Finding maximal meaningful modes in set of real values

Franco Marchesoni Acland
marchesoniacland@gmail.com

February 2021

1 Introduction

In this work we will tackle the problem of detecting maximum meaningful modes on a set of values $\{x_1, \ldots, x_M\}$ such that $\forall i,\ x_i \in \mathrm{R}$. This work will take the *a contrario* detection approach. Thus, a detection method and a background model will be defined. This allows to have a controlled number of false alarms, subject to the background model assumption.

The work is organized as follows. In Section 2, we revise the main concepts of the Course Notes [4]. In Section 3 we pose the problem as a classification one and discuss equivalent formulations. As a background model needs to be defined, we present our choice in Section 4. The formulation of the tests is presented in Section 5 and the experimental results are exposed in Section 6. Finally, we discuss other possible approaches in Section 7.

2 Review

The formulation found in Chapter 4 of the Course Notes assumes L ordered, discrete, possible values in the x-axis of the histogram. We will call these values, indexed by $a \in \{1, \ldots, L\}$, *gbins*. This is a little more general than traditional bins used for images also in Chapter 4, which are included into the definition of *gbins*. The main difference is that *gbins* are not required to have a fixed width, while bins usually share the same width.

An *a priori* distribution can be modeled by the probability distribution $p(X|I)$, where X is any index in $[1, L]$ and I is any previous information available. When the prior is uniform, we have that $p(X|I) = 1/L$, and $p(X \in [a, b]|I) = \frac{b-a+1}{L}$. The total number of intervals $[a, b]$ is $L(L + 1)/2$. To see this one can count: there are L possible intervals of the form $[a = 1, b \geq 1]$, $L - 1$ possible intervals of the form $[a = 2, b \geq 2]$, and so on. Then, for a fixed number of values M, one has that an interval $[a, b]$ is ϵ-meaningful if

$$\mathcal{B}(M, k(a, b), p(X \in [a, b]|I)) < \epsilon \frac{2}{L(L + 1)} \tag{1}$$

where $\mathcal{B}$ denotes the tail of the binomial distribution and k is the number of data points inside $[a, b]$.

By Hoeffding's inequality one has,

$$\mathcal{B}(M, k(a,b), p(a,b)) \leq \exp\left(-M \underbrace{\left[r(a,b) \log \frac{r(a,b)}{p(a,b)} + (1 - r(a,b)) \log \frac{1 - r(a,b)}{1 - p(a,b)}\right]}_{H_1([a,b])}\right) \tag{2}$$

where $r(a,b) = \frac{k(a,b)}{M} > p(a,b) = p(X \in [a,b]|I)$. Defining $H([a,b]) = H_1([a,b])\mathbb{1}_{p(a,b)<r(a,b)}$ one finally concludes that an interval $[a,b]$ is ϵ-meaningful in a large deviations sense if

$$H([a,b]) > \frac{1}{M} \log \frac{L(L+1)}{2\epsilon}. \tag{3}$$

A symmetrical derivation can be done to find gaps. Modes are defined as meaningful intervals that do not contain any gap. A maximal meaningful mode is defined as the most meaningful mode including any of the points it contains. We will use the a similar definition of meaningfulness soon, with the difference that the number of tests will not depend on a user defined number of bins L.

3 Maximum meaninful modes as a classification problem

In Chapter 4 some definitions were presented along with the methods. We now frame the problem as a classification problem to ilustrate the extrapolation capabilities of the methods.

Given data X, we want a function $f : x \mapsto \{0, 1\}$ where 1 represents a maximum meaningful mode. In the setting of Chapter 4, there is the implicit assumption that $f(x) = \mathbb{1}_{\bigcup_i x \in [a_i, b_i]}$, i.e. that $f(x) = 1$ for those x in some of the intervals $[a_i, b_i]$. For a new observation x', we just need to use $f(x')$ to classify it (e.g. if there were a new pixel value in the gray level quantization problem to be quantized). This is flexible enough to classify the points in the dataset and new points as well.

For continuous data X that was classified using some $f : x \in X \mapsto \{0, 1\}$, an extension to other points in $\mathbb{R}$ can be achieved easily. This is, given a classification over a set of real numbers X, one can extend it to new real numbers by finding the implicit intervals in the data. To do this, one must take an interval for every uninterrupted sequence of ordered points that were classified as a mode. There is some ambiguity, as for consecutive points $x_{s_i}, x_{s_{i+1}}$ such that $f(x_{s_i}) = 0, f(x_{s_{i+1}}) = 1$, the start of the interval a can be $x_{s_i} < a \leq x_{s_{i+1}}$. This ambiguity holds for the end points too, but it can be arbitrarily solved by taking the mean point between x_{s_i} and $x_{s_{i+1}}$.

This method returns the same than in Chapter 4: a set of non-overlapping intervals. It is easy now to define an f over all reals in an equal manner. The advantage of the formulation of the problem in terms of f is that it makes explicit three things: i) classifying real points is enough to create an extrapolation function f, ii) such a extrapolation is more general than just classifying in-sample and iii) one could directly define an f without loss of generality. In sum, defining a function f, providing intervals or simply classifying the datapoints are all equivalent, as one can go from one to the others for most cases (except if two identical observations are classified differently).

4 *a priori* distribution

When setting a distribution one usually estimates its parameters from data. The maximal entropy distribution is the one that is the least informative, i.e. it introduces the least amount of prior information. For the case of a support contained in an interval $[A, B]$, the maximal entropy distribution is the uniform distribution. For the case of unbounded support and known mean and variance, the maximal entropy distribution is the normal or central distribution. For simplicity we will restrict our analysis to the uniform distribution, estimating its parameters from the data. This decision is arbitrary, as the Laplace distribution, for instance, is the maximal entropy distribution when the average absolute deviation is known. Many parameters for many distributions can be estimated from data, and each of those distributions could be of maximal entropy when we assume that some quantity (support, variance, average absolute deviation, etc.) can be confidently estimated from the data.

4.1 Central distribution

For the central distribution case, the unbiased estimates of the parameters are:

$$\hat{\mu} = \bar{x} = \frac{1}{M}\sum_i x_i, \qquad \hat{\sigma^2} = \frac{1}{M-1}\sum_i (x_i - \bar{x})^2. \tag{4}$$

This derivation can be found in many places (e.g. here).

4.2 Uniform distribution

If the *a priori* distribution is assumed to be a uniform distribution $\mathcal{U}[\theta_1, \theta_2]$ with unknown parameters θ_1 and θ_2, a problem that arises is estimating θ_1 and θ_2. We can expose a simple derivation proposed by [1]: When observing the data X, one can create a sorted version $X_s = [x_{s_1}, \ldots, x_{s_N}]$. Set $\mathbf{w}$ to be the vector with the differences $\mathbf{w} = [x_{s_1} - \theta_1, x_{s_2} - x_{s_1}, \ldots, x_{s_N} - x_{s_{N-1}}, \theta_2 - x_{s_N}]$. The elements w_i are uniformly distributed and sum to $\theta_2 - \theta_1$ (see [3] for more details). The expected value for all of them is, by symmetry, $\frac{\theta_2 - \theta_1}{N+1}$.

This means that $\mathrm{E}[w_1] = \cdots = \mathrm{E}[w_{n+1}] = \frac{\theta_2 - \theta_1}{N+1}$, and then

$$\mathrm{E}[w_1] = \mathrm{E}[x_{s_1} - \theta_1] = \frac{\theta_2 - \theta_1}{N+1} \Rightarrow (N+1)\mathrm{E}[x_{s_1}] = N\theta_1 + \theta_2$$

$$\mathrm{E}[w_{N+1}] = \mathrm{E}[\theta_2 - x_{s_N}] = \frac{\theta_2 - \theta_1}{N+1} \Rightarrow (N+1)\mathrm{E}[x_{s_N}] = \theta_1 + N\theta_2$$

Then,

$$N(N+1)\mathrm{E}[x_{s_N}] - (N+1)\mathrm{E}[x_{s_1}] = N\theta_1 + N^2\theta_2 - N\theta_1 - \theta_2 =$$
$$= (N^2-1)\theta_2 = (N+1)(N-1)\theta_2$$
$$\Rightarrow N\mathrm{E}[x_{s_N}] - \mathrm{E}[x_{s_1}] = (N-1)\theta_2$$

Thus we can have an unbiased estimator of θ_2 by noting that

$$\mathrm{E}[\hat{\theta}_2] = \mathrm{E}\underbrace{\left[\frac{Nx_{s_N} - x_{s_1}}{N-1}\right]}_{\hat{\theta}_2} = \theta_2 \tag{5}$$

Equivalently,

$$\mathrm{E}[\hat{\theta}_1] = \mathrm{E}\underbrace{\left[\frac{Nx_{s_1} - x_{s_N}}{N-1}\right]}_{\hat{\theta}_1} = \frac{N}{N-1}\frac{N\theta_1 + \theta_2}{N+1} - \frac{1}{N-1}\frac{\theta_1 + N\theta_2}{N+1} = \theta_1 \tag{6}$$

We are now able to define the *a priori* uniform distribution as

$$p(X = x) = \begin{cases} \left(\hat{\theta}_2 - \hat{\theta}_1\right)^{-1} & \hat{\theta}_1 \leq x < \hat{\theta}_2 \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

where $\hat{\theta}_2 - \hat{\theta}_1 = \frac{N+1}{N-1}(\max_i x_i - \min_i x_i)$.

A short comment must be adressed now. This is not a pure *a priori* distribution, as it depends on the data. The true *a priori* knowledge being imposed is "we can confidently estimate the finite support of the generating distribution from the data". This part showed how to estimate such support in an unbiased way.

5 Formulation

If M is the number of real points, we will take as candidates the intervals between pairs of data points $x_i \leq x_j$. Although theoretically we could assume strict inequality because the probability of having the exact same value is 0, in computers this probability is non-zero and there can be cases (inclination of computer drawn parallel lines) in which the data points can be repeated, definitely not corresponding to the background model. This is one of the cases we want to detect. These points should not be considered in the estimation, that will be done as $r(x_i, x_j) = \frac{k(a,b)}{M-2}$, with $k(a,b)$ being the number of data points that satisfy $x \in [x_i, x_j],\ x \neq x_i, x_j$. The number of tests is the number of combinations of M taking 2, which is $M(M-1)/2$. An interval is then meaningful if

$$H([x_i, x_j]) > \frac{1}{M} \log \frac{M(M-1)}{2\epsilon}. \tag{8}$$

.

Gaps are defined equivalently. Every time we test the meaningfulness of an interval, we are testing it for gap meaningfulness too. This does not change the number of tests for meaningful intervals, this is just a detection that is done in parallel. We set the ϵ for gaps to 1, because using the same ϵ could end in undesired results: making ϵ lower makes detection harder (by NFA restriction) and easier (by detecting less gaps) at the same time.

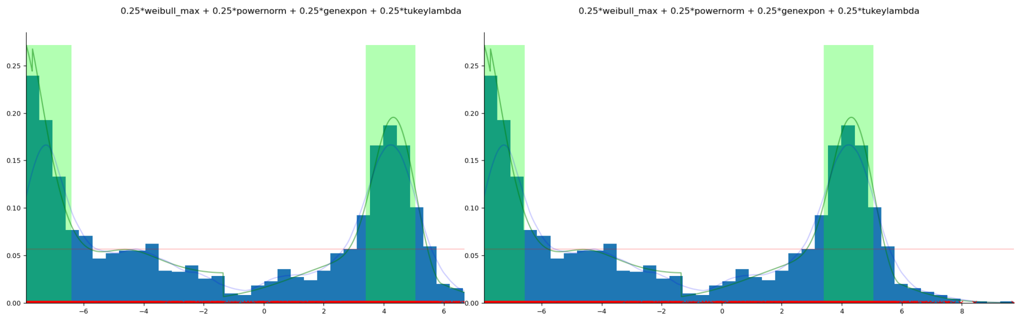
The NFA of maximal meaningful modes is not straightforward to deduce. It builds upon the concepts of meaningful interval, meaningful gap and maximality. We can note, however, that the NFA of maximal meaningful modes is upper bounded by the NFA of meaningful intervals: if the average number of false alarms is under ϵ for meaningful intervals, then the expected number of false alarms is also less than ϵ for maximal meaningful modes, because a meaningful interval is needed to detect a maximal meaningful mode. In other words, from data generated by the background model, we can only falsely detect a maximal meaningful mode if we have falsely detected a meaningful interval.

For large M the number of tests explodes because it is of order M^2. This makes it difficult to evaluate every interval x_i, x_j. Although we will not change the number of tests, we will describe some possible heuristics.

5.1 Heuristics

The main heuristic to be used is based on Kernel Density Estimation (KDE). With a predefined kernel one can estimate the probability distribution from the data and then find maximal meaningful modes on this PDF. Unfortunately, this heuristic relies on a `bandwidth` parameter that should be manually tuned. Automated procedures like cross validation showed, both from our experiments and researcher's experiences, that the the final `bandwidth` found with this procedure tends to be too large. In this work we have set `bandwidth=1`, default value in scikit-learn and visually appealing. Other decision to make is the kernel choice. We choose Epanechnikov's kernel because it is the best one between those provided in the standard data science libraries. Although it is the best non-negative kernel, lower errors can theoretically be achieved by using larger classes of kernels (see [2]). This is, however, good enough for our purposes.

The heuristic method starts at every local maximum of the PDF and starts expanding around it. This is, it is a speeded up version of the function `get_some_meaningful_intervals` that is itself a fast version of the function `get_all_meaningful_intervals`. The last one provides all meaningful intervals, its "fast" version provides only some (but discarding those that will not be maximal meaningful gaps). We have chosen to work in speeding up these functions because is the first step of the sequence: one has to find meaningful intervals before



(a) Fast implementation with heuristics

(b) Desired (slow) result

Figure 1: Comparison of results for heuristics vs reference. Legend: solid blue - data histogram, shaded green - detected maximal meaningful modes, red crosses - datapoints, blue line - estimated PDF via KDE, green line - generating PDF, red line - a priori distribution.

doing anything else. Furthermore, the remaining functions are sensible to the numbers of meaningful intervals detected.

More specifically, the proposed method estimates a PDF and uses it to get the estimated probability of each observed point. Next, we find local maxima, and for each local maxima we try to expand it to the left and we check if the meaningfulness improves. If it does not, we try to expand it to the right. As there are cases in which no improvement is found, we give the method a number of `livess`, and every time we lose one life we increase the expansion step in an exponential way. With 100 lives, our heuristic method yields a $\approx \times 50$ speedup (from $339.67s$ to $6.26s$) when compared to the first function mentioned. This timed example is shown in Figures 1a and 1b.

6 Experiments

We now illustrate the methods working for three different problems: gray-level quantization, synthetic histograms and orientation matching.

6.1 Gray-level quantization

This problem was presented in Chapter 4. The maximal meaningful modes are potentially good candidates to be painted by some gray color. We implemented one version the discrete method proposed in Chapter 4. We can see the implementation works well on this problem in Figure 2.

6.2 Synthetic data

We created an script that mixes a large variety of probability distributions and generates data from them. We used this script to analyze the speed of the

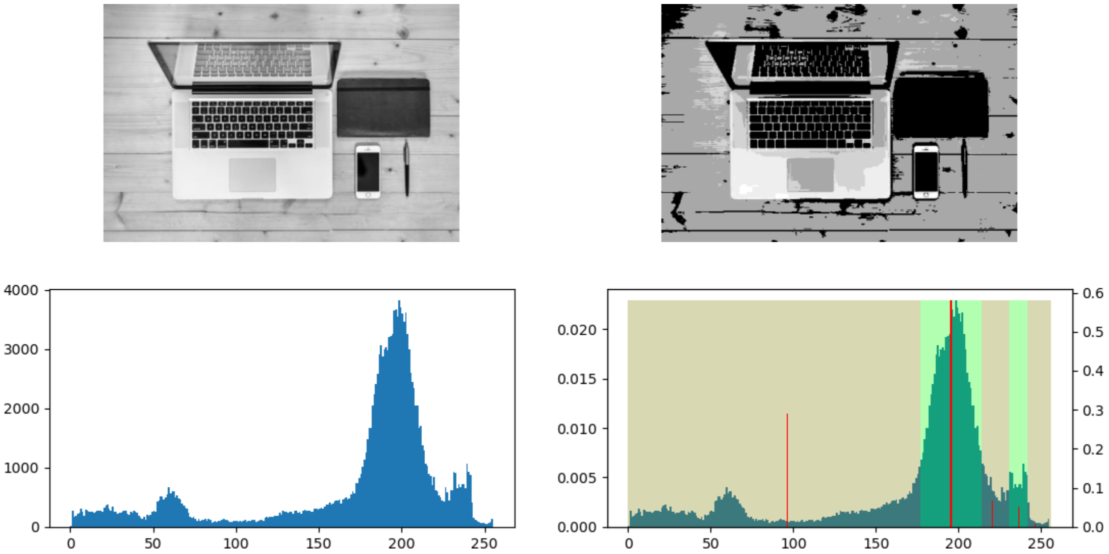


Figure 2: Gray-level quantization example. Legend: solid blue - image histogram, shaded gray - non maximal meaningful modes, green - maximal meaningful mode, red - gray value (mean).

heuristic presented above, because the orientation matching problem did not yield a large number of angles. Examples can be seen, for the fast version using heuristics, in Figures 3 and 4.

6.3 Orientation matching

This application required to add a `circular` parameter to the method, as the domain of the data was known $[-\pi/2, \pi/2]$. The idea is that the main directions in the image will be detected by the method. We see that this happens in Figures 5 and 6. We used the full version of the method in every case in which the number of edges is lower than 1000.

6.4 Code

The code is fast enough to run all the experiments in one minute on a laptop under 500EUR, but *it is not production code*. There are many code "enhancements" to make. The logic for some method is not obvious, but to keep the extension of this document controlled we refer the reader to the code to see implementation details (circle problem, search methods, parameters, etc.). The code is publicly available in this repository (click me).

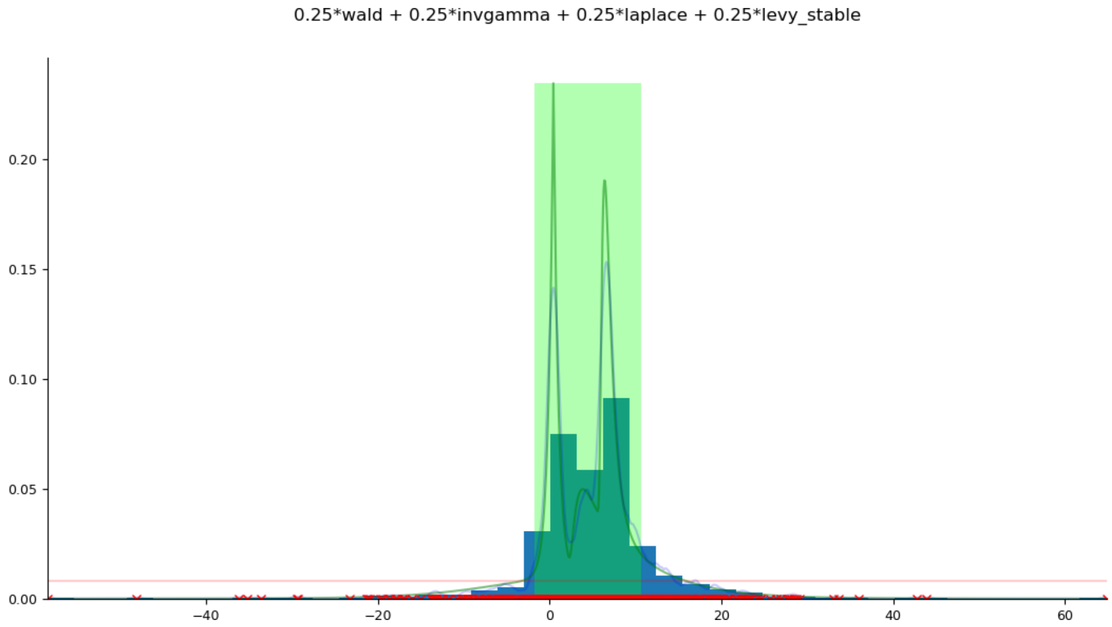


Figure 3: Synthetic data. Legend: solid blue - data histogram, shaded green - detected maximal meaningful modes, red crosses - datapoints, blue line - estimated PDF via KDE, green line - generating PDF, red line - a priori distribution.

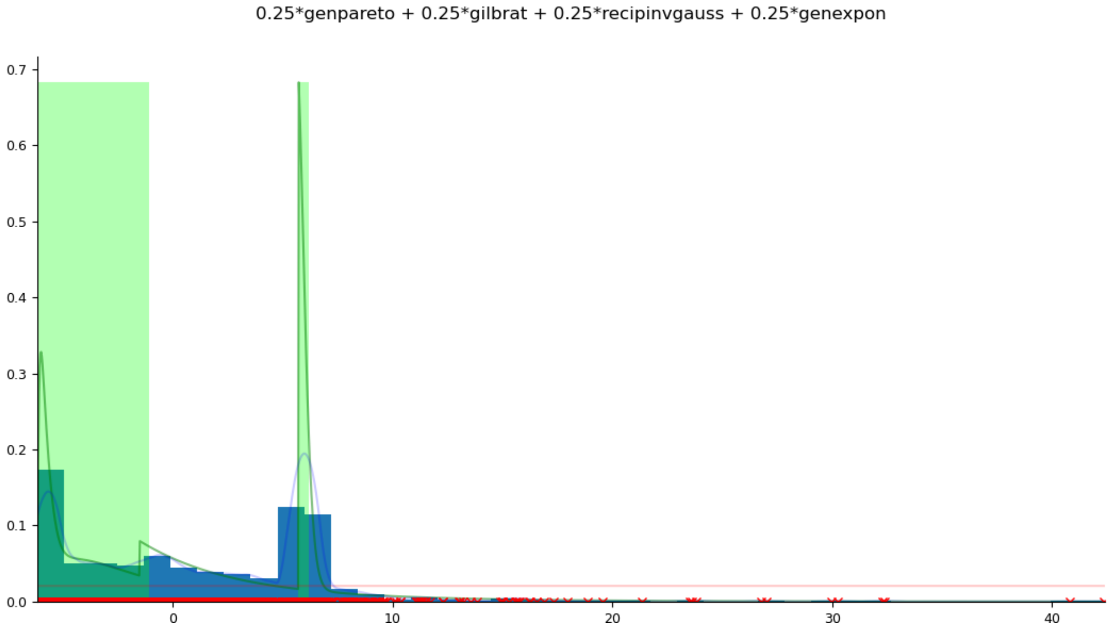


Figure 4: Synthetic data. Legend: solid blue - data histogram, shaded green - detected maximal meaningful modes, red crosses - datapoints, blue line - estimated PDF via KDE, green line - generating PDF, red line - a priori distribution.

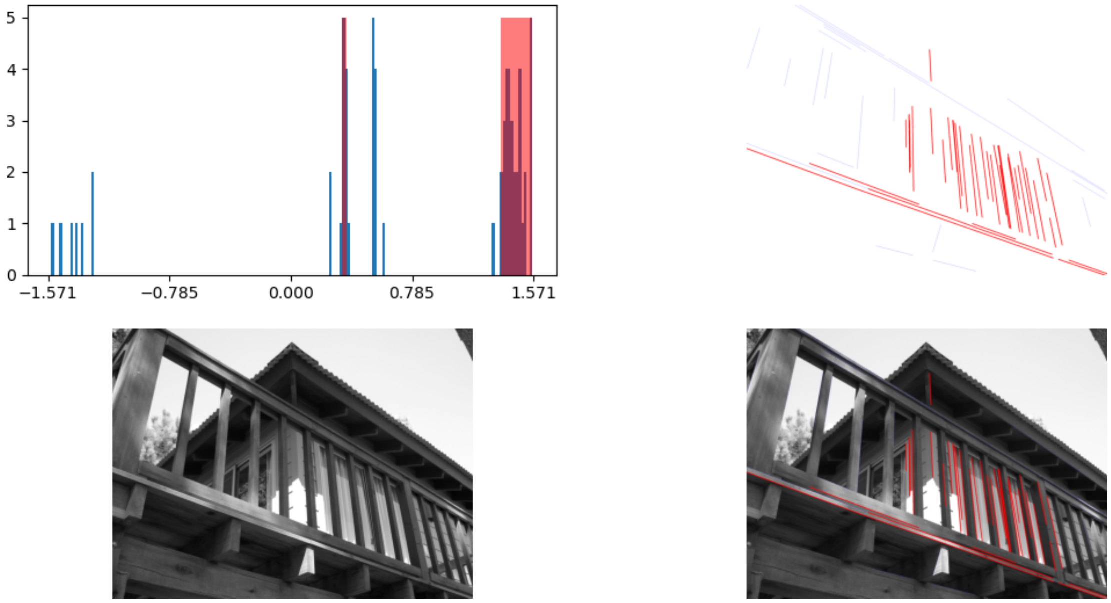


Figure 5: Orientation matching.

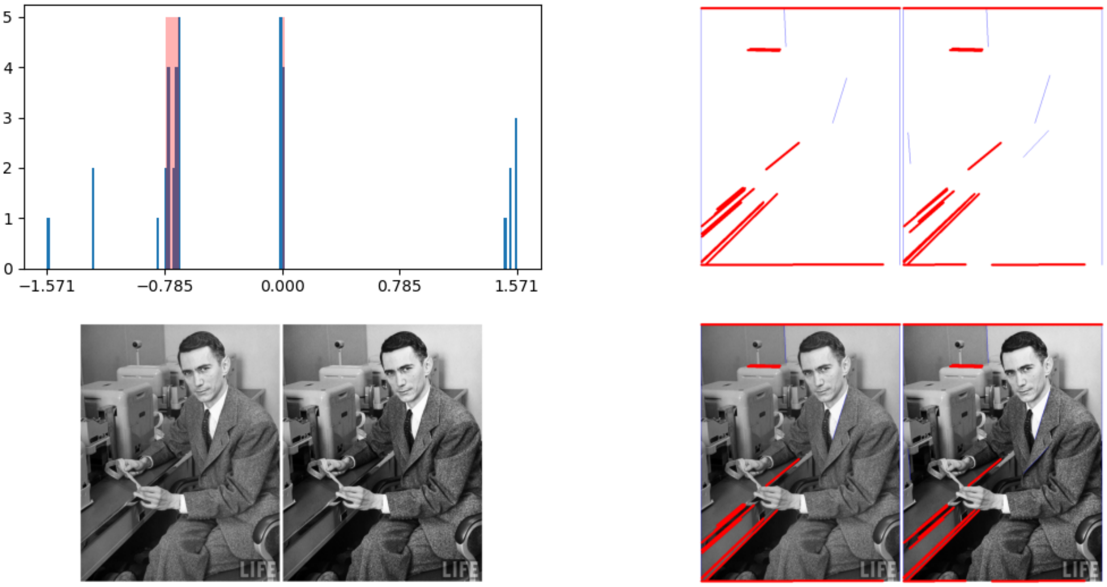


Figure 6: Orientation matching. Claude Shannon.

7 Other Approaches

Although we used an *a contrario* approach based on *gbins*, it is worth mentioning some other ad-hoc methods that could be used with the objective of detecting meaningful modes.

- First, one could use a clustering approach. Take, for instance, the DBSCAN algorithm, with parameters `eps` and `min_samples`. This algorithm assigns only some points to clusters, which is similar to the result we will get with our method. The number of false alarms could be studied experimentally: after defining a background model and chosen the two parameters, one could run simulations and count the number of false alarms under the background model.

- On the other hand, a density estimation approach could be used. For instance, using kernel density estimation. This approach gives us a probaiblity density function (PDF) and its formula, from which we could make numerical or analytic analysis to determine the maximum meaningful modes.

- Finally, taking bins is the default choice, although the objective of this project is to find another way that does not rely on a user-defined number of bins. This approach was developed in Chapter 4.

References

[1] FelixCQ (https://math.stackexchange.com/users/12767/felixcq). *Unbiased Estimator for a Uniform Variable Support.* Mathematics Stack Exchange. URL:https://math.stackexchange.com/q/60504 (version: 2011-08-30). eprint: `https://math.stackexchange.com/q/60504`. URL: `https://math.stackexchange.com/q/60504`.

[2] Chill2Macht (https://stats.stackexchange.com/users/113090/chill2macht). *If the Epanechnikov kernel is theoretically optimal when doing Kernel Density Estimation, why isn't it more commonly used?* Cross Validated. URL:https://stats.stackexchange.com/ (version: 2018-11-16). eprint: `https://stats.stackexchange.com/q/377418`. URL: `https://stats.stackexchange.com/q/377418`.

[3] whuber (https://stats.stackexchange.com/users/919/whuber). *Generate uniformly distributed weights that sum to unity?* Cross Validated. URL:https://stats.stackexchange.com/q/140 (version: 2014-10-20). eprint: `https://stats.stackexchange.com/q/14068`. URL: `https://stats.stackexchange.com/q/14068`.

[4] Rafael Grompone von Gioi Jean Michel Morel. *Detection Theory and Industrial Applications.* Course notes.