HEC MONTREAL

MASTER THESIS

Estimate The Cumulative Distribution Function with Wavelets

Author:
S. IFADIR

Supervisor:
Dr. J.F PLANTE

A supervised project submitted in fulfillment of the requirements for the degree of Master Degree

in the

HEC Montreal
DEPARTMENT OF DECISION SCIENCES

December 5, 2021

"Thanks to my solid academic training. I can write hundreds of words on statistics and mathematical topics. Proud of reaching this level humbly and looking forward to expressing more inner pleasure behind understanding the concept of formulae. Thank you all, those who insufflate the slight glow of statistical insight in all HEC and UQAM. Thank you, Jean-Francois Plante, for your trust and those by determinant remarks for the advancement of this work. I express truly my gratitude. "

Ifadir. S.

Contents

List of Figures

List of Tables

List of Abbreviations

CDF	**C**uml. **D**istr. **F**unction
HR	**H**aar **R**esolution
WDE	**W**avelet **D**ecomposition **E**rror
HDL	**H**aar **D**ecomposition **L**ist
KS	**K**olmogorov **S**mirnov
VM	**C**ramer **V**an **M**isses
SRS	**S**imple **R**andom **S**ampling

Chapter 1

Context and Tools

1.1 What

The data generated in distributed servers is large and complex. The architecture of a distributed system involves multiple computers, each is called a node, connected through a network. Any attempt to model a quantity described from data distributed across multiple computers runs into storage and synchronization problems. Beyond those technical challenges, data movement in a distributed system is the slowest between the interconnected nodes. Calculations in a distributed system should be designed to transfer as little data as possible across the nodes. Wavelet decomposition is a tool for compression that could significantly reduce the amount of data needed to achieve a modelization. In this work, we will focus on the problem of estimating the cumulative distribution function. Namely, this work aims to understand how to use wavelet compression to reduce the size of large data sets to a minimum sufficient to approximate the cumulative distribution function while keeping the generated error minimal. We will use R and a special wavelet package to write the module. We will use three goodness of fit statistics: The mean error (**ME**), the Kolmogorov Smirnov (**KS**) and Von Mises (**VM**) distances.

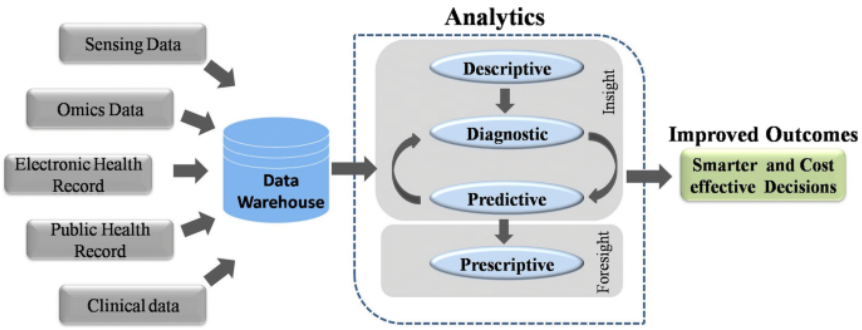

FIGURE 1.1: Different sources of Big data in healthcare [1].

Figure [1.1] depicts the different channels of data collection. Various significant sources of data are necessary before any pertinent cost-effective decisions can be made. Distributed systems are typically used as data warehouses. For large datasets, further analytics would be best performed if they could be calculated natively on that distributed system. A way to represent the distribution of a random variable is through its cumulative distribution function (**CDF**). In fact, many statistical methods can be seen as a functional of that function. The empirical CDF (**ecdf**) is mathematically easy to handle as it exhibits native regularities (monotonicity, continuity, and known hinge values) as well as appropriate asymptotic properties, notably uniform convergence. In addition, the ecdf is a non-parametric approach.

1.2 Proof of concept

Mathematical expansions are a powerful tool to get efficient approximations. We can expand a vector in the euclidian base. We can also expand a function into a polynomial base (Taylor series as an example) to approximate an irrational number. Similarly, wavelets are a basis for functions that can be used to obtain approximations that can capture abrupt jumps efficiently. To compress a **CDF**, we can expand it with the so-called **Haar wavelets**. The development of the function is made through the fast wavelet transform, which is akin to the Fast Fourier Transform. Compression consists in ignoring some coefficients in this representation, which saves memory at the cost of an approximation error. The basics of wavelets will be discussed in the second chapter.

In this work, we will explore the approximation of a cumulative distribution function by wavelets. The goal is to build an accurate approximation by reducing the number of points involved to transfer less information across nodes in a distributed system yet rebuild the **CDF** with precision. As a benchmark, we will compare the wavelet compression performance to an empirical **CDF** constructed from a sample of the data of commensurate size. The reduction of the number of points means reducing bandwidth requirement, which in our non-distributed exploration in this work corresponds to **memory usage** for storing that information. This reduction is paramount when modeling massive data issued from a distributed system. We will compare the quality of this approximation with the one made from simple random sampling to verify whether the wavelet approach is worth the effort. Figure [1.2] is the initial result that triggered our enthusiasm to explore the memory saving potential of wavelet compression for the Gaussian distribution function.

Numerically, wavelet decomposition is performed on a set of discrete points at which the function of interest is evaluated. For a given resolution, corresponding to a number of equidistant jumps, it represents the level on which we want to probe the function. By forcing a lower resolution, a coarser approximation is made.

Figure [1.2] displays the compromise between the amount of information (data size in bytes) used and the quality of the estimate. It plots the Kolmogorov Smirnov error(%) and the reduction in memory consumed (dataset size) as a function of the resolution of the wavelet. The resolution is the accuracy with which we probe the function locally with a window step function. Chapter two shows Fig.[2.2] representing geometrically two different resolutions of wavelets. Eq.[2.5] introduces its analytical formulation. Fig.[3.2] display insight about increased wavelet resolution for approximating a Gaussian **CDF**.

At low resolution, the error is important with significant memory savings. This result is understandable because we probe the function with a coarse wavelet. On the contrary, the error is almost zero at high resolution, but the memory saving is negligible as we probe with a finer wavelet. We observe good compromise around resolution **7**, where the error is about 5% with an overall memory usage of less than 5%. The area bounded by the orange rectangle on the figure is the region to consider for the resolution-error trade-off.

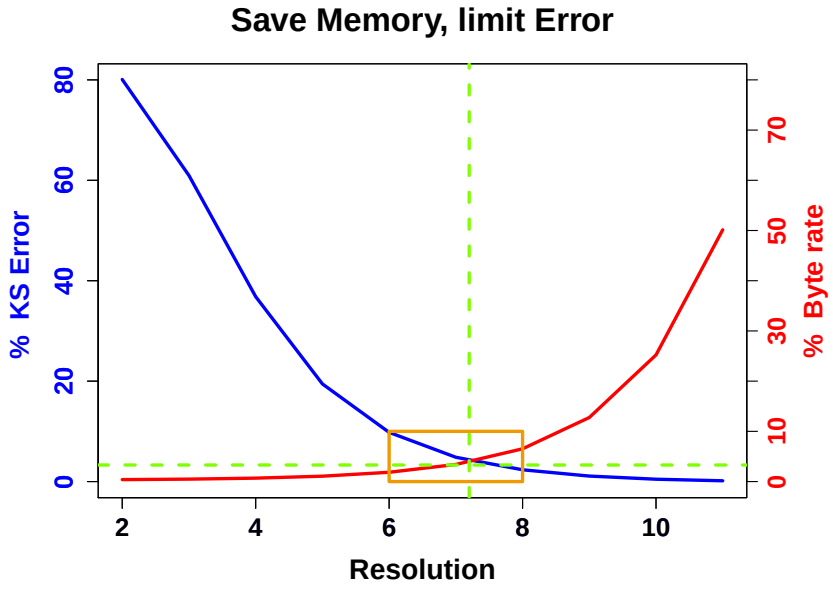

FIGURE 1.2: Example of results showing the compromise between saved memory and increased error. Blue: Error with increasing resolution. Red: The memory size of the input. Green dashed lines are for eye guidance to locate an optimal region. The orange rectangle delimits the area of optimized resolution with the lowest error. The denominator of the percentage of KS error is the maximum error (at resolution=1). For the memory size, the denominator is the actual data size of the Gaussian **CDF** curve to be approximated.

Let us examine the graph. The blue curve represents the errors or the distance of the approximation to the actual function: $\%\mathbf{KS\_Error} = \frac{\sup|\hat{\mathbf{F}}_\mathbf{n}(\mathbf{x})-\mathbf{F_0}(\mathbf{x})|}{\sup|\hat{\mathbf{F}}_\mathbf{1}(\mathbf{x})-\mathbf{F_0}(\mathbf{x})|}$, where $\hat{\mathbf{F}}_\mathbf{n}$ is the 2^n resolution wavelet, $\hat{\mathbf{F}}_\mathbf{1}$ is the lowest 1-resolution approximation (corresponding to 100% error) and $\mathbf{F_0}$ is the normal **CDF** generated by the function **pnorm** from R software. The red curve represents the size of the dataset in memory: $\%\mathbf{ByteRate} = \frac{\mathbf{size(wavelet)}}{\mathbf{size(dataset)}}$.

The error lies below 5% around resolution 7 (in fact 2^7 off a dataset size of 2^{12}). At that resolution, we intersect the red curve at a ridiculous value around 1% of the original dataset size. That is the amount of data needed to build the original function with the mentioned error. It suggests, therefore, that with $60Gb$ RAM, we can handle the equivalent of 6Tb of information! A trivial remark, when the resolution of the wavelet is equal to the number of the dataset points of the original function, the error collapse toward zero.
In short, we seek the maximum memory reduction, proportional to the number of data points we can divest, without affecting the quality of the wavelet approximation. As mentioned, The degree of precision in the case of wavelet is called a resolution. The error committed to approximate a function is linked to the wavelet resolution by which we approach that function.

We will introduce in the next section different statistical errors to assess the goodness of wavelet fit we will be using in chapter three.

1.3 Error Quantization

This paragraph introduces the different errors or distances we are using to assess the quality of the wavelet approximation. We present three distances to quantify the quality of approaching a CDF by wavelets.

In statistics, the Cramér-von Mises (**CVM**) criterion is used for judging the goodness of fit of a cumulative distribution function F_0 compared to a given empirical distribution function Fn or for comparing two empirical distributions. Kolmogorov Smirnov (**KS**) and Mean Average Error (**ME**) are also metrics used to estimate the distance from an estimated function to the actual one.

We use the following statistics because they measure the distance between distributions. We only use the metrics to assess the quality of our estimates. We will compute with Haar wavelet, the $\hat{\mathbf{F}}_\mathbf{n}$ approximations (**H**aar **D**evellopment **L**imit or **HDL**) for different resolutions to approach the true **F**. Fig.[3.2] illustrates the Haar wavelet for different resolutions). The resolution level n, we hope, small enough, yet sufficient to guarantee the proximity to the actual distribution.

- **Kolmogorov-Smirnov error (KS)** (illustration [1.3]): This statistical distance uses the supremum distance between the sample $\hat{F}_n$ and the reference probability distribution F_0:

$$D_n := \sup_{x \in \mathbb{R}} \left| \hat{F}_n(x) - F_0(x) \right| \tag{1.1}$$

  By the Glivenko-Cantelli theorem, if the sample comes from distribution F_0, then Dn converges to 0. Conversely, when $\hat{F}_n \neq F_0$, larger values of D_n are expected, hence does the error. **KS** distance measure is "extreme" in the sense that the error could deviate largely from F_0 because one of any points is out of a confidence limit.

- **Cramér-von Mises error (VM)** (illustration [1.4]): This statistical distance is the integral of the squared difference between the estimated $\hat{F}_n$ the reference F_0 distribution functions:

$$D_n := \sqrt{\int (\hat{F}_n(x) - F_0(x))^2 \, \mathrm{d}F_0(x)} \tag{1.2}$$

  This formula is nothing but Rieman-Stieltjes integral. in fact if X is random variable who's CDF is F_0 (and a density F_0'), then:

$$E(\hat{f}(X)) := \int \hat{f}(x) F_0'(x) dx = \int \hat{f}(x) \, \mathrm{d}F_0(x) \tag{1.3}$$

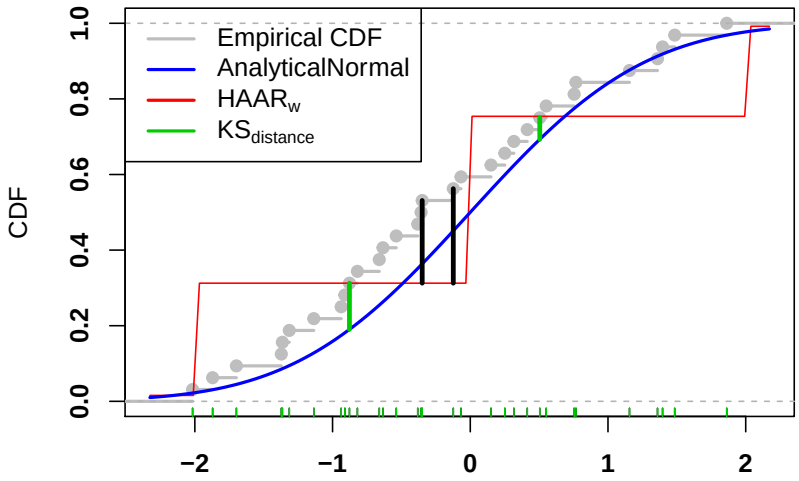

FIGURE 1.3: Illustration of the Normal CDF and its estimates considered in this project. Blue and red curves are Gaussian CDF, and it's wavelet approximation at resolution 5. Grey segments are the empirical CDF. Black lines are **surpremums** distances while approaching the ecdf from the right and left locally by the red Haar wavelet function. Green lines are **supremums** from the ecdf to theoretical cumulative.

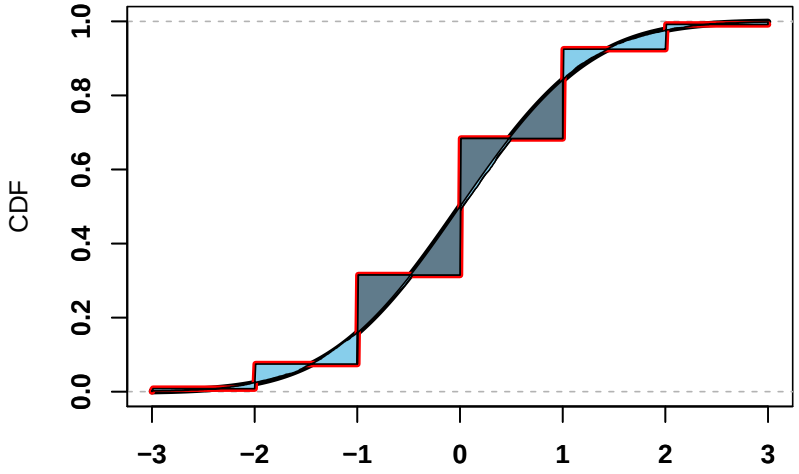

FIGURE 1.4: Illustration of the CVM approximation error. Points on the area of significant variation are given more importance in calculating the error, as shown by the slight color contrast.

Cramer Von Mises **VM** is the mean of the distance of the approximated distribution to the actual distributed weighted by the density of the concerned variable. The tail of the distribution will contribute less to the error and hence the weakness of this quantity if we are interested in a rare event or outlier quantification.

- **Mean Error distance (ME)** (illustration [1.5]):

$$D_n := mean(|\hat{F}_n(x) - F_0(x)|) \tag{1.4}$$

This distance is instead a local measure. **ME** takes the sum of difference point by point, and the error is the mean. This measure treats every observation equally and represents a good overall picture of the goodness of our approximation.

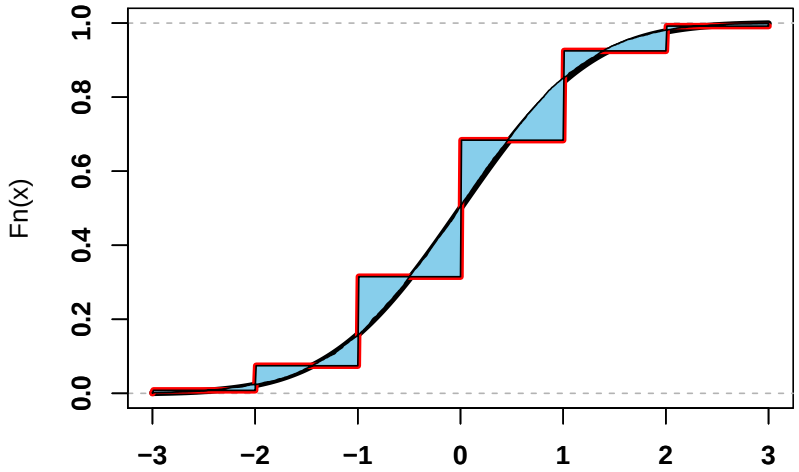

FIGURE 1.5: Error quantified by the area between the curves.

1.4 Introduction

The empirical cumulative distribution function **CDF** is used for non-parametric estimation methods. A robust and efficient data reduction method with limited error is of interest for large data. The data generated in distributed servers is large and complex. Any attempt to model a described quantity from data distributed across multiple computers should minimize the information traveling between nodes as this is the bottleneck of such systems.
Wavelet decomposition is a way to significantly reduce the amount of data needed to model the cumulative distribution function. This work aims to understand how to use wavelet compression and write an R module (github) that can approximates the cumulative distribution function from a large input dataset. To justify the validity of the wavelet approach, wavelet approximation error and simple random sampling

error for different standard distributions and different statistical distances are introduced. Kolmogorov Smirnov **KS**, Cramer Von Mises **VM**, and mean distance **ME** estimates are used to evaluate the quality of the wavelet and sampling fits. Common and mixed **CDF's** are used to test our approach. The quality of the fit with additive noise will be evoked.

1.5 Report Progress

In the next chapter, we will be discussing a few ideas about general signal processing as the source of the wavelet decomposition. We will talk about the Haar wavelet by inspiration of the Fourier transform. We will justify in the later chapters the interest to the cumulative distribution function with the illustration of Haar development limit (**HDL**). That limit is nothing by the minimum resolution (a threshold) required to generate the minimum error. We will discuss the memory reduction and quantify the generated error. The last chapter will be devoted to more realistic world noisy signals and compound distribution with their **HDL**. At the end of the document an appendix that calculates the errors generated from the wavelet and **SRS** approximations for Gaussian and mixed **CDF** distributions.

Chapter 2

Wavelets

Fourier decomposition represents a periodic function as a sum of sinusoidal functions. Wavelets extend that concept to a frequency and time domain providing efficient access to localized information representation. This chapter aims to introduce the wavelet decomposition by analogy to Fourier's. The goal is to approximate the cumulative distribution function by the **Haar** wavelet, as we will see in the next chapter. We will introduce the Haar wavelet, which, although simple, is most appropriate for the purpose of approximating a step function like the **ecdf**.

2.1 Transformation Techniques

Transformation can be viewed as a change of basis, which are abundant in Mathematics and sometimes used to approximate. The examples below will help to understand the concept of wavelets as a technique to approximate a function.

- **Geometry** – Perhaps the most straightforward transformation is projecting a 2**d** vector into two non-collinear fixed vectors. We can approximate that vector with one component if we judge the other component is much smaller.
- **Calculus** – In modern analysis, a function can be approximate locally by Taylor polynomials with increasing degrees. Upon satisfaction, one can stop that approximation to any desired power and discard the error. A second-order degree or a fundamental one-degree linear approximation is widely used in a typical situation if that system can behave under linear response.
- **Spectral Decomposition** – The inspiring idea behind wavelets. A periodic function is decomposed to a sum of sines and cosines function with increasing frequencies. In the case of wavelets, different functions are used depending on the function's spectrum and shape.

2.2 The essence

A special transformation conjugated with periodicity is behind the idea of decomposing a function (signal) into an (infinite) serie of sines and cosines. In signal processing, it is known as Fourier decomposition. The essence of that transformation is that any periodic bounded function f is expandable into an infinite additive set of monochromatic waves with different amplitudes and increasing frequencies, ensuring a base change from the spacial domain x to the frequency domain ω. The passage from one coordinate to the other is illustrated in Fig.[2.1].

$$f(x) = \sum_{n=0}^{+\infty} a_n \cos(\omega_n x) + b_n \sin(\omega_n x) \simeq \sum_{n=-\infty}^{+\infty} c_n e^{i\omega_n x} \tag{2.1}$$

Equation [2.1] is the discrete Fourier decomposition. The coefficients $\omega_n = \frac{2n\pi}{T}$ where T is the period and n is an integer are the singular frequencies (pulsations).

By analogy, wavelet decomposition is the expansion of a function into a serie of predefined simple functions similar to Fourier base $e^{i\omega_n x}$ in Eq.[2.1]. In this work, it is called **Haar base** (see next parag.).

The continuous version of the Eq.[2.1] is given by the first equation of the set.[2.2]. $\hat{f}$ is the f Fourier transform. The second equation (ref.[2]) shows that the total energy carried by the function is independent of the coordinate system or the basis used for the calculation (space coordinates x versus frequency coordinates ω). This result is known as the Parseval theorem[3]:

$$\begin{aligned} \hat{f}(\omega) &= \frac{1}{2\pi}\int_{-\infty}^{\infty} f(x)\exp(-i\omega x)dx \\ \int_{-\infty}^{\infty}|f(x)|^2 dx &= \frac{1}{2\pi}\int_{-\infty}^{\infty}\left|\hat{f}(\omega)\right|^2 d\omega \end{aligned} \tag{2.2}$$

In other words, the change of a coordinates system does not alter the total power or information carried by the function. This equality is the essence of data compression. Suppose there are some small Fourier coefficients (and there will be), neglecting them, truncating them with a predefined threshold is without significantly altering the integral's total value. In that case, we negligibly degrade the information carried the function f. The same principle applied to wavelets.

The incommensurable consequence of this theorem is none other than a data compression procedure used in communication protocols. When decomposing a function into a Fourier series, we ignore higher frequency coefficients without much influence on the original signal: the most significant components will carry as much information as the original signal (ref.[4]). Note that when the high-frequency components are neglected, the contribution of these components becomes even smaller due to the square power on the right side of the equation [2.2]. The idea of the wavelet approximation is a form of data compression recipe: higher resolution components are discarded with hopefully very negligible loss of accuracy. The wavelet decomposition carries as much the essential information. Note that the same technique is used to free signal from an environment noise that could affect it.

2.3 Wavelets Decomposition

The two graphs in the figure [2.1] illustrate the benefit of Fourier transformation but also the limitation of that decomposition to resolve a function at time and frequency simultaneously: we know precisely what frequencies, but we do not know at what time they occur. On the contrary, we know precisely the amplitude at a time, but we ignore which frequency occurs at that exact time.

The time signal, left figure, might appear to be noise, contains in fact four frequencies of comparable intensity. The graph on the left always provides the exact amplitude at a specific time. Thus, the Fourier decomposition resolves the signal in the frequency domain while the temporal representation resolves the signal in

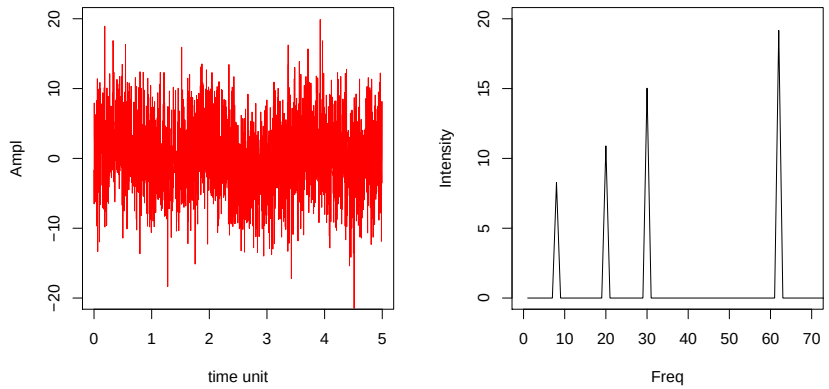

FIGURE 2.1: Base change from time (left) to frequency (right). Notice that the base change auscultates the function and, despite the noise, could extract four frequencies.

the time domain. The inability to accurately resolve the signal in both domains is known as the uncertainty principle (Heisenberg inequality in quantum physics context). This beautiful Fourier inequality is the mathematical translation of this fact (ref.[5]):

$$\int_{-\infty}^{\infty} x^2 \left|f(x)\right|^2 dx * \int_{-\infty}^{\infty} \omega^2 \left|\hat{f}(\omega)\right|^2 d\omega \geq 1/16\pi^2 \tag{2.3}$$

This inequality means that the narrower we ask the function f to be in the space domain, the broader it will become in the frequency domain to respect the upper bound.

Wavelet is a technique brought to overcome the limitation we just described. A wavelet is a window step function through which we scrutinize a function. Geometrically, the figure [2.2] displays a window step function. While translating at the desired scale, we can zoom coarser or finer at any specific interval and probe the function. The shape of the window function depends on the function and the application. The figure below is the simplest probing window called the **Haar window**.

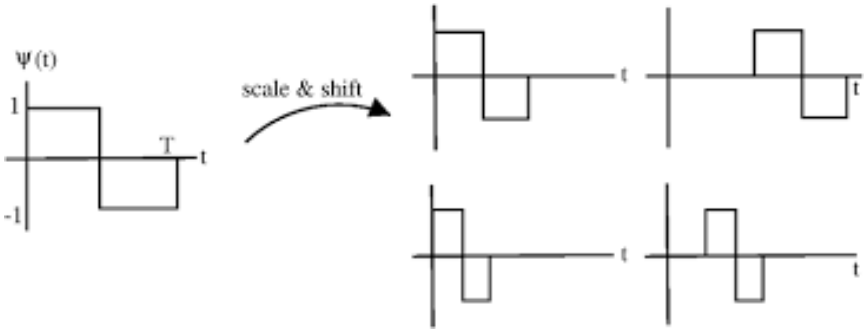

FIGURE 2.2: Left figure: Haar window or step function. The right figure: shifting or sliding the Haar wavelet for two different scales or resolutions. The step function is a local window applied to the desired signal to probe it locally.

The figure [2.2] explains the basic technique of how a wavelet probes a function: We choose the wavelet shape depending on the precision or scale (Haar window here as a probing function). We then slide that window in order to scan along with the function domain. Analytical definition of the Haar wavelet step function basis is presented in the next paragraph.

In our opinion, the word wavelet is misleading because it gives the impression that a function results from the composition of small waves. The wavelet decomposition is the successive approximation with a matched function (or a simple function like here Haar) is the wavelet decomposition. It has nothing to do with the intrinsic properties of the function (like the Fourier decomposition or any propagating wave notion). It is similar to Fourier, a projection on a basis (Haar basis in this work) with a translation to capture the local variation of the function is reminiscent of the notion of propagated wave. Depending on the nature of the function of interest, the choice of a specific wavelet is appropriate. A Haar wavelet is simply the local approximation of a function by a step function.

The numeric calculation used in this work is done with the help of the Daubechie wavelets. Daubechie is a set of different wavelets that include Haar and some more complex functions. In this work, we use Wavetresh R package (ref.[6]).

2.4 Haar Developpement Limit

Fourier developpement [7] has an orthonormal basis $= \left\{e^{i\omega_n t}\right\}$ (eq.[2.1]) whose elements are multiples of $e^{i\omega_0 t}$. Alfred Haar came with the idea of representing a function (continuous or not) in the form of discontinuous functions basis $\left\{\psi_{j,k}(t)\right\}$ indexed by a scale or a resolution j and a shift k [7]. We recommand highly the last reference for any endeavor to deepen the understanding of wavelets.

The Haar basis consists of the functions [7],[8], satisfying the relation:

$$\psi_{j,k}(x) = 2^{j/2}\psi_{j,k}(2^j x - k); k = 0, \pm 1...; j = 0, \pm 1...; \tag{2.4}$$

The space of those functions provided with the suitable scalar product form an orthonormal basis:

$$< \psi_{j,k}, \psi_{j',k'} >= \int \psi_{j,k}(x)\psi_{j',k'}(x)dx = \delta_{j,j'}\delta_{k,k'} \tag{2.5}$$

$\delta_{i,j} = 1$ if $i = j$, $\delta_{i,j} = 0$ if $i \neq j$ is the Kronecher function. The set ψ form a basis for all finite square integrable, $\int_{-\infty}^{+\infty} ||f(x)||^2 dx < \infty$, functions.

This means that we can represent such a function as called here Haar Devellopment (**HD**):

$$f(x) = \sum_{j,k} < f, \psi_{j,k} > \psi_{j,k}(x) \tag{2.6}$$

j is the resolution, k is the scale and $< f, \psi_{j,k} >$ is of the form [2.5]. When the function ψ takes the value:

$$\psi(x) = \begin{cases} 1 & 0 < x \leq 1/2 \\ -1 & 1/2 < x \leq 1 \\ 0 & otherwise, \end{cases}$$

The wavelet is called the Haar wavelet. The decomposition is called the Haar Developpement limit **HDL** when we truncate at a specific resolution j. The projection coefficients on the basis vectors $< f, \psi_{j,k} >$ are the Haar wavelet coefficients.

We mentioned earlier that we could compress a function with the help of the Fourier series by omitting the higher frequency terms as their contribution to the original (Power) function is not essential. The same procedure is used here. We define the **HDL** by omitting the higher terms. The important terms are stored, and the fewer contributions ones are pulled to null. Haar coefficients are calculated recursively. The reference [8] is the best accessible we found to explain how to obtain the coefficients $< f, \psi_{j,k} >$.

2.5 Application: Wavelet for a Distributed System

The data generated in distributed servers are large and complex. Any attempt to model a quantity described from data distributed across multiple computers runs into memory storage limits. Wavelet decomposition is a way to significantly reduce the amount of data needed to model the cumulative distribution function.

In the next chapter, we will justify the interest in the **CDF**. We will explain its approximation with a Haar wavelet. We will underpin how we calculate the Haar basis coefficients and discuss the algorithm complexity. We will show the error variation with different resolutions for the different distances presented in chapter one.

Chapter 3

Divide –> Develop –> Rule

This chapter is a description of the technique to use a wavelet to approximate a **CDF**. We will first review the concept of **CDF**. We will describe the Haar Wavelet, introduce the algorithm used for the wavelet coefficients, and discuss its complexity. We will explain the mechanism of data compression used to express a **CDF** with a minimum amount of data.

3.1 Empirical cumulative distribution function

A probability is a measure of an event occurrence. When the ensemble events are countable, it measures the frequency appearance of that event. On the other hand, when realizations are uncountable or continuous, the measure of single event realization is null. This result is intuitively related to the question that the probability of reaching a target is intimately associated with the degree of precision one wants to reach that target. A legitimate answer to the extreme question of infinite precision will lead fatally to a null chance to reach the target.

3.1.1 What is it?

To palliate the anomaly of finding the probability of a single value, we define the Cumulative Distribution Function as a measure of how much a variable accumulates in a continuous space. In such $X : \Omega \to \mathbb{R}$, where Ω is the universe events, then $P(X = x) = 0$ [9].

Proof. First we observe that subtracting the two equations

$$P(X \leq a) = \int_{-\infty}^{a} f_X(x)dx \tag{3.1}$$

$$P(X \leq b) = \int_{-\infty}^{b} f_X(x)dx \tag{3.2}$$

$$\text{gives, } \mathrm{P(X \leq b) - P(X \leq a)} = \int_{a}^{b} f_X(x)dx \tag{3.3}$$

$$= P(a < X \leq b) \tag{3.4}$$

$$\text{hence, } \mathrm{P(X = x)} \leq P(x - 1/n < X \leq x) = \int_{x-1/n}^{x} f_X(x)dx \tag{3.5}$$

$$\text{if } \mathrm{n \to \infty} : \mathrm{P(X = x) = 0} \tag{3.6}$$

□

Therefore, when the measurable probability space is uncountable, as of the impossibility of infinite precision, the probability of every precise realization is null $(P(X = x) = 0)$, we define the probability that the observation is less or equal than a given value to circumvent that abnormality:

$$F(x) = P(X \leq x) \tag{3.7}$$

The probablity of the realisation of a particular event $\{X = x\}$ is replaced then by that of the event $\{x < X \leq x + dx\}$ as follow:

$$P(x < X \leq x + dx) = F(X + dx) - F(X) \tag{3.8}$$

The empirical cumulative distribution function (ecdf) is a non-parametric estimator of the CDF. In practice, it assigns a probability of 1/n to each datum, orders the data from smallest to largest in value, and calculates the sum of the assigned probabilities up to and including each datum. The result is a step function that increases by 1/n at each datum [10].

$$\hat{F}_n(x) = \hat{P}_n(X \leq x) = n^{-1} \sum_{i=1}^{n} I(x_i \leq x) \tag{3.9}$$

3.1.2 Usefulness of cumulative distribution function

The space of monotonic increasing functions bounded in the unit interval such as a CDF is infinite. The graph [3.1] helps to visualize how different random variables could have visually similar CDF functions. Statistical error and or tests must be considered to assert whether an exact CDF describes a random variable. In our thinking, extra caution should be addressed when modeling a phenomenon with a CDF. I remind the reader that the different errors defined in the previous chapter are used to help assess the validity of the wavelet approximating a specific CDF.

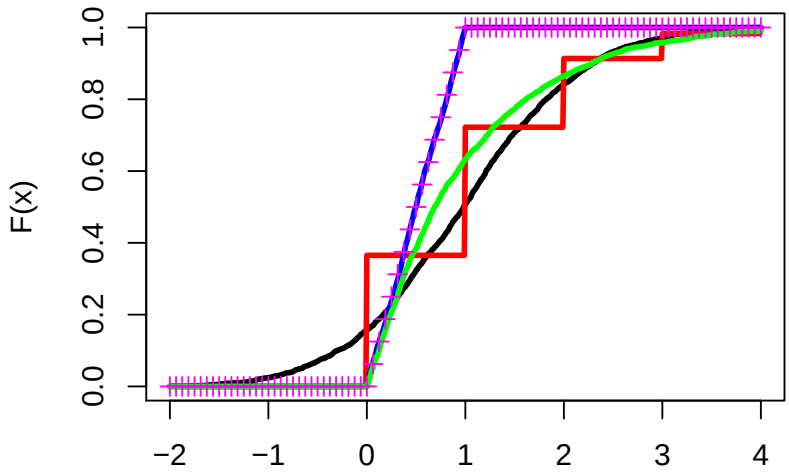

FIGURE 3.1: The curves show some similarities but describe completely different random variables: Gaussian, Poisson, Uniform, and Exponential distribution.

Once we modeled a CDF, statistical quantities can be accessed. Here are two important examples of the usefulness of the cumulative distribution function:

Moments: With this set of quantities, we can measure the mean, the variance, skewness and kurtosis. The moments are calculated with the cumulative F using the following expression:

$$\mathbb{E}[X^r] = \int_0^\infty x^r dF(x) - \int_0^\infty (-x)^r dF(-x) \tag{3.10}$$

$$= r \int_0^\infty x^{r-1} \left[1 - F(x) + (-1)^r F(-x)\right] dx. \tag{3.11}$$

Quantiles: Many real-world data are beautifully distributed with even a know probability distribution. A plethora of collected data doesn't fall in that category. The quantiles rank the score in the distribution. Those quantities are accessible through the **ecdf**. For example, the median is that score that corresponds to $50th$ percentile, i.e., the value taken by the random variable X such that:

$$\hat{F}(x) = \hat{P}(X \leq x) = 0.5 \tag{3.12}$$

Depending on the interest, any quantile can be calculated with the help of the **ecdf** (percentile, deciles, quartiles).

3.2 Haar Developpement Limit for CDF

Now we have defined the CDF and wavelets; we can approximate the first by the latter. Since wavelet decomposition exhibits no special mathematical conditions, and since the CDF exists all the time or to a lesser extent can be approximated by the empirical distribution, the wavelet approximation is mathematically possible.

The approximation of the cumulative **CDF** is performed a the light of Eq.[2.6]. As a reminder, an F (any function) can be expanded as:

$$F(x) = \sum_{k,n} < F, \psi_{k,j} > \psi_{k,j}(x)$$

We choose to formulate our approach by the following symbolic instead:

$$F(x) \sim \mathbf{HDL}[[j, k = 7]](x) \tag{3.13}$$

Where k is the level at which we choose to truncate the development or the resolution, and j is the translation coefficient. The choice of the value 7 is in reliance on the graph [1.2]. At that value, we had a minimum memory and a lower error.
We mentioned earlier that we could compress a function with the help of the Fourier series by omitting the higher frequency terms as their contribution to the original (Power) function is not essential. The same procedure is used here: We define the **HDL** by omitting the higher terms. The essential terms are stored, and the more minor contributions ones are put to null.

3.2.1 Algorithmic and Complexity

The method used to calculate the wavelet coefficients is known as a pyramid algorithm. The algorithm for general wavelets is the discrete wavelet transform due to Mallat (1989b). Interested reader can consult [6] for the details.
Time and size complexity is generally expressed as a function of the size of the input. Since this function is difficult to compute precisely, one commonly focuses on the behavior of the complexity when the input size increases-that is, the asymptotic behavior of the complexity. The time complexity of 1D wavelet decomposition uses big O notation, typically $O(n)$. In contrast and to give a comparison level, the FFT is $O(n \log n)$, where n is the input size in units of bits needed to represent the input. The study of complexity is out of scope. In laptops nowadays and for a resolution less than 2^{14}, the calculation time is in the order of seconds.

3.2.2 Convergence

Wavelets series are illustrated by components taken from the simple case (resol=:1), to an increased maximum (resol:=11, fig.[3.2]). The data are of 2^{12} points. As a reminder, the goal is never to reach the maximum resolution but to settle for a very low one to shrink down the memory size, yet at the cost of a low limited error on the estimation of the cumulative distribution function.

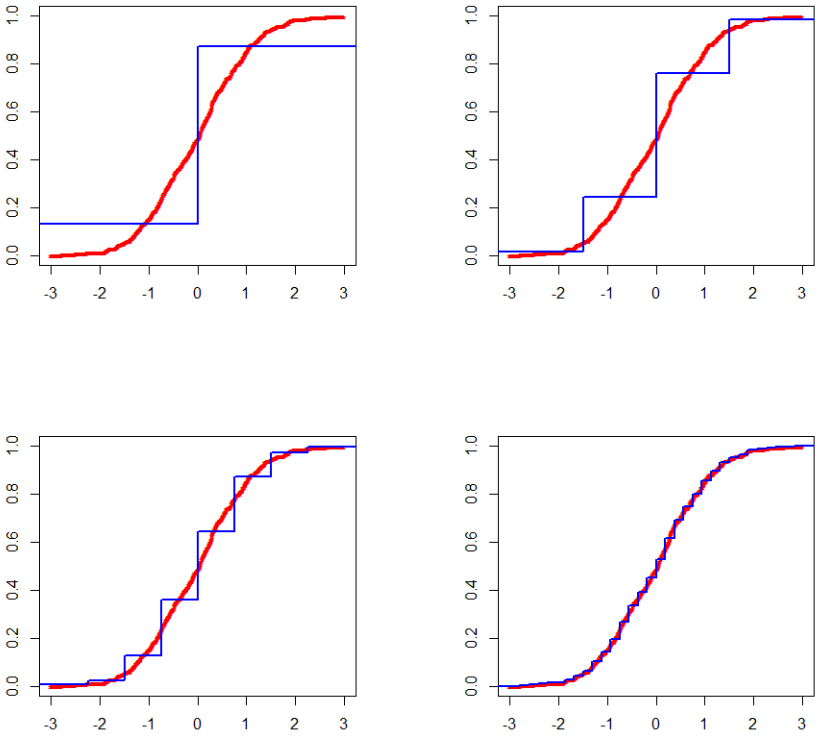

FIGURE 3.2: Different scaled Haar wavelets transated across the curve.

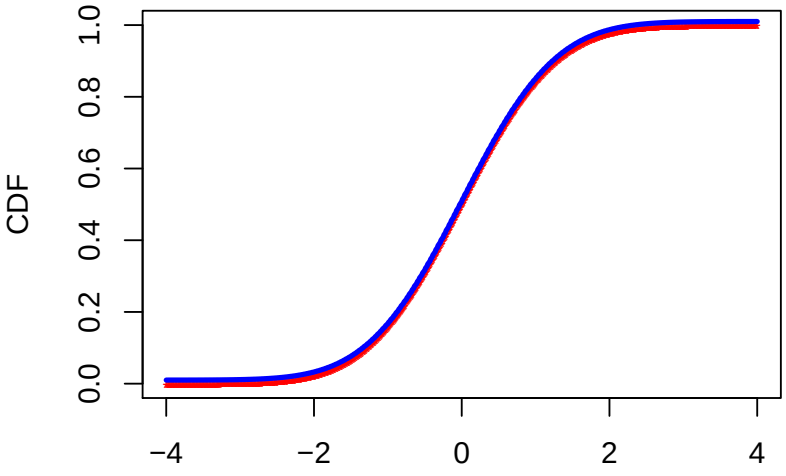

FIGURE 3.3: Ultimate resolution leads to collapse the function with it's wavelet!

In the figure [3.3], red curves are nothing but CDF generated by the "pnorm" command in R. The blue curve is the wavelet "approximation" taken to its highest resolution. The graphs appear to approximate the function by a stepwise. That visualization is misleading but partially true, as we explained before: the scaled wavelet is translated along the curve.

3.3 Error-Memory Trade

Considering the method of wavelet approximation to reduce a big dataset size to gain computational memory is a winner. However, a sophisticated algorithm should outperform any easier one to respect parsimony and common sense thinking. Sampling data and developing a model is a simple way to solve a function fit. This section compares the use of simple random sampling fit and wavelet fit with respect to their committed errors.
We approach a Gaussian **CDF** with an increased resolution Haar wavelet (see Fig.[3.2] for insight). We will present the variation of the three distance **KM**, **CVM** and **MAE** with the resolution. The Boxplots represent the dispersion of the same errors obtained the same amount (same resolution) of data points drawn with simple random sampling.

3.3.1 Kolmogorov Smirnov distance

Two errors are calculated here. the first is the KS error $KS_n =: \sup_{x\in\mathbb{R}} \left|\hat{F}_n(x) - F_0(x)\right|$ where $\hat{F}_n(x)$ is the 7-resolution wavelet approximation. the second is the error generated from sampling the reference Gaussian cumulative $F_0(x)$ represented by the boxplots.

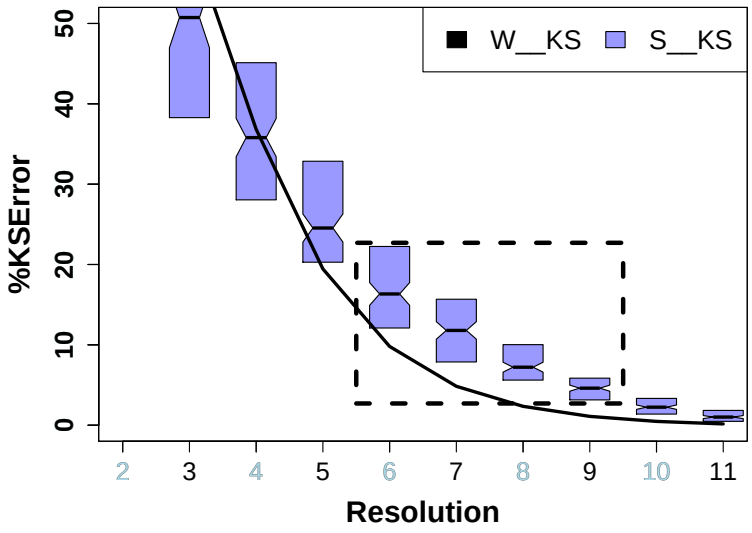

FIGURE 3.4: KS error from SRS (**S**_KS) and wavelet (**W**_KS) approximations. The black horizontal segment is the SRS sampling median error.

The curve in the figure [3.4] shows that the **KS error** [Eq.1.1] decreases while approaching a Gaussian CDF with the Harr wavelet with increasing resolution. The boxplot is the dispersion, drawn from 1000 samples, of the error while sampling the same normal CDF with increased selection by SRS data point number from 2^2 to 2^{11} (total of 2^{12}), in coincidence with the wavelet resolution. The black rectangle delimits the region where the wavelet (W_KS) error is below the SRS (S_KS). For example, the resolution 7 (2^7), the error is around 5% while the median of the boxplot is well above 10%. The median errors are obtained from 1000 samples. Beyond resolution four, the median errors lay higher than the wavelet error curves demonstrating the worthiness of our approach: it is better to approximate with a wavelet of at least 2^4 resolution than to take an approximation with a sample of the same amount of point drawn from the original available data.

3.3.2 Van Mises distance

Two errors are calculated. the first is the VM error $\mathrm{VM}_n := \sqrt{\int (\hat{F}_n(x) - F_0(x))^2 \, \mathrm{d}F_0(x)}$ where $\hat{F}_n(x)$ is the 7-resolution wavelet approximation. the second is the error generated from sampling the reference Gaussian cumulative $F_0(x)$ represented by the boxplots.

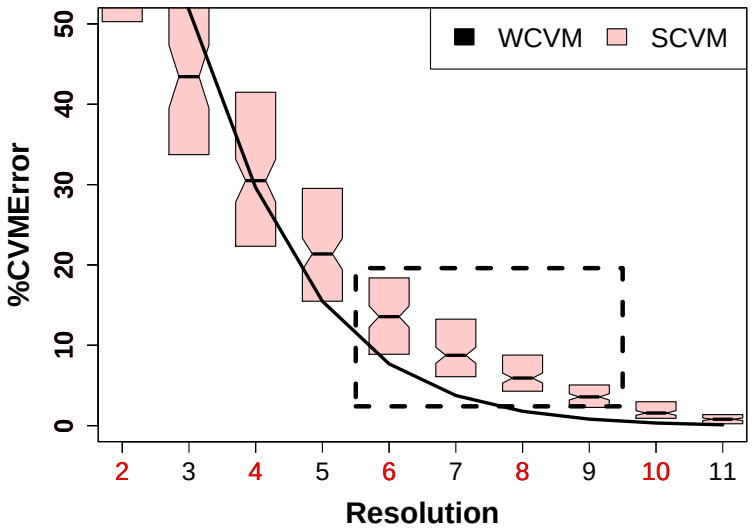

FIGURE 3.5: CVM error from SRS (**S**CVM) and wavelet (**W**CVM) approximations. The black horizontal segment is the median error.

The curve in the figure [3.5] shows the **CVM error** [Eq.1.2] decrease while approaching a normal CDF with the Harr wavelet with increasing resolution. The boxplot is the dispersion, drawn from 1000 samples, of the error while sampling the same normal CDF with increased selection by SRS data point number from 2^2 to 2^{11} (total of 2^{12}), in coincidence with the wavelet resolution. The black rectangle delimits the region where the wavelet VM error is below the SRS VM one. For example, the resolution 7 (2^7), the error is around 5% while the median of the boxplot is beyond 10%.

3.3.3 Mean Average distance

Two errors are calculated here. the first is the ME error ME_n: $=mean(|\hat{F}_n(x) - F_0(x)|)$ where $\hat{F}_n(x)$ is the 7-resolution wavelet approximation. the second is the error generated from sampling the reference Gaussian cumulative $F_0(x)$ represented by the boxplots.
The curve in the figure [3.6] shows the **MA error** Eq.[1.4] decrease while approaching a normal CDF with the Harr wavelet with increasing resolution. The boxplot is the dispersion, drawn from 1000 samples, of the error while sampling the same normal CDF with increased selection by SRS data point number from 2^2 to 2^{11} (total of 2^{12}), in coincidence with the wavelet resolution. The black rectangle delimits the region where the wavelet KS error is below the SRS KS. For example, the resolution 7 (2^7), the error is around 5% while the median of the boxplot is beyond 10%.

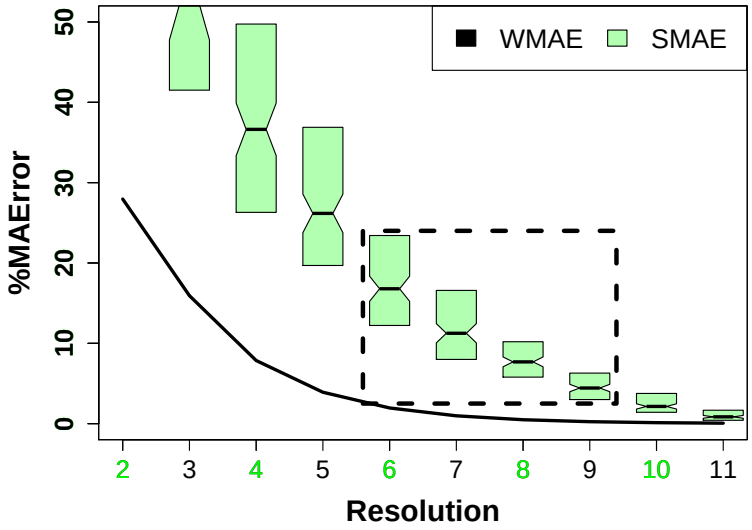

FIGURE 3.6: MA error from SRS (**S**MAE) and wavelet (**W**MAE) approximations. The black horizontal segment is the median error.

3.4 Synthesis

To better compare the distances between the original function and its wavelet fit for the same resolution, which coincides with the SRS sample size, we include the last three figures in one. That is Fig.[3.4] for **KS**, Fig.[3.5] for **CVM** and Fig.[3.6] for **MAE**, in the figure [3.7]. The figure assembles The error generated by the ***wavelet approximation*** and the ***SRS approximation***: the three distances and the two ways of calculating it.
At the resolution of 2^7, (we remind that the scale of the y-axis is to the power of 2), the three curves representing the three wavelets errors is strictly under the corresponding boxplots representing the dispersion of the errors obtained by sampling rate (point 2^7 out of 2^{12}.) At that resolution, the memory consumed is around 5%.

The percentage of the memory size in Bytes involved in the process increases with the resolution while errors decrease. At low resolution, the number of points used

is meager to build the wavelet. On the opposite side, when using a high-resolution wavelet (2^{11}), the errors are smaller, but the memory consumed is considerable. Note the number of data points is 2^{12}. The black dashed box delineates the region where the wavelet errors are much smaller than the median SRS errors, with almost no overlap between the curve and the boxplots. The precise error quantization will be given in the next chapter when the individual CDFs are approximated.

FIGURE 3.7: Curves: HDL errors. Boxplots: SRS errors.

The graph [3.8] below is the zoom delimited by the dashed rectangle in the Fig.[3.7]. We kept the KS and CVM errors. The graph shows that we can consider the wavelet fit starting from resolution 5 as both red and blue lines are under the SRS median error at the cost of moderate precision loss. At 7-resolution and beyond, the SRS sampling fit generates an error greater than wavelet one almost for all the drawn samples (2^7 points out of a 2^{12} total data points).

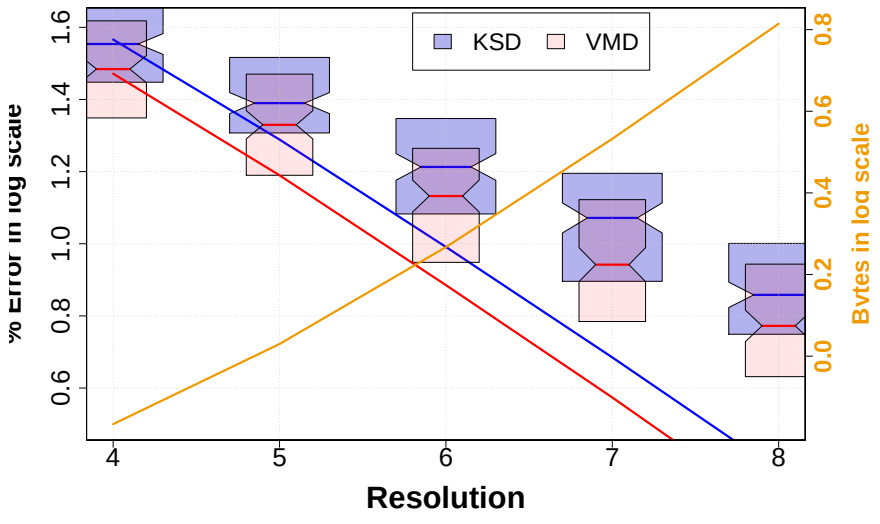

FIGURE 3.8: Dashed region at the figure [3.7] in logarithm scale.

A vital remark, starting from resolution 6 (2^6), the low error generated and very low memory size is consumed when approximating. We choose resolution 7 to study more CDFs in the next chapter as that resolution. The three errors are much lower than the median errors generated from simple random sampling.
We would argue furthermore that the **HDL** is better starting of $2^4 = 64$ resolution than the **SRS** with a sample of the sample size. The will be no surprise by outliers, even at low resolution, while guaranteeing lower or comparable error at very low resolution. Those outliers were removed explicitly for a cleaner graph. We should bear in mind that they will be there and can generate unexpectedly lower or high errors. We leave it to the reader to forge an opinion from the graph [3.7] to how far he wants to sacrifice the precision for the small memory size.

In conclusion, the Haar wavelet limit approach performs better, especially around and above the 2^6 resolution. For lower resolution, **SRS** and **HDL** are comparable. The error induced by the reduction of the data size (points) is less than 10%. The data size is reduced to 5%. The errors made by **HDL** with three types of error stays below the same level of randomly sampled data.

Chapter 4

Compound Distribution and Noise

This chapter will present the error generated upon replacing a theoretical CDF with its wavelet of $128 = 2^7$ components along with an SRS with the same sample size. Wavelet shows lower errors hence giving better function approximation.

4.1 Common Distributions Approximation

Recall that we note indistinctively mention the error or the distance from a distribution to its wavelet approximation by $\mathrm{VM}_{n\alpha}$, $\mathrm{KS}_{n\alpha}$ and $\mathrm{ME}_{n\alpha}$. We remind the expressions to calculate the distances as follows:

$$\begin{aligned}
\mathrm{VM}_{n\alpha} &:= \Big(\int (F_n(x) - F_0(x))^2 \, \mathrm{d}F_0(x)\Big)^{\frac{1}{2}} \\
\mathrm{KS}_{n\alpha} &:= \sup_{x\in\mathbb{R}} |F_n(x) - F_0(x)| \\
\mathrm{ME}_{n\alpha} &:= mean(|F_n(x) - F_0(x)|)
\end{aligned}$$

The subscript n denotes the wavelet resolution. The subscript α represents the method of calculation that will be replaced by the letter **w** or **s** to specify if the distance is calculated to the wavelet approximation or the sample approximation. The table below displays the result for a Gaussian with different resolutions. Please check the graph [3.7] to understand the choice of the three resolutions.

4.1.1 School Case Gaussian CDF

The following table is the **VM** and **KS** errors for a gaussian **CDF**. We display the errors for different **HDL** resolutions and the same amount of points drawn 1024 times by SRS. Appendice at chap.[5] shows how the **KSw** and **VMw** distances are calculated numerically for the resolution 7. It also shows **KSs** and **VMs** distances for different sampling levels.

TABLE 4.1: **HDL** and **SRS** (=: **wavelet.** and **sampling.**) errors for three resolutions.

Resolution	KSs	KSw	Pks	VMs	VMw	Pvm
2^5	.141	.064	.99	.059	.048	.71
2^6	.101	.032	1.0	.043	.024	.95
2^7	.071	.016	1.0	.031	.012	1.0

The table[4.1] is read as follows: a resolution or a sample size of 2^5, the **KSs** is the **KS** error made with the sampling, and **KSw** the error made when we approximate by Haar wavelet. Pks denote the empirical probability of how often the sampling error exceeds the wavelet error for a total of 1024 drawn samples. The same stands for **VM** distance with **VMw**, **VMs** and Pvm.
The resolution 2^5 provides a much lower error KSw (.06) than KSs (.14) but comparable VM error (.06 for sampling versus .05 for the wavelet). 2^7 resolution reaches much lower error. For the intermediate resolution, the KS error for wavelet is better than SRS, and VM is comparable.
The strategy wavelet performs better, at least in the KS sens. Overall, it is robust and the error is predictable, at most comparable to SRS error or lower. As a reminder, I refer you to the graph [3.7] to be convinced about the choice of the range of the wavelet resolution.

4.1.2 More School Cases

The errors displayed at the table [4.2] are generated by the 2^7 wavelet approximation for differents chosen distributions with random parameters. Only 128 points regardless of the amount of the original data could approximate the original distribution with at most a **KSw** error of .04. This number is considered small knowing that for any **CDF** $F(x), 0 \leq F(x) \leq 1, \forall x$.
The example of approximating a Gaussian CDF gives a KSw (.01 HDL versus .07 SRS). SRS gets only 0% chances to beat HDL (Pks is the chance that the median sampling KS error est bigger than the HDL error). The mean wavelet error MEw is very low (.003). The error generated by sampling is the median error calculated from a 1024 drawn sample.

TABLE 4.2: Distances from the theoritical CDF to their 2^7 wavelet.

Dist	**KSs**	**KSw**	**Pks**	**VMs**	**VMw**	**Pvm**	**MEw**
$Normal(0,1)$	.07	.01	1.0	.03	.01	1.0	.003
$LogNorm(1,1)$	.06	.04	.95	.03	.02	.84	.003
$Weibull(3,1)$	.07	.01	1.0	.03	.01	1.0	.003
$Gamma(1,2)$	.49	.03	1.0	.33	.01	1.0	.003
$Logit(2,1)$	.07	.02	1.0	.03	.01	.99	.004

To help read the table [4.2], we take the **Weibull(3,1) CDF** in the third row. The **HDL** KSw error is .01 vers .07 with **SRS**. VM error is .01 for an **SRS** of .04. The sampling error is the median sample error. The Probabilities Pks and Pvm are the empirical chances that the HDL error is lower than the median SRS error. For that case, for every sample, the HDL error is lower than the SRS one. Apart from lognormal with slightly comparable errors (but still smaller for the wavelet), the HDL approach perform better for every distribution.

We choose the most common distributions with different parameters. A 2^7 randomly chosen points from every **CDF** generate the error noted by a subscript **s**, while the sample resolution generate the subscript **w**, I refer you to the figure [3.7]. The KS and VM errors are lower for the HDL in comparison to SRS.

4.2 Mixing Approximation

A more realistic situation would be a mixing of two different distributions. We present here the CDF graph of two Gaussians [4.1] along with its approximation.

$$F(x) = .4 * F_{Norm(1.2,.5)}(x) + .6 * F_{Norm(-1.4,.4)}(x) \tag{4.1}$$

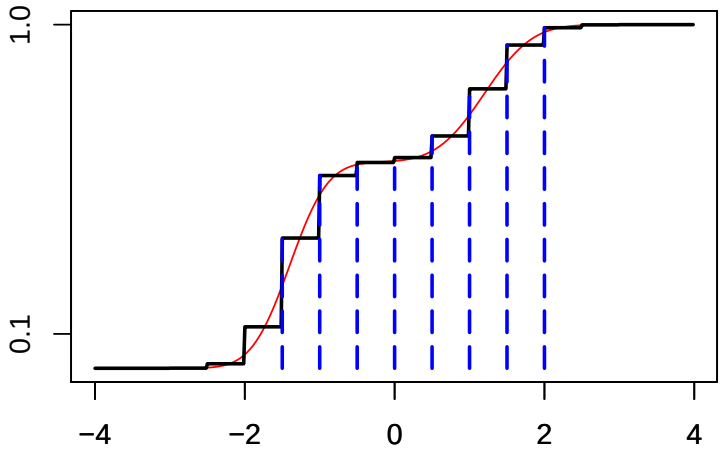

FIGURE 4.1: Curve of compound normales CDF eq.[4.1] with its 2^5 resolution wavelet.

This example **illustrate** how the wavelet can track the **CDF** inflexion. The table below shows the error induced upon the approximation of this special distribution.

TABLE 4.3: Error with used parameters of compound dist.

Dist	**Param**	**KSs**	**KSw**	**Pks**	**VMs**	**VMw**	**Pvm**
Norma	$\{(0;1.0)+(0.2;2.0)\}$	.07	.01	1.0	.03	.00	1.0
LNorm	$\{(1;1.0)+(3.3;.08)\}$	.07	.06	.75	.03	.02	.86
Weibul	$\{(3;4.2)+(4.5;5.0)\}$	.07	.03	1.0	.03	.01	1.0
Gamma	$\{(1;2.0)+(1.9;1.4)\}$	.07	.05	.91	.03	.01	1.0
Logit	$\{(2;1.0)+(3.0;1.4)\}$	.07	.03	1.0	.03	.01	.99

The table [4.3] displays a clear lower error of HDL versus SRS from the error KS column (KSs > KSw) and VM column (VMs > VMw) except for the lognormal, in which case the errors are equivalent. We still plead for the Wavelets as they generate predictable and deterministic lower errors.

To explore the less convincing capability of the **HDL** to approximate a mixed lognormal in comparison to **SRS**, we plot its *CDF* describe as:

$$F(x) = .4 * F_{LogNorm(1,1)}(x) + .6 * F_{LogNorm(3.3,.08)}(x) \tag{4.2}$$

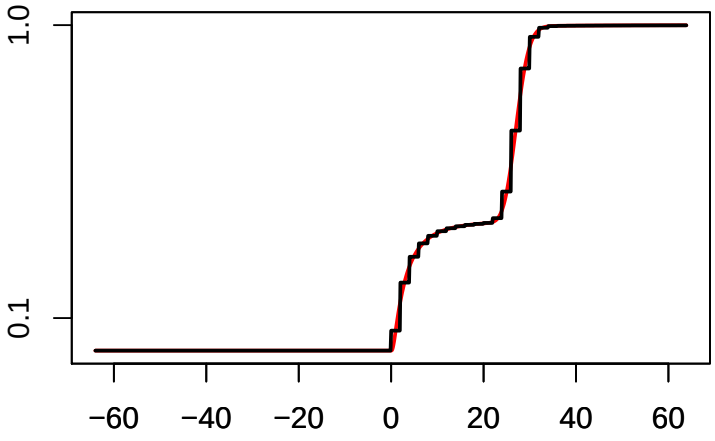

FIGURE 4.2: Mixed Weibull eq.[4.2] with its 2^6 Haar wavelet.

The plot display also the wavelet approximation (red) and how it **HDL** performs. We suspect the strong inflections highly penalize the calculated errors. We can argue that with the same type of error (error from an HDL and the one from an SRS), we would still prefer the **HDL** as for its deterministic character.

4.3 Noise and wavelet Shrinkage

Statistical noise is unexplained variability within a data sample. The term noise, in this context, came from signal processing used to refer to unwanted electrical or electromagnetic energy that degrades the quality of signals and data. The noise means that the sampling results might not be duplicated if the process were repeated. When noise intervenes, as most likely in all experimental studies, our capabilities to extract precise information are put to the test. In the next section, we will introduce the evolution of the error with gradually increasing noise. Modeling a CDF, especially from big data, would be highly exposed to noise.

The idea is that one obtains observations, $y = (y1, ..., yn)$, that arise from the following model [6]:

$$y(x) = ecdf(x) + \epsilon \tag{4.3}$$

The noise denoted ϵ is the unexplained variability within a sample. One usual way to remove noise is through Fourier transform. We keep only the frequencies beyond a power threshold. The two graphs illustrate a function affected by the noise. Fourier

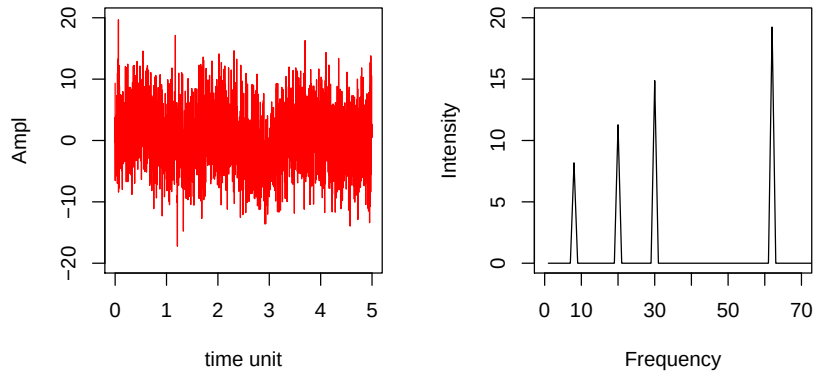

FIGURE 4.3: Time and frequency diagramme of $s(t)$.

transform could resolve the following function as shown in the figure [4.3].

$$s(t) = .2 * sin(8t) + .5 * sin(20t) + sin(30t) + \varepsilon \tag{4.4}$$

where ϵ is a Gaussian. The Fourier de-noising method removes all the frequencies components above a certain threshold.

When approaching a noisy **ecdf** by HDL, akin to Fourier transform, the model proposed is:

$$\mathrm{CDF_{n,k}(x) = HDL[[n]](x) + N(0,k)(x)}$$

HDL is the Haar development truncated at the n^{th} resolution, k is a noise level introduced by a gradual increase of the variance of the gaussian perturbation N(0,k).

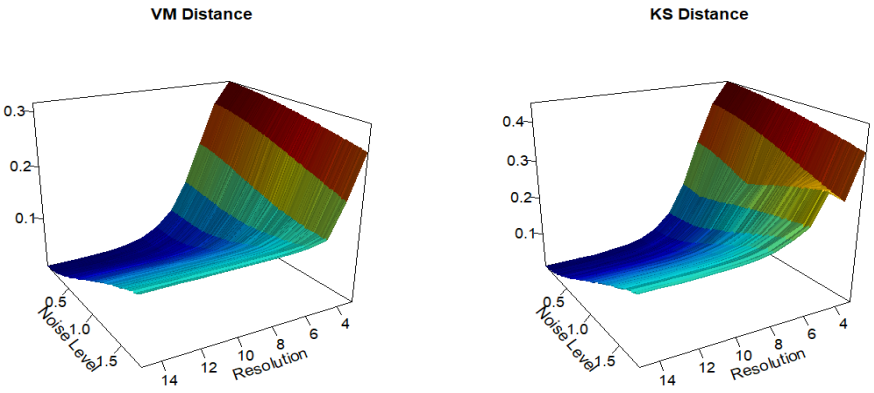

FIGURE 4.4: Vertical axis is the error. Horizontal axis are noise level and wavelet resolution.

The two graphs represent the error KS and VM with different levels of noise and resolution. As expected, the error is negligible for low noise and high resolution.

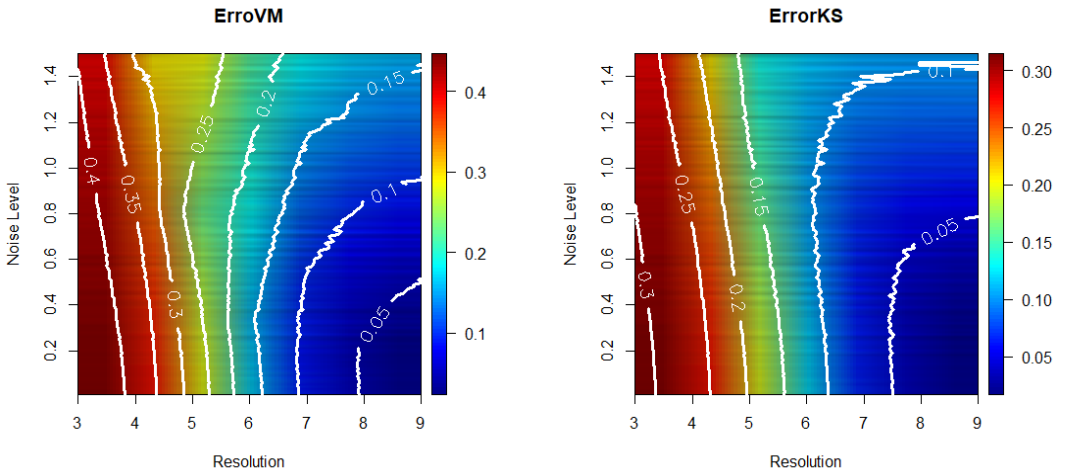

FIGURE 4.5: Contour plot: color intensity represents the error.

The contour plot represents the error with different levels of noise and resolution. These graphics show that any resolution below 6 is not enough to model a normal CDF. HDL can resolve the function down to an error below .1 even under a strong noise. The blue color is the region of a low error where we want to constraint it. The Intresting region is between resolutions 6 and 8.

4.4 Conclusion

"All models are wrong" is a common aphorism in statistics often expanded as "All models are wrong, but some are useful [11]." The model explored in that work was to approach a cumulative distribution function by Haar wavelet to limit the amount of data used for the approximation without a significant loss of precision using thresholding. A comparison is made between the wavelet and simple random sampling to prove the worthiness of the wavelet approximation. The challenge to synthesize and bring a complex subject to the "vernacular" was one of the difficulties faced in that work. A future endeavor would be applying the approach in a distributed system by using the concept to noisy real-world data.

Chapter 5

Appendix

This chapter is summary of the R code to calculate a Haar Wavelet Devellopment HDL[[7]] for different **CDF's** : Gaussian, Gaussian mix and Weibull mix with 2^7 resolution. Wavelet methods are from the **WaveThresh** package in R by G.P. Nason [6]. The errors are displayed as KSw and VMw. The first code is for a Gaussian, it's generated wavelet approximation **HDL** and the sampling approximation **SRS**. Code available on github.

5.1 HDL Code for $N(0,1)$

5.1.1 wavelet

```
> mr=2^12;lev=7;#mr maxreso; lev the waveleth treshold
> escalw=function(sig,resol=3,x){
+   dd <- wd(sig, filter.number=1, family= "DaubExPhase" );
+   esc=accessC(dd,resol)/sqrt(2)^(log2(length(sig))-resol);#length(esc)
+   t=seq(min(x),max(x),(max(x)-min(x))/(2^(resol)) );
+   t=t[1:2^resol];#length(t);(esc)
+   iso=isoreg(t,esc);
+   stepfun(t,c(iso$y[1],iso$y));#rm(t)
+ }
> shapeDist=function(pdist,parm1=1,parm2=1){
+   x=seq(-200,200,.0001);y=pdist(x,parm1,parm2);
+   maxseq<-2*(x[which(y>.98)[1]]);minseq<-2*(x[which(y>.001)[1]]);
+   x=NULL;x<-seq(minseq,maxseq,(maxseq-minseq)/mr);x<-x[1:mr];
+   list(x,pdist(x,parm1,parm2));
+ }
> HaarWGene=function(cdffun){
+   x=cdffun[[1]];y=cdffun[[2]];
+   HDL=lapply(seq(1,7,1),function(resol){escalw(y,resol,x)});
+ }
> KSfw=function(refcdf=HDL[[6]],cdftheo=pnorm,p1=.01,p2=1){
+   x1=knots(refcdf);x2=shift(x1,1);x2[1]=x1[1];
+   DnP=abs(cdftheo(x1,p1,p2)-refcdf(x1));DnM=abs(cdftheo(x1,p1,p2)-refcdf(x2));
+   max(DnP,DnM)
+ }
> VMfw=function(refcdf=HDL,cdftheo=pnorm,p1=.01,p2=1){
+   x1=knots(refcdf);F0=cdftheo(x1,p1,p2);Fn=refcdf ;#hist(F0);#hist(Fn(x));
+   F0=shift(F0,1);F0[1]=0;F0[length(F0)+1]=1;
+   dF0=diff(F0);F0=cdftheo(x1,p1,p2);
+   sqrt(sum( ((Fn(x1)-F0)^2)*dF0 ))}
```

```
>   HDL=HaarWGene(shapeDist(pnorm,0,1))[[lev]];
>   KSw=round(KSfw(HDL,pnorm,0,1),4)
>   VMw=round(VMfw(HDL,pnorm,0,1),4)
>   cat("The generated KS and VM wavelet errors are:\n","KSw:=",KSw,";","VMw:=",VMw)

The generated KS and VM wavelet errors are:
 KSw:= 0.0161 ; VMw:= 0.0122
```

5.1.2 Sampling

We present here the error generated by **SRS**. Erors are displayed as KSs and VMs. Pvs and Pvm are the probabilities that the generated error median of the sample is greater than the one generated from the wavelet.

```
>   KSfnorm=NULL;vmfnorm=NULL;lev=2^7
>   VMfs11=function(sampdist=ecdf(rnorm(lev,0,1)),cdftheo=pnorm,p1=0,p2=1){
+     if(!is.numeric(x2))x2<<-shapeDist(cdftheo,p1,p2)[[1]];
+     F0=cdftheo(x2,p1,p2);Fn=sampdist;
+     F0=shift(F0,1);F0[1]=0;F0[length(F0)+1]=1;# cdf limits.
+     dF0=diff(F0);F0=cdftheo(x2,p1,p2);
+     sqrt(sum( ((Fn(x2)-F0)^2)*dF0 ))
+   }
>   KSfs1=function(sampdist=ecdf(rnorm(lev,0,1)),cdftheo=pnorm,p1=0,p2=1){
+     if(!exists("x1"))x1<<-shapeDist(cdftheo,p1,p2)[[1]];
+     x2=shift(x1,1);x2[1]=x1[1];
+     DnP=abs(cdftheo(x1,p1,p2)-sampdist(x1));
+     DnM=abs(cdftheo(x1,p1,p2)-sampdist(x2));
+     max(DnP,DnM)
+   }
>   p=1024;KSfnorm=NULL;vmfnorm=NULL;
>   for(i in 1:p) KSfnorm=(c(KSfnorm,KSfs1(ecdf(rnorm(lev,0,1)),pnorm,0,1)));
>   for(i in 1:p) vmfnorm=(c(vmfnorm,VMfs11(ecdf(rnorm(lev,0,1)),pnorm,0,1)));
>   cat("The generated sampling KS and VM errors are:\n","KSs:=",
+       c(round(median(KSfnorm),4),"VMs:=",round(median(vmfnorm),4)))

The generated sampling KS and VM errors are:
 KSs:= 0.0721 VMs:= 0.0327

>   cat("The probabilies Pks and Pvm are the chances that sample
+       approximation generates an error greater than the wavelet approximation:")

The probabilies Pks and Pvm are the chances that sample
      approximation generates an error greater than the wavelet approximation:

>   cat("Pks=",round(sum(KSfnorm>KSw)/p,2),";","Pvm=",round(sum(vmfnorm>VMw)/p,2))

Pks= 1 ; Pvm= 1
```

5.2 HDL Code for mixed distribution

This chapter is summary of the R code to obtain a Haar Wavelet Devellopment HDL[[7]] along with the errors for mixed **CDF's**: mixed Gaussian and mixed Weibull. Code available on github.

```
> mr=2^12;lev=7;
> rweib=rweibull;rlnor=rlnorm;rgamm=rgamma;rlogi=rlogis;
> pweib=pweibull;plnor=plnorm;pgamm=pgamma;plogi=plogis;
> vmfnorm=NULL;vmflnor=NULL;vmfweib=NULL;vmfgamm=NULL;vmflogi=NULL;
> KSfnorm=NULL;KSflnor=NULL;KSfweib=NULL;KSfgamm=NULL;KSflogi=NULL;

> escalcom=function(sig,resol=3,x){
+   #sig=dist[[2]];x=dist[[1]];resol=4
+   dd <- wd(sig, filter.number=1, family="DaubExPhase");
+   esc=accessC(dd,resol)/sqrt(2)^(log2(length(sig))-resol);#length(esc)
+   t=seq(min(x),max(x),(max(x)-min(x))/(2^(resol)) );
+   t=t[1:2^resol];#length(t);(esc)
+   iso=isoreg(t,esc);
+   stepfun(t,c(iso$y[1],iso$y));#rm(t)
+ }
> shapeComp=function(dist1,dist2,p1=3,p2=4.2,p3=4.5,p4=3.2){
+   #dist1=pweibull(3,4.2);dist2=pweibull(4.5,3.2);#x1=dist1[[1]];x2=dist2[[1]]
+   x1=shapeDist(dist1,p1,p2)[[1]];x2=shapeDist(dist2,p3,p4)[[1]];
+   x1=round(max(abs(min(x1,x2)),abs(max(x1,x2))))
+   xseq=seq(-x1,x1,2*x1/2^10);xseq=xseq[1:2^10]
+   list(xseq,.4*dist1(xseq,p1,p2)+.6*dist2(xseq,p3,p4));
+ }#shapeComp(pweibull,pweibull,3,4.2,4.5,3.2);
> KSf=function(refcdf=HDL[[6]],cdfcomp){
+   x1=cdfcomp[[1]];x2=shift(x1,1);x2[1]=x1[1];
+   DnP=abs(cdfcomp[[2]]-refcdf(x1));DnM=abs(cdfcomp[[2]]-refcdf(x2));
+   max(DnP,DnM)
+ }
> VMf=function(refcdf=HaarWGene(c1)[[4]],cdfcomp=c1){
+   x1=cdfcomp[[1]];F0=cdfcomp[[2]];Fn=refcdf ;#hist(F0);#hist(Fn(x));
+   F0=shift(F0,1);F0[1]=0;F0[length(F0)+1]=1;
+   dF0=diff(F0);F0=cdfcomp[[2]];
+   sqrt(sum( ((Fn(x1)-F0)^2)*dF0 ))
+ }
> MEf=function(wvlet=HaarWGene(mix)[[1]],cdfcomp=mix){
+   x=cdfcomp[[1]]
+   return( mean(abs(wvlet(x)-cdfcomp[[2]])) )
+ }
> cat("dist(theorCDF,HDL) :=");

dist(theorCDF,HDL) :=

> mix=shapeComp(pnorm,pnorm,0,1,3.0,1.4);
> cat("The generated errors for mixed Gaussian:\n",
+     "  KS:=",KSf(HaarWGene(mix)[[lev]],mix),
+     "; VM:=",VMf(HaarWGene(mix)[[lev]],mix),
+     "; ME:=",MEf(HaarWGene(mix)[[lev]],mix),"\n")
```

```
The generated errors for mixed Gaussian:
   KS:= 0.01887749 ; VM:= 0.007938658 ; ME:= 0.001953125

> mix=shapeComp(pweib,pweib,3,4.2,4.5,5);
> cat("The generated errors for mixed Weibull:\n",
+     "  KS:=",KSf(HaarWGene(mix)[[lev]],mix),
+     "; VM:=",VMf(HaarWGene(mix)[[lev]],mix),
+     "; ME:=",MEf(HaarWGene(mix)[[lev]],mix))

The generated errors for mixed Weibull:
   KS:= 0.03641575 ; VM:= 0.01425041 ; ME:= 0.001953132
```

Bibliography

[1] Sabyasachi Dash et al. "Big data in healthcare: management, analysis and future prospects". In: *Journal of Big Data* 6.1 (2019). DOI: 10.1186/s40537-019-0217-0.

[2] Riccardo Pascuzzo. *Demonstration heisenberg inequality fourier transform*. https://appliedmath.brown.edu.

[3] A. Papoulis. *The Fourier integral and its applications*. Mc Graw Hill, 1962.

[4] Steven L. Brunton and Jose Nathan Kutz. *Data-driven science and engineering: machine learning, dynamical systems, and control*. Cambridge University Press, 2019.

[5] Elias M. Stein and Rami Shakarchi. *Fourier analysis: an introduction*. Princeton University Press, 2003.

[6] Nasson G.P. *Wavelet methods in statistics with R*. Springer, 2008.

[7] Vikram M. Gadre and Aditya S. Abhyankar. *Multirate Signal Analysis*. Mc Graw Hill, 2017.

[8] *The Haar basis*. https://users.math.yale.edu/pub/wavelets/software/xwpl/html/manual/node28.html.

[9] Geoffrey R. Grimmett and Dominic James Anthony Welsh. *Probability: an introduction*. Clarendon Press, 1986.

[10] *Exploratory Data Analysis: Conceptual Foundations of Empirical Cumulative Distribution Functions | The Chemical Statistician*. https://chemicalstatistician.wordpress.com. June 2013.

[11] *All models are wrong - Wikipedia*. https://en.wikipedia.org/wiki/All_models_are_wrong.