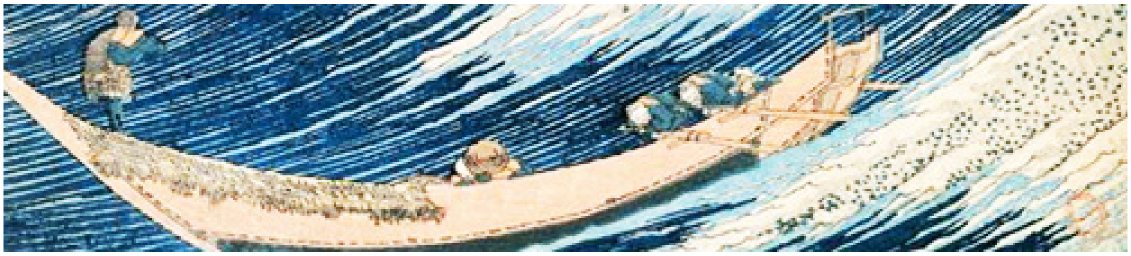
Mathematical Foundations of Data Sciences



Gabriel Peyré
CNRS & DMA
École Normale Supérieure
gabriel.peyre@ens.fr
https://mathematical-tours.github.io
www.numerical-tours.com

September 11, 2024

Chapter 1

Shannon Coding Theory

1.1 Source Coding

Uniform coding. We consider an alphabet $(s_1, \ldots, s_K)$ of K symbols. For instance, if one samples and quantizes a bounded signal $0 \leqslant f_0 < 1$ using a step size $1/K$, then one can consider $s_k = k$ to be integer symbols. For text, these symbols include the letter plus extra punctuation symbols and blank. It is of course possible to code a sequence of such symbols using a uniform code (e.g. using the base 2 expansion) with $\lceil \log_2(K) \rceil$ bit per symbols. For instance if $K = 4$ and the symbols are $\{0, 1, 2, 3\}$, then the code words are $(c_0 = 00, c_1 = 01, c_2 = 10, c_3 = 11)$.

This uniform coding strategy is however extremely inefficient if the symbols are not uniformly distributed (i.e. if some symbols are more frequent than others, which is likely to be the case). We aim to design better codes.

Prefix coding. A code $c_k = c(s_k)$ associate to each symbol s_k a code word $c_k \in \{0,1\}^{\mathbb{N}}$ with a varying length $|c_k| \in \mathbb{N}^*$. A prefix code $c_k = c(s_k)$ is such that no word c_k is the beginning of another word c'_k. This is equivalent to being able to embed the $(c_k)_k$ as leaves of a binary tree T, with the code being the output of a traversal from root to leaves (with a convention that goes to a left (resp. right) child output a 0 (resp. a 1). We denote $c = \text{Leaves}(T)$ such prefix property.

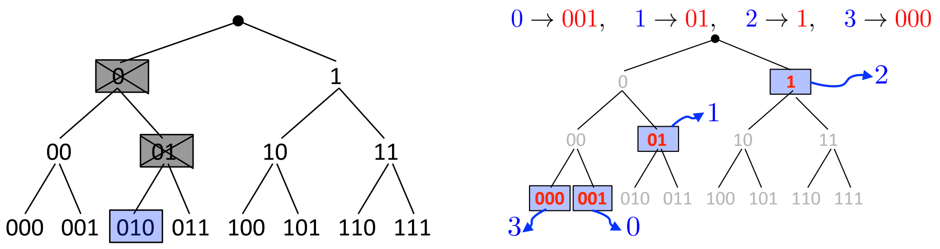


Figure 1.1: Left: complete tree of all codes of length 3; right: example of prefix code.

This tree-based representation is useful to decode a binary stream by simply performing tree traversal. One follows the tree, from top to bottom, and outputs a symbol each time a leaf is reached (and then restarts at the top).

Probabilistic modeling. We aim to design the most possible compact code c_k. We assume at our disposal some probability distribution over this alphabet, which is just a histogram $p = (p_1, \ldots, p_K) \in \mathbb{R}_+^K$ in the simplex, i.e. $\sum_k p_k = 1$. In practice, this probability is usually the empirical probability of the appearance of the symbols x_k in the data to be coded.

The entropy of such an histogram is

$$H(p) \overset{\text{def.}}{=} -\sum_k p_k \log_2(p_k)$$

with the convention $0 \log_2(0) = 0$.

Denoting $h(u) = -u \log_2(u)$, $h'(u) \propto -\log(u) - 1$, $h''(u) \propto -1/u < 0$ so that H is strictly concave. The definition of the entropy extends to continuous density $f(x)$ for x on some measure space with reference measure $\mathrm{d}x$ (e.g. Lebesgue on $\mathbb{R}^d$) by setting $H(f) \overset{\text{def.}}{=} -\int f(x) \log(f(x)) \mathrm{d}x$.

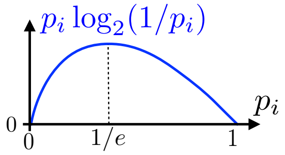
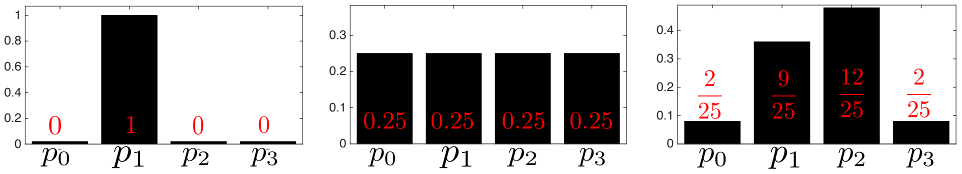


Figure 1.2:



$H(p) = 0$ $\quad$ $H(p) = \log_2(2) = 1$ $\quad$ $H(p) = 1.62$

Figure 1.3: Three examples of probability distributions with corresponding entropies.

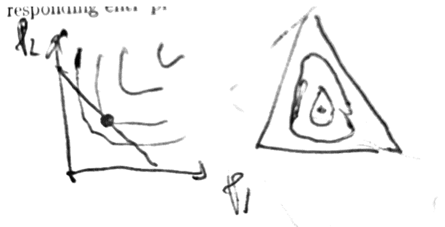


Figure 1.4: Linked extrema for the entropy.

Lemma 1. *One has*

$$0 = H(\delta_i) \leqslant H(p) \leqslant H(1/K) = \log_2(K)$$

where δ_i is the Dirac histogram distribution at i.

Proof. First one notes that $-u \log_2(u) \geqslant 0$ for $u \in [0,1]$ (see figure above), so that $H \geqslant 0$. Then we show the following inequality, for any histogram q

$$H(p) \leqslant -\sum_i p_i \log(q_i) \quad \Leftrightarrow \quad \sum_i p_i \log(q_i/p_i) \leqslant 0.$$

This follows from $\log(u) \leqslant u - 1$, since

$$\sum_i p_i \log(q_i/p_i) \leqslant \sum_i p_i(q_i/p_i - 1) = \sum_i q_i - \sum_i p_i = 0.$$

Applying this inequality to $q_i = 1/K$ gives

$$H(p) \leqslant -\sum_i p_i \log(1/K) = \log(1/K).$$

□

The quantity $KL(p|q) \overset{\text{def.}}{=} \sum_i p_i \log(p_i/q_i) \geqslant 0$ is called the Kulback-Leibler divergence, or relative entropy between p and q. It is positive and zero if and only if $p = q$, so it is a "distance like" (beware it is nonsymmetric and does not satisfy the triangular inequality). This is a quantity of fundamental importance in statistics (because it arises when computing maximum likelihood estimator) and also in machine learning (because it is the most common loss function used to perform classification).

1.2 Shannon Source Coding Theorem

Assuming that $(p_k)_k$ is the empirical probability of the appearance of the symbols x_k in the data to be coded, the average symbol length associated with some code c is

$$L(c) \overset{\text{def.}}{=} \sum_k p_k |c_k|.$$

The goal is to design the best possible c so that $L(c)$ is as small as possible. Shannon's theorem of entropic coding, proved below, gives both lower and upper bounds for this question.

Theorem 1. *(i) If $c = Leaves(T)$ for some tree T, then*

$$L(c) \geqslant H(p).$$

(ii) Conversely, there exists a code c with $c = Leaves(T)$ such that

$$L(c) \leqslant H(p) + 1.$$

Before proving this theorem, we prove Kraft inequality, which describes the set of prefix codes using an inequality.

Lemma 2 (Kraft inequality)**.** *(i) For a code c, if there exists a tree T such that $c = Leaves(T)$ then*

$$\sum_k 2^{-|c_k|} \leqslant 1. \tag{1.1}$$

(ii) Conversely, if $(\ell_k)_k$ are such that

$$\sum_k 2^{-\ell_k} \leqslant 1 \tag{1.2}$$

then there exists a code $c = Leaves(T)$ such that $|c_k| = \ell_k$.

Proof. $\Rightarrow$ We suppose $c = \text{Leaves}(T)$. We denote $m = \max_k |c_k|$ and consider the full binary tree. Bellow each c_k, one has a sub-tree of height $m - |c_k|$, see Figure 1.5, left. This sub-tree has $2^{m-|c_k|}$ leaves. Since all these sub-trees do not overlap, the total number of leaf do not exceed the total number of leaves 2^m of the full binary tree, hence

$$\sum_k 2^{m-|c_k|} \leqslant 2^m,$$

hence (1.1).

$\Leftarrow$ Conversely, we assume (1.2) holds. Without loss of generality, we assume that $\ell_1 \leqslant \ldots \leqslant \ell_K =: m$. We first construct trees of height $2^{m-\ell_k}$. We then put then side by side from left to right. Since $\sum_k 2^{m-\ell_k} \leqslant 2^m$, these sub-trees occupy a space of less than the number of leaf in the full binary tree of depth m. The proof follows from the fact (which can be seen on Figure 1.5, right) that because the ℓ_k are increasing, the height of the trees are decaying, so that this ordering of trees form a sub-tree of this full binary tree of depth m. □

Shannon theorem. First, we consider the following optimization problem

$$\min_{\ell=(\ell_k)_k} \left\{ f(\ell) \overset{\text{def.}}{=} \sum_k \ell_k p_k \; ; \; g(\ell) \overset{\text{def.}}{=} \sum_k 2^{-\ell_k} \leqslant 1 \right\}. \tag{1.3}$$

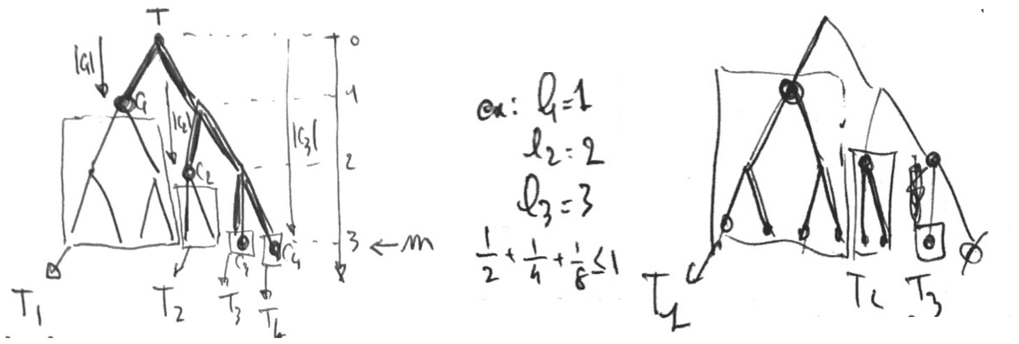


Figure 1.5: Left: full binary tree obtained by completing the tree associated to the code $(c_1 = 0, c_2 = 10, c_3 = 110, c_4 = 111)$. Right: packing sub-trees associated to code length to form the left part of the full tree.

We fist show that at an optimal $\ell^\star$, the constraint is saturated, i.g. $g(\ell^\star) = 1$. Indeed, if $g(\ell^\star) = 2^{-u} < 1$, with $u > 0$, we define $\ell'_k \overset{\text{def.}}{=} \ell^\star_k - u$, which satisfies $g(\ell') = 1$ and also $f(\ell') = \sum_k (\ell^\star_k - u)p_k < f(\ell^\star)$, which is a contradiction. So we can restrict in (1.3) the constraint to $g(\ell) = 1$ and apply the linked extrema theorem, which shows that necessarily, there exists $\lambda \in \mathbb{R}$ with $\nabla f(\ell^\star) = \lambda \nabla g(\ell^\star)$, i.e. $(p_k)_k = -\lambda \ln(2)(2^{-\ell^\star_k})_k$. Since

$$1 = \sum_k p_k = -\lambda \ln(2) \sum_k 2^{-\ell_k} = -\lambda \ln(2)$$

we deduce that $\ell^\star_k = -\log(p_k)$.

(i) If $c = \text{Leave}(T)$, the by Kraft inequality (1.1), necessarily $\ell_k = |c_k|$ satisfy the constraints of (1.3), and thus $H(p) = f(\ell^\star) \leqslant f(\ell) = L(\ell)$.

(ii) We define $\ell_k \overset{\text{def.}}{=} \lceil -\log_2(p_k) \rceil \in \mathbb{N}^*$. Then $\sum_k 2^{-\ell_k} \leqslant \sum_k 2^{\log_2(p_k)} = 1$, so that these lengths satisfy (1.2). Thanks to Proposition 2 (ii), there thus exists a prefix code c with $|c_k| = \lceil -\log_2(p_k) \rceil$. Furthermore

$$L(c) = \sum_k p_k \lceil -\log_2(p_k) \rceil \leqslant \sum_k p_k(-\log_2(p_k) + 1) = H(p) + 1.$$

□

Note that this proof is constructing, i.e. it gives an algorithm that constructs an almost optimal c, and this code is often called the Shannon-Fano code. It is usually a good code, although it is not necessarily the optimal code with the smallest $L(c)$. Such an optimal code can easily be computed in almost linear time (only sorting of the probability is needed, so it is $K(\log(K))$) by Huffman's dynamic programming algorithm (invented in 1952). The proof of the correctness of this algorithm is however a bit tedious. Figure 1.6 shows an example of the application of this algorithm.

`Associated code: coding/test_text.m`

1.3 Probabilistic modelling

In practice, such entropic coding, although optimal, is not very efficient when one of the symbols has a large probability p_k. This is because $2^{-p_k} \ll 1$, but one cannot allocate a fractional number of bits. This is why $L(c)$ can be as large as $H(p) + 1$. A simple workaround is to artificially increase the size of the alphabet from K to K^r by grouping together sets of r consecutive symbols and then performing Shannon coding over this new, larger alphabet.

Under a proper probabilistic model, and when the number n of coded symbols goes to $+\infty$, one can show that this reduces the gap to $H(p) + 1/r$. This is not in contradiction with Shannon's theorem because the code is not a prefix code over the initial alphabet.

We give here an informal argument. If we assume the symbols are generated randomly from the probability vector $p = (p_k)_k$, then as $n \to +\infty$, the (empirical) entropy of the observed symbols tends to $H(p)$. Then, if we assume independence of the symbols, the probability $P_{k_1,\ldots,k_r}$ of groups of r consecutive tuples is

$$P_{k_1,\ldots,k_r} = p_{k_1} p_{k_2} \ldots p_{k_r}$$

i.e., $P = p \otimes \ldots \otimes p$ is the r-fold tensor product. Then, the entropy is

$$H(p \otimes \ldots \otimes p) = -\sum p_{k_1} p_{k_2} \ldots p_{k_r} \log(p_{k_1} p_{k_2} \ldots p_{k_r}) = rH(p).$$

In fact, the independence assumption is not needed in the reasoning; one can show that, in general,

$$H(P) \leqslant rH(p),$$

where equality holds only if the tokens are independent. For large N, if one applies Shannon coding, one sees that the code length is reduced from $N(H(p) + 1)$ to

$$\frac{N}{r}(H(P) + 1) = N\left(H(p) + \frac{1}{r}\right),$$

hence, when r is large, one can get arbitrarily close to $H(p)$.

A useful byproduct of this reasoning is that if the tokens are not independent, the improvement can be much higher. This idea is also reflected in the following.

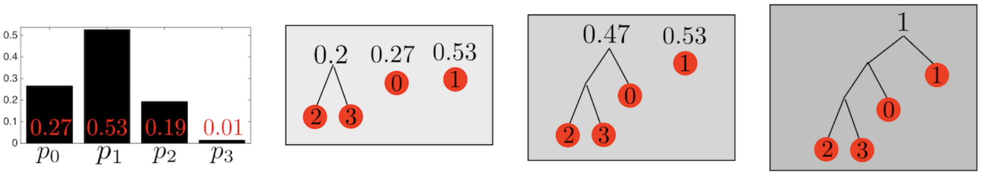


Figure 1.6: Huffman coding algorithm in action.

Impact of using the wrong probability distribution Note that if one chooses a different probability q and uses it to code the sequence, one necessarily obtains a worse average coding length, and this is reflected by the positivity of the so-called relative entropy (beware that it is a convex function while the entropy is concave), which is often called the Kulback-Leibler divergence

$$\mathrm{KL}(p|q) = -\sum_k p_k \log q_k - H(p) = \sum_k p_k \log \frac{p_k}{q_k} \geqslant 0.$$

This KL divergence is similar to a distance in the sense that $\mathrm{KL}(p|q) = 0$ if and only if $p = q$ (note however that KL is not symmetric and does not satisfy the triangular inequality). It also has the remarkable property that it is jointly convex in (p, q). It is of paramount importance to compare probability distributions and measures and form the basis of the fields of information theory and information geometry.

1.4 Doing better leveraging non-independance

One can wonder if it is possible to go below the entropy bound. By Shannon's theorem, it is not possible if one can only code in sequential order the symbols themselves. From a statistical perspective, it is as if the symbols were considered to be independent. If there is some redundancy in the sequence of symbols (for instance if they are discretization of a smooth function, so that two consecutive symbols are likely to be equal), it is possible to re-transform (in a bijective way) the sequence to make them “more independent”. A simple illustration of this idea is given in Figure 1.7, where one computes successive difference of a 1D sequence of symbols (beware to also retain the initial symbol to be able to do the decoding.

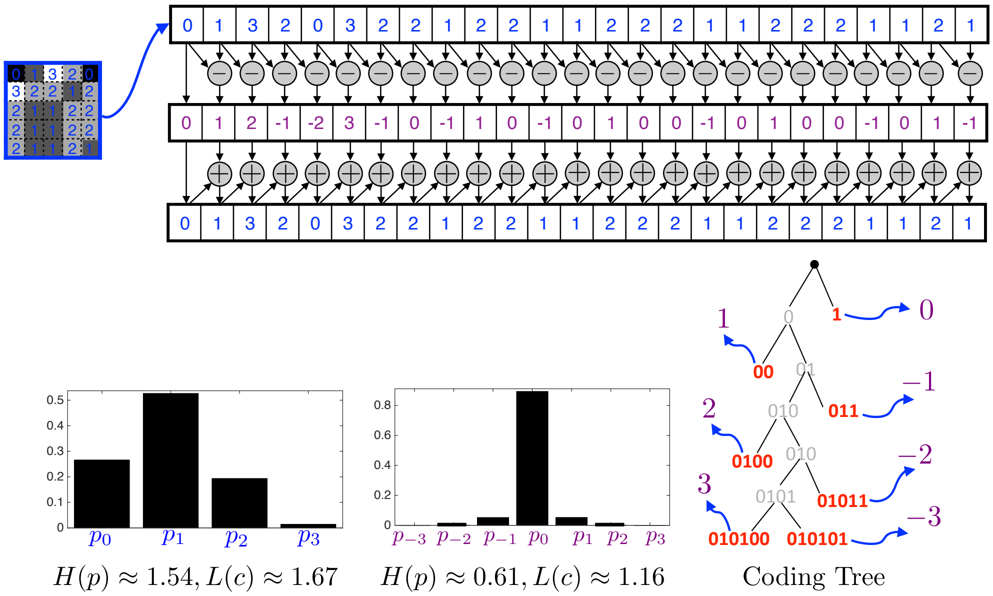


Figure 1.7: Top: retransformation by successive differences. Bottom: comparison of histograms of pixel values and differences, and a code tree for these differences.

Another, more systematic way to leverage such temporal redundancy is by performing run-length coding, which operates by grouping a sequence of similar symbols and thus coding first a symbol and then the length of the associated group (which is coded entropically). If the sequence is generated by a Markov chain, this method can be shown to asymptotically reach the Shannon bound where now the entropy is the entropy associated to the distribution of the Markov chain on infinite sequences (which can be computed as the limit of the entropy for finite sequences).

Bibliography

[1] Francis Bach. Breaking the curse of dimensionality with convex neural networks. *The Journal of Machine Learning Research*, 18(1):629–681, 2017.

[2] Francis Bach. Learning theory from first principles. 2021.

[3] Andrew R Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information theory*, 39(3):930–945, 1993.

[4] Amir Beck. *Introduction to Nonlinear Optimization: Theory, Algorithms, and Applications with MATLAB.* SIAM, 2014.

[5] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

[6] Stephen Boyd and Lieven Vandenberghe. *Convex optimization.* Cambridge university press, 2004.

[7] E. Candès and D. Donoho. New tight frames of curvelets and optimal representations of objects with piecewise C^2 singularities. *Commun. on Pure and Appl. Math.*, 57(2):219–266, 2004.

[8] E. J. Candès, L. Demanet, D. L. Donoho, and L. Ying. Fast discrete curvelet transforms. *SIAM Multiscale Modeling and Simulation*, 5:861–899, 2005.

[9] A. Chambolle. An algorithm for total variation minimization and applications. *J. Math. Imaging Vis.*, 20:89–97, 2004.

[10] Antonin Chambolle, Vicent Caselles, Daniel Cremers, Matteo Novaga, and Thomas Pock. An introduction to total variation for image analysis. *Theoretical foundations and numerical methods for sparse recovery*, 9(263-340):227, 2010.

[11] Antonin Chambolle and Thomas Pock. An introduction to continuous optimization for imaging. *Acta Numerica*, 25:161–319, 2016.

[12] S.S. Chen, D.L. Donoho, and M.A. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20(1):33–61, 1999.

[13] Philippe G Ciarlet. Introduction à l'analyse numérique matricielle et à l'optimisation. 1982.

[14] P. L. Combettes and V. R. Wajs. Signal recovery by proximal forward-backward splitting. *SIAM Multiscale Modeling and Simulation*, 4(4), 2005.

[15] George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

[16] I. Daubechies, M. Defrise, and C. De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Commun. on Pure and Appl. Math.*, 57:1413–1541, 2004.

[17] D. Donoho and I. Johnstone. Ideal spatial adaptation via wavelet shrinkage. *Biometrika*, 81:425–455, Dec 1994.

[18] Heinz Werner Engl, Martin Hanke, and Andreas Neubauer. *Regularization of inverse problems*, volume 375. Springer Science & Business Media, 1996.

[19] M. Figueiredo and R. Nowak. An EM Algorithm for Wavelet-Based Image Restoration. *IEEE Trans. Image Proc.*, 12(8):906–916, 2003.

[20] Simon Foucart and Holger Rauhut. *A mathematical introduction to compressive sensing*, volume 1. Birkhäuser Basel, 2013.

[21] Stephane Mallat. *A wavelet tour of signal processing: the sparse way.* Academic press, 2008.

[22] D. Mumford and J. Shah. Optimal approximation by piecewise smooth functions and associated variational problems. *Commun. on Pure and Appl. Math.*, 42:577–685, 1989.

[23] Neal Parikh, Stephen Boyd, et al. Proximal algorithms. *Foundations and Trends® in Optimization*, 1(3):127–239, 2014.

[24] Gabriel Peyré. *L'algèbre discrète de la transformée de Fourier.* Ellipses, 2004.

[25] J. Portilla, V. Strela, M.J. Wainwright, and Simoncelli E.P. Image denoising using scale mixtures of Gaussians in the wavelet domain. *IEEE Trans. Image Proc.*, 12(11):1338–1351, November 2003.

[26] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Phys. D*, 60(1-4):259–268, 1992.

[27] Otmar Scherzer, Markus Grasmair, Harald Grossauer, Markus Haltmeier, Frank Lenzen, and L Sirovich. *Variational methods in imaging.* Springer, 2009.

[28] C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423, 1948.

[29] Jean-Luc Starck, Fionn Murtagh, and Jalal Fadili. *Sparse image and signal processing: Wavelets and related geometric multiscale analysis.* Cambridge university press, 2015.