Campus Internacional
excelencia UAM CSIC+

UNIVERSIDAD AUTONOMA
DE MADRID

Máster en

Fisica Teórica: Astrofísica y Física del Cosmos.

Fully Adaptive Bayesian Algorithm for Data Analysis FABADA

Pablo Manuel Sánchez Alarcón

FACULTAD DE CIENCIAS

Yago Ascasibar Sequeiros
Universidad Autónoma de Madrid
Departamento de Física Teórica

Trabajo Fin de Máster. Curso 2019 - 2020

Universidad Autónoma de Madrid

Fully Adaptive Bayesian Algorithm for Data Analysis FABADA

Pablo M. Sánchez Alarcón, Universidad Autónoma de Madrid. *pablom.sanchez@estudiante.uam.es*
Yago Ascasibar Sequeiros, Department of Theoretical Physics, UAM. *yago.ascasibar@uam.es*

***Abstract*—The aim of this work is to implement a new automatic procedure to the previous developed algorithm being able to generate a new automatic noise reduction method. This method emerged from a new point of view of Bayesian statistics that allows us to evaluate the possible smoothed models of the data via Bayesian inference, obtaining a model that is statistically compatible with them. The new implementation evaluates the probabilities of the smooth models produced to elaborate the estimation of the noisy data. Throughout this work, we will explain in detail the mathematical development that underpins the methodology of the algorithm together with the new automation process, and the two variations that arose from this new approach, FABADA's single and Bayesian model generations. Then we will evaluate the implementation of the new algorithm, in terms of the mean square error. To assess its performance we will compare the quality of the recovers from a battery of real astronomical spectra and images with different standards and novel methods. The source code needed to reproduce all the results presented in this report along the implementation of the method is publicly available at https://github.com/PabloMSan/FABADA.**

***Index Terms*—noise reduction, Bayes' inference, peak signal to noise ratio.**

I. Introduction

THE acquisition of any kind of experimental data is afected by several sources of statistical error, which ultimately translate into a random noise component in the signal to be recorded. There are different types of noise depending on their physical origin, both related to electronic (thermal noise, fluctuations) and mechanical (defective lenses, antennas, etc.) measuring devices. In astronomy, for example, errors can be produced in the acquisition of the images due to defects in the optics of the telescopes and also in the reading process of the detector (typically a CCD) in charge of transforming the light captured by the telescope into an electrical signal. The noise introduced can sometimes be noticeable enough to bury our signal, and that is why different algorithms have been developed to reduce its impact and try to recover the information in the most reliable way possible.

Smoothing, where measurements are weighted at nearby spatial or temporal points (using different schemes to assign weights), is one of the most popular techniques to mitigate the effects of random noise (e.g., Cleveland, 1979; Savitzky and Golay, 1964). Nowadays there is a large number of smoothing algorithms, based on many different techniques, such as central moving average, data grouping/segmentation (e.g., Dabov et al., 2007), fitting smooth functions, different types of statistical analysis, partial differential equations, wavelenght transformation filters, linear and nonlinear filtering, and for the later epochs artificial neural networks (see Jing et al., 2019). Generally, all these methods rely on some explicit or implicit assumptions about the true (noise-free) signal in order to separate it properly from the random noise. An usual assumption is that the signal being retrieved varies gradually along the dimensions on which the process is carried out and that the data must fit a smooth function (see Katkovnik et al., 2006). Generally, this distribution is sought that fits the experimental data and the deviation of the measurements from the proposed model being evaluated using one of the methods mentioned above. Many techniques analyze the probability that the data correspond to a random Gaussian realization of the model that attempts to describe the underlying signal (e.g., El Helou and Susstrunk, 2020), but few or none do so from a strictly Bayesian point.

In the work of last year Bachelor's final thesis we used Bayesian statistical inference to evaluate and combine different models being able to elaborate, through statistical analysis, a smoothing that better fits the data, obtaining a new method of noise reduction. This new Bayesian technique have been remodel to incorporate an automatically selection criteria, which is based in the statistical properties of the models produced converting this new technique in a non-parametric method.

For this purpose, several algorithms have been developed, implemented in Python language, and their ability to recover the underlying signal from an experimental data set with its corresponding errors has been evaluated and compared with different standard methods. Although the ultimate motivation of the study is directed to the application in the field of astronomy, the algorithm has been focused in an absolutely general way, and it is possible to generate a smoothed model for any type of data.

II. FABADA

For the development of this new algorithm, we assume that the real image has been contaminated with Gaussian distributed white noise. This lets us generate a likelihood function to evaluate the models produced. With the first set of a constant prior probability, we iterative evaluate different smooth versions of the posterior probabilities until a certain condition is reached. Then, this new method can be separated into two different steps. The first one, the iterative models, conceived in the previous work, generates several smooth versions of the data and evaluates them with Bayes inference. And the model selection, which is the new implementation developed in this work, evaluates the evidence produced in the previous step imposing an evidence-based criterion to stop the smoothing process and generate the estimation of the real data.

A. Iterative models

$FABADA$ appears as a new technique for noise reduction either for one dimensional data sets, such as astronomical spectra and for two dimensional images which can be used as a previous step to the analysis of such data. Focusing on the mechanism that makes this algorithm work, $FABADA$, as his name indicates, is a fully automatic algorithm which it only takes as inputs the data set and his error associated, $\vec{I} = \{\vec{D}, \vec{E}\}$, which can have one or two dimensions. In order to simplify, the method will be explained in one dimensional since the two dimensional is analogous. Therefore our data set with his associated errors has a length N_D, i.e.,

$$\vec{D} = \{D_d\}_{d=1,..,N_D} \qquad \vec{E} = \{E_d\}_{d=1,..,N_D} \tag{1}$$

where D_d is the d-th value of the data set and E_d his associated error, which can be any type of information.

$FABADA$ is a general purpose algorithm, which means that you can introduce any kind of data to it, with the same aim of achieving a smoothed version of the data set. Due to this property we will not have any previous knowledge about these data. In other words, our prior probability distribution will be homogeneous in the range of all possible values, i.e. $p(\vec{D}) = 1$ using the annotation describe in the table I. If $\vec{M} = \{M_d\}$ is the model we want to create, which will be a well fitted smoothed version of the data, then we have that the prior probability distribution should be $p(M_d) = 1$ (all values are equally possible) for all every data d. As we will see later, we are interested in the prior probability of our model being a probability of its own, and the one we have just selected is not, since the integral over the whole real line diverges. Therefore, it is necessary to limit the range of possible values for M_d to a finite range. As we do not have any prior information about the data we are going to smooth, we adopt that

$$p(M_d) = \frac{1}{\max(\vec{D} + 3\vec{E}) - \min(\vec{D} - 3\vec{E})} \equiv p_0 \tag{2}$$

which, despite not being strictly Bayesian (since you access information from the data to select the prior), allows us to have our own prior probability of the model, commensurate with any range of values (and units) provided by the user.

Assuming that the measurement errors are Gaussian, the likelihood function $\mathcal{L}$ for our model is given by

$$p(\vec{D}|\vec{M}, \vec{E}) = \mathcal{L}(\vec{D}|\vec{M}, \vec{E}) = \prod_{d=1}^{N_D} \frac{e^{-\frac{(D_d - M_d)^2}{2 \cdot E_d^2}}}{\sqrt{2\pi} \cdot E_d}. \tag{3}$$

In this first iteration, where the prior probability distribution for $\vec{M}$ is uniform, the Bayes' theorem indicates that the posterior distribution is

$$\begin{aligned} \mathcal{P}(\vec{M}|\vec{D}, \vec{E}) &= \frac{p(\vec{M}) \cdot \mathcal{L}(\vec{D}|\vec{M}, \vec{E})}{\mathcal{E}} = \\ &= \prod_{d=1}^{N_D} \frac{e^{-\frac{(D_d - M_d)^2}{2 \cdot E_d^2}}}{\sqrt{2\pi} \cdot E_d} \end{aligned} \tag{4}$$

is, as expected, a Gaussian centered on the data set itself (i.e. the expected value of M_d is $\mu_d \equiv \langle M_d \rangle = D_d$) with variances $V_d = E_d^2$ determined by the corresponding errors. In this case, the overall evidence $\mathcal{E} = \int p(\vec{M}) \cdot \mathcal{L}(\vec{D})\, d\vec{M}$ of our model is simply $\mathcal{E} \simeq p_0^{N_D}$.

Furthermore, we want to see if a smoothed version of this first model would be statistically consistent with the data. To check this, we apply the central moving average

$$\mathcal{MM}(\mu_d) = \frac{\mu_{d-1} + \mu_d + \mu_{d+1}}{3} \tag{5}$$

as a local smoothing filter. The $\mathcal{MM}$ averages the value of the neighbour points, helping us to build a new smoothing model from the previous one. In other words, we are going to use the information from the adjacent points to update our knowledge about the correct value of M_d. In principle, by updating our prior distribution, we are changing our knowledge about the data and we would have to rebuild the input data set in order to update the posterior distribution. Once again, we are forsaking the strict Bayesian philosophy (the prior distribution should be, as its name indicates, totally independent from the data) for the sake of a practical result, and this is precisely where the magic of this algorithm lay; we are not going to change the data, but the models, repeating the process a number N_i of iterations until a better fit is achieved. On section II-B we will discussed the different criteria used to automatically select the number of iterations N_i to achieve the best fit. For now, lets supposed that we already know the value of N_i. In this way, the inputs of our algorithm are $\vec{I} = \{\vec{D}, \vec{E}\}$, and the set of models is extended to

$$\begin{aligned} \vec{M} &= \{M_d^i\}_{d=1,\ldots,N_D}^{i=0,\ldots,N_i} \\ \vec{\mu} &= \{\mu_d^i\}_{d=1,\ldots,N_D}^{i=0,\ldots,N_i} \\ \vec{V} &= \{V_d^i\}_{d=1,\ldots,N_D}^{i=0,\ldots,N_i} \end{aligned} \tag{6}$$

to include the iterations $i = 1, ..., N_i$. Taking into account the *mise en place* $i = 0$, the dimensionality of the problem (total number of values for each variable) amounts to $N_d \times (N_i + 1)$. Once we have established the basis of the Bayesian inference that we are going to apply, we can let the algorithm iterate on the data and calculate the successive models, its expected values and its variances. For each iteration $i > 0$, the prior probability distribution

$$p_i(M_d^i) = \frac{e^{-\frac{(\mathcal{MM}(\mu_d^{i-1}) - M_d^i)^2}{2 \cdot V_d^{i-1}}}}{\sqrt{2\pi \cdot V_d^{i-1}}} \tag{7}$$

is updated as a Gaussian centered on the smoothed values of the expected value of the previous iteration, using the variance as a measure of our uncertainty. For the case of the next iteration $i = 1$ we will have a prior centered on a first smoothing of the data (since $\mu_d^0 = D_d$) and we expect that M_d^i will vary in an environment of the order of $\sqrt{V_d^0} = E_d$ around that value. Through Bayesian inference we will be able to compute the posterior probability distribution of our smoothed model by applying the Bayes'

	Symbol	**Definition**
DIMENSIONS	d	Length of the data set $\{1, 2, .., N_D\}$
	i	Iteration number $\{1, 2, , .., N_i\}$
INPUTS	$\vec{I} = \{\vec{D}, \vec{E}\}$	Set of inputs of the algorithm
	$\vec{D} = \{D_d\}$	Data set to smooth
	$\vec{E} = \{E_d\}$	Errors associated to the data set
MODELS	$\vec{M} = \{M_d^i\}$	Set of all possible models computed with shape (N_D, N_i).
	$\vec{\mu} = \{\mu_d^i\}$	Expectation value of each element of the models
	$\vec{V} = \{V_d^i\}$	Associate variance of the expectation value
	$\mathcal{L}(\vec{D}\|\vec{M}, \vec{E})$	Likelihood of the model
	$\mathcal{E}(\vec{M})$	Evidence of the model
PRIORS	$p(\vec{M})$	Prior probability distribution of all possible values of the model
	$\mathcal{MM}$	Central moving mean
POSTERIORS	$\mathcal{P}(\vec{M}\|\vec{D}, \vec{E})$	Posterior probability distribution of all possible values of the model
	$\vec{k} = \{k_d^i\}$	Scale factor arised from the product of two Gaussians
OUTPUT	$\vec{S} = \{S_d\}$	Smoothed data set

TABLE I: List of all symbols used to describe the $FABADA$ algorithm in section II-A.

Theorem

$$\mathcal{P}_i(M_d^i|D_d, E_d) = \frac{1}{\mathcal{E}_d^i} \cdot \frac{e^{-\frac{(\mathcal{MM}(\mu_d^{i-1}) - M_d^i)^2}{2 \cdot V_d^{i-1}}}}{\sqrt{2\pi \cdot V_d^{i-1}}} \cdot \frac{e^{-\frac{(D_d - M_d^i)^2}{2 \cdot E_d^2}}}{\sqrt{2\pi} \cdot E_d}$$
$$= \frac{1}{\mathcal{E}_d^i} \cdot k_d^i \cdot \frac{e^{-\frac{(\mu_d^i - M_d^i)^2}{2V_d^i}}}{\sqrt{2\pi V_d^i}} \tag{8}$$

leading to a product of two Gaussian (our prior and the plausibility of the data), which remains in another Gaussian (the later $\mathcal{P}$) centered on

$$\mu_d^i = V_d^i \cdot \left[\frac{\mathcal{MM}(\mu_d^{i-1})}{V_d^{i-1}} + \frac{D_d}{E_d^2} \right] \tag{9}$$

with variance

$$\frac{1}{V_d^i} = \frac{1}{E_d^2} + \frac{1}{V_d^{i-1}} \quad . \tag{10}$$

The constant

$$k_d^i = \frac{e^{-\frac{[\mathcal{MM}(\mu_d^{i-1}) - D_d]^2}{2\left(E_d^2 + V_d^{i-1}\right)}}}{\sqrt{2\pi \cdot \left(E_d^2 + V_d^{i-1}\right)}} \tag{11}$$

associated with the product of two Gaussian (the mathematical development can be found, for example, in Bromiley (2003)) corresponds to the evidence of the i model for the d data

$$\mathcal{E}_d^i = \int_{-\infty}^{\infty} p_i(M_d^i) \cdot \mathcal{L}_i \; dM_d^i =$$
$$= \int_{-\infty}^{\infty} k_d^i \cdot \frac{e^{-\frac{(\mu_d^i - M_d^i)^2}{2V_d^i}}}{\sqrt{2\pi V_d^i}} \; dM_d^i = k_d^i \tag{12}$$

so that the posterior probability distribution is correctly normalized. In the figure 1 we have a random example to represent the mathematical process we have just explained. The probability distributions are shown on one side (right) next to the expected value of one of these distributions (left). It can be seen that the product of two Gaussians is still a Gaussian as in (8), and that the expected value of the subsequent distribution for the new smoothed model (μ_d^i) is more conservative (closer to μ_d^{i-1}) than the expected value obtained previously by means of $\mathcal{MM}(\mu_d^{i-1})$. As the number of iterations increases, the model becomes gradually smoother and the variance V_d^i decreases monotonously, as can be deduced from the equation (10).

As long as the model remains relatively close to the data (within an environment of the order of $\sqrt{E_d^2 + V_d^i}$), the evidence in its favour will be high, but if it departs significantly, the exponential term that appears in the constant k_d^i will indicate that we have reached the maximum smoothing that is statistically compatible with the data. If we look at figure 1, we can see that the Gaussian in the posterior is much lower than the two previous ones, because they are both relatively separate. From the physical point of view, this means that our prior idea of

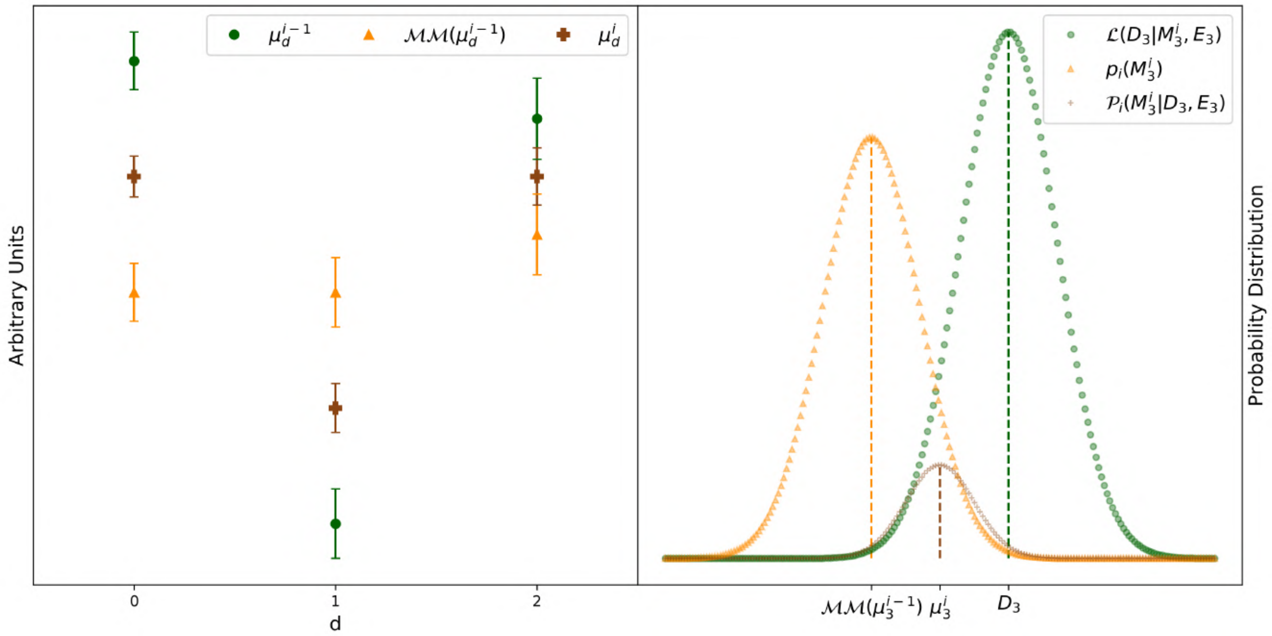


Fig. 1: Graphical representation of an iteration of the algorithm for a random data set. On the left, the data set to be smoothed is shown μ_d^{i-1} (with $d = 1, 2, 3$) together with the smoothed values $\mathcal{MM}(\mu_d^{i-1})$, and the expected value of the probability distribution after applying Bayes' Theorem μ_d^i. On the right, the prior and posterior distributions set to the model's Likelihood.

the value that M_d should adopt is relatively far (several sigma) from the measured data D_d. This disagreement is mathematically expressed by the scale factor k_d^i, which plays no role in calculating the expected values $\vec{\mu}$ and their variances $\vec{V}$, but is critical in calculating the evidence of each model.

B. *Model selection*

The last step of FABADA will be to automatically decide a stopping condition (i.e, the total number of iterations N_i) and then combine the different models produced at each iteration in order to produce the final result to be returned on output. We have considered two different approaches: selecting the model with maximal evidence and the Bayesian combination of all the models up to N_i. Although we evaluate individual evidences for every data point, both criteria are based on the mean value

$$\langle\mathcal{E}^i\rangle = \frac{1}{N_D}\cdot\sum_{d=0}^{N_D}\frac{e^{-\frac{[\mathcal{MM}(\mu_d^{i-1})-D_d]^2}{2\left(E_d^2+V_d^{i-1}\right)}}}{\sqrt{2\pi\cdot\left(E_d^2+V_d^{i-1}\right)}} \tag{13}$$

over the model computed at iteration i.

1) Single Model (SM): In this approach, the number of iterations N_i is set to the maximum average evidence, and the algorithm will stop as soon as $\frac{d\langle\mathcal{E}^i\rangle}{di} = 0$. Since the evidence provides a measure of the probability of obtaining the measurements D_d, the model $M_d^{N_i}$ with the maximum evidence is the most likely to represent the smooth distribution $\vec{S} = \{S_d\}$ from which the noisy data D_d were measured. Therefore, one may select its posterior probability distribution

$$\mathcal{P}(S_d) = \mathcal{P}(M_d^{N_i}|D_d, E_d) \tag{14}$$

with expected value

$$\mu_d^{\vec{S}} = \mu_d^{N_i} \tag{15}$$

as the most likely description of the input data set.

2) Bayesian Model (BM): In this case, the results of the different iterations will be weighted according to Bayes' theorem. To do this we must choose a prior probability for each of the models M_d^i. Using the same criterion as in equation 2, we say that $p_i = \frac{1}{N_i+1}$, so that all of them are equiprobable, the probability distribution is proper (sum normalized to unity), and one can apply the same reasoning as in equation 4, this time for a discrete distribution, where the mean evidence $\langle\mathcal{E}^i\rangle$ represents the likelihood of each model. Thus, the posterior probability of the i-th model would be given by the expression

$$\mathcal{P}_d^i = \frac{\mathcal{E}_d^i}{\sum_{i=0}^{N_i}\mathcal{E}_d^i} \tag{16}$$

while, for the value of the final smooth model S_d at point d, each and every model would have to be combined, using this probability as a weight:

$$\mathcal{P}(S_d) = \sum_{i=0}^{N_i}\mathcal{P}_d^i\cdot\mathcal{P}(M_d^i) \tag{17}$$

whose expected value is none other than

$$\mu_d^{\vec{S}} = \sum_{i=0}^{N_i}\mathcal{P}_d^i\cdot\mu_d^i. \tag{18}$$

In other words, our final model $\vec{S}$ would be a weighted sum of all the previous models, weighed by their respective evidence.

It is important to note that these values are calculated for each data element d separately, so not all models have the same weight at all points. This allows for adaptation to the structure of the data, as some regions may admit a greater degree of smoothing than others. If any region has been smoothed more or less, the evidence will give more weight to the models that better describe it, achieving the objective of the algorithm to develop a smooth model, compatible with the error bars, that adapts to the structure of the data.

In principle, one should take into account an arbitrarily large number of iterations N_i, but this is of course not feasible in practice. Instead, we will try to sum over a representative range of models, within the vicinity of the most probable one, assuming that they will dominate the total result. The iteration process won't stop at the maximum of the average evidence (equation 13), but will continue until it reaches an inflection point, $\frac{d^2\langle\mathcal{E}^i\rangle}{di^2} = 0$, which acts as a proxy for the end of the evidence peak and the transition towards an asymptotic regime where the evidence will slowly decline. After this point, the models change very little, and they produce similar probabilities, slightly smaller than the previous ones, and they add little useful information. Therefore, the algorithm will now stop at the iteration N_i in which the inflection point has been reached and combine the $N_D \times (N_i + 1)$ expected values μ_d^i according to equation 18.

III. Synthetic tests

In order to assess the performance of the algorithm, we develop a battery of synthetic tests based on real astronomical data. More precisely, we will apply FABADA to a set of astronomical spectra and images, where different levels of Gaussian random noise have been added, and compare the quality of the reconstructed signal (in terms of the mean square error) as well as the execution time with other methods available in the literature.

A. *Other Algorithms*

Over the last decades, lots of effort have been placed into the development of different applications to help in the analysis of digital images in different fields. Noise reduction is one of the basic problems in this context, and we have attempted to provide a fair comparison of our algorithm with other methods that are representative of the current state of the art. In this section we briefly describe the main principles and free parameters, if any, of all the techniques that we have considered. A brief summary is provided in Table II.

1) Median filter: One of the classical non-linear digital filtering techniques, it is still often used to remove noise from an image or signal. The main idea of the median filter is to run through the data, replacing each point with the median of neighboring entries. The number of neighbors used in the median is called the ”window”, which slides, entry by entry, over the entire signal. So for our data set D_d the median filter is computed as

$$S_d^{Mdn} = Median(D_d, w) \qquad (19)$$

Method	**Parameters**	**1D**	**2D**
Median	Window size (w)	X	X
SGF	Window (w) and order (o)	X	X
LOWESS	Fraction window	X	-
LPFF	Radius (R_0)	X	X
Photoshop©	Intensity (i) and detail (det)	-	X
BM3D	—	-	X
FABADA	—	X	X

TABLE II: List of all noise reduction methods used to compared with FABADA. It is shown the initials that we will used to reference the different methods explained. *SGF*, *LOWESS*, *LPFF*, *BM3D* stands for Savitzky-Golay filter, locally weighted scatterplot smoothing, low-pass frequency filter, and the block-matching and 3D filtering method respectively. Along the initials we show their parameters and their space implementations (one or two dimensions).

where S_d is the smoothed result, and $Median$ represents the median of the data set inside the window of size w. For each data point D_d, the region used to computed the median contains, for one dimensional data, $(w-1)/2$ neighbors on each side, whereas for two dimensions it corresponds to a square of size w centered in D_d. This implies that this method has one parameter, which is the size of the window w.

2) Savitzki-Golay filter: As first noted by Savitzky and Golay (1964), a smoothed version of the data may be obtained by fitting successive sub-sets of adjacent points with a low-degree polynomial using the method of least squares. When the data are equally spaced, the solution of the least squares (i.e., the coefficients of the polynomials) is analytical and independent of the data to be smoothed. Thus,

$$S_d^{SG} = SGF(D_d, w, o) = \sum_{i=\frac{1-w}{2}}^{\frac{w-1}{2}} C_i^o(w, o) \cdot D_{d+i} \qquad (20)$$

for $\frac{w-1}{2} \leq d \leq N_D - \frac{w-1}{2}$, where the two free parameters of the method are the window length w of the data region, i.e., the number of data points to be fitted, and the order o of the polynomial. C_i^o are the $w \geq o$ Savitzky-Golay coefficients, and S_d^{SG} is the smoothed result of the filter at position d.

3) LOWESS: A popular variant of the Savitzky and Golay method is the locally weighted scatterplot smoothing (LOWESS), where the regions to be fitted are not evenly spaced and the least squares procedure takes into account weighted values of the data, according to their distance from the point to be evaluated. This scheme involves computing the coefficients of the fitted polynomial each time, producing a less efficient algorithm. For the purpose of comparing with FABADA, we have used the implementation explained in Cleveland (1979), which uses a linear fit and can only be used for one dimensional data. This implementation has only one parameter, which is the fraction of data points used to

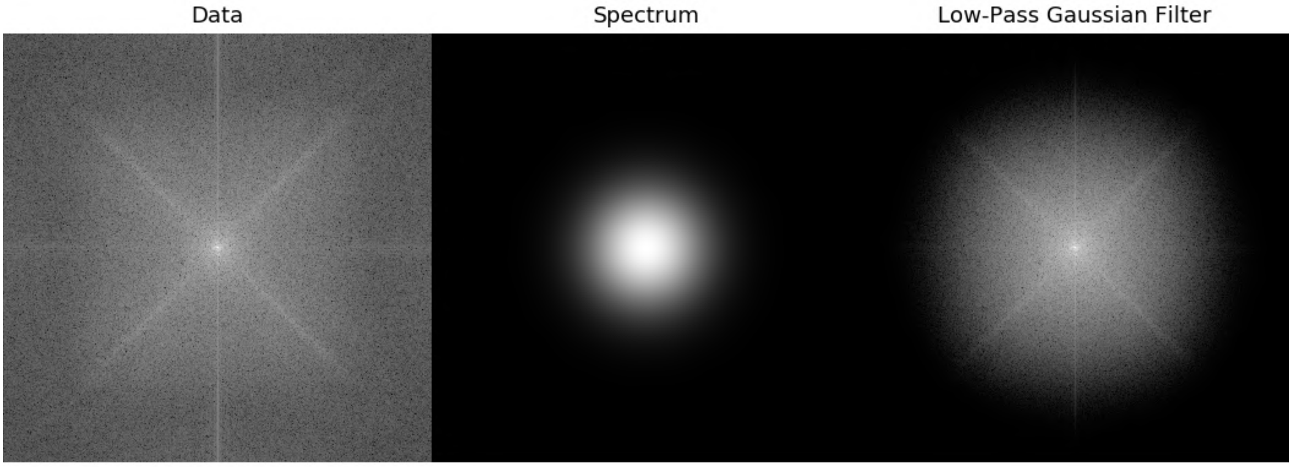


Fig. 2: On the left is shown the spectra of one of the image sample used in this work. This spectra is computed as the module of the fast Fourier Transform of the image. In the middle, one example of the shape of the Gaussian Low-Pass filter with radius 30 pixels used in the Low-Pass Frequency Filter. On the right, the filtered result of the spectra.

accomplish the linear regression at each point.

4) Low-Pass Frequency Filter (LPFF): Another classical technique of noise reduction consists in filtering the high frequency components of the data. The fast Fourier Transform (FFT) is the most computational efficient way to convert the data D_d to the frequency domain $D(w)_d$. Once we have the spectrum of the image, defined as the amplitude of the FFT of the data (see figure 2), we can apply a low pass Gaussian filter to discard the highest modes:

$$LP(D(w)_d, R_0) = e^{-r(D(w)_d)^2/2R_0^2} \tag{21}$$

where $r(D(w)_d)$ is the distance from the center to the data $D(w)_d$ in the spectrum image, and R_0 is the radius of the filter. We can construct the smoothed data S_d^{LPFF} by computing the inverse fast Fourier Transform (iFFT) of the filtered spectrum, which yields

$$\begin{aligned} S_d^{LPFF} &= LPFF(D_d, R_0) = \\ &= iFFT(LP(D(w)_d, R_0) * FFT(D_d)) \end{aligned} \tag{22}$$

where $LPFF$ stands for Low-Pass frequency filter. In figure 2 there is an example of how this process in done, showing the spectra of one image (left), the low-pass filter (middle), and the filtered spectrum (right). This is again a one parametric method in which only we have to select the radius of the low pass filter.

5) Photoshop©: In addition to classical methods, we have also considered the noise reduction filter implemented in the Adobe Photoshop© raster graphics editor software (Knoll and Knoll, 1990-2017). Although neither the mathematical foundations nor the source code of this method are publicly available, we consider it representative of current industrial standards, and it has been used to gauge the commercial potential of our algorithm. The intensity i of the noise reduction applied (an integer number that controls the aggressiveness of the smoothing) is one of the two parameters of this filter. Furthermore, one can choose the number of features preserved in the process by changing the detail conservation parameter det. Another feature of this filter would be the capacity to focus the details of the smoothed image, but it has not been activated because FABADA does not include any such procedure yet. As this is an image editor, we have only been able to use it in two dimensions.

6) BM3D: Finally, we also included the Block-Matching and 3D filtering (Dabov et al., 2007) algorithm, which arguably represents the state of the art in the research field of image analysis (Zhang et al., 2017). A detailed account of this method, where image denoising is implemented as two-step process, can be found in Dabov et al. (2007).

For the first step, the noisy image is divided into equal-size square blocks. For each of these blocks, a three-dimensional group is formed with blocks of other regions that are consistent with being similar to the reference one (see figure 3), and the process is then repeated for each block in the image. When these 3D groups are formed via block-matching, they attenuate the noise by hard-thresholding of the coefficients of a 3D transform, which they called *"Collaborative hard-thresholding"*. By preforming the inverse 3D transform, BM3D produces an estimation of the denoised image for each block of the inverted group. Thanks to this group filtering via the hard-threshold, they are able to form more than one estimator for the different blocks of the image, and these overlapping estimations are combined in a weighted average for each block.

With the basic estimate formed in step one, they are able to produce more accurate groups of similar blocks and then apply a 3D transform the groups formed in both steps. With the 3D transform of both groups, they are able to perform Wiener filtering on the noisy one using the energy spectrum of the basic estimate as to the true (pilot) energy spectrum. The inverse transformation produces again the estimates of the blocks to their original positions. By weighted average, they generate the final smoothed result

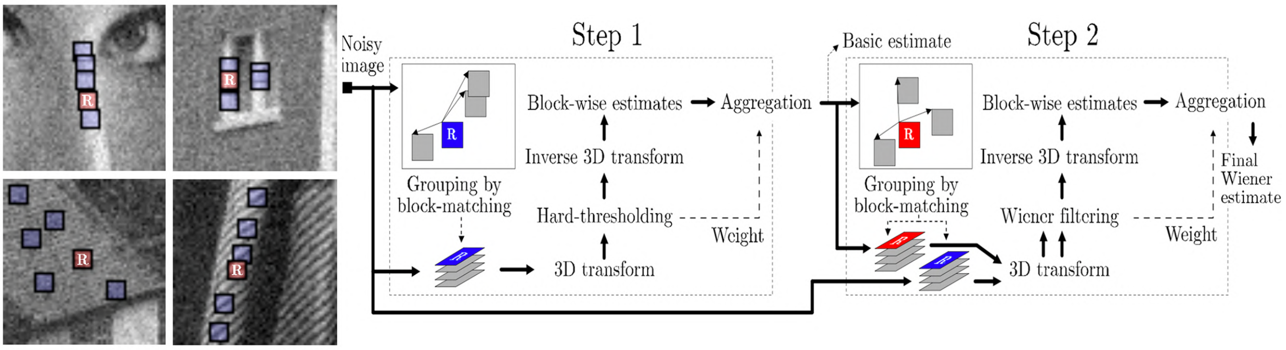


Fig. 3: On the left, there is an illustration of grouping blocks from the BM3D method explained from noisy natural images corrupted by white Gaussian noise with a standard deviation of 15 counts and zero mean. Each fragment shows a reference block marked with "R" and a few of the blocks matched to it. On the right there is the flowchart of the BM3D image denoising algorithm. The operations surrounded by dashed lines are repeated for each processed block (marked with "R"). Both of the figures are taken from Dabov et al. (2007).

of the image. This second process is also shown in the flowchart of the figure 3.

B. Data Sample

All the methods explained in the previous section are applied to a set of test data in one and two dimensions (astrophysical spectra and monochromatic images, respectively) with different levels of Gaussian random noise.

An important aspect in the recovery of spectra is the conservation of their features, such as the Balmer break or emission and absorption lines, after noise reduction. For this purpose, we have considered three different spectra (represented in Figure 4) that show these characteristics in different degrees. The first spectrum (left) is a Kurucz stellar atmosphere model with an effective temperature $T_{eff} = 11500K$, metallicity $Z = 0.1$ and surface gravity $\log g = 5.0$, typical of an O/B type star, with a prominent Balmer break and some strong absorption lines. The spectrum of a supernova remnant, plotted on the middle panel, is a composite of 5 different observations from the Faint Object Spectrograph (FOS) instrument of the Hubble Space Telescope (HST). This high-resolution spectrum (0.9 Å/pixel) is characterized by very prominent emission lines, useful for inferring different physical properties of these objects. The last spectrum (right) is taken from observations of the interacting galaxy pair Arp 256, and it contains a combination of emission and absorption lines with a stellar continuum. The Kurucz model and the Arp 256 spectra have been extracted from the ASTROLIB PYSYNPHOT (Lim, 2015) Python package that simulates photometric data and spectra as they are observed with the Hubble Space Telescope (HST). The aim of using these different spectra is to obtain a good representation of the possible features that can appear in one-dimensional astrophysical data and see how the different algorithms perform in digging up spectral features out of a noisy signal.

On the other hand, for astronomical images, we would also like to cover all the possible ranges in this field, including planets, stars, diffuse nebulae, and galaxies, either alone or in potentially blended groups. We have thus considered a variety of images, formed from the eight different targets displayed in Figure 5, intended to sample almost all possible combinations of these objects. Saturn is arguably the target whose features (e.g. sharp edges) are more similar to the ordinary test images (e.g natural landscapes, human subjects) that are often used in the context of digital image processing. In addition, our sample includes two examples of nebulae (Crab and Bubble) dominated by the gaseous component, two with a more significant contribution of the stellar population (Eagle and Ghost nebulae), and a globular cluster full of stars with different brightness. There is also an image with a galaxy pair (NGC 4302 & 4298) in which we can see two different orientations of the galaxies, as well as a galaxy cluster with a wide variety of morphologies and apparent sizes. All of these images have been taken from the Hubble Space Telescope gallery and have been compressed to 8-bit images, with a maximal dynamical range of $0 - 255$ counts and 512×512 pixels size to lighten up the computational load. For simplicity, we have also normalized the astronomical spectra to 255 in order to have the same dynamical range and represent the noise in terms of this value for both dimensions.

C. Test Statistics

We applied different levels of Gaussian random noise η with constant variance σ^2 to the real data R_d:

$$D_d = R_d + \eta_d \quad ; \quad E_d = \sigma \tag{23}$$

where $\eta_d = \mathcal{N}(0, \sigma)$, the subscript $1 \leq d \leq N_D$ denotes independent measurements (spectrum wavelengths or image pixels), and we assume that statistical errors are correctly characterized in the input data. Once D_d is computed, an softened estimation S_d of the real data R_d is carried out using the different algorithms explained above and different noise levels, from $\sigma = 5$ counts to $\sigma = 95$ counts out of the 255 maximal value that sets the dynamical range of our data (i.e. of the order of $\approx 2 - 40\%$ relative errors).

Afterwards, we evaluate the quality of the reconstruction in terms of the Peak Signal to Noise Ratio (PSNR) of the estimators S_d, following common practice in the signal processing literature. By definition, the PSNR (usually

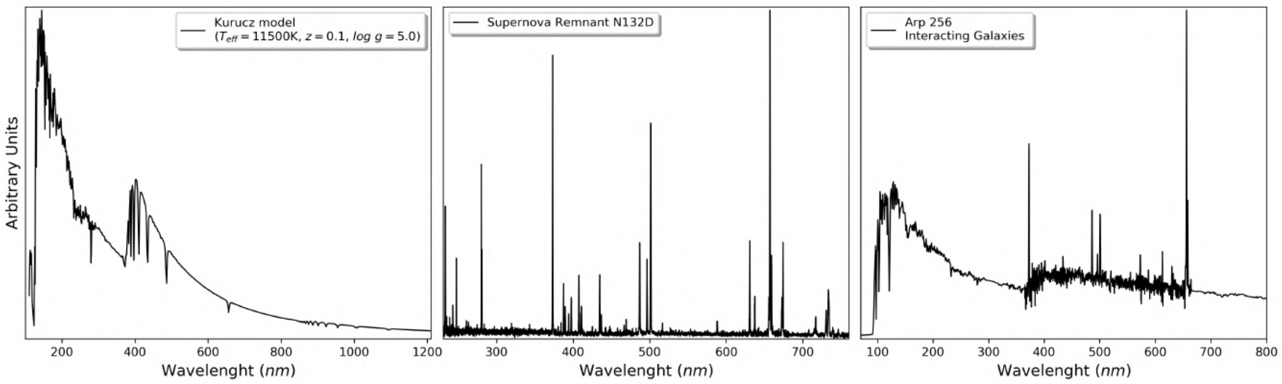


Fig. 4: All the spectra sample used to compare the performance of the algorithms explain along the one proposed in this work, FABADA. The left spectrum is a Kurucz model with an effective temperature of $T_{eff} = 11500K$, metallicity $z = 0.1$ and a surface gravity of $log\ g = 5.0$. The middle one is from the supernova remnant N132D in the Large Magellanic Cloud. An the left one is from a pair of interacting galaxies (Arp 256) in the constellation of Cetus. The first and last spectra were generated with the ASTROLIB PYSYNPHOT (Lim, 2015) Python package and the middle one is a composition of five observations of the Faint Object Spectrograph (FOS) instrument of the Hubble Space Telescope (HST).

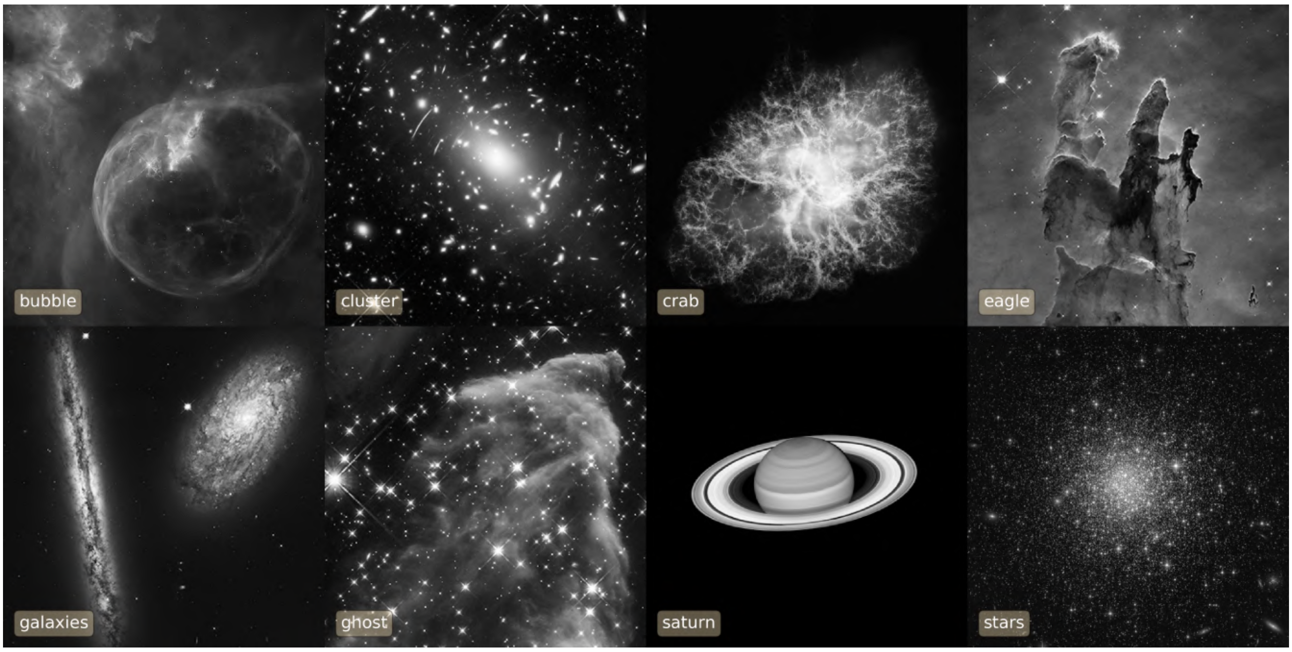


Fig. 5: Battery of images used for the comparison procedure. From left to right, up to bottom the objects shown in this figure are the Bubble nebula (NGC 7635), a galaxy cluster (Abell S1063), the Crab nebula (M1), the Eagle nebula (M16), a spiral galaxy pair (NGC 4302 & 4298), the Ghost nebulae (IC 63), Saturn and a globular star cluster (NGC 1466). All these images are taken from the Hubble Space Telescope gallery and have been pre-processed for the purpose of time efficiency. In the left bottom corner is shown the name used to reference the image in this work.

expressed in decibels, dB) is related to the Mean Square Error (MSE)

$$\mathrm{MSE}(\vec{S}) = \frac{1}{N_D}\sum_{d=1}^{N_D}(S_d - R_d)^2 \tag{24}$$

as

$$\mathrm{PSNR}(\vec{S}) = 10 \cdot \log_{10}\left(\frac{255^2}{\mathrm{MSE}}\right), \tag{25}$$

where 255 is the maximal possible value in our data. In principle, a more faithful recovery of the real data should yield smaller values of the MSE and higher values of the PSNR. It is very important to note here that, as can be readily seen in table II, only BM3D and FABADA are completely parameter-free. In all other cases, the parameters of each algorithm have been optimized to to minimize the MSE (maximize the PSNR) of each particular realisation of the input data. This implies that their results are the best possible and should be regarded as an upper limit to the performance of these algorithms, since this kind of optimisation is only possible when the correct solution is known.

Another metric that we have considered is the CPU time used to generate the estimation of the real data. This time corresponds to the final execution time for the given noise level in the Python implementation of the algorithms, once the optimal parameters have been found, but it does *not* include the time invested in the exploration of the relevant parameter space, wich is considerably larger. All runs have been carried out on a single processor with a 2.40 GHz Intel i9-9980HK CPU along with 16Gb DDR4 2400 MHz RAM memory.

IV. Results

In this section, we present and discuss the experimental results obtained by our algorithm. First of all, we investigate the merits and drawbacks of the two model selection prescriptions described in section II-B: the Single Model $SM(\mathcal{E}_{MAX})$ that maximizes the average evidence, and the Bayesian Model $BM(\mathcal{E}_{IP})$ that combines all iterative models, weighted by their average evidence, until this quantity reaches an inflection point. Then we evaluate the quality of signal recovery and compare it with the results of other algorithms in terms of the Mean Square Error (MSE) / Peak Signal-to-Noise Ratio (PSNR). As we optimize the free parameters of other methods, we also consider the number of iterations that minimizes the MSE in each case, $SM(MSE_{min})$ and $BM(MSE_{min})$. Finally, we also assess the performance of the different algorithms in terms of CPU time efficiency.

A. Model Selection

Figure 6 shows a comparison between the average evidence of our models for each iteration of the recovery process of the *bubble* image with the MSE obtained with the two variants of the algorithm, $SM(\mathcal{E}_{MAX})$ and $BM(\mathcal{E}_{IP})$. Different panels illustrate four different levels of noise ($\sigma = 5, 25, 50, 95$) with the MSE normalized by the noise level. Vertical lines indicate the iterations where the maximum and the inflection point of the average evidence are achieved, along with the minimum normalized MSE for both model selection variants.

This figure provides a visual example of the differences between the two automatic stopping criteria implemented in FABADA, based on the average evidence $\langle\mathcal{E}^i\rangle$, and the optimal solutions in terms of MSE. The interpretation of the overall shape of these curves is fairly straightforward. As the number of iterations increases, the quality of the reconstruction initially improves (higher evidence and lower MSE), but at some point smoothing starts to blur the structures present in the image an the quality of the solution drops.

The ideal correlation of these two parameters would be that they reached the maximum and the minimum at the same time, but as expected, life is more complicated. For low noise levels, the SM might even overestimate the optimal number of iterations, whereas the BM will stop relatively close to its optimum. The difference in MSE is not very high in either case. However, as the level of noise increases, both prescriptions tend to severely underestimate the optimal degree of smoothing, yielding a much more significant difference in terms of the MSE of the reconstructed signal. We also see that the $SM(\mathcal{E}_{MAX})$ model always achieves its optimal solution slightly faster than $BM(\mathcal{E}_{IP})$ for any noise level, with comparable or better MSE except at the lowest noise level $\sigma = 5$ counts.

These trends are verified by all our test cases, as summarized in figure 7, where the results in one dimension (spectra) are plotted on the top panels, whereas the analyses of astronomical images are displayed in the bottom row. The left column compares the iteration number where our models $SM(\mathcal{E}_{MAX})$ and $BM(\mathcal{E}_{IP})$ have stopped with the iteration where the optimal MSE is achieved (a one-to-one relation is illustrated by a dotted line), whereas relative differences in iteration number and model MSE are shown in the middle column. In these panels, symbol colors indicate the noise value (σ) of the input data. The right column plots the normalized MSE achieved by each model selection prescription as a function of the noise value.

As one may expect, fewer iterations are needed to find the optimal solution for smaller values of the noise. While our two stopping criteria correctly capture this trend, they only coincide with the optimum MSE when it is reached near ~ 10 iterations. Below that number (i.e. high signal-to-noise ratio), the data are smoothed too much, whereas the opposite happens as the noise increases. Even if the selected models diverge from the optimal solution, there seems to be a correlation, somewhat stronger in two-dimensional images than in one-dimensional spectra, that can be roughly approximated by a pure power law whose slope would be steeper for the single model selection $SM(\mathcal{E}_{MAX})$ than the Bayesian model $BM(\mathcal{E}_{IP})$. Although one could try to fit this correlation and find an empirical recipe to estimate the optimal number of iterations, we are not sure that minimizing the MSE is the best strategy to keep all the information in the data (see the discussion below).

In any case, it is seen both for the spectra data and for the images that the Bayesian model requires more iterations to achieve convergence, under any selection criterion. This will always be true, since the Bayesian prescription keeps memory of all previous estimates, and therefore it has a larger 'inertia' with respect to the original input data. While this is clearly a disadvantage regarding the total execution time of the algorithm, it helps it to adapt better to the structure of the data, providing more smoothing where appropriate while keeping relevant features, if present.

In terms of MSE (middle column), an excessive smoothing (typical at high SNR) is roughly as bad (up to the order of a few) as an underestimation of the optimal stopping condition, which can be as large as an order of magnitude. In practice, though, the problem is encountered more often at high noise levels, as evidenced by both the central and rightmost panels. It is only for a few instances that the $SM(\mathcal{E}_{MAX})$ severely oversmooths the data. In general, $BM(\mathcal{E}_{IP})$ tends to perform better than $SM(\mathcal{E}_{max})$ when the signal-to-noise ratio is high, achieving slightly lower values of the MSE. The situation reverses at high levels of random noise, where the single models performs a more aggressive smoothing. In any case, both prescriptions could achieve significantly lower values of the MSE by carrying out a much larger number of iterations, at the expense of potentially missing informative features in the

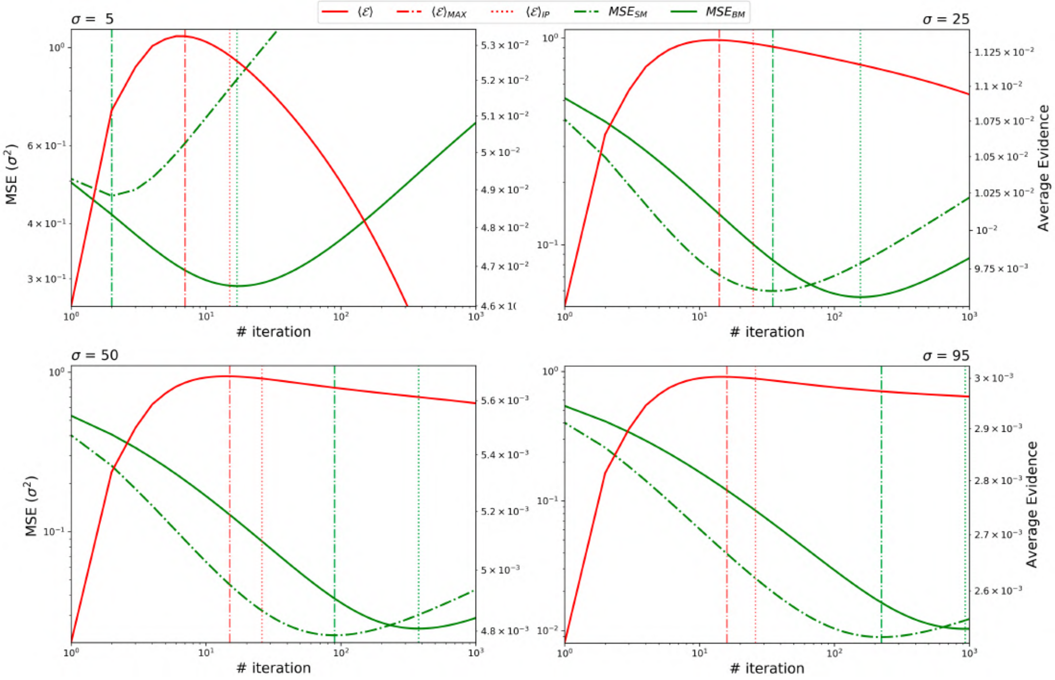


Fig. 6: Representation of the average evidence (13) for the models computed in the recovery process of the *bubble* image as a function of the the number of iterations done by the implementation of FABADA for four different noise levels. From left to right, top to bottom the noise levels are $\sigma = 5, 25, 50, 95$ counts respectively. The iteration where the maximum of the average evidence is reached is shown with a red dashed-dotted line while the inflection point is pointed out with a red dotted line. The normalized, by the level of noise, Mean Square Error MSE (24) achieved by the two different criteria, the single selection model $SM(\mathcal{E}_{MAX})$ (green dashed-dotted line) and the Bayesian combination model $BM(\mathcal{E}_{IP})$ (green solid line) is also represented. The minimum of MSE for these two models are shown by the dash-dotted vertical line and the dotted line, respectively. Both of the axis are in logarithmic scale.

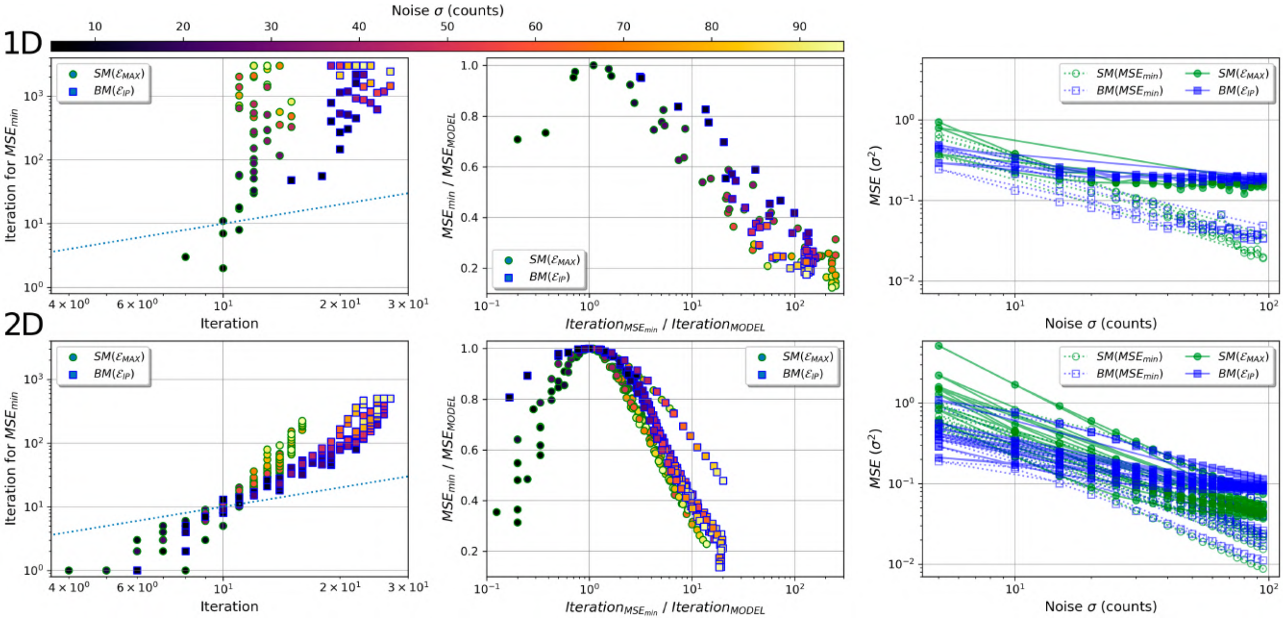


Fig. 7: Normalized Mean Square Error (MSE) and number of iteration produce by both of the models explained, the single model $SM(\mathcal{E}_{MAX})$ (green circles) and the Bayesian model $BM(\mathcal{E}_{IP})$ (blue squares) for the recover of all the data sample, 1D astronomical spectra (top) and 2D astronomical images (bottom). In the left figure is shown the iteration at which the models have stopped against the iteration where the minimum MSE is achieved. On the middle is represented the difference in normalized MSE between the models and the optimal solution with the difference in iteration. The color code for these figure shows the noise level of the input of the image in counts. In the right, is shown the normalize MSE for the models (filled symbols) and the optimal solutions (unfilled symbols) for different noise levels. All of the axis are shown in logarithmic scale.

data.

Table III lists the MSE values achieved by each criterion for every test problem and noise level. The $BM(\mathcal{E}_{IP})$ only supersedes $SM(\mathcal{E}_{MAX})$ in 46 out of the 209 examples (i.e. 22%). However, when the number of iterations is set to optimize the MSE (not shown), this fraction increases to

σ / PSNR	bubble	crab	eagle	galaxies	ghost	cluster	saturn	stars	Kurucz Model	Arp256	SN132D
5 / 34,15	36,19 / 38,60	32,02 / 35,24	32,02 / 35,78	32,41 / 36,73	29,77 / 34,88	31,01 / 36,12	31,75 / 37,04	27,02 / 30,42	34,41 / 37,26	35,12 / 37,66	38,42 / 39,48
10 / 28,13	34,61 / 35,39	30,62 / 31,95	30,63 / 32,96	31,54 / 33,48	28,79 / 32,21	29,81 / 33,02	30,94 / 32,93	25,84 / 28,27	32,33 / 32,99	32,69 / 33,71	**34,61 / 34,36**
15 / 24,61	**33,40 / 33,35**	29,29 / 30,04	29,63 / 31,03	30,64 / 31,46	27,93 / 30,35	28,81 / 30,98	29,63 / 30,51	24,98 / 26,73	**30,59 / 30,47**	31,03 / 31,12	**31,96 / 31,47**
20 / 22,12	**32,32 / 31,63**	28,46 / 28,66	28,94 / 29,57	29,87 / 29,92	26,99 / 28,91	27,93 / 29,24	28,55 / 28,80	24,32 / 25,59	**28,77 / 28,45**	**28,81 / 28,51**	**29,85 / 29,23**
25 / 20,17	**31,32 / 30,25**	**27,75 / 27,57**	28,29 / 28,39	**29,12 / 28,81**	26,53 / 27,72	27,22 / 27,96	**27,58 / 27,54**	23,79 / 24,66	**27,86 / 27,11**	**27,91 / 27,33**	**27,99 / 27,28**
30 / 18,59	**30,35 / 29,06**	**27,14 / 26,72**	**27,69 / 27,36**	**28,45 / 27,75**	25,98 / 26,68	26,54 / 26,92	**26,89 / 26,55**	23,35 / 23,89	**26,43 / 25,71**	**25,97 / 25,38**	**26,41 / 25,68**
35 / 17,25	**29,58 / 27,84**	**26,52 / 25,90**	**27,17 / 26,43**	**27,76 / 26,86**	25,46 / 25,80	**26,04 / 25,91**	**26,26 / 25,71**	22,97 / 23,30	**24,78 / 24,12**	**25,02 / 24,39**	**25,17 / 24,41**
40 / 16,08	**28,78 / 26,97**	**26,02 / 25,17**	**26,53 / 25,65**	**27,23 / 25,94**	**25,04 / 25,01**	**25,49 / 25,15**	**25,61 / 24,91**	22,63 / 22,77	**23,76 / 23,08**	**24,18 / 23,38**	**24,07 / 23,30**
45 / 15,06	**28,15 / 26,05**	**25,57 / 24,41**	**26,09 / 25,01**	**26,64 / 25,26**	**24,60 / 24,26**	**25,01 / 24,39**	**25,06 / 24,20**	**22,35 / 22,23**	**23,65 / 22,87**	**22,95 / 22,06**	**23,05 / 22,22**
50 / 14,16	**27,40 / 25,24**	**25,04 / 23,91**	**25,55 / 24,29**	**26,16 / 24,64**	**24,20 / 23,56**	**24,49 / 23,75**	**24,61 / 23,51**	**22,05 / 21,77**	**22,05 / 21,30**	**21,46 / 20,78**	**22,33 / 21,54**
55 / 13,32	**26,78 / 24,58**	**24,66 / 23,26**	**25,18 / 23,66**	**25,56 / 23,96**	**23,76 / 22,86**	**24,11 / 23,06**	**24,14 / 22,94**	**21,74 / 21,33**	**21,77 / 20,97**	**21,06 / 20,23**	**21,33 / 20,53**
60 / 12,57	**26,00 / 23,91**	**24,24 / 22,76**	**24,67 / 23,03**	**25,02 / 23,24**	**23,33 / 22,41**	**23,65 / 22,56**	**23,68 / 22,42**	**21,53 / 20,94**	**20,02 / 19,43**	**20,28 / 19,55**	**20,55 / 19,62**
65 / 11,87	**25,42 / 23,18**	**23,72 / 22,18**	**24,18 / 22,42**	**24,76 / 22,77**	**23,06 / 21,93**	**23,19 / 21,98**	**23,26 / 21,84**	**21,27 / 20,54**	**20,01 / 19,24**	**19,92 / 19,01**	**19,93 / 19,16**
70 / 11,22	**24,83 / 22,58**	**23,48 / 21,63**	**23,71 / 21,96**	**24,21 / 22,00**	**22,66 / 21,45**	**22,92 / 21,44**	**22,93 / 21,46**	**21,05 / 20,20**	**19,27 / 18,49**	**19,92 / 18,90**	**19,14 / 18,32**
75 / 10,62	**24,45 / 22,16**	**23,05 / 21,29**	**23,30 / 21,31**	**23,81 / 21,62**	**22,29 / 20,85**	**22,49 / 21,03**	**22,59 / 20,92**	**20,79 / 19,73**	**18,42 / 17,66**	**18,38 / 17,55**	**18,57 / 17,80**
80 / 10,06	**24,04 / 21,25**	**22,62 / 20,77**	**22,84 / 21,05**	**23,27 / 21,10**	**21,89 / 20,40**	**22,10 / 20,53**	**22,16 / 20,45**	**20,49 / 19,43**	**19,22 / 18,18**	**18,48 / 17,51**	**18,17 / 17,39**
85 / 09,54	**23,59 / 20,93**	**22,45 / 20,28**	**22,46 / 20,44**	**22,93 / 20,61**	**21,67 / 20,08**	**21,82 / 20,07**	**21,76 / 20,05**	**20,25 / 19,06**	**17,34 / 16,46**	**17,86 / 16,97**	**17,87 / 17,05**
90 / 09,07	**22,86 / 20,50**	**22,06 / 19,97**	**22,26 / 20,03**	**22,38 / 20,11**	**21,40 / 19,70**	**21,54 / 19,56**	**21,43 / 19,62**	**19,99 / 18,83**	**17,16 / 16,34**	**17,32 / 16,61**	**16,83 / 16,07**
95 / 08,57	**22,69 / 20,14**	**21,68 / 19,52**	**21,86 / 19,59**	**22,05 / 19,87**	**21,03 / 19,21**	**21,19 / 19,31**	**21,04 / 19,20**	**19,88 / 18,43**	**16,22 / 15,57**	**16,87 / 16,02**	**16,64 / 15,76**

TABLE III: Peak signal to noise ratio values for the result of $SM(\mathcal{E}_{MAX})$ (left value) and $BM(\mathcal{E}_{IP})$ (right value) for all the data sample and the noise range. The first column indicates the value of the variance of the noise (σ) in counts and the PSNR of the noisy image. The **bold** values indicates that the highest values is achieved with $SM(\mathcal{E}_{MAX})$ method while the non-bold is achieved by $BM(\mathcal{E}_{IP})$.

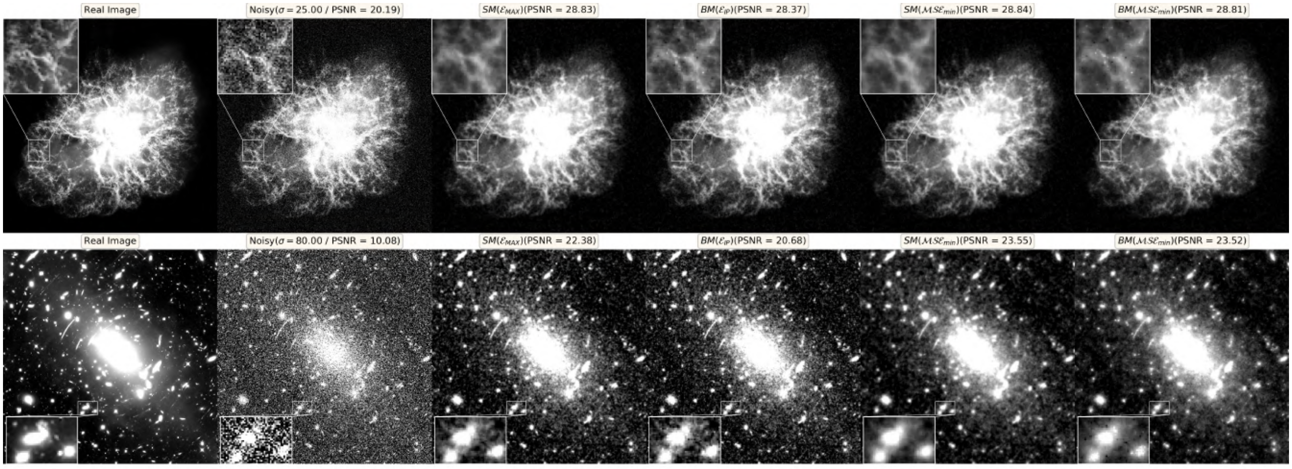

Real Image
Noisy(σ = 25.00 / PSNR = 20.19)
$SM(\mathcal{E}_{MAX})$(PSNR = 28.83)
$BM(\mathcal{E}_{IP})$(PSNR = 28.37)
$SM(\mathcal{MSE}_{min})$(PSNR = 28.84)
$BM(\mathcal{MSE}_{min})$(PSNR = 28.81)
Real Image
Noisy(σ = 80.00 / PSNR = 10.08)
$SM(\mathcal{E}_{MAX})$(PSNR = 22.38)
$BM(\mathcal{E}_{IP})$(PSNR = 20.68)
$SM(\mathcal{MSE}_{min})$(PSNR = 23.55)
$BM(\mathcal{MSE}_{min})$(PSNR = 23.52)

Fig. 8: Estimations compute for the *Crab* nebula (top) and galaxy *cluster* (bottom) by FABADA with noise levels of $\sigma = 25, 80$ counts, respectively. From left to right is shown the original image, the noisy image, the $SM(\mathcal{E}_{MAX})$ and $BM(\mathcal{E}_{IP})$ models, and also the optimal solutions of FABADA, $SM(MSE_{min})$, and $BM(MSE_{min})$. In top of the image is shown the PSNR (25) in decibels (dB) computed for that estimation.

62% (130 out of 209), implying that the Bayesian model has more potential to reduce the MSE, especially when the original signal-to-noise ration is high. Its smoothing is more conservative and requires more iterations, but the reconstructed model is closer to the original solution. For noisy data, the MSE metric always favors the more aggressive smoothing carried out by the single model $SM(\mathcal{E}_{MAX})$.

B. *Quality of the reconstruction*

We will now assess the ability of the different variants of our algorithm to recover the underlying signal. In order to facilitate the comparison with previous results reported in the literature, we will use the Peak Signal-to-Noise Ratio (PSNR) defined in expression (25) which is just a measure of the Mean Squared Error (MSE), expressed in decibel (dB).

The results of the different model selection criteria are illustrated in figure 8, using the images of the *Crab* nebula and the galaxy *cluster* as an example. The noise levels are $\sigma = 25$ and 80 counts, respectively, and the PSNR achieved by each method is quoted at the top of the corresponding panel. All models yield fairly similar reconstructions for the high signal-to-noise image of the *Crab* Nebula, and FABADA has been able to increase the PSNR from the 20.19 dB of the noisy data to 28.83 dB for $SM(\mathcal{E}_{MAX})$ and 28.37 dB for $BM(\mathcal{E}_{IP})$. This represents an improvement of almost one order of magnitude in MSE, and it is close to the optimum model when the number of iterations is adjusted *a posteriori*. For the noisy image of the galaxy *cluster*, with an initial PSNR of 10.07 dB, the effect of noise reduction is more noticeable. The maximum possible PSNR would be 23.55 dB for $SM(MSE_{min})$, once again with very little difference (0.03 dB) with respect to the Bayesian model $BM(MSE_{min})$. However, we notice that $SM(\mathcal{E}_{MAX})$ is able to achieve a PSNR of 22.38 dB, whereas $BM(\mathcal{E}_{IP})$ only reaches 20.68 dB.

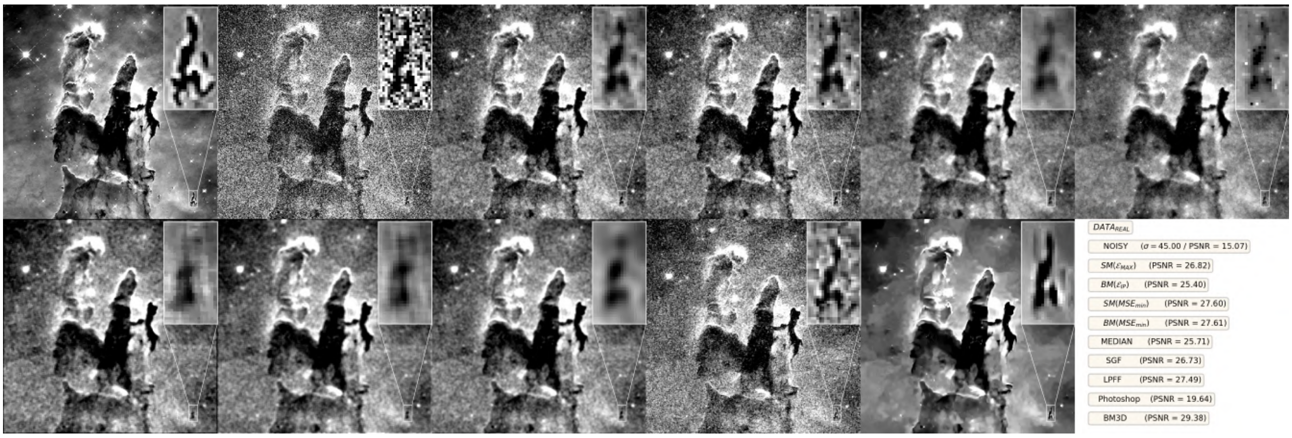

$DATA_{REAL}$
NOISY (σ = 45.00 / PSNR = 15.07)
$SM(\mathcal{E}_{MAX})$ (PSNR = 26.82)
$BM(\mathcal{E}_{IP})$ (PSNR = 25.40)
$SM(MSE_{min})$ (PSNR = 27.60)
$BM(MSE_{min})$ (PSNR = 27.61)
MEDIAN (PSNR = 25.71)
SGF (PSNR = 26.73)
LPFF (PSNR = 27.49)
Photoshop (PSNR = 19.64)
BM3D (PSNR = 29.38)

Fig. 9: Comparison example of the *Eagle* nebula for all the methods explained in III-A for a noise level of $\sigma = 45$ counts. From top to bottom, the first row of images shown are the real image, the noisy image, the recoveries found with $SM(\mathcal{E}_{MAX})$ and $BM(\mathcal{E}_{IP})$ model, and the optimal solutions $SM(MSE_{min})$, and $BM(MSE_{min})$. The second row again from left to right the images shown are the recovery using the Median, Savitzky-Golay, low-pass frequency, and Photoshop filter while the last one of this row is the BM3D recovery. In the last block is listed, from top to bottom, and left to right the PSNR of these recoveries.

In any case, the improvement in image quality is notable for both models. The images returned by $SM(\mathcal{E}_{MAX})$ are smoother, while $BM(\mathcal{E}_{IP})$ is more conservative. Thus, the MSE (or PSNR) tends to favor the former, although visual inspection hints that some details, as well as the overall sharpness of the image, may be somewhat blurred in the smoothest models. The Bayesian models do a better job at keeping any significant structures and/or jumps in the original data, losing less information, at the expense of a less powerful noise reduction that results in more significant image 'grain'. This effect is highlighted in the close-ups of the images, focusing on one of the characteristic filaments of the *Crab* nebula and a small group of galaxies in the *cluster*; all these structures seem to be slightly more defined in the Bayesian than in the single models.

Similar trends are observed in other images and noise levels (a further example is provided in figure 9) and one-dimensional spectra (figure 10), where $BM(\mathcal{E}_{IP})$ seems to adapt more faithfully to spectral features such as emission and absorption lines, even in presence of significant noise. These results make us think that perhaps the MSE is not the optimal metric to gauge the quality of the recovered solution or, more likely, that it should be complemented with another test statistic that quantifies information loss and/or gives more weight to informative features.

Figures 9 and 10 also plot the results obtained by the other algorithms described in Section III-A: the median, Savitzky-Golay (SGF), low-pass frequency (LPFF), Photoshop noise reduction, and block-matching (BM3D) filters for the images, and the median, SGF, locally weighted scatterplot smoothing (LOWESS), and LPFF for spectra.

Figure 9 shows the recovery of the *Eagle* nebula image, along the PSNR values obtained and a close-up of some structures to see whether the smoothing methods can reproduce their shape and edges. The best reconstruction in terms of the MSE is by the BM3D method, that improves the PSNR from $15.0.7$ dB to 29.38 dB, around one and a half orders of magnitude of noise reduction. This is around 50% more than the recovery with the optimal solution of FABADA (27.60 dB), and around twice the recovery of FABADA's models, $BM(\mathcal{E}_{IP})$ (25.40 dB) and $SM(\mathcal{E}_{MAX})$ (26.82 dB). While the optimal solutions of FABADA have closer results to the state of the art method (BM3D) for this example, there is still a lot of room for improvement for the $BM(\mathcal{E}_{IP})$ and $SM(\mathcal{E}_{MAX})$ models.

The classical filters (median, SGF, LPFF) yield values comparable to $SM(\mathcal{E}_{MAX})$ and slightly above $BM(\mathcal{E}_{IP})$ when their parameters are tuned to minimize the MSE, and about ≈ 1 dB ($\sim 25\%$) below FABADA's optimal solutions. The best possible solution for Photoshop is well below any of the other methods in terms of the MSE, with a difference of almost an order of magnitude.

On the other hand, it is interesting to note how, regardless of the MSE statistic, the median, SGF, and LPFF virtually miss the entire shape of the zoomed structure, while FABADA, Photoshop, and BM3D recover a more realistic shape. Lacking a quantitative comparison, our subjective opinion based on visual inspection would be that $BM(\mathcal{E}_{IP})$ and Photoshop provide a somewhat better reconstruction than BM3D in terms of fidelity to the original structure. In particular, several edges in the latter seem to be a little bit more blurry, artificially 'straightened' and/or aligned with the image axes; more importantly, the top left 'arm' is an artifact that has been amplified from a random statistical fluctuation of the noise, but was not present in the original data. Regarding overall visual appearance of the whole image, BM3D has been extremely successful in eliminating the high-frequency 'grain', albeit the smooth areas of the real image are transformed into more staggered gradients. This feature might be inherent to the BM3D algorithm, whose aim is to classify similar square sections of the image into groups, thus resulting in patches with relative constant gradients and/or straight edges. One may imagine that Photoshop, on the other extreme, tries to avoid this kind of visual effect, and therefore carries an extremely conservative smoothing procedure, but this hypothesis is difficult to test due to the private nature of the algorithm and the lack of proper documentation. The fully automatic model

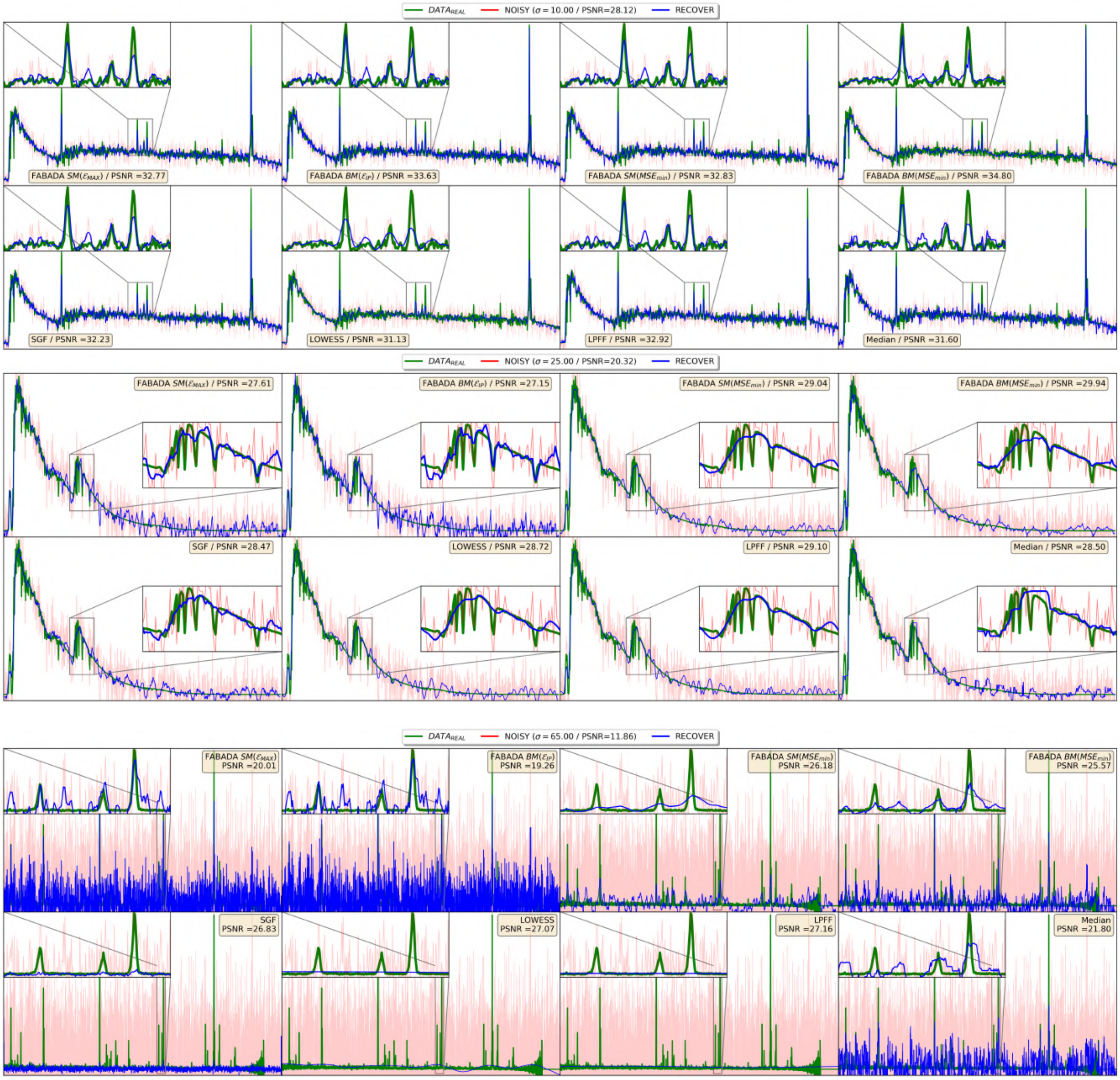
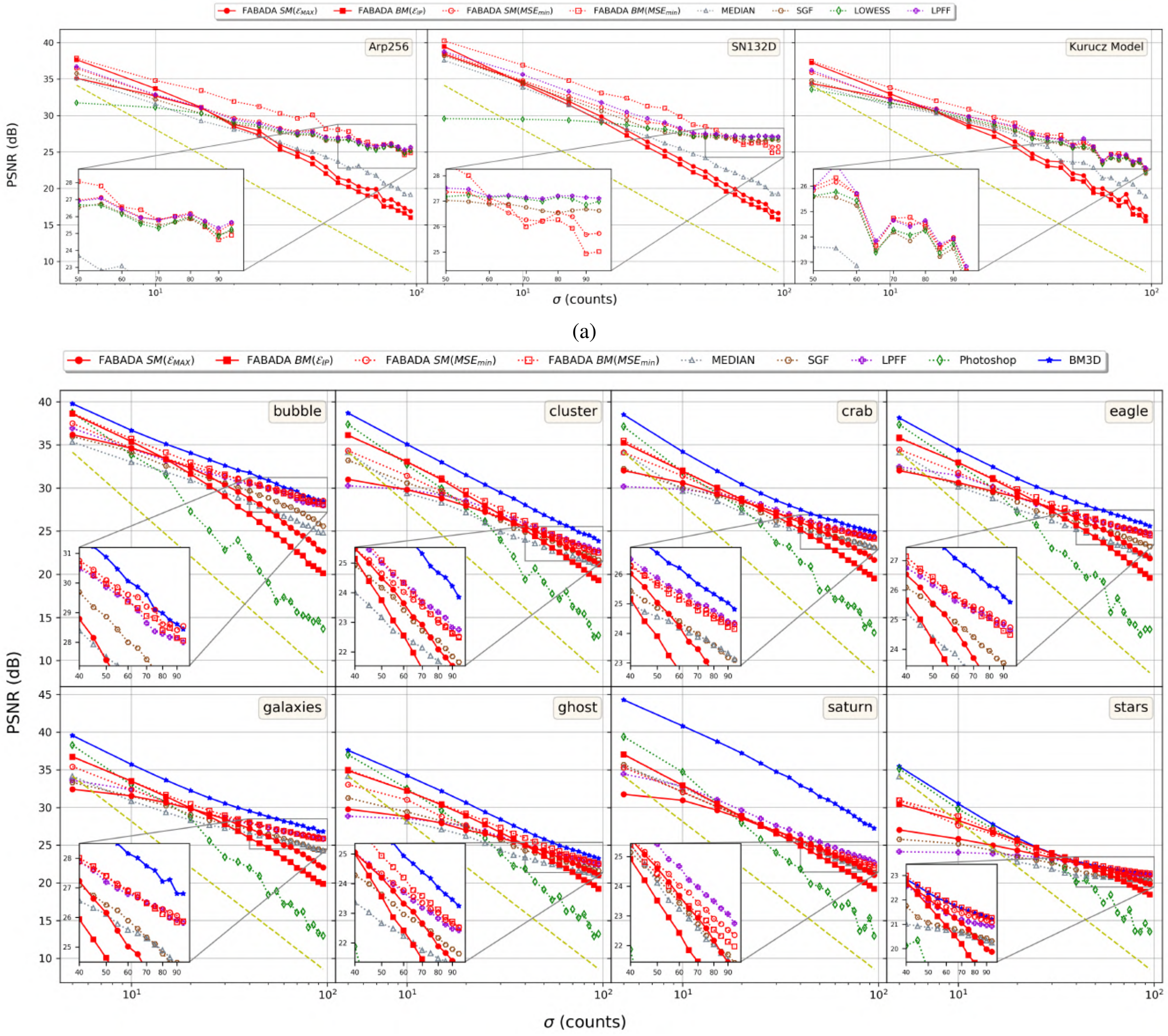


Fig. 10: Comparison example of the spectra data sample for all the methods explained in III-A. The top figure are the recoveries (blue line) found of the *Arp 256* spectrum (blue line) with a noise level of $\sigma = 10$ (red line). In the first row is represented the FABADA models and optimal solutions either for the single and Bayesian methods. In the second row are shown the recoveries found with the standards methods explained in III-A. From left to right, the Savitzky-Golay, the locally weighted scatterplot smoothing, the low-pass frequency filter, and the median methods are shown. With the same color code, and the same distribution, the middle set of figures are the recovery of the *Kurucz* model with a noise level of $\sigma = 25$. And the last one for the *SN132D* spectrum with a noise level of $\sigma = 50$.

selection methods implemented in FABADA, based on the mean evidence, seem to offer a reasonable compromise solution. Models $SM(MSE_{min})$ and $BM(MSE_{min})$ are able to reduce the MSE at the cost of significantly blurring the image. The effect is not as severe as in the standard methods (especially the median and SGF), but we do think that this aspect of the image reconstruction is not adequately captured by the MSE, and complementary test statistics should be sought.

This discussion becomes even more relevant in the context of astrophysical spectra. Figure 10 displays several examples, for the three different spectra with different noise levels: $\sigma = 10$ counts for the *Arp 256* spectrum (top), 25 for the *Kurucz* model (middle), and 50 for *SN132D*.

At high signal-to-noise ratio, all algorithms display not only a similar performance, but actually converge to very similar solutions. The highest value of PSNR is obtained by the optimized Bayesian model $BM(MSE_{min})$, but the improvement with respect to the originally high quality of the data is necessarily modest in all cases. Regarding the ability to reproduce the relevant structures, the close-up shows a zoom into the OIII-H$_\beta$ window at $\sim$ 5000 Å, where all algorithms are able to correctly trace the presence of the three emission lines that are present, albeit with somewhat reduced amplitude, and fail, as expected, to realize that one of the peaks in the noisy measurements



(a)

(b)

Fig. 11: Peak Signal to Noise Ratios (PSNR) obtained for all the data samples and noise ranges considered in the comparison process. In the top set of figures (a) is shown the results from the spectra sample while in (b) is represented the results from the image sample. Each figure of both groups is labeled with the reference name given. The red color corresponds to FABADA's models and optimal solutions proposed in this work and the dashed yellow line represents the PSNR of the noisy data. Dotted lines with unfilled symbols refers to the optimized methods $SM(MSE_{min})$ (○), $BM(MSE_{min})$ (□), SGF (⬡), LOWESS in 1D and Photoshop in 2D (◊), LPFF (✢) and median (△) while solid lines with filled symbols refers to the non-parametric methods $SM(\mathcal{E}_{MAX})$ (●), $BM(\mathcal{E}_{IP})$ (■) and BM3D (★).

is due to a random statistical fluctuation.

More significant differences appear for the *Kurucz* stellar atmosphere on the middle row, where it becomes more difficult to discriminate significant spectral features from Gaussian random noise. Zooming into the spectral region of the Balmer break at $\sim$ 4000 Å, all models display a similar behavior regarding the overall shape of the continuum, and they are able to correctly reproduce the break. The Balmer absorption lines, though, are much more difficult to recover, and all the methods that are tuned to optimize the MSE fail to trace them, although hints of H$_\beta$ and H$_\gamma$ are still present in the smoothed models. Only the evidence-based criteria $SM(\mathcal{E}_{MAX})$ and $BM(\mathcal{E}_{IP})$ are able to provide a good description of these features with this level of noise in the input data, albeit further lines along the Balmer series are completely lost.

If we now turn to the recovery ot the *SN132D* spectrum, we can see how the PSNR of LPFF, SGF and LOWESS is higher than the FABADA models, but basically all the spectral information is gone. At these high levels of noise, the MSE metric is absolutely inadequate to assess the quality of reconstructed emission line spectra, because it incurs in minimal penalty for failing to reproduce a handful of peaks that are barely statistically significant. Both criteria in the standard FABADA implementation are much more conservative, and keep a lot of the random fluctuations (hence their lower PSNR) together with the actual signal. It is somewhat remarkable that, even when optimized in terms of the MSE, FABADA and the median filter manage to recover some of the brightest lines, at variance with the other methods. Let us please stress once again that this does not necessarily imply a failure of the latter, but of the MSE as a goodness-of-fit indicator. On the other hand, it does highlight the robustness of FABADA and the median filter in this respect.

Focusing on the PSNR alone, the results for all data samples, noise levels, and methods are plotted as a function of σ in figure 11a for one-dimensional spectra and figure 11b for two-dimensional images.

In one dimension, we can easily see that the highest values of PSNR are achieved with the optimized Bayesian model of FABADA ($BM(MSE_{min})$), specifically 30 of the 57 (53%) tries, followed by the (also optimized) low-pass frequency filter (LPFF), who obtained the best PSNR for 24 (42%) estimations. The remaining 3 (5%) correspond to the optimize single model of FABADA ($SM(MSE_{min})$). If we don't take into account the optimal models of FABADA, 51 (89%) of the highest values are achieved with the optimal models of LPFF and the rest 6 (11%) with FABADA's Bayesian model ($BM(\mathcal{E}_{IP})$).

The LPFF achieves the best estimation evaluated with the MSE for the highest values of noise, specifically for $\sigma \gtrapprox 50$. We can again see how the models of FABADA diverge from the optimal solution for these higher values of noise. However, for lower noise levels, FABADA's models have similar performance than the optimal solutions of the standard models until $\sigma \lessapprox 25$. Above this level, the PSNR values are lower than all the standard models. Although high PSNR values can directly associate with best recoveries, we have to take into account that for almost all of the standard methods, the solution achieved for higher values of noise in the $SN132D$ spectrum has lost all the information contained. As we have just shown, these optimized solutions were not able to recover any emission line and were more similar to a straight line than to an astrophysical spectrum (see figure 10).

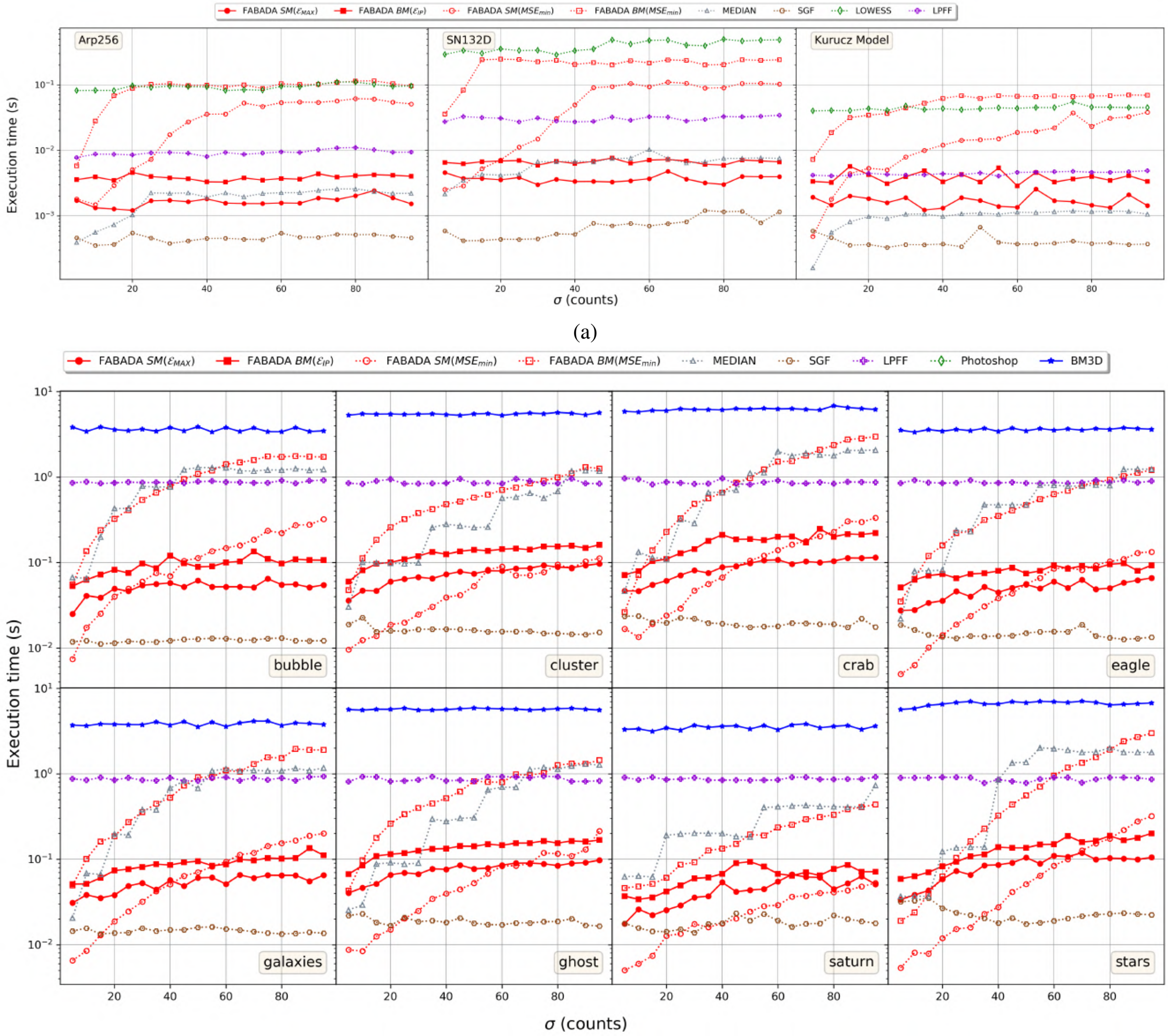
For the two-dimensional images (figure 11b) we can see that BM3D reaches the highest values of PSNR for almost all combinations of target and noise levels. For 139 out of 152 tests (91%), the highest values were achieved by BM3D. The remaining 13 (9%) correspond to FABADA's optimal models, $BM(MSE_{min})$ (12) and $SM(MSE_{min})$ (1). Although this gives an average of 1.9 dB higher values of PSNR than the best FABADA result, we can see that this difference varies for different images. For example, in the *Saturn* image the values of PSNR are about 10 dB higher than the rest, one order of magnitude in terms of the MSE, while for the *star* images there is a much lower difference and even for higher levels of noise does not dominate. This variance is driven by the grouping method, where for periodic images (or images with repetitive patterns), bigger groups can be formed that perform better in a collaborative filter. Such patterns can be easily found in *Saturn*'s image (in the atmosphere of the planet as well as its rings), whereas they are virtually absent in the *stars* image.

The difference between FABADA models and the other (optimized) solutions is much lower than for the spectra. Excluding BM3D from the comparison, the highest values are distributed in the following way: $BM(MSE_{min})$ accounts for 68 (45%), followed by LPFF with 43 (28%), then $SM(MSE_{min})$ with 29 (19%), and finally Photoshop with 12 (8%) in its favor. We can see that almost all the optimized solutions of the standard models behave in the same way, and have obtained similar results except for Photoshop. Photoshop achieves higher values of PSNR for lower values of noise ($\sigma \lessapprox 15$) while for higher noise levels it tends to perform much worse. This might be due to an optimization for lower noise levels, as they are arguably the most common in its domain of application.

Discarding also the optimized FABADA models in order to compare our evidence-based criteria with the optimized standard methods, we find that $BM(\mathcal{E}_{IP})$ and $SM(\mathcal{E}_{MAX})$ only reach the highest PSNR for 24 (16%) and 7 (5%) recoveries, respectively, although the behavior of all algorithms is fairly similar, and the average difference in PSNR in favor of one or the other is often not more than $1 - 1.5$ dB. In general, FABADA's $SM(\mathcal{E}_{MAX})$ and $BM(\mathcal{E}_{IP})$ models tend to have higher values of PSNR for $\sigma \lessapprox 40$. Above this level, the evidence-based models start to diverge from the optimal MSE solution. As can be seen in the zoomed insets, may be still considered competitive with SGF and the median filter in this regime, but they are below LPFF in terms of MSE.

C. Time Efficiency

For the whole sample of the recoveries process shown in the previous section, we have computed the time consumed for each of the methods to produce the estimation of the real data. The results can be shown, with the same color codes, in the figures 12a for the spectra sample and 12b for the image sample. For the optimized standard models, we have only taken into account the time consumed by the run with the optimal parameters, the optimization process is not considered. We weren't able to measure the time process for Photoshop. In table IV is listed, from lowest to highest, the average times taken in one dimension $\langle t_{1D} \rangle$ and for two dimension $\langle t_{2D} \rangle$ data sample. We can see that the behavior of the implementations is similar for both of the dimensions for all the methods. As expected, the fastest method is the SGF, due to its simple implementation where it only has to carry out a convolution between the coefficients, theoretically computed, and the data. Then is followed by FABADA's single $SM(\mathcal{E}_{MAX})$ and Bayesian $BM(\mathcal{E}_{IP})$ models, and followed by the median. In the one dimensional implementation, the median filter oscillates between FABADA's models, which depends on the box size used. The next fastest algorithm is the LPFF which also varies between the spectra data sample due to the difference in the length of the data. While *SN132D* had around 8000 data points, *Arp 256* had 2300 data points, and the *Kurucz* model only had around 1000. We also see that for the FABADA optimal models the amount of time spent has a dependence on the noise levels, being slower for higher values of noise. This can be understood with the results already shown, where we see that for the single and Bayesian models, there is no much difference in the iterations done for most noises while for the optimal models it requires many iterations for higher noise levels,



(a)

(b)

Fig. 12: Execution time, in seconds, taken to produce the recovered results as a function of the noise level (σ) for each of the data sample and methods. In the set of figures 12a is shown the results for each of the spectra sample while in the set of figures 12b is shown the results obtained for the images sample. Each figure is labeled with the reference name of the data, and the color and symbols codes used are the same as in figure 11.

producing the divergence of the iteration number between the models and the optimal solution. It also varies with the image, where it can vary the maximum and the inflection point of the average evidence and the iterations need to find the optimal solution. It is also interesting to see how this also affects highest values obtained in the one dimensional average. While the MSE is trying to destroy the information and achieve a smoother spectrum without any spectral line, our models is trying to save these features with the effects of the evidence. This is translate into much more iterations and even more for the Bayesian optimal solution, where its harder to get rid of these features due to the weighted average. The LOWESS algorithm have also a big dependence on the number of data points like LPFF, and it more or less took the same time than the slowest FABADA's optimal model $BM(MSE_{min})$. The biggest difference seen is with the BM3D model, where is the only model which is always above the second, with the highest average of $4.3s$. This implies a difference of about four times two orders of magnitude, in average, for FABADA's single and Bayesian model and for the single optimal model while for the optimal Bayesian model the difference oscillates from two orders of magnitude to 5 times due to the high oscillation of these models. This might not be seen as an important as it is, taking into account that we are on the second's scale, but these results have been obtained with a 512×512 pixel image which corresponds to a really low resolution for nowadays. The common resolution for images is around $\approx 3000 - 4000$ pixels which are not only resolutions for professional CCDs but also for most nowadays telephones. This is an increase in almost one order of magnitude per side, two from the total number of pixels. Assuming an ideal linear escalation with the number of pixels N,

Method	$\langle t_{1D} \rangle$	$\langle t_{2D} \rangle$
SGF	$0.5 \cdot 10^{-3}$	$1.7 \cdot 10^{-2}$
$SM(\mathcal{E}_{MAX})$	$2.3 \cdot 10^{-3} s$	$6.5 \cdot 10^{-2}$
$BM(\mathcal{E}_{IP})$	$4.8 \cdot 10^{-3}$	0.12
Median	$3.1 \cdot 10^{-3}$	0.7
LPFF	0.015	0.87
$SM(MSE_{min})$	0.09	0.08
$BM(MSE_{min})$	0.3	0.77
LOWESS	0.17	—
BM3D	—	4.81

TABLE IV: Average time taken to generate the recoveries for method given. These time have been computed from the results of figure 12. The values are listed from lowest to highest values, i.e, from fastest to slowest method.

BM3D would take around 7 min for an image with this resolution while for FABADA would only take around 20 s. Using the original resolution of the $Crab$ Nebula image, 4291 × 4291 pixels, $BM3D$ took 10 minutes while for $BM(\mathcal{E}_{IP})$ model took only $40s$ which implies it might have even bigger scaling. If we scale up a little more, and if we want to process the whole images of a survey, let's say like Pan-STARRS, this relationship would become more important. Pan-STARRS has 3π steradians of the sky covered and a scale of 0.26 $''/pixel$. This survey contains around $\approx 6 \cdot 10^{12}$ pixels and assuming a linear escalation with the number of pixels, which in reality is probable bigger, BM3D would take, at least, 3.48 years while FABADA's $BM(\mathcal{E}_{IP})$ would take only 31 days. In these kinds of automatic processes the only possible models used have to be nonparametric and the time efficiency plays a major role. While fast algorithms as FABADA, could deal with this heavy operation with multiple processors nowadays computers within no more than some days of CPU time, low methods would need to be executed in supercomputers in order to decrease the execution time to days which would become the main reason to be ruled out of this kind of tasks.

V. Conclusions and future work

In this work we present two new approaches to automate model selection in our Fully Adaptive Bayesian Algorithm for Data Analysis (FABADA): the single model $SM(\mathcal{E}_{MAX})$ maximizes the average evidence and the Bayesian model $BM(\mathcal{E}_{IP})$ combines all the iterations of the algorithm according to Bayes' theorem. We compare both prescriptions with other methods that are representative of the current state of the art in image analysis and digital signal processing.

Our results support the viability of FABADA as a competitive alternative, especially taking into account that is completely parameter-free. The two prescriptions developed in the present M.Sc. thesis appear to produce reliable smoothing results that adapt to the structure present in the input data. For the sake of a fairer comparison, we also investigate the FABADA solutions that minimized the Mean Square Error (MSE), as we did to set the parameters of the standard methods. The differences are minimal at high signal-to-noise ratios, but they become more significant as the noise increases.

However, the optimal solution in terms of the MSE might discard valuable information contained in the input data. As seen in figures 9 and 10, the MSE might not be the best metric for quantifying the recovery of physical structures within the data, leading to over-smoothing and loss of significant structures at low SNR. FABADA was meant to recover information in a statistically robust way, and our automatic prescriptions perform a more conservative smoothing. Even when the algorithm is optimized with respect to the MSE, traces of the most prominent features can still be found in the recovered solution. On the other hand, it is also of major importance to find a metric that strongly penalizes the introduction of artificial features that might be later interpreted as a real signal. This is indeed a potential pitfall for BM3D, whose results in two dimensions clearly stand out over all other methods, including FABADA, in terms of MSE.

Besides the quality of the reconstruction, execution time might become a significant issue for sufficiently large data sets. BM3D is fairly computationally intensive in this respect. The standard algorithms are also very fast, but the optimization process (which is not possible in a real case) would require a potentially time-consuming exploration of their parameter space. The two prescriptions present here are parameter-free, and they can considerably improve the PSNR within modest CPU times. These two properties make FABADA a good candidate, for instance, for video noise reduction: for 2 minutes at 4K resolution and 20 frames per second, an extrapolation of the results shown in figure 12b yields 1.4 h in a single processor without any previous analysis of the video.

Further work along these lines should focus on several improvements. First, our results evidence that the quality of the FABADA solution could benefit from aspects borrowed from the other methods, such as including information about the gradients of the image into the prior probabilities (BM3D) or using another smoothing techniques (SGF, LPFF) instead of the moving median to produce the priors of each new iteration. Other ideas that would be worth exploring would be to test the statistical distribution of the residuals and adapt the iteration process to each pixel evaluating the optimal number of iterations on an individual basis. For the time being, a new, fully functional automatic method of noise reduction has been generated, which can be applied in its current state to any type of one and two-dimensional data.

References

P Bromiley. Products and convolutions of gaussian distributions. *Division of Informatics, Imaging and Data Sciences, School of Health Sciences, University of Manchester*, pages 1–2, 01 2003.

William S. Cleveland. Robust locally weighted regression and smoothing scatterplots. *Journal of the American Statistical Association*, 74(368):829–836, 1979. doi: 10.1080/01621459.1979.10481038.

URL https://www.tandfonline.com/doi/abs/10.1080/01621459.1979.10481038.

K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, 16 (8):2080–2095, 2007.

Majed El Helou and Sabine Susstrunk. Blind universal bayesian image denoising with gaussian noise level learning. *IEEE Transactions on Image Processing*, 29: 4885–4897, 2020. ISSN 1941-0042. doi: 10.1109/tip.2020.2976814. URL http://dx.doi.org/10.1109/TIP.2020.2976814.

Yongcheng Jing, Yezhou Yang, Zunlei Feng, Jingwen Ye, Yizhou Yu, and Mingli Song. Neural style transfer: A review. *IEEE Transactions on Visualization and Computer Graphics*, 2019.

V. Katkovnik, V.I.A. Katkovnik, K. Egiazarian, and J. Astola. *Local Approximation Techniques in Signal and Image Processing*. SPIE Press monograph. SPIE Press, 2006. ISBN 9780819460929. URL https://books.google.to/books?id=YChTAAAAMAAJ.

Thomas Knoll and John Knoll. Adobe Photoshop CC 2018 V.19, 1990-2017. URL https://www.adobe.com/products/photoshop.html.

Diaz R. I. Laidler V. Lim, P. L. Pysynphot user's guide (baltimore, md: Stsci), 2015. URL http://synphot.readthedocs.io/en/latest/.

Abraham. Savitzky and M. J. E. Golay. Smoothing and differentiation of data by simplified least squares procedures. *Analytical Chemistry*, 36(8):1627–1639, 1964. doi: 10.1021/ac60214a047. URL https://doi.org/10.1021/ac60214a047.

Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, Jul 2017. ISSN 1941-0042. doi: 10.1109/tip.2017.2662206. URL http://dx.doi.org/10.1109/TIP.2017.2662206.