Quantization algorithm

Enrico Maria Di Mauro
matr. 0622701706
Allegra Cuzzocrea
matr. 0622701707

Acronyms

RV $\rightarrow$ Random Variable

PDF $\rightarrow$ Probability Density Function

Notation

X is a RV

x are values assumed by X

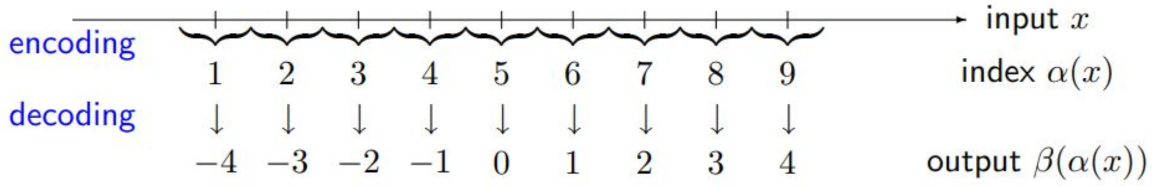
$f(\cdot)$ is the PDF of a RV

SUMMARY

Quantization

Quantization is the process that consists in mapping a continuous set of values into a discrete one

- The **encoder** maps each point x in an input space (like the real line, the plane, Euclidean vector space, function space) into an integer index $i = \alpha(x)$. It produces a partition $S = \{S_i\ ;\ i \in I\}$ of the input space into separate indexed pieces $S_i = \{x : \alpha(x) = i\}$
- The **decoder** maps each index into an output $\beta(i)$. It produces $C = \{\beta(i)\}$



Distortion

- **Sampling**: the sampling theorem guarantees that under appropriate hypothesis it is possible to reconstruct the original signal
- **Quantization**: a **distortion** is introduced.

We focus on **lossy codes**: discrete representations of continuous objects are inherently lossy

In lossy case it is required a **measure of quality** of a quantizer quantifying the loss of the resulting reproduction in comparison to the original one

Distortion measures the difference between the continuous signal and the discrete one

Small distortion → Good quality
Large distortion → Bad quality

Average distortion

To be useful, distortion should be:

- easy to compute
- tractable for analysis
- meaningful for perception or application

Given RV X and its quantized version $Q(X)$, the average distortion is defined as follows:

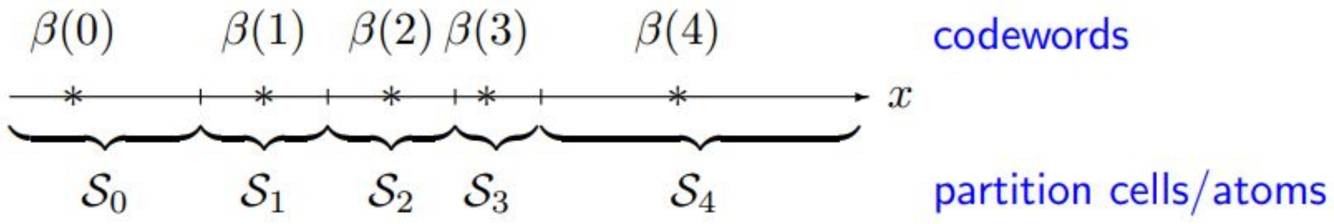
$$D = E_d[(X, Q(X))] = \int_{-\infty}^{+\infty} d(x, Q(x)) f(x) dx$$

A commonly used type of distortion is the **MSE** (Mean Squared Error):

$$D = E_d\left[(X - Q(X))^2\right] = \sum_{i=1}^{N} \int_{R_i} (x - y_i)^2 f(x) dx$$

Scalar quantization

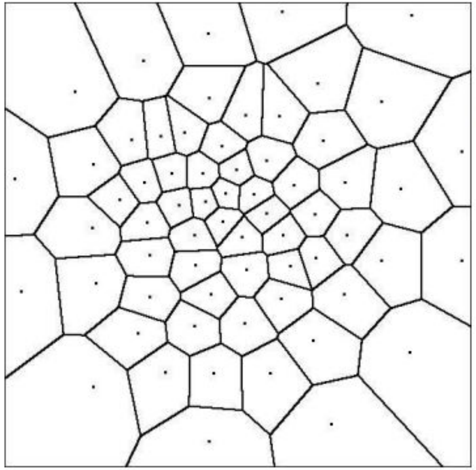
We talk about scalar quantization if the input is scalar



Vector quantization

We talk about vector quantization if the input is a vector

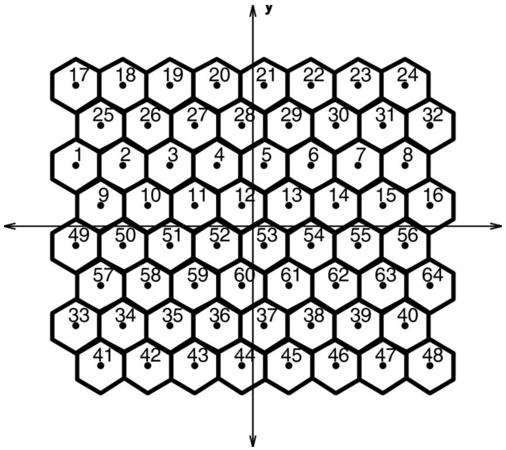
A two-dimensional example of vector quantization is the Voronoi diagram.



Vector quantization

Talking about 2-dimensional spaces, quantization using **hexagons** produces less distortion than quantization using squares. Using squares means separately quantizing abscissa and ordinate, while using hexagons means binding them

To realize a quantizer for two independent RV X and Y, it can be proven that **vector quantization is more efficient than single scalar quantizations**



Rate-distortion theory

The **quantization rate** is the number of different indexes obtained from the quantization

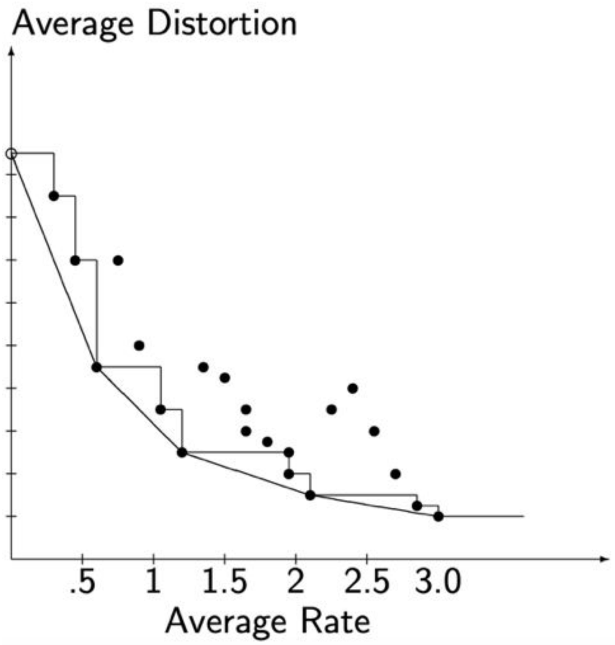
There are fixed rate codes and variable rate codes. For this reason it is better to introduce the **code average rate**

Given a tolerated distortion value, the **rate-distortion theory by Shannon** provides the **minimum rate**

In particular, this theory leads to the computation of a **curve** that gives the relationship between distortion and minimum possible rate

Rate-distortion theory

It can be proven that the rate-distortion curve has the following trend:



The distance between a point and the curve gives the distance from the optimum

Note that a distortion equal to zero corresponds to an average rate equal to the entropy

Quantization noise

A quantizer can be seen as a **noise source**. This source can be divided in 2 parts:

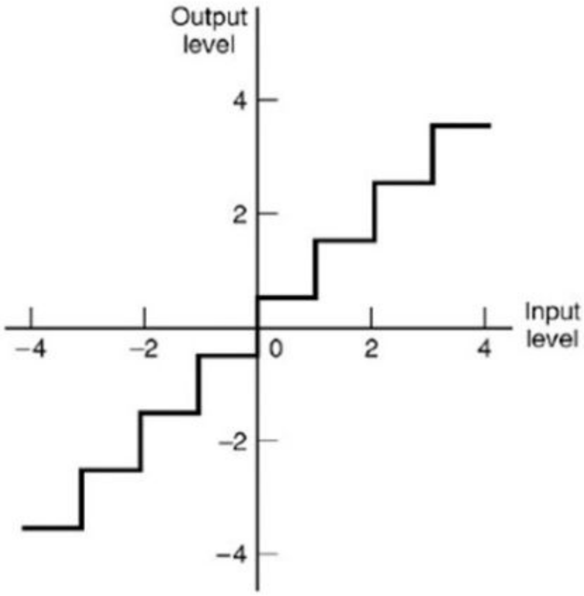
- **granular noise**: it is limited since it concerns finite length intervals. It can not be greater than the semi-amplitude of finite intervals
- **overload noise**: it is unlimited since it concerns infinite length intervals

The **quantization noise** can be defined as the difference between the quantized value and the original one

$$\epsilon = Q(x) - x$$

Uniform quantizer

A uniform quantizer adopts a constant step size between restitution levels



A lot of sources are characterized by non uniform distribution so the uniform quantizer is not convenient

It is simpler to design a uniform quantizer and then model non linearities than to design a new quantizer

For this reason, typically a uniform quantizer is designed and then a distortion function that corrects the quantizer is applied

Lloyd-Max algorithm

The Lloyd-Max algorithm is an iterative quantization algorithm that allows to realize a quantizer if the following assumptions are satisfied:

- the statistical distribution of the input source is known
- the distortion is known (quadratic distortion is commonly used)

Lloyd-Max algorithm

The algorithm is characterized by the following steps:

1. choose random restitution levels
2. given the restitution levels, compute the intervals extremes of the partition. In particular compute the middle point of each interval
3. compute the new restitution levels minimizing the distortion $E[x \mid x \in R_i] = y_i$
4. Iterate the second and the third step until an acceptable distortion value or a certain number of iterations is reached

It can be proven that the steps 2. and 3. are necessary conditions for the optimum quantizer

In some cases they are also sufficient conditions for the optimum quantizer

Implementation structure

```python
def lloyd_max_algorithm(samples, threshold, levels_num=8):
    samples = sorted(samples)
    diff = threshold + 1 #difference between levels and new_levels
    levels = compute_initial_levels(samples, levels_num)
    extremes = []
    iter_num = 0
    while diff > threshold:
        extremes = compute_extremes(samples, levels)
        new_levels = compute_levels(samples, extremes)
        diff = max(np.abs(new_levels-levels))
        levels = new_levels
        iter_num += 1
    extremes = [samples[e] for e in extremes]
    level_extremes = get_level_extremes(levels, extremes)
    return level_extremes
```

`compute_initial_levels`: computes the initial levels by dividing the interval occupied by the source samples equally

`compute_extremes`: computes the extremes for the given levels

`compute_levels`: computes the new levels by minimizing the distortion

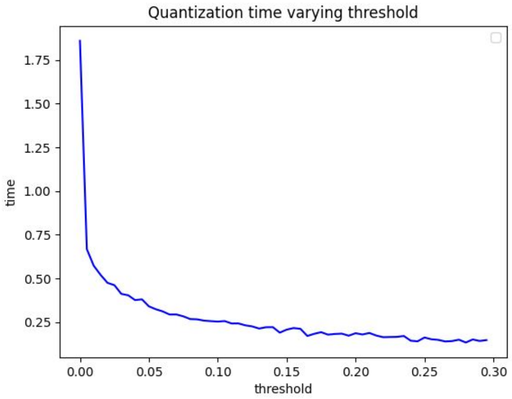
`get_level_extremes`: builds a dictionary that associates to each level its corresponding extremes

Benchmarks and considerations

The curves below have been realized using a source with a normal distribution and consisting of 100000 samples

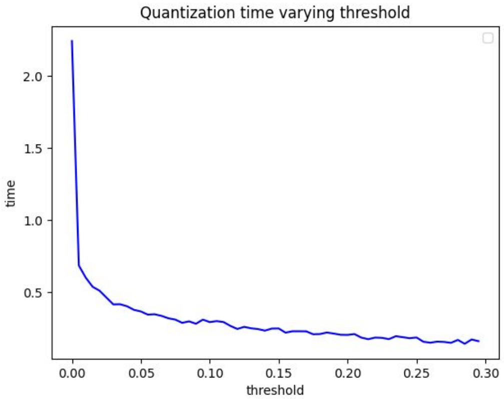
l_1 norm

$$l_1 = |x_1 - x_1'| + |x_2 - x_2'| + \ldots + |x_n - x_n'|$$



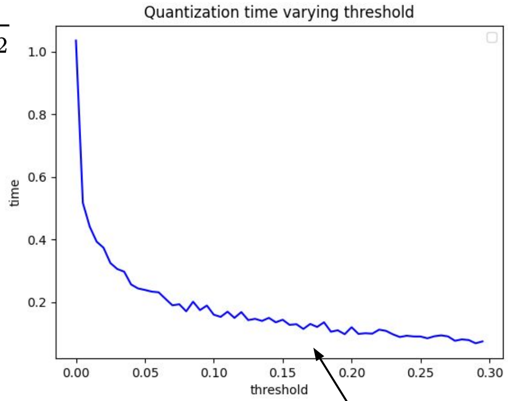
l_2 norm

$$l_2 = \sqrt{(x_1 - x_1')^2 + (x_2 - x_2')^2 + \ldots + (x_n - x_n')^2}$$



l_∞ norm

$$l_\infty = \max(|x_1 - x_1'|, |x_2 - x_2'|, \ldots, |x_n - x_n'|)$$



l_∞ has the best time performances

Benchmarks and considerations

In the previous slide three different norms for the computation of the difference between the new levels and the old ones are compared in terms of time and the l_∞ ones produces the best performances

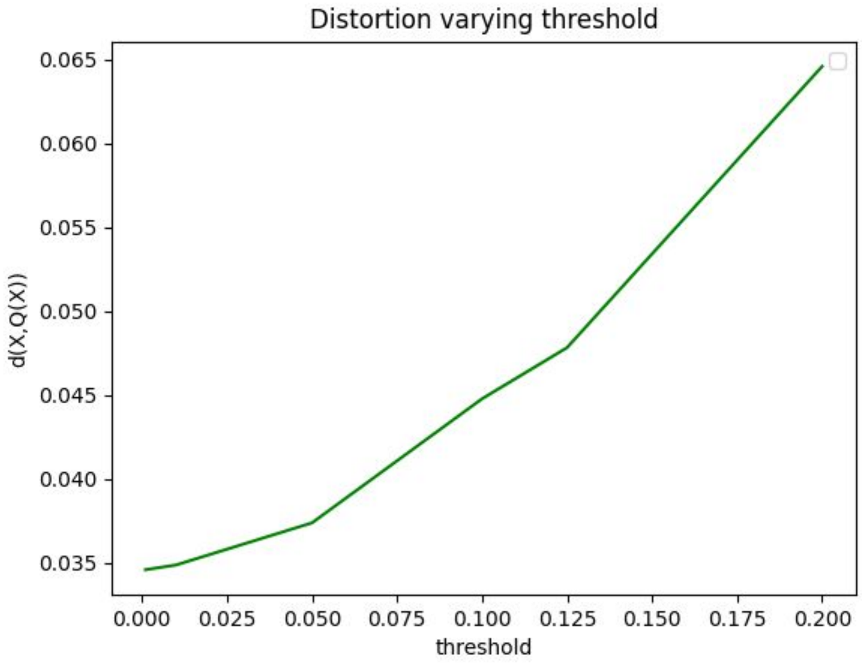
The same thing has been done in terms of distortion but negligible differences have been obtained

For these reasons l_∞ norm has been chosen for the benchmarks in the next slides

Notice that in the implemented version of Lloyd-Max algorithm l_2 norm is used to minimize the distortion, but l_∞ norm is used to compute the difference between the new levels and the old ones. This difference is compared to the threshold to stop the algorithm

Theoretically the same l_2 norm has to be used but after some tests it is possible to conclude that this combination produce better performances

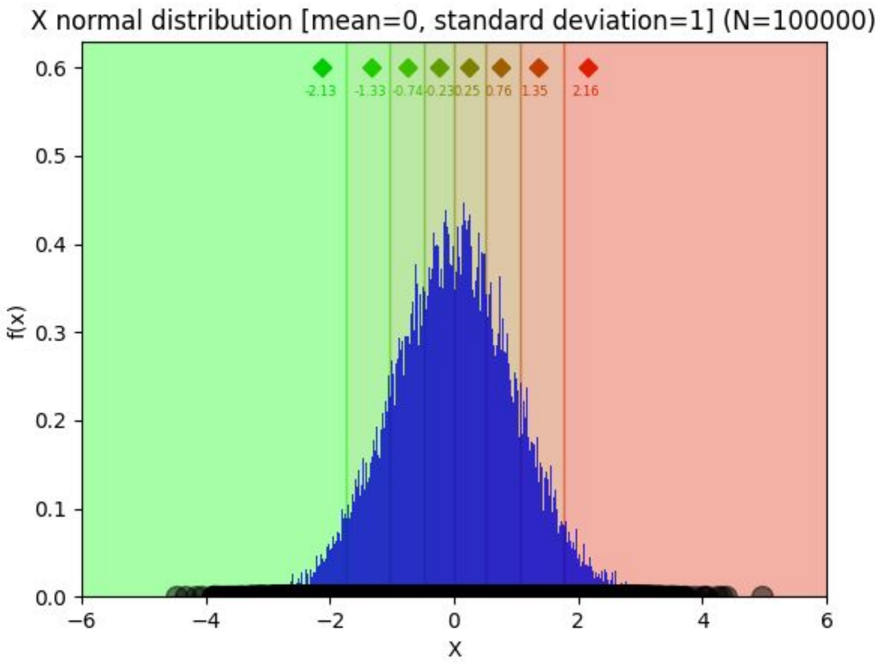
Benchmarks and considerations



The following curve has been realized using l_∞ norm and a source with a normal distribution and consisting of 100000 samples

As expected, as the threshold increases the distortion increases

Benchmarks and considerations

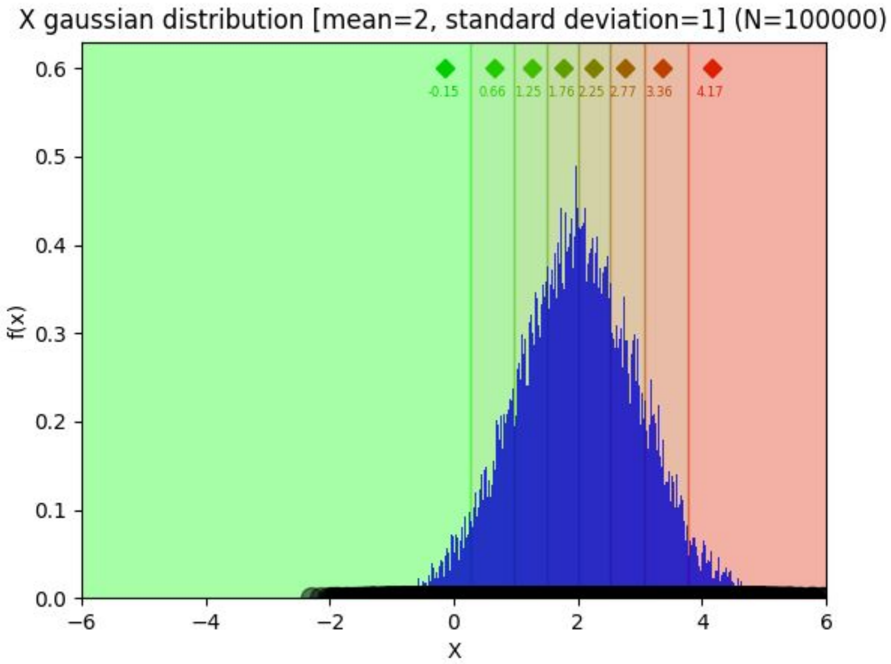


N = 100000
Threshold = 0.000001
Restitution levels = 8

Distortion = 0.03451

The levels follow the distribution curve accurately and they are closer around the mean

Benchmarks and considerations

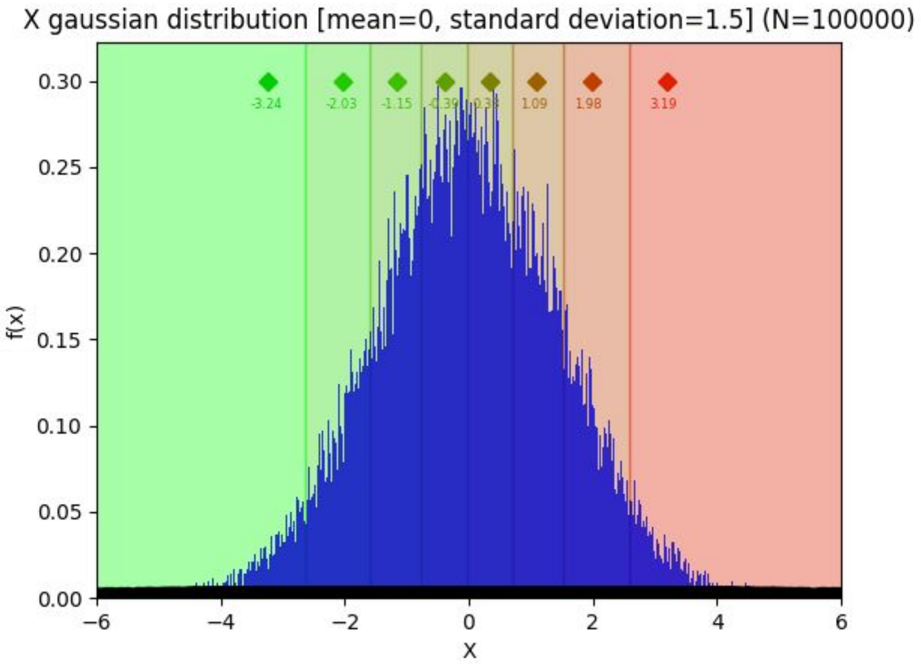


N = 100000
Threshold = 0.000001
Restitution levels = 8

Distortion = 0.03464

The levels follow the distribution curve accurately and they are closer around the mean

Benchmarks and considerations



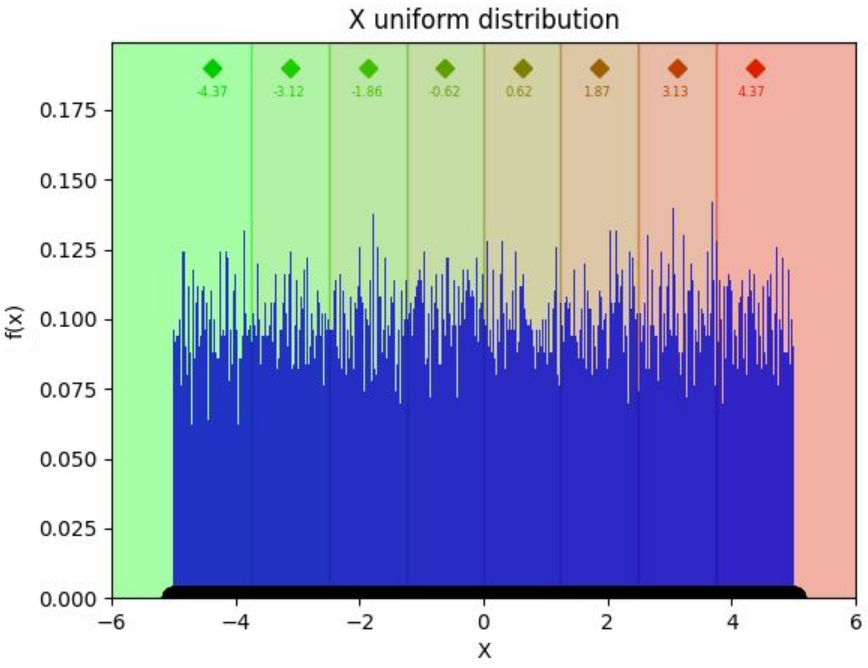
N = 100000
Threshold = 0.000001
Restitution levels = 8

Distortion = 0.07894

The levels follow the distribution curve accurately and they are closer around the mean

The levels range is greater than the case with standard deviation equal to 1. This is because the standard deviation is higher

Benchmarks and considerations



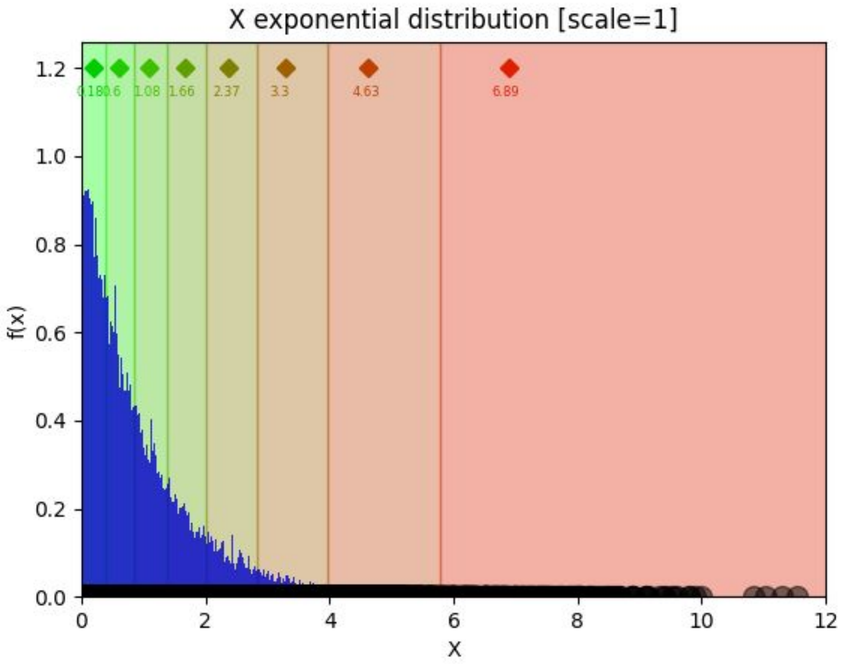
N = 100000
Threshold = 0.000001
Restitution levels = 8

Distortion = 0.12999

The levels follow the distribution curve accurately

The levels are distributed equally

Benchmarks and considerations



N = 100000
Threshold = 0.000001
Restitution levels = 8

Distortion = 0.03171

The levels follow the distribution curve accurately

Thank you for the attention

Enrico Maria Di Mauro
matr. 062270**1706**
Allegra Cuzzocrea
matr. 062270**1707**