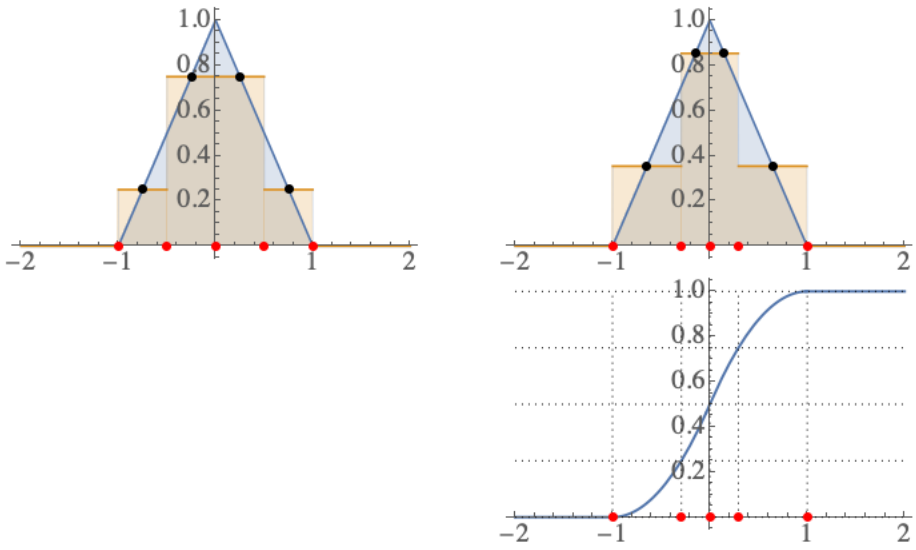

FIGURE 23 – Example of two strategies for dividing the x-axis of the probability density $p(x)$, which is triangular here on $[-1, 1]$: on the left, it is uniform quantization, and on the right, it is the division performed to obtain constant probability in each bin. The latter is achieved by considering the cumulative distribution function $F(x)$ (bottom right) and a uniform division of the y-axis.

this constraint is not equivalent to fixing the number of quantization bins (whose optimal solution would have been different).

If we perform a "naive" coding and fix the number of bins (K), then the number of bits is roughly $\log_2 K$. But, we know we can do better because, to within an epsilon, the average number of bits for the optimal code is given by the entropy $\mathbb{H} = -\sum_k p_k \log_2 p_k$. What we will observe is that ***the optimal quantizer is the constant quantizer***. Why? The reason lies in the small argumentation at the beginning of Section 6.2. Taking bins of variable size as described above implicitly optimizes the fact that the probability of each bin is uniform, which results in high entropy. In contrast, taking a uniform quantizer reflects the probability $p(x)$ in each bin, and it significantly reduces entropy (in the limit, if $p(x)$ is in only one bin, entropy is zero). One could argue that we lose in error with the uniform quantizer solution. So, we need to do the math...

NDJE To numerically illustrate the point, we can refer to the example in Figure 23.

In the case of uniform quantization (graph on the top left), the entropy and distortion values are approximately $\mathbb{H} = 3/4(1 + \log_2(8/3)) \approx 1.81$ *and* $D = 1/48 = 0.0208$*, while for the uniform probability quantization scheme (graphs on the right),* $\mathbb{H} = 2$ *and* $D = (2-\sqrt{2})/24 = 0.0244$*. Although the values of* $\mathbb{H}$ *and* D *are approximately the same for both quantization schemes, uniform quantization is the better choice. Note that the differential entropy in this context is* $2/\log 16 \approx 0.72$*.*

Theorem 21 ***(Optimal Quantizer)***
Let $p(x)$ *be the probability density of* X*, with differential entropy*

$$\mathbb{H}_d[X] = -\int p(x) \log_2 p(x) dx \tag{247}$$

Under the assumption of high-resolution quantization, we can define the source entropy of $Q(X)$*, which is also the number of bits* R *required to encode it*

$$\mathbb{H}[Q(X)] = -\sum_k p_k \log_2 p_k = R \tag{248}$$

then

$$\mathbb{H}[Q(X)] \geq \mathbb{H}_d[X] - \frac{1}{2}\log_2(12D) \tag{249}$$

and equality holds if and only if the quantization is uniform with step size Δ*, which minimizes* $\mathbb{H}$*, and we have*

$$\boxed{R = \mathbb{H}[Q(X)] = \mathbb{H}_d[X] - \log_2 \Delta} \tag{250}$$

Thus, the constant quantizer is the optimal quantizer.

Through this theorem, we obtain the connection between differential entropy (continuous framework) and entropy (discrete framework). If we remember that the volume of a typical set is approximately $V \approx 2^{nH_d(x)}$, and the volume in n dimensions of the bin around an element of this set $\delta V \approx \Delta^n$, then the relation above actually gives us $\log_2 N_b = n\mathbb{H}$, where N_d is the number of quantization bins needed to tile the entire typical space. Let's move on to the proof, but before that, it would be better to get an idea of the result to understand it better.

Proof 21. So, we want to calculate $\mathbb{H}$, which is

$$\mathbb{H} = -\sum_k p_k \log_2 p_k \tag{251}$$

under the high-resolution quantization assumption, so $p_k = p(a_k)\Delta_k$, and over the interval $]y_{k-1}, y_k]$, the probability is constant, $p(x) = p(a_k)$. Thus,

$$\begin{aligned}
\mathbb{H} &= -\sum_k \log_2(p(a_k)\Delta_k) \times \int_{y_{k-1}}^{y_k} p(x)dx \\
&= -\sum_k \int_{y_{k-1}}^{y_k} \log_2(p(x)\Delta_k)p(x)dx \\
&= -\sum_k \int_{y_{k-1}}^{y_k} p(x)\log_2 p(x)dx - \sum_k \int_{y_{k-1}}^{y_k} \log_2(\Delta_k)p(x)dx \\
&= \mathbb{H}_d - \sum_k p_k \log_2(\Delta_k)
\end{aligned} \tag{252}$$

Given that the probability density is fixed, $\mathbb{H}$ and $\mathbb{H}_d$ are fixed, and only the values of $(\Delta_k)_k$ and p_k remain to be optimized, knowing that $\sum_k p_k = 1$. So, we have an optimization under constraint. But the function to minimize is the average of the logarithm, that is, $-\sum_k p_k \frac{1}{2}\log_2(\Delta_k^2)$. Now, since $-\log(x)$ is a strictly convex function, using Jensen's inequality (Theorem 12), we get

$$\mathbb{H} \geq \mathbb{H}_d - \frac{1}{2}\log_2\left(\sum_k p_k \Delta_k^2\right) = \mathbb{H}_d - \frac{1}{2}\log_2(12D) \tag{253}$$

This gives the expected result, but we will agree that if we had only proceeded with the proof, its meaning would not have been immediately revealed. Furthermore, Jensen's inequality becomes an equality if $\Delta_k = Cte = \Delta$. This provides the second result. **The optimum is achieved for the constant quantizer.** ■

NDJE. In the case (constant quantizer) of the example in Figure 23, we have $\mathbb{H}_d = 2/\log(16) \approx 0.72$*, so* $\mathbb{H}_d - 1/2\log_2(12D) = 1.72$*, while* $\mathbb{H} \approx 1.81$*. If we increase the number of quantization bins from 4 to 10, then* $\mathbb{H}_d - 1/2\log_2(12D) \approx 3.04$*, while* $\mathbb{H} = 3.06$ *(relative agreement to* $7\ 10^{-3}$*), and for 100 bins, the relative agreement is accurate to* $5\ 10^{-5}$*. Thus, the agreement becomes perfect asymptotically.*

If we return to the coding problem, if the distortion (error) D is fixed, then the average number of bits is given by

$$R(D) = -\sum_k p_k \log_2 p_k = \mathbb{H} = \mathbb{H}_d - \frac{1}{2}\log_2(12D) \tag{254}$$

or if R is fixed, we obtain a minimum error given by

$$\boxed{D(R) = \frac{1}{12} 2^{2(\mathbb{H}_d - R)}} \tag{255}$$

This results in ***exponential decay*** of D as a function of R: e.g., if we add 1 bit, the quadratic error is divided by a factor of 4. Thus, this result gives us the error when we have ***a real random variable coded with the optimal quantizer***. This is the basic result. However, the practical problem that arises is that of a signal, possibly in high dimensions, which has ***redundancy and structure***, i.e., two characteristics that we would like to exploit. So, we will try to find a representation that is best suited for these signals. The simplest and algorithmically efficient technique is the use of ***orthonormal bases***. This theme was the subject of the 2021 Course, relating ***Sparsity, Regularity, and Approximation***. This time, we will take the approach of ***coding by orthogonal transform***. Recall that this coding is very well suited for signals whose structure is not sufficient to attempt dedicated parametric models. Before that, we need to quantize the signal.

8.4 Scalar Quantization

Let Y be a vector of size N ($0 \leq n < N$) with components denoted as $Y[n]$, and let $\mathscr{B}$ be an orthonormal basis in $\mathbb{R}^N$, $\mathscr{B} = \{g_m\}_{0 \leq m < N}$ with

$$\langle g_m, g_{m'} \rangle = \sum_n g_m[n]\ g^*_{m'}[n] = \delta(m - m') \tag{256}$$

The decomposition of the vector Y can be expressed as

$$Y = \sum_m \langle Y, g_m \rangle\ g_m = \sum_{m=0}^{N-1} A[m]\ g_m \tag{257}$$

So, instead of coding the components of Y in a canonical Dirac basis, ***we have the freedom to choose a basis $\mathscr{B}$ and code the coefficients of Y in it*** (i.e., coding the inner products). Be aware that in the process, $A[m]$ becomes random variables because

$$A[m] = \sum_{n=0}^{N-1} Y[n]\ g_m^*[n] \tag{258}$$

meaning that $A[m]$ ***is a linear combination of the N random variables*** $Y[n]$. In the following, uppercase letters X, Y represent random variables, while lowercase letters represent scalars.

Therefore, the problem of coding the vector Y boils down to solving the coding problem for $A[m]$. We begin by assuming that

$$\mathbb{E}[A[m]] = 0 \tag{259}$$

If this is not the case, we assume that these means can be subtracted, and both the transmitter and receiver of Y can store them once and for all.

The next step is to perform ***scalar quantization***, where we operate on one component at a time, meaning

$$\boxed{\hat{A}[m] := Q(A[m])} \tag{260}$$

We could consider the block of N values $A[m]$ and attempt to adapt the quantization box in N dimensions to the probability distribution. Indeed, in N dimensions, there is no obstacle to taking non-regular tessellations adapted to the probability distribution. However, scalar quantization is much simpler; it amounts to taking cubes in N dimensions. One might think this simplification is abrupt and that there is something better to do. What S. Mallat tells us is that after a lot of research in this area, optimizing the tessellation in high dimensions ultimately doesn't bring much. In fact, ***the gain from these tessellation optimizations is offset by the optimization of the representation, which is more complex than orthonormal bases, while still using scalar quantization***. Until 3-4 years ago, the problem seemed fixed, with standards in audio/imaging, etc., but since then, there have been new developments because neural networks are better at compressing signals, so they have captured something that was missing from previous representations. However, even in this (new) context, quantization is still scalar.

This quantization introduces an error, as we saw in the previous section, and the vector $\hat{Y}$ constructed from $\hat{A}[m]$ (the received signal at best) is given by

$$\hat{Y} = \sum_m \hat{A}[m] g_m \tag{261}$$

From this, the mean squared error between Y and $\hat{Y}$ is easily calculated thanks to the orthonormality of the basis $\mathscr{B}$

$$D = \mathbb{E}[\|Y - \hat{Y}\|^2] = \mathbb{E}[\sum_m \|A[m] - \hat{A}[m]\|^2] = \sum_m \underbrace{\mathbb{E}[\|A[m] - \hat{A}[m]\|^2]}_{D_m} \tag{262}$$

This means that the total distortion is the sum of the quantization distortions on each of the components $A[m]$ of Y in the basis $\mathscr{B}$. The total number of bits required to code Y is the sum of the number of bits needed to code each quantized component, denoted as R_m. Therefore, finally,

$$D = \sum_m D_m \quad \text{and} \quad R = \sum_m R_m \tag{263}$$

Thus, we encounter the problem developed for one random variable, this time in the extended context of N random variables, namely, ***what is the total distortion*** D ***considering a total number of bits*** R (and vice versa). This is ***a bit allocation problem where we have coding, quantization, and ultimately, base selection issues***.

8.5 Bit Allocation

So, we want to fix the total number of bits to code the vector Y (or rather its components in my chosen basis), but at the same time, this leaves us some freedom to optimize the number of bits required for each of the N directions. What we know is that each $\hat{A}[m]$ will be encoded with a constant quantizer to be optimal. So, what remains to be optimized are the N quantization steps $(\Delta_m)_{m<N}$. Thus, the problem boils down to whether we should favor certain directions to reduce the error? Again, Jensen's inequality will provide the answer, and once again, the result is very simple and intriguing.

Theorem 22 *(**Optimal Allocation**)*
Under the assumptions of high-resolution quantization, given a fixed total distortion D, the total number of bits R is minimized if we set all the steps Δ_m to a single value Δ such that

$$D = \sum_m D_m = N\frac{\Delta^2}{12} \Leftrightarrow \boxed{\Delta^2 = \frac{12D}{N}} \tag{264}$$

Furthermore, if the number of bits per coefficient is denoted as $\bar{R} = R/N$, and the average differential entropy is defined as

$$\overline{\mathbb{H}}_d := \frac{1}{N}\sum_m \mathbb{H}_d(A[m]) \tag{265}$$

then the total distortion is given by the expression

$$\boxed{D(\bar{R}) = \frac{N}{12}2^{2(\overline{\mathbb{H}}_d - \bar{R})}} \tag{266}$$

Note that $\overline{\mathbb{H}}_d$ is fixed once we have chosen the decomposition basis and for reference, the $A[m]$ (the inner products) are random variables. The theorem then tells us that in N dimensions with high-resolution quantization, the single constant quantization for all components is optimal. We proceed with tiling small hypercubes with edge length Δ.

Proof 22. The number of bits is defined by entropy (entropy coding), so for each quantized component $\hat{A}[m]$, it is given by Theorem 21. That is,

$$R_m = \mathbb{H}(\hat{A}[m]) = \mathbb{H}_d(A[m]) - \log_2 \Delta_m = \mathbb{H}_d(A[m]) - \frac{1}{2}\log_2(12D_m) \tag{267}$$

If we take the average over the N components, we get

$$\bar{R} = \overline{\mathbb{H}}_d - \frac{1}{2}\left(\frac{1}{N}\sum_m \log_2(12D_m)\right) \tag{268}$$

To optimize the D_m, again, we are faced with an average of logs. So, due to the strict convexity of $-\log$, we have

$$\bar{R} \geq \overline{\mathbb{H}}_d - \frac{1}{2}\log_2\left(\frac{12}{N}\sum_m D_m\right) = \overline{\mathbb{H}}_d - \frac{1}{2}\log_2\left(\frac{12D}{N}\right) \tag{269}$$

Equality holds, and thus optimization is achieved when we reach the lower bound if $D_m = cte = D/N$, which implies that all Δ_m are equal to Δ such that

$$\Delta^2 = \frac{12D}{N} \tag{270}$$

which is the first result. Similarly, by writing the equality, we get

$$\bar{R} = \overline{\mathbb{H}}_d - \frac{1}{2}\log_2\left(\frac{12D}{N}\right) \tag{271}$$

providing the second result of the theorem. ■

So, this theorem tells us that ***in the context of high-resolution quantization***, the optimal solution is the simplest one, namely the one obtained with a single step, either with fixed D or fixed R. ***We have regular tiling in hypercubes along the axes of the orthonormal basis***. So, the question that arises now is that of ***choosing the basis***.

How are we going to proceed? When we look at the theorem above, we notice that the only connection with the basis is the factor $\overline{\mathbb{H}}_d$, i.e., the differential entropy of the inner products of Y with the basis vectors. Now, $\overline{\mathbb{H}}_d$ is the average of the differential entropies of the probability distributions of the (non-quantized) components $A[m]$ ($p_m(x)$). We know that entropy is smaller when the probability is concentrated around its mean value [81]. By construction, the mean is zero ($\int p_m(x)dx = 0$), so the expected concentration is around the value 0. Now, ***this is precisely what happens when we have a form of sparsity generated by a sparse decomposition***. Note that since there is energy conservation in an orthonormal basis, if there are many 0s, there are large coefficients elsewhere (but they are few). The problem we will see is that by pushing the sparsity reasoning to the extreme, it will conflict with the high-resolution assumption, which will lead us to revisit the results obtained so far.

81. Let's recall the small example from section 6.2.

But before we go too far, let's make use of the results obtained to optimize the orthonormal basis because they formed the basis for all coding ideas until the 1990s.

8.6 Choosing the Orthonormal Basis

If we recall the expression for R_m (Eq. 267), we need to be cautious about the presence of differential entropy because it can be negative (unlike its non-differential counterpart). We haven't imposed that R_m should be positive, or even that it should be an integer. So, in terms of the number of bits, we need to be careful. In a way, we need to find equivalents to Huffman coding (Sec. 7.3) that satisfy $R \in \mathbb{N}$. For example, we will use "greedy" algorithms [82] (*NDJE: see an example in Sec. 8.7*) that allocate bits successively among the different components while minimizing the overall distortion. We find the optimal solution because the problem is convex. What we also notice is an asymptotic behavior (N large) comparable to that of Theorem 22.

Another aspect to consider before addressing the choice of basis is that, for certain applications, taking errors into account for some components is unnecessary. For example, in the case of audio, if we end up with coding that generates large errors for frequencies beyond 20 kHz, it's not critical because the human ear will be unable to detect them. So, the perception of the receiving device needs to be taken into account. Translated into more mathematical terms, this amounts to asking whether the $L2$ norm is suitable for our case. The way ***we can account for perception is by using weighted norms***. Therefore, we replace the expression for D (Eq. 263) with

$$D_w = \sum_m \frac{D_m}{w_m^2} = \sum_m D_m^w \tag{272}$$

with weights $1/w_m^2$. By the way, if we think of m as a frequency axis (or equivalent), then w_m takes into account the receiver's frequency response.

82. In general, a "greedy" algorithm is one that makes the optimal choice at each step without concern for what has come before or what will happen afterward, hoping to achieve the optimal overall result but without guarantees. Huffman coding is an example of this. Other examples outside of coding include giving change, optimizing room occupancy, finding the traveling salesman's route, determining the shortest path in a network...

Now, how do the previous results transform if we optimize D_w? Note that

$$D_m^w = \mathbb{E}\left[\frac{1}{w_m^2}\|A[m] - \hat{A}[m]\|^2\right] = \mathbb{E}\left[\left\|\frac{A[m]}{w_m} - \frac{\hat{A}[m]}{w_m}\right\|^2\right] \tag{273}$$

This means weighting the coefficients by w_m. But be careful, $Q(w_m^{-1}A[m]) \neq w_m^{-1}Q(A[m])$ in general. However, what is optimal based on what we've seen in the previous theorems is to perform a *constant quantization* (uniform) of $A[m]w_m^{-1}$ with a step equivalent [83] to quantizing $A[m]$ with a step

$$\Delta_m = w_m\Delta \tag{274}$$

So, we develop ***"non-uniform" quantizations adapted to the specific problem***, not because of the optimal bit allocation strategy in the case of an error in $L2$ norm but because, on the contrary, ***we adapt the metric by weighting the distortions, which allows allocating more error to certain channels***. The larger $1/w_m^2$ is (small value of w_m like a better definition of signal perception), the greater the contribution to D, so we need to allocate a smaller quantization step. Conversely, with large values of w_m (poorer definition), $1/w_m^2$ is small, hence a small contribution to the total error, and we can quantize with a large step. We will come back to this point because there are cases where neurological physiology suggests that it makes sense to adapt the allocated error for each channel (frequency) depending on the signal itself.

To choose the orthonormal basis, we will exploit ***the signal's redundancy***. We assume that the signal is ***piecewise regular***, and let's assume that this is in terms of time. We want to find an orthonormal basis in which the differential entropy of the signal is as small as possible. Now, a very regular signal implies that the decay of Fourier coefficients at high frequencies is rapid [84]. From the spectral decay, we can read the regularity class of the function (in the Sobolev sense). Thus, on each interval (in time), we discretize into N points, and the orthonormal basis of discrete Fourier on $\mathbb{R}^N$ is given by the set of vectors

83. This simply means that $(A[m]w_m^{-1})/\Delta = A[m]/(w_m\Delta)$.
84. See, for example, Course 2021 Sec. 3.3, Course 2019 Sec. 5.3.1, Course 2018 Sec. 5.2.3.

$\{g_k\}_{k<N}$ as follows [85]:

$$\boxed{\mathscr{B}_F = \left\{ g_k(n) = \frac{\exp\left\{ i\frac{2\pi k}{N} n \right\}}{\sqrt{N}} \right\}, \qquad (k,n) \in [\![0, N-1]\!]} \tag{275}$$

But will this work? Or rather, can we be certain that there will be no high frequencies? The point to note is that implicitly, the definition of $g_k(n)$ imposes a periodicity condition on $\mathbb{Z}$ outside of $n = 0, \ldots, N-1$. However, the signal in question in the interval is not periodic, and there is ***a discontinuity at the boundaries***, which then leads to a spectrum in $1/\omega$ or $1/k$ ($\omega_k = 2\pi k$). This consequence results in spending a lot of bits coding the discontinuity. Therefore, we will have to opt for ***the cosine basis***.

8.7 NDJE: Example of a Greedy Bit Allocation Algorithm

I will illustrate Theorem 22 in the case where the $A[m]$ are independent random variables, each following a Gaussian distribution $\mathcal{N}(0, \sigma_m^2)$. Under these conditions, the differential entropy of each component is given by

$$\mathbb{H}_d(m) = \frac{1}{2}\log_2(2\pi e \sigma_m^2) \tag{276}$$

And the distortion D_m associated with the optimal quantization is related to the allocated number of bits R_m by

$$D_m = \frac{1}{12} 2^{2(\mathbb{H}_d(m) - R_m)} = \frac{\pi e}{6} \sigma_m^2 2^{-2R_m} \tag{277}$$

Note that if $R_m = 0$, then $D_m \propto \sigma_m^2$ (where $c = \pi e/6$ is the proportionality constant), and adding one bit reduces it by a factor of 4. Therefore, the problem is, given the values $(\sigma_m^2)_m$, how to allocate the R_m bits to each component m, knowing that $\sum_m R_m = R$ is fixed. A simple "greedy" algorithm [86] is to iteratively allocate one additional bit to the component with the largest distortion to reduce it by a factor of 4:

85. *NOTE: To match the notations from earlier sections, you should probably consider $(g_m)_m$ instead of $(g_k)_k$.*

86. See the notebook `Allocation_de_bits.ipynb`.

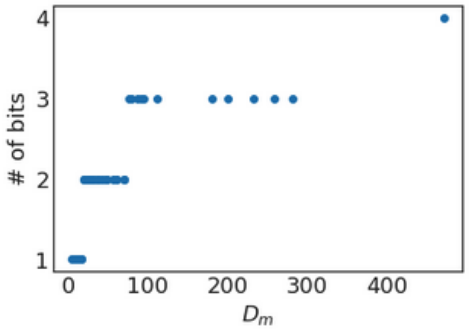

FIGURE 24 – Result of the bit allocation algorithm 1 for $N = 50$ and $R = 100$.

Algorithm 1 *(**Bit Allocation**)*

Initialization *: For all $m = 0, \ldots, N-1$, set $D_m = \sigma_m^2$ and $R_m = 0$.*

While-loop *: Under the condition $\sum_{m=0}^{N-1} R_m < R$, proceed successively with:*

- *Find $m = \underset{n}{\text{argmax}}\ D_n$*
- *Update $R_m \leftarrow R_m + 1$*
- *Update $D_m \leftarrow D_m/4$*

An example of the result of this algorithm is shown in Figure 24.

Next, we can compare the value of the total distortion obtained by summing all the D_m values to the optimized expression of D:

$$D^{algo} = \sum_{m=0}^{N-1} D_m \qquad\qquad D^{optim} = N\frac{\pi e}{6}\left(\prod_{m=0}^{N-1}\sigma_m^2\right)^{1/N} 4^{-R/N} \tag{278}$$

In the example, for a specific configuration of σ_m^2 values, we obtain the following distortion values: $D^{algo} \approx 88.25$ and $D^{optim} \approx 81.08$, resulting in an efficiency of 91.9%.

9. Lecture 9 Mar.

9.1 Recap from the Previous Session

In this session, we will explore applications of the theoretical framework developed during this year's course. We will discover unexpected results that challenge this framework. To understand the origins of these differences, we will revisit sparsity, which is at the heart of the problem. Through this, we will connect with previous years' courses, particularly the one from 2021 on representations.

In the last session, we motivated the use of ***compression by orthogonal transformation*** when there is no underlying model for the generation of observations/data (e.g., audio, which covers all types of sounds, in contrast to speech, which can be modeled). Thus, if we have a vector Y with N components $Y[n]$ $(0 \leq n < N)$, its representation in an orthonormal basis is given by the inner products of Y with the unit vectors of the basis $\mathscr{B} = \{g_m\}_{m<N}$, denoted as $\langle Y, g_m \rangle = A[m]$. We also saw that ***constant (uniform) quantization of these components is optimal***, and we were able to develop a simple ***greedy*** bit allocation algorithm that is nearly optimal.

The signal reconstructed from the quantized components is not identical to the original signal:

$$\hat{Y} = \sum_m \hat{A}[m] g_m \tag{279}$$

To optimize quantization, we used mean square error as the measure of ***distortion***, which is simply the sum of distortions on each of the components:

$$D = \mathbb{E}[\|Y - \hat{Y}\|^2] = \sum_m \mathbb{E}[\|A[m] - \hat{A}[m]\|^2] = \sum_m D_m \tag{280}$$

And if we fix the total number of bits R to code Y, then $R = \sum_m R_m$, the sum of bits allocated for each component. Optimization led us to relate R_m to the differential entropy (Eq. 267), which I'll recall here for reference:

$$R_m = \mathbb{H}(\hat{A}[m]) = \mathbb{H}_d(A[m]) - \log_2 \Delta_m = \mathbb{H}_d(A[m]) - \frac{1}{2}\log_2(12 D_m) \tag{281}$$

We saw that all quantization steps Δ_m are equal to Δ (a constant step for all components).

Theorem 22 gives us the formula that relates D to the average number of bits $\bar{R} = R/N$, resulting in a scaling of $D \propto 4^{-\bar{R}}$. It's essential to remember that all this theory relies on a ***"high-resolution" assumption*** (Eq. 242), which states that for all quantization intervals, we can approximate the probability density $p_m(x)$ of $A[m]$ by a constant.

The remaining degree of freedom is related to the pre-factor of the distortion D (Th. 22), which is the average differential entropy:

$$\overline{\mathbb{H}}_d := \frac{1}{N} \sum_m \mathbb{H}_d(A[m]) \tag{282}$$

Indeed, the more concentrated the distribution of $A[m]$ is around its mean (which is zero by construction), the smaller the differential entropy. This criterion of concentration around 0 implies sparsity related to a ***sparse representation of*** Y ***in the orthonormal basis*** $\mathscr{B}$. Therefore, we are addressing the choice of the orthonormal basis.

9.2 Piecewise Regular Signals: The DCT

Let's consider the case of piecewise regular signals. Typically, we divide the time frame into intervals, for example, of size N, on which we perform signal coding. What we discussed at the end of the last session is that the discrete Fourier basis that comes to mind,

$$\mathscr{B}_F = \left\{ g_m(n) = \frac{\exp\left\{ i \frac{2\pi m}{N} n \right\}}{\sqrt{N}} \right\}, \qquad (m,n) \in [\![0, N-1]\!] \tag{283}$$

will pose a problem due to the implicitly imposed periodicity (N) by the g_k. However, the signal itself does not need to follow this periodicity. ***Discontinuities appear at the boundaries of the intervals, generating a spectrum with high-frequency components*** that are entirely counterproductive. Not only do we expect the spectrum of a regular signal to decay rapidly, which is not the case here, but we also spend bits coding these artificial discontinuities. ***We need to smooth out these discontinuities***.

To do this, instead of directly forcing the periodicity of the extracted signal interval (N samples), we start by ***symmetrizing***, which eliminates discontinuities of order 0. Periodizing with a period of $2N$ leaves discontinuities at the edges concerning the derivative (order 1), which is less troublesome (Fig. 25). Moreover, the symmetry is around a

half-integer, which ultimately motivates the use of the basis

$$\mathscr{B}_{FSym} = \left\{ g_m(n) = \frac{\exp\left\{ i\frac{\pi m}{N}(n+1/2) \right\}}{\sqrt{2N}} \right\}, \qquad (m,n) \in [\![0, N-1]\!] \tag{284}$$

Let $\tilde{x}(n)$ be the symmetrized sampled signal,

$$\tilde{x}(n) = \begin{cases} x(n) & \text{if } 0 \leq n < N \\ x(-n-1) & \text{if } -N \leq n < -1 \end{cases} \tag{285}$$

It decomposes into the $\mathscr{B}_{FSym}$ basis, and if we separate the real and imaginary parts:

$$\tilde{x}(n) = \sum_{m=0}^{N-1} \alpha_m \cos\left(\frac{\pi m}{N}(n+1/2)\right) + \sum_{m=0}^{N-1} \beta_m \sin\left(\frac{\pi m}{N}(n+1/2)\right) \tag{286}$$

Now, $\tilde{x}(n)$ is even with respect to $n = -1/2$, so the sum over sines is identically zero. Thus, the natural orthonormal basis for piecewise regular signals is the one of cosines:

$$\boxed{\mathscr{B}_{cos} = \left\{ g_m[n] = \lambda_m \sqrt{\frac{2}{N}} \cos\left(\frac{\pi m}{N}(n+1/2)\right) \right\}, \qquad (m,n) \in [\![0, N-1]\!]} \tag{287}$$

with the factor λ_m that adjusts the normalization:

$$\boxed{\lambda_m = \begin{cases} 1/\sqrt{2} & \text{if } m = 0 \\ 1 & \text{otherwise} \end{cases}} \tag{288}$$

Note that we have the following relations:

$$\langle g_m, g_{m'} \rangle = \delta(m - m') = \sum_{n=0}^{N-1} g_m[n] g_{m'}[n] \qquad \sum_{m=0}^{N-1} g_m[n] g_m[n'] = \delta(n - n') \tag{289}$$

Thus, the so-called ***DCT*** (Discrete Cosine Transform), derived from the FFT, requires

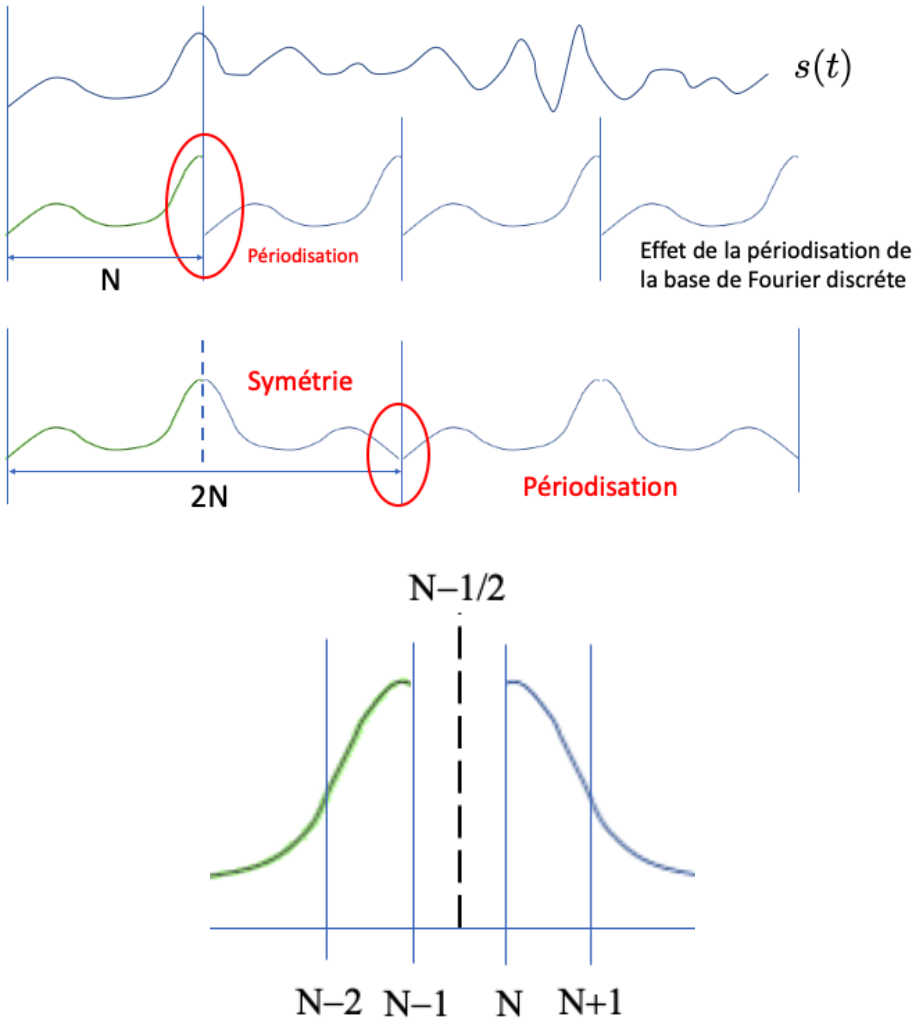

FIGURE 25 – Top: Schematic of the symmetrization procedure to counteract the effect of implicit periodicity in a discrete Fourier basis. Directly periodizing the extracted signal introduces significant junction discontinuities. After symmetrization, there are discontinuities in the derivative but much less troublesome. Bottom: Zooming in on the junction point of symmetrization shows that symmetry occurs around a half-integer.

$O(N \log_2 N)$ operations and is defined as follows[87]:

$$x[n] = \sum_{m=0}^{N-1} \check{x}[m] g_m[n] = \sqrt{\frac{2}{N}} \sum_{m=0}^{N-1} \check{x}[m] \lambda_m \cos\Big(\frac{\pi m}{N}(n+1/2)\Big)$$
$$\check{x}[m] = \langle x, g_m \rangle = \sqrt{\frac{2}{N}} \lambda_m \sum_{n=0}^{N-1} x[n] \cos\Big(\frac{\pi m}{N}(n+1/2)\Big)$$

9.3 Audio Case: MPEG Standard

How do we apply this transformation to audio coding? The first thing to do is to divide the time frame into intervals of about 1024 samples. For this, we use a fixed-size *sliding window*[88] of size $N = 1024$

$$w[n] = 1 \quad \text{if } 0 <= n < N \tag{290}$$

Thus, extracting samples between $[\![pN, (p+1)N - 1]\!]$ is given by the multiplication of $x[n]$ by $w[n - pN]$, and then we apply the DCT as described above. In particular, this sequence of operations introduces the concept of ***block basis***

$$\Big\{ w[n - pN] \cos\Big(\frac{\pi m}{N}(n+1/2)\Big) \Big\} \qquad \forall m < N, \forall p \in \mathbb{Z} \tag{291}$$

There is a possible extension in which different window sizes are used, but the essence is there.

Now, we need to adjust this basis to achieve minimal distortion. However, the mean squared error is not entirely suitable for the perceptual system, especially because of ***masking phenomena***[89]. We would like to account for errors in the budget only when they originate from stimuli above the perception threshold. However, the threshold itself depends on the height of the stimulus itself. In essence, if we stimulate the ear with a sine

87. NDJE: I have opted for a symmetric form here; there are other practical definitions, and I recommend referring to the documentation of each library.

88. NDJE: I use the notation w for *window* to differentiate it from the g_m of the basis. It may also be wise to use windows with softer edges than a rectangle.

89. See, for example, http://www.cochlea.eu/son/psychoacoustique. See also the 2020 Course Sec 7.3 *Naturalistic Digression*.

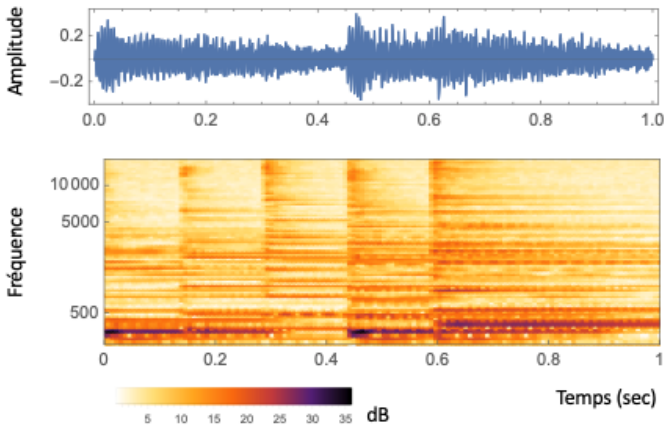

FIGURE 26 – Example of energy calculation for critical frequency bands and a time interval of 1024 samples.

wave at frequency ω, there is a frequency band around ω that is less well perceived, while outside of this band, hearing is not affected. In practice, below 700Hz, there are 7 critical masking bands (each with a width of 100Hz), and above 700Hz, the bands increase in size (logarithmically constant in scale) as the frequency increases. The organ of Corti at the center of the cochlea is covered with cilia bathed in a liquid, and the response of these cilia can be modeled as ***bandpass filters*** that strongly resemble those of wavelets with a constant width in log scale. Therefore, the auditory signal is the result of convolution with these wavelet filter banks (and of constant width at low frequencies).

The algorithms proceed as follows:

— After extracting the samples in an interval of width N, we calculate the frequency energy for each critical band (MEL filter banks [90] as illustrated in Figure 26.

— We will encode the m-th component of the signal in the DCT basis ($\check{x}[m]$) such that the coding error D_m is imperceptible. However, the perception threshold depends on the energy in each band. Since $D_m \propto \Delta_m^2$, the quantization step Δ_m is calculated based on the energy for each critical band.

So, in the end, it appears that we adjust the error to the perception problem with

90. See the 2020 Course Sec. 7.4 MFC (Mel-Frequency Cepstrum).

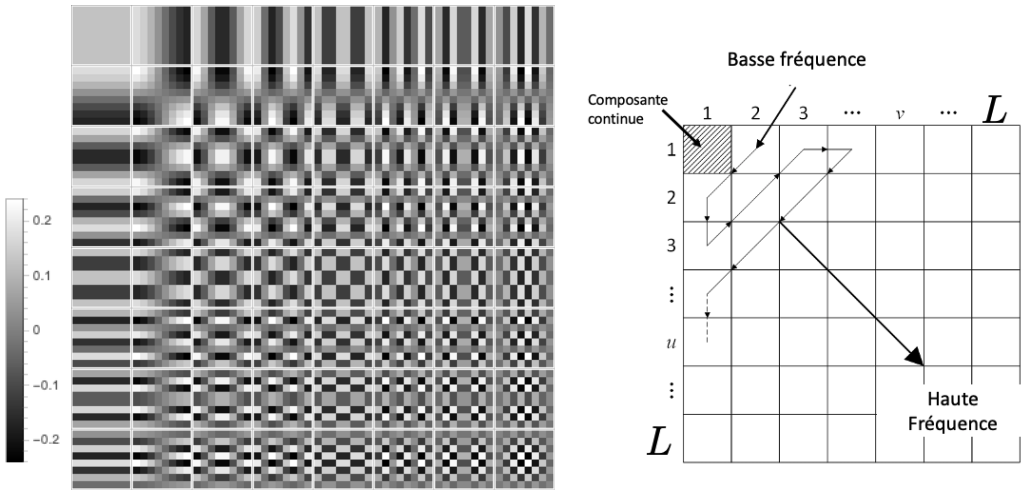

FIGURE 27 – The 64 elements of the 2D cosine basis (Eq. 293) with $L = 8$. Going from low to high frequencies by moving from top left to bottom right in a *zigzag ordering*. The color of the square $m_1 = m_2 = 0$ is given by the constant pixel values of $1/8$.

a variable step. However, we saw last session that the use of weighted quadratic norms accounts just as well for the phenomenon. It's then the weighting coefficients that are adjusted based on the energy in the critical bands. These are weights that are not fixed in advance but are calculated based on the signal at hand.

In practice, for audio in the $[0, 20k]$Hz range, 25 critical bands are defined, and standards allow for achieving compressions that result in a data rate of 100kbits/sec, or about a 7x compression factor while maintaining excellent sound quality.

9.4 Image Case: JPEG Standard

Once again, we need to adapt the orthonormal basis. As in the case of sound in the previous section, we start by dividing the $N \times N$ image into blocks of $L \times L$ pixels, typically with $L = 8$ for JPEG and $L = 16$ in video. If we have small patches of uniform intensity, it is natural to decompose them into a cosine basis in both directions (u, v) of the image. We will then use the following theorem:

Theorem 23

If $\mathscr{B} = \{g_m[n]\}_{m<L}$ is an orthonormal basis of $\mathbb{R}^L$, we can obtain a separable orthonormal basis of $\mathbb{R}^L \times \mathbb{R}^L$ by taking the product

$$\{g_{m_1,m_2}[n_1,n_2] := g_{m_1}[n_1]g_{m_2}[n_2]\}_{(m_1,m_2)<L} \tag{292}$$

Thus, the elements of the 2D basis are given by

$$\mathscr{B}_{cos,2D} = \left\{ g_{m_1,m_2}[n_1,n_2] = \lambda_k \lambda_j \frac{2}{L} \cos\left(\frac{\pi m_1}{L}(n_1+1/2)\right) \cos\left(\frac{\pi m_2}{L}(n_2+1/2)\right) \right\}_{(m_1,m_2)<L} \tag{293}$$

with the λ_k, λ_j defined previously for the 1D cosine basis. The pair (n_1, n_2) identifies a pixel in the $L \times L$ patch. The 64 elements of the 2D basis with $L = 8$ are shown in Figure 27. The patch extracted from the original image is decomposed into this cosine product basis as follows:

$$x[n_1,n_2] = \sum_{m_1,m_2} \langle x, g_{m_1,m_2} \rangle \; g_{m_1,m_2}[n_1,n_2] \tag{294}$$

Note that we can group the pair of indices (n_1, n_2) into a single index n, and we find the same type of expression used in 1D. And, just like in 1D, we can define an orthogonal basis over the entire image by moving a sliding 2D window.

Now, ***the goal is to obtain a sparse representation of the signal*** (in this case, the image). The question that arises is when do we have large coefficients? An example is given in Figure 28. It is clear that in both cases, the coefficient corresponding to $m_1 = m_2 = 0$ is the largest, representing the sum of the pixels in each image within a small factor [91]. But apart from this coefficient, in the randomized image, the coefficients are close to zero. In contrast, for the image with 2 uniform regions, there are 2 or 3 other non-zero coefficients. So, ***we have more non-zero coefficients when there are transitions/discontinuities***. However, we would like to reduce the number of these non-zero coefficients as much as possible. Therefore, we need patch sizes adjusted to avoid intensity transitions. Why not reduce the size to $L = 2$? The reason is that we cannot obtain lower frequencies than the size of the patch. So, for a uniform area of sufficient size, we will not capture the

91. With the definition of the basis taken, the factor is $1/L$. For an image with an average pixel value of 128, the value of the first coefficient is approximately $128 * L \approx 1024$.

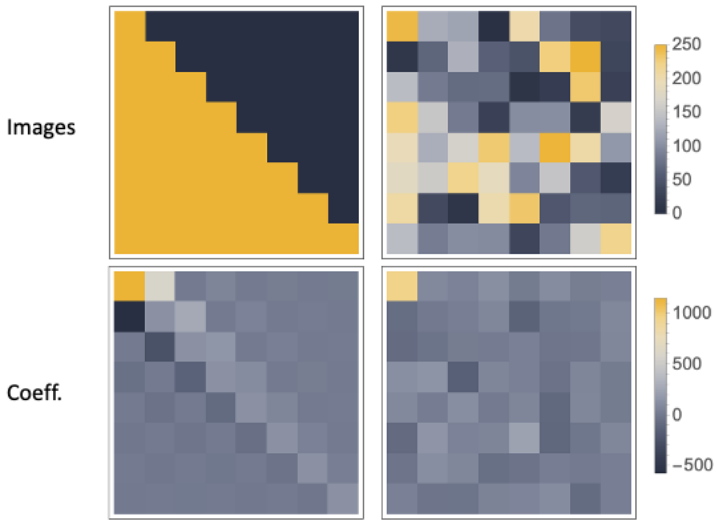

FIGURE 28 – Examples of calculation of decomposition coefficients for patches in the basis of 64 elements shown in Figure 27. At the top: 8×8 images with intensity encoded on 8 bits, with a transition image between 2 uniform regions on the left and a random uniform image on the right. At the bottom, the coefficient table for each image. The scales are common for either the images or the coefficients.

redundancy between patches, and we will encode the low-frequency coefficient multiple times unnecessarily (Fig. 29). On the other hand, if the patches are too large, then each one may contain a discontinuity, which generates several non-zero coefficients to code per patch, resulting in a loss of compression rate. ***So, we need to use the largest possible patches with compromises to achieve the best compression rate.*** After testing, it seems that for typical images taken in everyday life, a size of 8×8 is a good compromise.

We can index the coefficients to obtain a progression from low to high frequencies (*zigzag ordering*) and produce an equivalent of a spectrogram as shown in Figure 30. We can clearly observe the rapid decrease in amplitude of the coefficients as a function of "frequency" (index of the coefficient). Thus, ***the information retained for coding*** is as follows:

— ***The position of coefficients whose quantization is non-zero***. This is done first through a binary vector of size L^2, where each 1 indicates the index of the coefficient with non-zero quantization. Then, this vector is compressed using a *Run*

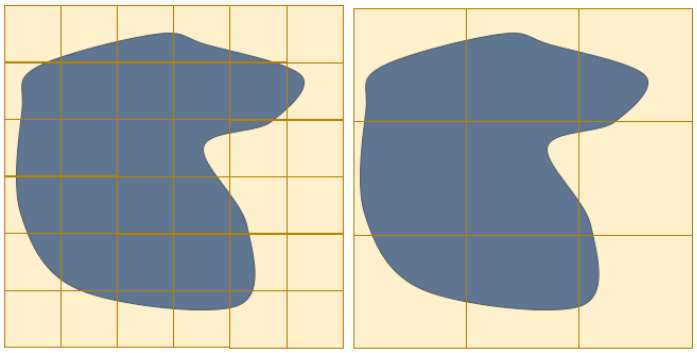

FIGURE 29 – Example of two different-sized patchings of the same underlying image (in blue). If the patches are too small, as on the left, then the unique low-frequency coefficient is repeated multiple times (around 9 times) unnecessarily. If the patches are too large, as on the right, then there are discontinuities in each patch, resulting in several non-zero coefficients to code per patch, and a loss of compression rate.

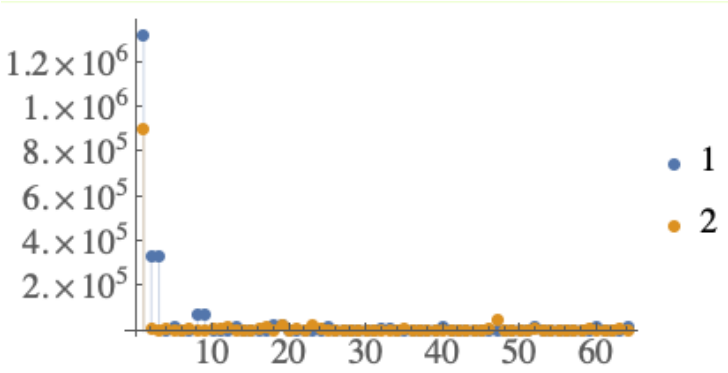

FIGURE 30 – The squared magnitude values of the 64 coefficients in the decomposition of the two images in Figure 28: "1" corresponds to the image with a discontinuity, and "2" is for the random image.

FIGURE 31 – Left: JPEG compression rate is 0.2 bpp. Right: JPEG compression rate is 0.5 bpp.

Length Code, which encodes the lengths of segments with the same value, I for 1s and Z for 0s, using entropy coding. Finally, it ends with a special word *end-of-block*, indicating that there are only 0s.

— ***The quantized value of these coefficients***.

There are libraries that allow you to compress in JPEG format by specifying a quality factor, such as `ImageMagick` or `cjpeg` on Linux [92]. Therefore, you need to calibrate to establish the correspondence between this factor and the number of bits per pixel. Figure 31 shows two levels of compression (0.2bpp and 0.5bpp) of the same 512×512 image initially encoded at 8 bits per pixel (bpp).

Contrary to what you might think at first, ***you can restore the image with good visual quality even with fewer than 1 bit per pixel***.

Why is this possible? The fundamental reason is that we have used spatial redundancy through the use of the orthogonal transform. If we had considered pixels as independent of each other, we could only binaryize the image, which would correspond to

92. https://imagemagick.org/script/convert.php, https://www.unix.com/man-page/linux/1/cjpeg/

encoding each pixel with either the value 0 or 1 (1bpp) [93]. The gain from the transform is at least a factor of 2. Note that if you zoom in on the image, you observe oscillations (*Gibbs phenomenon* [94]) because we have removed high frequencies.

Now, ***changing the compression rate is done by changing the quantization step***. However, the basis of cosine products remains the same, meaning that L is always 8 pixels. In audio, we were more subtle by using a weighted norm to adapt to the perception of the ear. Therefore, if we switch from an equal step Δ_1 to a larger step $\Delta_2 > \Delta_1$, all coefficients smaller than Δ_2 that were non-zero with Δ_1 are set to zero, degrading the image reconstruction quality by destroying high frequencies, resulting in ***blocking artifacts***. It's a bit better for the boat's mats because the intensity is greater, so there are more bits to represent the signal in those cases. The effect is much more visible in image regions where there are not many structures, such as the sky.

How can we achieve higher compression rates without degrading the image quality? We need to use different scales and exploit redundancies at all levels. This is done ***using orthonormal wavelet bases***.

9.5 Using Wavelets: JPEG2000 Standard

Note: For the introduction to 1D and 2D wavelet bases, it wouldn't have been appropriate to copy-paste from the 2021 course. You can refer, for example, to sections 5.3, 6.3, 8, and 9.3. You can also find additional information in the 2020 course.

Just for reference, wavelet decomposition is performed using a fast algorithm in filter banks with complexity $O(N)$, faster than the FFT. ***The only non-zero coefficients are those for which the wavelet, localized both in frequency and in space while respecting the Heisenberg inequality, signals the presence of a discontinuity***, as can be seen in Figure 32. Then, we proceed with quantization of the non-zero coefficients somewhat similar to JPEG. The result in Figure 33 demonstrates the efficiency of wavelet decomposition for a compression rate of 0.2 bpp by comparing JPEG and JPEG2000 images [95].

93. See the 2021 Course Figure 57.
94. See the 2021 Course, footnote in Section 8.3
95. Note: The images were obtained using the convert/ImageMagick tool on Linux/Mac by adjusting the "-define jp2:rate=x" option in addition to the "jp2:nomct" and "jp2:numrlvls=4" options. Then, as *pdflatex* does not understand the "jp2" format, I converted the files to "png" format and verified that the rendering was the same.

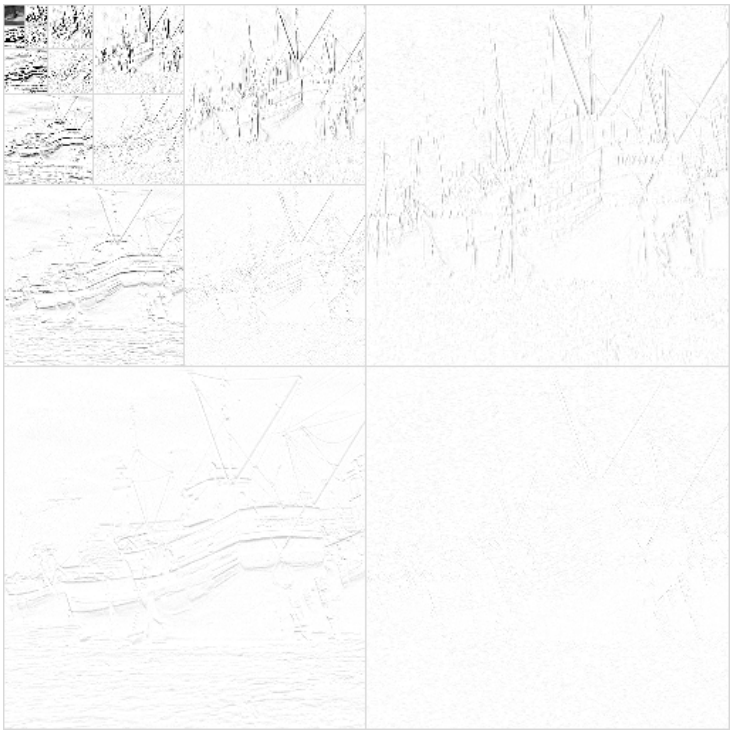

FIGURE 32 – Example of wavelet decomposition of a boat image. The colormap has been reversed to highlight the sparse non-zero coefficients, which are few in number and localized at hue discontinuities.

FIGURE 33 – Examples showing the difference in image quality when using the same compression rate of 0.2 bpp, either with the JPEG standard on the left or JPEG2000 on the right. It can be observed that at higher compression rates, the use of wavelets allows capturing redundancies at all scales to achieve better image reconstruction.

FIGURE 34 – Images in JPEG2000 format with a compression rate of 0.05 bpp.

If we push the compression by an additional factor of 4 (Figure 34) to achieve a rate of 0.05 bpp, some distortions become visible. These are not block artifacts like in JPEG but rather a result of the lack of high-frequency coefficients, resembling the Gibbs phenomenon present in JPEG. However, these effects are less pronounced in JPEG2000.

9.6 Confrontation of Theory with a Real Case

After showing examples of JPEG and JPEG2000 compression, let's examine their alignment with theory, particularly the expression of distortion D in terms of the bits per pixel $\bar{R}$ from Theorem 22. To do this, we introduce a quality indicator, the PSNR (*peak signal-to-noise ratio*). For images containing N pixels with values encoded in 8 bits (maximum pixel value equals 255), we have

$$\boxed{PSNR(\bar{R}, \overline{\mathbb{H}}_d) := 10 \log_{10} \frac{255^2}{D(\bar{R}, \overline{\mathbb{H}}_d)/N}} \tag{295}$$

where $\overline{\mathbb{H}}_d$ is given to reflect the dependence on signal properties. Thus, we expect ***a linear relationship*** of the form

$$PSNR(\bar{R}, \overline{\mathbb{H}}_d) = (20 \log_{10} 2)\bar{R} + C(\overline{\mathbb{H}}_d) \tag{296}$$

In Figure 35, we show the evolution of PSNR as a function of $\bar{R}$ for the boat image compressed with JPEG and JPEG2000 formats. What can be observed is indeed a linear behavior [96] for $\bar{R} > 1$. However, as shown in the right figure, ***for $\bar{R} < 1$, the behavior is more linear in*** $\log_2(\bar{R})$. Therefore, it is necessary to amend the theory and understand the origin of this image quality loss phenomenon.

Why is it important to consider the $\bar{R} < 1$ region? In practice, this is the region of interest, as explained following Shannon's theorem (Th. 16). The reason lies in the distribution of wavelet coefficients, as shown in Figure 36. What is noticeable is that these distributions are sharply peaked at 0, which challenges ***the high-resolution assumption*** that requires a constant distribution over a scale Δ. The error here is due to quantization within the bin $[-\Delta/2, \Delta/2]$.

9.7 Behavior When $\bar{R} < 1$

Let's establish a connection with the 2021 course. Consider a signal to which we apply quantization on the decomposition coefficients in an orthonormal basis:

$$\hat{x} = \sum_m Q(\langle x, g_m \rangle)\ g_m \tag{297}$$

The distortion is given by:

$$\begin{aligned} D &= \sum_m |\langle x, g_m \rangle - Q(\langle x, g_m \rangle)|^2 \\ &= \sum_{|\langle x, g_m \rangle| \leq \frac{\Delta}{2}} |\langle x, g_m \rangle|^2 + \sum_{|\langle x, g_m \rangle| > \frac{\Delta}{2}} |\langle x, g_m \rangle - Q(\langle x, g_m \rangle)|^2 \end{aligned} \tag{298}$$

96. Note: At the time these notes were written, these curves are still preliminary because whether using `ImageMagick/convert` or `cjpeg`, the slope is not as expected with $20 \log_{10} 2$ but approximately half of that for $\bar{R} > 1$

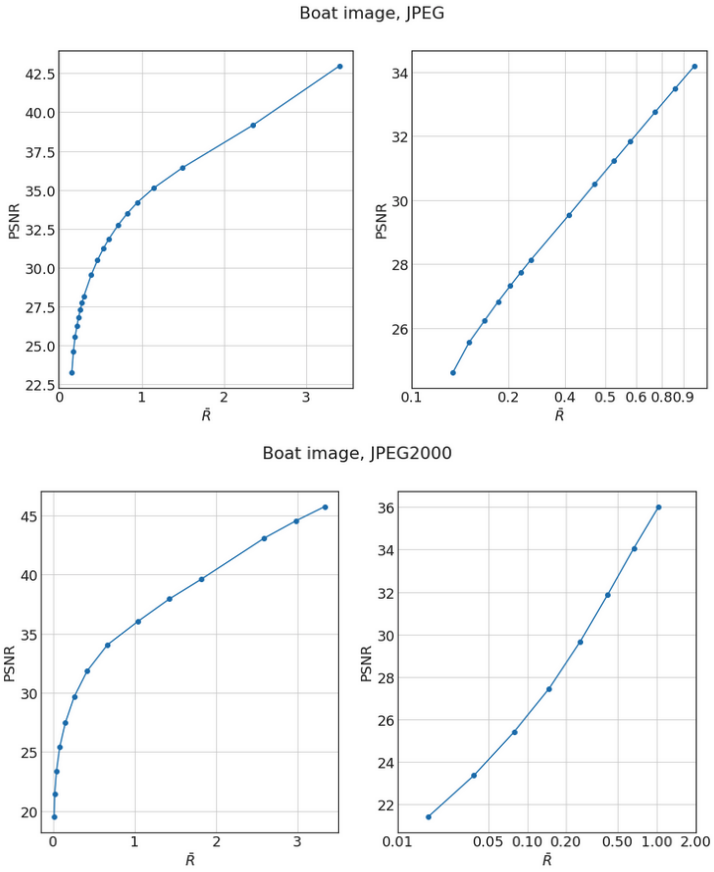

FIGURE 35 – Evolution of PSNR as a function of bits per pixel for the boat image compressed in JPEG (top) and JPEG2000 (bottom). On the right, the scale of $\bar{R}$ is in $\log_2$, while on the left, it is linear.

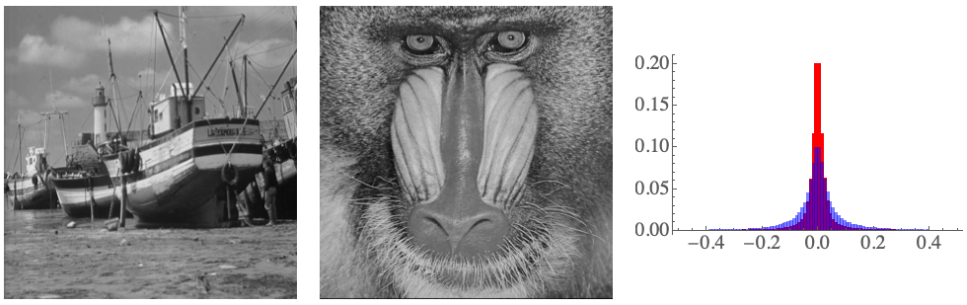

FIGURE 36 – On the left and in the middle are two images. On the right, in red, is the normalized histogram of wavelet coefficients (details only) for the boat image, and in blue for the one showing more textures, the monkey image.

Let M be the number of coefficients whose magnitude is greater than $\Delta/2$. Following an argument from 2021, the approximation that only keeps M coefficients of the decomposition [97]:

$$\tilde{x}_M = \sum_{m \in I(M)} \langle x, g_m \rangle \; g_m \tag{299}$$

generates an error:

$$\|x - \tilde{x}\|^2 = \sum_{m \notin I(M)} |\langle x, g_m \rangle|^2 \tag{300}$$

To minimize this error, the absolute values of scalar products should be minimized for $m \notin I(M)$ and maximized for $m \in I(M)$. In particular, a threshold $T(M)$ is set, which keeps only the M largest coefficients to define $I(M)$:

$$I(M) = \{m \; / \; |\langle x, g_m \rangle| > T(M)\} \quad \text{and} \quad T(M) \text{ such that } |I(M)| = M \tag{301}$$

97. NDJE: $I(M)$ is a set that, in Fourier, would be, for example, the M low-frequency coefficients (linear approach: M does not depend on the signal). In Wavelets, this would be the set of coefficients whose magnitude is greater than a certain threshold (non-linear effect: because the threshold depends on the signal itself).

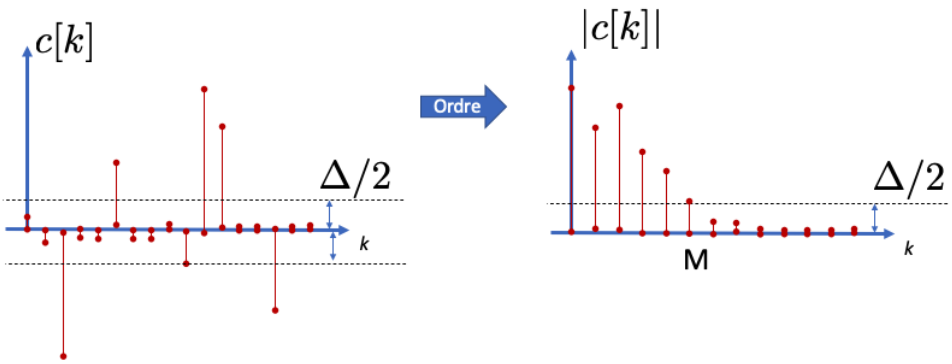

FIGURE 37 – Illustration of sorting coefficients $c[k]$ and the relationship between the number of coefficients M and the threshold $\Delta/2$.

By setting the threshold to $T(M) = \Delta/2$, we not only fix the number M of retained coefficients but also the level of approximation error. Let x_M be the approximation of x such that:

$$x_M = \sum_{|\langle x, g_m \rangle| \geq \frac{\Delta}{2}} Q(\langle x, g_m \rangle)\ g_m \tag{302}$$

(the sum contains only M terms). Then, the distortion D can be bounded as follows:

$$\boxed{\|x - x_M\|^2 \leq D \leq \|x - x_M\|^2 + M\frac{\Delta^2}{4}} \tag{303}$$

Indeed, the error $\|x - x_M\|^2$ is given by the power of coefficients for which $|\langle x, g_m \rangle| \leq \Delta/2$, and an upper bound of $|\langle x, g_m \rangle - Q(\langle x, g_m \rangle)|$ is $\Delta/2$.

To obtain a relationship of the type $D(\bar{R})$, we need to relate the number of coefficients M to $\bar{R}$ and $\|x - x_M\|$ to M. To achieve this, we need an assumption about the scalar products, guided by the profiles of their distributions in practice.

Theorem 24

If we order the coefficients of the decomposition of x in the basis in descending order (Fig. 37):

$$c[k] = |\langle x, g_k \rangle|, \quad c[k] \geq c[k+1] \tag{304}$$

and assume that the decrease is of the form:

$$|c[k]| = ck^{-\alpha} \quad \alpha > 1/2, \ c > 0 \tag{305}$$

then the distortion behaves as follows:

$$\boxed{D(R) \approx \left(\frac{R}{(\alpha - 1) \log_2 R + O(\log N)} \right)^{1-2\alpha}} \tag{306}$$

Proof 24. We will outline the proof. First, let's calculate $\|x - x_M\|^2$. According to the order of coefficients and the fact that x_M retains only the first M coefficients (the largest ones):

$$\|x - x_M\|^2 = \sum_{k=M+1}^{N} |c[k]|^2 \approx \sum_{k=M}^{N} c^2 k^{-2\alpha} \approx c^2 M^{1-2\alpha} \tag{307}$$

Now, we need to relate M to R, the number of bits. To do this, we distinguish in R the contribution R_0 that encodes the position of non-zero coefficients, and the contribution R_1 that encodes their values. The non-zero coefficients satisfy the decreasing relation, so a priori $c[k] \in]-c, c[$, an interval divided into boxes of width Δ. Therefore, the number of quantization boxes is:

$$K \approx \frac{2c}{\Delta} \tag{308}$$

and $\log_2 K$ gives the number of bits for a coarse coding of a coefficient. Thus:

$$R_1 = M \log_2(2c/\Delta) \tag{309}$$

Now, the relationship between M and Δ is roughly given by:

$$c[M] = \frac{\Delta}{2} \approx cM^{-\alpha} \Rightarrow R_1 \approx \alpha M \log_2 M \tag{310}$$

To obtain R_0, we need to calculate the entropy of a random variable that takes the value 1 with probability p and the value 0 with probability $1 - p$. This is the entropy of a Bernoulli distribution:

$$\mathbb{H}_B = -(1-p)\log_2(1-p) - p\log_2(p) \tag{311}$$

Now, $p = M/N$ is the frequency of occurrence of non-zero coefficients among the N coefficients of the decomposition. Thus [98], $R_0 = N\mathbb{H}_B$. Finally, in the case of a high compression rate $M/N \ll 1$, then:

$$R/M \approx (\alpha - 1)\log_2 M + \log_2 N + o(\log_2 N) \tag{312}$$

Inverting the relation gives M as a function of R, which yields:

$$M \approx \frac{R}{(\alpha - 1)\log_2 R + \log_2 N + o(\log_2 N)} \tag{313}$$

And since $M(\Delta/2)^2 \approx c^2 M^{1-2\alpha}$, just like $\|x - x_M\|^2$, then $D(R)$ indeed follows a law:

$$D(R) \approx \left(\frac{R}{(\alpha - 1)\log_2 R + \log_2 N + o(\log_2 N)} \right)^{1-2\alpha} \tag{314}$$

98. Note: There are as many coefficients as samples of the signal, which is N.

■

What will be crucial at high compression rates ($\bar{R}$ small) is indeed the number of non-zero coefficients M, as the number of bits is *roughly proportional* to it, just like the error, which should be as small as possible. ***And ultimately, once we know how to perform efficient coding, the most challenging part is finding the basis that best compresses the information.*** This optimal basis eliminates data redundancy by finding regularity patterns, which was the subject of the 2021 course, navigating the triangle: ***Regularity, Approximation, Sparsity***.

10. Epilogue

Finally, a century after the publication of Fisher's article, we are still operating within the framework of his program to find a parametric model of the distribution that best reflects the observations. This lies at the heart of Machine Learning, and in a way, neural networks are parameterized systems designed to ultimately maximize likelihood, all while performing gradient descent to determine the parameters. Of course, what has fundamentally changed over a century is the complexity of the models.

This probabilistic aspect, according to S. Mallat, is essential for understanding the perplexing results of neural networks. Let's say that the probabilistic viewpoint, as opposed to the deterministic one, is indeed the conceptual framework in which we hope to understand the statistical properties. After all, if these networks can estimate these parameterized distributions in very high dimensions, it means they are capturing highly relevant information contained in the observations. This simplification is undoubtedly related to the concentration of probability, which is a reflection of the fact that the observations are not arbitrary; they "live" in Shannon's typical sets.

Between the 1950s and 2000s, Information Theory fully exploited the use of entropy, especially. However, the problem that emerged was that, except for some simple cases involving Gaussians for brevity, we didn't know how to calculate the entropy of systems. It's from the 2000s onwards, and increasingly with neural networks with a colossal number of

parameters, that we are much better able to grasp these typical sets of probability distributions. So, while the theoretical framework remains the same, the (pleasant) surprise is that we are getting closer to these typical sets, which allows us to tackle new problems, such as delving into approximation errors, for example, using harmonic analysis.

Now, Shannon's viewpoint, which ignores any form of parameterization to focus solely on the intrinsic information of observations, and Fisher's viewpoint, are ultimately in a natural relationship. In statistical physics, particularly when dealing with observables from which moments are drawn (broadly speaking, averages), if we want to infer distributions by maximizing entropy (Jaynes' Maximum Entropy Principle), this leads to modeling by the exponential family, which is a formalism of Fisher-parameterized probabilities. Among the questions that arise, for example, is whether we can do better in the context of data compression with neural networks. Perhaps we will see new standards emerging in the near future.