

An introduction on bayesian parameter inference and its applications in cosmology

Luis Padilla-Albores,^{1,*} Alberto Vazquez-Gonzalez,^{1,†} and Luis O. Tellez^{1,‡}

¹*Departamento de Física, Centro de Investigación y de Estudios Avanzados del IPN, A.P. 14-740, 07000 México D.F., México.*

(Dated: October 14, 2018)

In this paper we review the basic concepts on Bayesian statistics for parameter inference and how we can use it in cosmology. This work is organized in such a way that if the reader is only interested in the parameter inference procedure for a model, but not in the cosmological part, he/she can make use of it ignoring only the last part of the paper.

First, we start by giving the basic differences between Bayesian and Frequentist statistics. Then we continue with a not so formal introduction to the basic mathematical concepts necessary for a Bayesian parameter inference procedure. Later, we review the most common computational tools at our disposal that can help us to simplify this task. Finally we show how this new concepts can be used in cosmology.

INTRODUCTION

The beginning of the standard cosmology as it is known today emerged after 1920 when the Shapley-Curtis debate was carried out [1]. This debate was held between the astronomers Harlow Shapley and Heber Curtis, resulting in a revolution for astronomy at that time: “The universe had a larger scale than the Milky Way galaxy”. Several observations at that epoch established the size and the dynamics of the cosmos that could only be explained by Einstein’s General Theory of Relativity. In its childhood, Cosmology was a speculative science that was based only on a few data sets and characterized by a dispute between two cosmological models: the steady state model and the Big Bang (BB) theory. It was not until 1990 when the amount of data was enough to eliminate competing theories, being awarded the BB model as the most accepted cosmological theory. In that same decade David Schramm heralded the “Golden Age of Cosmology” at a National Academy of Sciences colloquium.

Once the new age of cosmological experiments arrived with a very large variety of cosmological data, it was necessary to confront all the cosmological models with said data. The way that it is usually done is using a statistical confrontation. First of all we need to notice that, since we only have one Universe, we can not consider a frequentist interpretation of statistics (we can not create multiple Universes to make a frequentist inference of our models). An alternative interpretation that can help us is the Bayesian statistics. In Bayesian statistics probability is interpreted as a “degree of belief” and it can be useful when no repetitive processes need to be considered.

The main objective of this work is to give the reader an introduction on bayesian parameter inference and how we can use it in cosmology. We assume that the reader is familiarized with the basic concepts of statistics, but not necessarily with Bayesian statistics. Then, we give a not so formal general view of it, enough to understand the basic concepts of our main objective. This general view is written in a generic way so that if the reader is not interested in the cosmological section but in the parameter inference one he/she can draw on the bayesian part.

This paper is organized as follows. We start in the section [Bayesian vs Frequentist statistic] mentioning the most important differences between the Bayesian statistics and the Frequentist one. Then, in section [A first look on the Bayesian statistics] we present the basic mathematical concepts in Bayesian statistics that are going to be necessary at the moment that we want to estimate a model parameter. Once we have the mathematical concepts, we continue in section [Numerical tools] with the numerical tools at our disposal that can help us simplify our homework. Such numerical tools are very important given the fact that, in general, it is not possible to apply the analytical ones when several parameters of our models need to be confronted with data. In section [Bayesian statistics and cosmology] we show how these tools can be used in cosmology, mentioning the different numerical codes free to download on web that are programmed to do this homework and applying them to specific examples. Finally, in section [Conclusions] we conclude this paper.

BAYESIAN VS FREQUENTIST STATISTICS

Fundamentally, the main difference between Bayesian and Frequentist statistics is their definition of probability. In a frequentist point of view probability has meaning in a limiting case of repeated measurements

$$P = \frac{n}{N} \quad (1)$$

where n denotes the number of successes and N the total number of attempts. Frequentist statistics define probability as the limit for the number of independent trials going to infinity (**cambiar definicion, suena rara**). Then, **for frequentist statistics, probabilities are fundamentally related to frequencies of events**. On the other hand, in bayesian statistics the concept of probability is extended to cover degrees of certainty about a statement. **For Bayesian statistics, probabilities are fundamentally related to our knowledge about an event**.

On this part we introduce the basic concepts necessary to understand the basic consequences that this discrepancy entails. For an extended review see [2], [3], [4], [5] or/and [6].

If we consider that x is a random variable related to a particular event and $P(x)$ its corresponding probability distribution, for both cases the same rules of probabilities apply¹:

$$P(x) \geq 0 \quad (2a)$$

¹ This rules are defined for a continuous variable; however, the corresponding discrete definition can be given immediately by replacing $\int \rightarrow \sum$.

$$\int_{-\infty}^{\infty} dx P(x) = 1 \quad (2b)$$

For mutually exclusive events

$$P(x_1 \cup x_2) = P(x_1) + P(x_2) \quad (2c)$$

Generally

$$P(x_1, x_2) = P(x_1)P(x_1|x_2) \quad (2d)$$

The last rule can be read as: the probability of x_1 and x_2 to happen is equal to the probability of x_1 times the probability of x_2 given that x_1 has already happened.

These rules for probability distributions must be fulfilled by both frequentist and bayesian statistics. But what are the consequences derived by the fact that these two scenarios have a different definition of probability? In the next section we will try to answer this question.

Frequentist statistics

Any frequentist inferential procedure relies on three basic ingredients: the data, a model and an estimation procedure. The main assumption in Frequentist statistics is that the data has a definite, albeit unknown, underlying distribution to which all inference pertains.

The *data* is a measurement or observation, denoted by X , that can take any value from a corresponding *sample space*. A *sample space* of an observation X can be defined as a measurable space (x, \hat{B}) that contains all values that X can take upon measurement.

In Frequentist statistics it is considered that there is a probability function $P_0 : \hat{B} \rightarrow [0, 1]$ in the sample space (x, \hat{B}) representing the “true distribution of the data”

$$X \sim P_0$$

Now we have the model. For Frequentist statistics a model Q is a collection of probability measurements $P : \hat{B} \rightarrow [0, 1]$ in the sample space (x, \hat{B}) . The distributions P_θ are called model distributions. In this approach θ is unchanged.

A model Q is said to be well-specified if it contains the true distribution of the data P_0 , i.e.

$$P_0 \in Q$$

Finally, we need a point-estimator(or estimator) for P_0 . An estimator for P_0 is a map $\hat{P} : x \rightarrow Q$, representing our “best guess” $\hat{P} \in Q$ for P_0 based on the data X .

Hence, the Frequentist statistics is based on trying to answer the following questions: “what does the data clarify about P_0 ?” or “from the data, what can we say about the mean of P_0 ?”.

Bayesian statistics

As it is explained in [2], in bayesian statistics, data and model form two elements of the same space, i.e. no formal distinction is made between measured quantities X and parameters θ . One may envisage the process of generating a measurement’s outcome $Y = y$ as two draws, one draw for Θ (or Q) to select a value of θ (or distribution P_θ) and a subsequent draw for P_θ to arrive at $X = x$ (que es draw en este contexto?). This perspective may seem rather absurd in view of the definitions for a frequentist way of thinking, but in a bayesian one, where probabilities are related to our own knowledge, it results natural to associate probability distributions to our parameters. In this way an element P_θ of the model is interpreted simply as the distribution of X given the parameter value θ , i.e. as the conditional distribution $X|\theta$.

Frequentist	Bayesian
Data are a repeatable random sample. There is a frequency	Data are observed from the realized sample
Underlying parameters remain constant during this repeatable process	Parameters are unknown and described probabilistically
Parameters are fixed	Data are fixed

TABLE I: Main differences between the bayesian and Frequentist interpretations.

A bit clearer

It is very important to understand the differences between both approximations. For it, let us review in this subsection the mandatory example that can help us to easily understand these differences.

In table I we have a little review of the most important differences about the two approximations. In the next example we present an experiment seen from both points of view. Since we are interested in knowing both descriptions, we show only the basic results of the procedures and we analyze them in the point of view of both the Frequentist and Bayesian statistics.

Example.- In this experiment we have a coin that has a probability p to land as heads and a probability $1 - p$ to land tails. Trying to estimate p (that must be $p = 0.5$ since we have two possible states) we flip the coin 14 times, obtaining heads in 10 of them. Now we are interested in the next two possible events. To be precise: “What is the probability that in the next two tosses we will get two heads in a row?”

- *Frequentist approximation.* As mentioned previously, in Frequentist statistics probability is related to the frequency of events, then our best estimate for p is $P(head) = p = \text{No.heads}/\text{No.total} = 10/14$. So, the probability of having 2 heads in a row is $P(2heads) = P(head)P(head) \simeq 0.51$.
- *Bayesian approximation.* In the bayesian approach p is not a value, it is a random variable with its own distribution. This distribution must be defined by the existing evidence. In this example a good distribution for p is a binomial distribution. Then, by considering a uniform distribution as our prior information (we do not know anything about p) we have that the probability of having two heads is

$$P(2heads|D) = \frac{B(13, 5)}{B(11, 5)} = 0.485$$

where $B(x, y)$ is the beta function. This distribution appears given the fact that $B(1, 1)$ is the uniform distribution which we consider as our prior (reescribir esta oracion, pero no se me ocurre como :,c).

We can see that both approximations arrive at different results. In one of them, since we consider probability as a frequency of events we have a bigger probability to have two heads in a row than when we consider probability as our knowledge about the experiment, however, in both cases, the probability differs from the real one ($P(2heads) = 0.25$) because we don't have enough data for our estimations.

Note: If you are not familiarized with Bayesian statistics, please don't be scared of the last example. In the next section we will introduce the basic concepts to understand the bayesian part and then we will go back to this example to solve it using the new tools acquired.

(Tratar de escribir esta parte un poco más bonito)

A FIRST LOOK AT BAYESIAN STATISTICS

Before we start with the applications of bayesian statistics in cosmology it's necessary to learn the most important tools for the bayesian analysis. In this section, we review them in an informal way, keeping in mind that the reader can look for the formal treatment in literature.

Bayes theorem, priors, posteriors and all that stuff

When a statistician is interested in working with the Bayes framework, there are several concepts that are necessary to understand before he can interpret the results obtained. In this section we quickly review these concepts and then we take back the example about the coin toss given in the last section.

The Bayes theorem. The Bayes theorem is a direct consequence of the axioms of probability (2). We can see that, from (2d), without loss of generality, we can rewrite $P(x_2, x_1) = P(x_2)P(x_2|x_1)$. Considering that the relation $P(x_1, x_2) = P(x_2, x_1)$ has to be fulfilled, we arrive at the Bayes formula

$$P(x_2|x_1) = \frac{P(x_2)P(x_1|x_2)}{P(x_1)} \quad (3)$$

As it was mentioned, in the Bayes framework, data and model are part of the same space. In this manner, considering $x_1 \rightarrow D$ as a set of data, and $x_2 \rightarrow H$ as our "hypothesis", we can rewrite the above equation in Bayesian statistics as

$$P(\theta, H|D) = \frac{P(\theta, H)P(D|\theta, H)}{P(D)} \quad (4)$$

Here we have added the extra term θ in order to specify that our hypothesis can depend on one or several parameters. The above equation is the so-called *Bayes theorem* and it's the most important tool in a Bayesian inference procedure. In this result, $P(\theta, H|D)$ is called the *posterior* probability for the model. $L(D; \theta) \equiv P(D|\theta, H)$ is called the *Likelihood* and it will be our main focus in a future section. $P(D|\theta, H)$ is called the *prior* and expresses what we know about our model before acquiring the data. This prior can be fixed depending on either previous experiment results or the theory that we are working with. $P(D)$ is the evidence of our model. We notice that this evidence acts only as a normalizing factor

$$P(D) = \int d\theta P(D|\theta, H)P(\theta, H). \quad (5)$$

This quantity is usually ignored for practical reasons when testing the parameter space of a unique model. On the other side, if what we are interested in is a comparison between 2 or more models, the evidence plays an important role when we choose which model is more likely to be the "real one", whatever the parameters are. For the purpose of this work, we are not interested in this scenario and we will ignore it at all times.

The reader can notice that we have added here a new ingredient in our Bayes description: the hypothesis. An hypothesis is our best guess at which model best fits the data, $H = Q_{best}$.

We can see that Bayes theorem has an enormous implication with respect to an statistical inferential point of view. In a typical scenario we can collect some data and hope to interpret it with a given model, however, what we can usually do is the opposite, that is, first we have a set of data and then we can confront a model considering what is the probability that our model fits the data. As we can see from (4), Bayes theorem gives us a tool that allows us to relate both scenarios. Then, thanks to the Bayes theorem, in principle, we can know what is the "real" model that best fits the data.

Example.- We consider again the example shown in the last section: the coin toss. What we are interested in is knowing what is the probability $P(2heads|D)$ (D = the previous 14 coin tosses acting as data) to have 2 heads in a row given the data. For simplicity, we do not update p between the two tosses and we assume that both tosses are independent from each other. To obtain the result that we expect we need to calculate

$$P(2heads|D) = \int_0^1 P(2heads|p)P(p|D)dp \quad (6)$$

where $P(p|D)$ is the probability of the data given our model p and $P(2heads|p)$ is the probability of our model p given that we had 2 heads. Since we are considering that both tries are independent we have

$$P(2heads|p) = [P(head|p)]^2 \quad (7)$$

where $P(head|p)$ is the probability of our model p given that we obtained heads once. A good model for our experiment, given the data, is a binomial distribution

$$P(D|p) = \binom{14}{10} p^{10} (1-p)^4 \quad (8)$$

Then

$$P(head|p) = p \Rightarrow P(2head|p) = p^2 \quad (9)$$

Now we need to compute $P(p|D)$. Using the bayes theorem we have

$$P(p|D) = \frac{P(D|p)P(p)}{P(D)} \quad (10)$$

A very convenient prior distribution for this scenario is the *beta distribution* $Beta(p; a, b)$ defined as

$$Beta(p; a, b) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} p^{a-1} (1-p)^{b-1} \quad (11)$$

where Γ is the gamma function. So

$$P(p) = Beta(p; a, b) \quad (12)$$

We are interested in the explicit form of $P(p|D)$ in such case we need to compute $P(D)$. Introducing (8) and (12) into (5) we have

$$P(D) = B(10+a, 4+b) \equiv \frac{\Gamma(10+a)\Gamma(4+b)}{\Gamma((10+a)+(4+b))} \quad (13)$$

and then

$$P(p|D) = \frac{p^{10+a-1} (1-p)^{4+b-1}}{B(10+a, 4+b)} \quad (14)$$

(revisar todo este ejemplo de la moneda, hay algo que no me quedo claro)

Now we need to know the values of a and b . If we assume that we know nothing about p , we can consider our prior as a uniform distribution, this means that $a = b = 1$. Notice that, from figure 1, our posterior result (12) doesn't agree with the real result ($p=0.5$). We would expect that our posterior distribution is centered at $p = 0.5$ with a very thin distribution. This disagreement can be fixed if we increase our experimental data.

FIG. 1: Prior and posterior distributions for the coin example. The black line corresponds with the known real value of p .

Solving the integral in (6) with (9) and (12) we arrive at the final result obtained in the previous section

$$P(2heads|D) = \frac{B(13, 5)}{B(11, 5)} = 0.485 \quad (15)$$

Updating the probability distribution for a model

As seen in the coin example, we couldn't arrive to the real value of p because we didn't have enough data. If we want to be closer to the real value, we would have to keep flipping the coin until the amount of data was big enough. Let's continue with the coin example: after throwing it 100 times we obtain, let's say, 56 heads, or after throwing it 500 times we obtain 246 heads, we should obtain a tinier distribution with the center close to $p = 0.5$ (see fig. 2). Given this, it is clear that, in order to confront a parameter model and be more accurate about the most probable (or "real") value of said parameter, it is necessary to increase the amount of data (and the precision) in any experiment. Then, what we have here are some model's parameters that have to be confronted with different set of data. This can be done in two ways: first one is, regarding the sum of all the sets of data that we have; second one is, consider each data set as the new data, but our prior information has to be set by what we know about the previous information. The important thing in Bayesian statistics is that it doesn't matter which one of the 2 possibilities we choose. In the coin toss example it means that it is equivalent to start with the prior given in figure 2-a and considering the 500 data we can arrive at the posterior 2-d, or start with 2-c as our prior and consider the last 400 data to obtain the same posterior 2-d.



FIG. 2: Posterior distributions of p when our data is increased. Notice that while we continue increasing the experimental results, the posterior distribution starts to be more localized near the real value $p = 0.5$.

In fact, if we rewrite Bayes theorem so that all probabilities are explicitly dependent on some prior information I

$$P(\theta, H|DI) = \frac{P(\theta, H|I)P(D|\theta, HI)}{P(D|I)} \quad (16)$$

and then we consider a new set of data D' , letting the old data become part of the prior information $I' = DI$, we arrive at [3]

$$P(\theta, H|D'I') = \frac{P(\theta, H|I)P(DD'|\theta, HI)}{P(DD'|I)} = P(\theta, H|[DD']I) \quad (17)$$

where we can explicitly see the equivalence of the 2 different options.

About the Likelihood

We mentioned that the evidence in the Bayes theorem is usually not important when we try to do any inference procedure in the parameter space of a single model. Then, we can fix it as $P(D) = 1$ without loss of generality. If we ignore the prior² we can identify the likelihood with $P(D|H) = L(D; H)$ and thus, by maximizing it, we can find the most probable model (or model's parameters) for the given data. However, having ignored $P(D)$ and the prior, this approach cannot give a good fit and thus cannot give an absolute probability for a given model, but it can give relative probabilities. On the other side, it is possible to report results independently of the prior by using the *Likelihood ratio*. The likelihood at a particular point in the parameter space can be compared with the one that best fit the observations, L_{max} . Then we can say that a model is acceptable if the likelihood ratio

$$\Lambda = -2 \ln \left[\frac{L(D; H)}{L_{max}} \right] \quad (18)$$

² It is expected that the real value of a given parameter is independent of the prior.

is bigger than a given value.

On the other side, let us assume we have a posterior distribution, which is single-peaked. We consider that $\hat{\theta}$ is the peak of the distribution (most probable value) or the mean

$$\hat{\theta} = \int d\theta \theta P(\theta, H|D). \quad (19)$$

If our model is well-specified and the expectation value of $\hat{\theta}$ corresponds with the real value θ_0

$$\langle \hat{\theta} \rangle = \theta_0, \quad (20)$$

then we say that $\hat{\theta}$ is *unbiased*. Considering a Taylor expansion of the *log likelihood*

$$\ln L(D; H) = \ln L(D; H_0) + \frac{1}{2}(\theta_\alpha - \theta_{0\alpha}) \frac{\partial^2 \ln L}{\partial \theta_\alpha \partial \theta_\beta} (\theta_\beta - \theta_{0\beta}) + \dots \quad (21)$$

where H_0 corresponds with the real model (and/or parameter of the model). In this manner, we have that the likelihood can be expressed as a multi-variable likelihood given by

$$L(D; H) = L(D; H_0) \exp \left[-\frac{1}{2}(\theta_\alpha - \theta_{0\alpha}) H_{\alpha\beta} (\theta_\beta - \theta_{0\beta}) \right] \quad (22)$$

where

$$H_{\alpha\beta} = \frac{\partial^2 \ln L}{\partial \theta_\alpha \partial \theta_\beta} \quad (23)$$

is called the *Hessian matrix* and it controls whether the estimates of θ_α and θ_β are correlated or not. If it is diagonal, these estimates are uncorrelated.

The above expression for the likelihood is a good approximation as long as our posterior distribution possesses a single-peak. However, in a general way, the likelihood may not be well described by a gaussian expression at levels which set the interesting credibility levels. It is worth mentioning that, if the data errors are normally distributed, then the likelihood for the data will be a Gaussian function as well. In fact, this is always true if the model is linearly dependent on the parameters [5]. On the other side, if the data is not normally distributed we can resort to the central limit theorem. In this way, the central limit theorem will tell us that the resulting distribution will be best approximated by a multi-variate Gaussian distribution [5].

Justifying the neglect of the priors

In this section we are interested in justifying the neglect of the prior in the Bayes theorem. For this, we follow the example given in [4]. In this example there are 2 persons, A and B, that are interested in the measurement of a given physical quantity θ . A and B have different prior beliefs regarding the possible value of θ . This discrepancy could be given by the experience, such as the possibility that A and B have made the same measurement at different times. Let us denote their priors by $P(\theta|I_i)$, ($i = A, B$), and let us assume that they are described by two Gaussian distributions with mean μ_i and variance Σ_i^2 . Now, A and B make a measurement together and they obtain the value $\theta_0 = m_1$. If we consider that the experiment is such that our parameters are uncorrelated we can rewrite (22) as

$$L(D; HI) = L_0 \exp \left[-\frac{1}{2} \frac{(\theta - m_1)^2}{\sigma^2} \right]. \quad (24)$$

By replacing the hypothesis H by the continuous variable θ the Bayes theorem for the model of A and B becomes

$$P(\theta|m_1) = \frac{L(m_1; \theta I_i) P(\theta|I_i)}{P(m_1|I_i)} \quad (25)$$

where we have used the notation given in (16). Then, the posterior of A and B are (again) Gaussian with mean

$$\hat{\mu}_i = \frac{m_1 + (\sigma/\Sigma_i)^2 \mu_i}{1 + (\sigma/\Sigma_i)^2} \quad (26)$$

and variance

$$\tau_i^2 = \frac{\sigma^2}{1 + (\sigma/\Sigma_i)^2}, \quad (i = A, B) \quad (27)$$

Thus, if the likelihood is more informative than the prior i.e. $(\sigma/\Sigma) \ll 1$ the posterior means of A and B will converge towards the measured value, m_1 . As more and more data are obtained one can simply replace the value of m_1 in the above equation by the mean $\langle m \rangle$ and σ^2 by σ^2/N (que es sigma minuscula?). Then, we can see that the initial prior μ_i of A and B will progressively be overridden by the data. This process is ilutrated in figure 3.

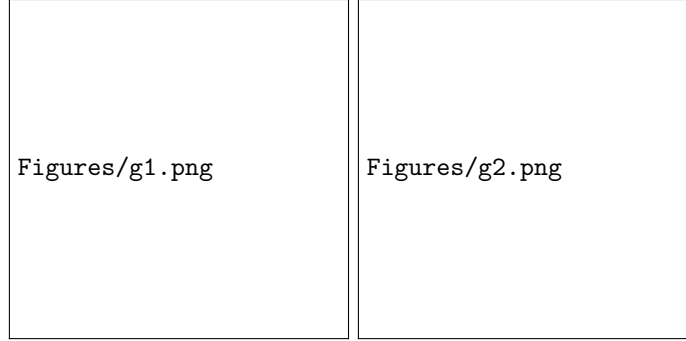


FIG. 3: Converging views in Bayes inference, taken from [4]. A and B have different priors $P(\theta|I_i)$ about a value of θ (panel (a)). Then, they observe one datum with likelihood $L(\theta; HI)$ (panel (b)), after which their posteriors $P(\theta|m_1)$ (panel (c)) are obtained. Then, after observing 100 data, it can be seen how both posteriors are practically indistinguishable (panel (d))

Chisquare and goodness of fit

We mentioned that it is necessary to maximize the likelihood in order to obtain the most probable model (or model parameters) given the data. If we consider the Gaussian approximation given in (24) we can see that the likelihood will be maximum if the quantity

$$\chi^2 \equiv (\theta_\alpha - \theta_{0\alpha})H_{\alpha\beta}(\theta_\beta - \theta_{0\beta}) \quad (28)$$

is minimum. The quantity χ^2 is usually called *chi-square* and is related to the Gaussian likelihood via $L = L_0 e^{-\chi^2/2}$. We can say that the maximizing of a Gaussian likelihood procedure and the minimizing of a chisquare procedure are equivalent. However, as we mentioned before, there are some circumstances where the likelihood can't be well specified by a Gaussian distribution, in those cases the chi-square and the likelihood are no longer equivalent.

We can consider a probability distribution for different values of χ^2 around its minimum. This is the χ^2 distribution for $v = n - M$ degrees of freedom where n is the number of independent data points and M the number of parameters. Hence, we can calculate the probability that an observed χ^2 exceeds by chance a value $\hat{\chi}$ for the correct model. This probability is given by [8] $Q(v, \hat{\chi}) = 1 - \Gamma(v/2, \hat{\chi}/2)$, where Γ is the incomplete Gamma function. Then, the probability that the observed χ^2 (even the correct model) is less than a given value $\hat{\chi}^2$ is $1 - Q$. This statement is strictly true if the errors are Gaussian and the model is a linear function of the likelihood, i.e., for Gaussian likelihoods.

If we evaluate the quantity Q in the best-fit chi-square (i.e. its minimum) we can have a measure of the goodness of the fit. If Q is small (small probability) we can interpret it as:

- The model is wrong and can be rejected
- The errors are underestimated
- The measurement errors are not normally distributed.

On the other side, if Q is too large there are some reasons that could cause such overestimation:

- Errors have been overestimated
- Data are correlated or non-independent
- The distribution is non-Gaussian.

Contour plots and confidence regions

Once the best fit parameter values are obtained we would like to know if there are confidence regions where other values could be considered as good candidates for our model. The most logical election is to consider values inside a compact region around the best fit value. Then, a natural choice is to consider regions with constant χ^2 boundaries. In the case that χ^2 possesses more than 1 minimum, it is said that we have more than one non-connected confidence region. For multi-variate Gaussian distributions (as the likelihood approximation (24)) these are ellipsoidal regions. In this section we exemplify how to calculate the confidence regions following [5].

We can consider a little perturbation from the best fit of chisquare $\Delta\chi^2 = \chi^2 - \chi_{best}^2$. Then we can use the properties of χ^2 distribution to define confidence regions for variations on χ^2 to its minimum. In Table II we can see the typical 68.3%, 95.4% and 99.5% confidence levels as a function of number of parameters for the joint confidence level. In the case of Gaussian distribution (as the likelihood) these correspond to the conventional 1, 2 and 3 σ (revisar esta parte y la tabla).

σ	p	$M = 1$	$M = 2$	$M = 3$
1σ	68.3%	1.00	2.30	3.53
2σ	95.4%	4.00	6.17	8.02
3σ	99.73%	9.00	11.8	14.20

TABLE II: $\Delta\chi^2$ for the conventional 68.3%, 95.4% and 99.73% as a function of the number of parameters for the joint confidence level.

The general cooking recipe to compute constant χ^2 confidence regions is as follows: After finding the best fit by minimizing χ^2 (or maximizing the likelihood) and if Q for the best parameters is acceptable, then:

1. Let M be the number of parameters, n the number of data and p be the confidence limit desired.
2. Solve the equation:

$$Q(n - M, \min(\chi^2) + \Delta\chi^2) = p \quad (29)$$

3. Find the parameter region where $\chi^2 \leq \min(\chi^2) + \Delta\chi^2$. This defines the confidence region.

Marginalization

It is clear that a model can (in general) depend on more than one parameter. However, most of this parameters θ_i may be uninteresting. For example, these parameters can correspond to nuisance parameters like calibration factors or it may be that we are interested in constraints on only one parameter at a time rather than on the joint constraints on 2 or more parameters simultaneously. Then we marginalize over the uninteresting parameters by

$$P(\theta_1, \dots, \theta_j, H|D) = \int d\theta_{j+1} \dots d\theta_m P(\theta, H|D) \quad (30)$$

where m is the total number of parameters in our model and $\theta_1, \dots, \theta_j$ denote the parameters that we are interested in.

Fisher Matrix

Once we have a set of data it is important to know how to accurately estimate parameters. Fisher [7] proposed a way to solve this issue 70 years ago. In this section we review the main results of his work following the procedure given in [5].

First of all, we consider again a gaussian likelihood. As we can notice, the Hessian matrix $H_{\alpha\beta}$ has information on the parameters' errors and their covariance. More specifically, when all parameters are fixed except one (e.g. the i th parameter), the error on that one parameter is $1/\sqrt{H_{ii}}$. These errors are called conditional errors, although they are rarely used.

A quantity that arises naturally with gaussian likelihoods to forecast the precision of a model is the so-called *Fisher information matrix*

$$F_{ij} = - \left\langle \frac{\partial^2 \mathcal{L}}{\partial \theta_\alpha \partial \theta_\beta} \right\rangle \quad (31)$$

where

$$\mathcal{L} = \ln L \quad (32)$$

It's clear that $F = \langle H \rangle$. The average is made with observational data.

As we can see from eq. (2d), when we have independent data sets, the complete likelihood is the product of the likelihoods, and the fisher matrix for independent data sets is the sum of the individual fisher matrices.

A pedagogical and easy case is the one with one-parameter θ_i considering a gaussian likelihood. In this scenario we have that

$$\Delta\mathcal{L} = \frac{1}{2}F_{ii}(\theta_i - \theta_{0i})^2 \quad (33)$$

when $2\Delta\mathcal{L} = 1$ and by identifying it with the $\Delta\chi^2$ corresponding to 68% confidence level, we notice that $1/\sqrt{F_{ii}}$ yields the $1 - \sigma$ displacement for θ_i . In the general case

$$\sigma_{ij}^2 \geq (F^{-1})_{ij} \quad (34)$$

Thus, when all parameters are estimated simultaneously from the data, the marginalized error is

$$\sigma_{\theta_i} \geq (F^{-1})_{ii}^{1/2} \quad (35)$$

The beauty of the Fisher matrix approach is that there is a simple prescription for setting it up knowing only the model and measurement uncertainties, and that, under the assumption of a gaussian likelihood, the Fisher matrix is the inverse of the covariance matrix. So, all you have to do is set up the Fisher matrix and then invert it to obtain the covariance matrix (that is, the uncertainties on your model parameters). In addition, if it can be computed quickly, it also enables one to explore different experimental set ups and optimize the experiment.

The whole point of the Fisher matrix formalism is to predict how well the experiment will be able to constrain the parameters of the model before doing the experiment and without even simulating the experiment in any detail. We can then forecast the results of different experiments and look at trade-offs such as precision versus cost. In other words, we can engage in experimental design.

The \leq in (34) is called the Kramer-Rao inequality. One can see that the Fisher information matrix represents a lower bound for the errors. Only when the likelihood is normally distributed, the \leq is transformed in $=$. However as we saw in [About the likelihood] a gaussian likelihood is only applicable to some circumstances, being generally impossible to be applied, so the key is to have a good understanding of our theoretical model in such a way that we can construct a gaussian likelihood.

Importance Sampling

We call importance sampling (IS) to the different techniques of determining properties of a distribution by drawing samples from another one. The basic idea of this procedure is to consider that the distribution one draws from should be (for a larger number of samples) representative of the distribution of interest. In such case, we should infer different quantities of it. In this section we review the basic concepts necessary to understand what IS is following [42].

Suppose we are interested in computing the expectation value $\mu_f = E_p[f(X)]$, where $p(x)$ is a probability density of a random variable X and the sub-index p means average over the distribution p . Then, if we consider a new probability density $q(x)$ that satisfies $q(x) > 0$ whenever $f(x)p(x) \neq 0$, we can rewrite the mean value μ_f as

$$\mu_f = \int f(x)p(x)dx = \int f(x)\frac{p(x)}{q(x)}q(x)dx = E_q[f(X)w(x)] \quad (36)$$

where $w(x) = p(x)/q(x)$, and now we have an average over q . So, if we have a collection of different draws $x^{(1)}, \dots, x^{(m)}$ from $q(x)$, we can estimate μ_f using this draws as

$$\hat{\mu}_f = \frac{1}{m} \sum_{j=1}^m w(x^{(j)})f(x^{(j)}) \quad (37)$$

If $p(x)$ is known only up to a normalizing constant, the above expression can be calculated as a ratio estimate

$$\hat{\mu}_f = \frac{\sum_{j=1}^m w(x^{(j)})f(x^{(j)})}{\sum_{j=1}^m w(x^{(j)})}. \quad (38)$$

For the strong law of large numbers, in the limit when $m \rightarrow \infty$ we will have that $\hat{\mu}_f \rightarrow \mu_f$.

In Bayes analysis it can be useful to compute the ratio between evidences for two different models

$$\frac{P'(D)}{P(D)} = E \left[\frac{P'(\theta, D)}{P(\theta, D)} \right]_{P(\theta|D)} \simeq \frac{1}{N} \sum_{n=1}^N \frac{P'(D|\theta_n)P'(\theta_n)}{P(D|\theta_n)P(\theta_n)} \quad (39)$$

where the samples $\{\theta_n\}$ are drawn from $P(\theta|D)$.

An important result for importance sampling is that, if we have a new set of data which is broadly consistent with the current data (in the sense that the posterior only shrinks), we can make use of importance sampling in order to quickly calculate a new posterior including the new data.

Combining datasets: Hyperparameter method

Suppose we are dealing with multiple datasets, $\{D_1, \dots, D_N\}$, coming from a collection of different surveys $\{S_1, \dots, S_N\}$. Of course we can't be sure that, a priori, all our data surveys are consistent with one another, or if there is one or more that are likely to be erroneous. If we were sure that all this datasets are consistent, then it should be enough to update the probability as was seen in [Updating the probability distribution for a model] in order to calculate the new posterior distribution for the parameters that we are interested in. However, because we are usually not sure about this, a way to have any information about how useful is a data survey is by introducing the **hyperparameter method**. This method was first proposed by [39] and [40] in order to perform a joint estimation of cosmological parameters from combined datasets and it can be used as long as we can consider every survey independent from each other.

In this section we review the main steps necessary to understand the hyperparameter method following ref. [40]. If the reader is interested in a more explicit explanation of it, they can consult [39] or [40].

The main feature of this process is the introduction of a new set of "hyperparameters" α in our bayesian procedure in order to avoid extra freedom in the parameter estimation process. These hyperparameters are equivalent to nuisance parameters in such case we need to marginalize over the hyperparameters α in order to recover the posterior distribution, i.e.

$$P(\theta, H|D) = \frac{1}{P(D)} \int P(D|\theta, \alpha, H) P(\theta, \alpha, H) d\alpha \quad (40)$$

where we have used the Bayes theorem. Now, it is necessary for the method to assume that the hyperparameters α and the parameters of interest θ are independent, i.e. $P(\theta, \alpha, H) = P(\alpha)P(\theta, H)$, it is also necessary to assume that each hyperparameter α_k is independent from other hyperparameters, i.e. $P(\alpha) = P(\alpha_1)P(\alpha_2)\dots P(\alpha_N)$. In this way we can rewrite the above expression as

$$P(\theta, H|D) = \frac{P(\theta, H)}{P(D|H)} \left[\prod_{k=1}^N \int P(D_k|\theta, \alpha_k, H) P(\alpha_k) d\alpha_k \right] \quad (41)$$

Here, the quantity inside square brackets is the marginalized likelihood over hyperparameters $L(D; \theta, H)$, and then we can identify the quantity inside the integration as the individual likelihood $L(D_k; \theta, \alpha, H)$, for every α_k and D_k ; $P(D|H)$ is the evidence and, in a typical parameter inference procedure, works as a normalized function, i.e. $P(D|H) = \int P(\theta, H) L(D; \theta, H)$. Notice that, by considering $P(\alpha_k) = \delta(\alpha_k - 1)$, we rely on the standard approach, where no hyperparameters are used.

We add this hyperparameters in order to, in a way, weight every dataset and take away the importance of the data that doesn't seem to be consistent with other ones. Then, how can we know whether the data support the introduction of hyperparameters? Of course, if the datasets are consistent, the introduction of hyperparameters could make it difficult to make our estimation or give rise to large uncertainties. A way to answer this question is given by the bayesian evidence.

Suppose that we have two models, one that considers the introduction of hyperparameters, called H_1 , while the second one doesn't, called H_0 . The bayesian evidence $P(D|H_i)$ is an important quantity if we are interested in making a comparison between two different models. In fact, by defining the ratio between two Bayesian evidences

$$K = \frac{P(D|H_1)}{P(D|H_0)} \quad (42)$$

K value	Strenght of evidence
< 1	Negative (supporting H_0)
1 to 3	Weak
3 to 10	Substantial
10 to 30	Strong
30 to 100	Very Strong
> 100	Decisive

TABLE III: Criteria for the ratio between two Bayesian evidences, taken from [41].

we can estimate if it's necessary to introduce the hyperparameters to our model using the criteria given in the next table:

If we consider a gaussian likelihood and a maximum entropy prior, and assuming that, on average, the hyperparameters' weight are unity, we can rewritte the marginalized likelihood function $L(D; \theta, H_1)$ for model H_1 as

$$P(D; \theta, H_1) = \prod_{k=1}^N \frac{2\Gamma(\frac{n_k}{2} + 1)}{\pi^{n_k/2} |V_k|^{1/2}} (\chi_k^2 + 2)^{-(\frac{n_k}{2} + 1)} \quad (43)$$

obtaining an explicitly functional form for K given by

$$K = \prod_{k=1}^N \frac{2^{n_k/2+1} \Gamma(n_k/2 + 1)}{\chi_k^2 + 2} e^{-\chi_k^2/2}. \quad (44)$$

Here, χ_k is the one given by 28 for every dataset and n_k is the number of points contained in D_k . Notice that, if we have a set of independent samples for H_0 , we can compute an estimate for K with the help of equation (39).

NUMERICAL TOOLS

In a typical scenario it's not possible to compute the posterior distribution analytically. It's important to know the numerical tools at our disposal that can help us in our parameter estimation. Of course we have several candidates that could help us in this task, but in this section we present only the usual one (and the easiest) used in cosmology: the Markov Chain Monte Carlo (MCMC) with the Metropolis Hasting algorithm (MHA). Additionally, in this section we present some useful details that we have to take into account if we want to make efficient our computation (no se si cambiar la palabra “computation”).

MCMC techniques for parameter inference

The purpose of a MCMC algorithm is to build a sequence of points (called “chain”) in a parameter space in order to evaluate the posterior of eq. (4) for the usual case where analytical solutions do not exist or are insufficiently accurate. In this section we review the basic results for this procedure in a simplistic way, but for curious readers it is recommendable to check [9],[10], [11], or [12] only for Markov chains.

A sequence X_1, X_2, \dots of random elements of some set is a *Markov chain* if the conditional distribution of X_{n+1} given X_1, \dots, X_n depends on X_n only. In other words, a Markov chain is a process where we can do predictions of the future based only in the information given at the present. An important property of a Markov chain is that they can be shown to converge to a stationary state were successive elements of the chain are samples from the target distribution, in our case it converges to the posterior $P(\theta, H|D)$. In this way we can estimate all the usual quantities of interest from it (mean, variance, etc). The number of points required to get good estimates in MCMC is said to scale linearly with the number of parameters, so this method becomes much faster than grids as the dimensionality increases.

The target density is approximated by a set of delta functions

$$p(\theta, H|D) \simeq \frac{1}{N} \sum_{i=1}^N \delta(\theta - \theta_i) \quad (45)$$

Then, the posterior mean is calculated as

$$\langle \theta \rangle = \int d\theta \theta P(\theta, H|D) \simeq \frac{1}{N} \sum_{i=1}^N \theta_i \quad (46)$$

where \simeq follows because the samples θ_i are generated from the posterior by construction. Then, we can estimate any integrals (such as the mean, variance, etc) as

$$\langle f(\theta) \rangle \simeq \frac{1}{N} \sum_{i=1}^N f(\theta_i) \quad (47)$$

As was mentioned, in a Markov chain it is necessary to generate a new point θ_{i+1} in our chain from the present point θ_i . However, as it is expected, we need a criteria for accepting (or refusing) this new point depending on if it turns out to be better for our model or not. On the other side, if this new step is worse than the previous one, we may accept it, since it could be the case that, if we only accept steps with better probability, we could be converging into a local maximum in our parameter space and, therefore, not completely mapping it. The simplest algorithm that contains all this information in its methodology is known as the Metropolis-Hastings algorithm and now we will explain it.

Metropolis-Hastings algorithm

In the *Metropolis-Hastings algorithm* (MHA) [13] it is necessary to start from a random initial point θ_0 , with an associated posterior probability $p_0 = p(\theta_0, H|D)$. We need to propose a candidate θ_c by drawing from the *proposal distribution* $q(\theta_0, \theta_c)$. Then, the probability of acceptance of a new point is given by

$$p(\text{acceptance}) = \min \left[1, \frac{p_c q(\theta_c, \theta_0)}{p_0 q(\theta_0, \theta_c)} \right] \quad (48)$$

If the proposal distribution is symmetric the algorithm is reduced to the *Metropolis algorithm*

$$p(\text{acceptance}) = \min \left[1, \frac{p_c}{p_0} \right] \quad (49)$$

In this way the complete algorithm can be expressed by the following steps:

1. Choose a random initial condition θ_0 in parameter space and compute the posterior distribution
2. Generate a new candidate from a proposal distribution in the parameter space and compute the corresponding posterior distribution
3. Accept (or not) the new point with the help of the Metropolis Hasting algorithm
4. If the point is not accepted, repeat the previous point in the chain
5. Repeat steps 2-4 until you have a large enough chain.

A first example in parameter inference using a MCMC technique with a MHA

In order to exemplify the numerical tools learned in this section, let us go back to the coin toss example seen in subsection [Bayes theorem, prior and posterior distributions]. Since it's of our interest that the reader understands what is the basic procedure given in this section, let us try to estimate what is the value of p (or region of values for p) that best matches our data (the 14 times that the coin was thrown). For it we will calculate the posterior distribution (12) using the MHA.

As before, we consider a Likelihood given by a binomial distribution (8) and a normal distributed prior (11) ($a = b = 1$). As our first "guess" for p we take $p_0 = 0.1$. We generate a new candidate p_c as $p_c = p_i + G(p_i, \sigma)$, where $G(p_i, \sigma)$ is our proposed gaussian distribution centered at p_i and with variance $\sigma = 0.1$; p_i is the current value of p , for our first step is $p_i = p_0$. Then, we compute the Metropolis-Hastings algorithm in a Python code as can be seen

in appendix [A]. Our final result, fig. 4, was a posterior distribution that matches very well the posterior that was calculated analytically. Notice that we have plotted our 95% confidence regions (black line).

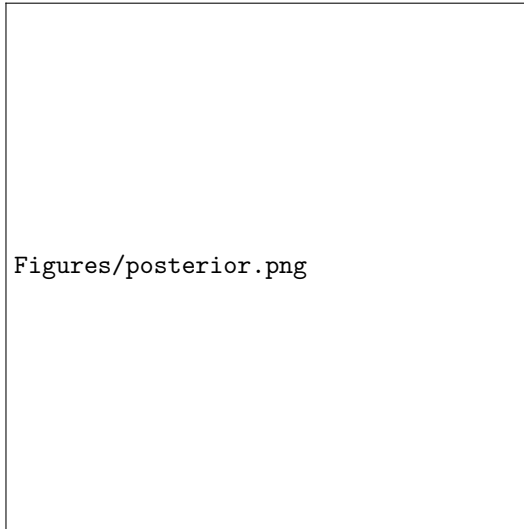


FIG. 4: Posterior distribution for our example. We plot the prior distribution (blue), true posterior (dashed-red) and the posterior calculated by the MHA (red). We plot 95% confidence region for the estimation of p .

To complete the example we show in figure 5 the Markov Chain generated by our code. It's easy to see that the chain oscillates around a middle value. This behaviour is expected due to the fact that we don't have enough data to constrain more accurately the value of p .

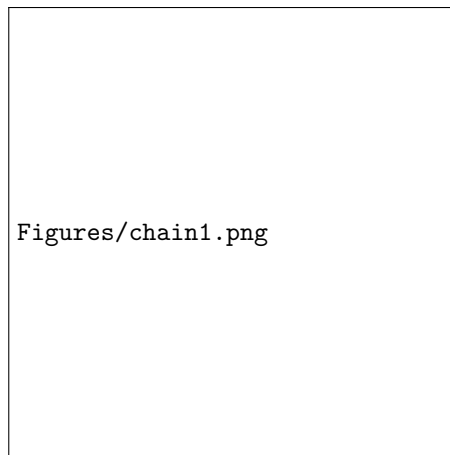


FIG. 5: Markov chain. We use $p_0 = 0.1$ as our first "guess" for p .

Note: In appendix [A] we computed our MCMC algorithm using an explicit code of the MCMC process. However, in Python there are some modules that can help us simplify this task. For example, PyMC is a Python module that implements statistical models and fitting algorithms, including the MCMC algorithm. We use this module at the end of this section. Applying the tools learned in a complete work session.

Convergence test

It is clear that we need a test to know when our chain has converged. However, we need to be warned that the point in our chain is not in a "false convergent point" or a locally maximum point. In this sense, we need that our rule takes into account this possible difficulty. The simplest way (the informal way) to know if our chain is converging is running several chains starting with different proposal initial points for the parameter that we are interested in estimating. Then, if we see, by naked eye, that all the chains seem to converge in a single region of the possible value for our parameter, we may consider that our chains are converging to that region.

Taking yet again the example of the coins, we can run several chains for the above example and try to estimate if the value (the region) of p that we found is an stationary value (region). In figure 6 we plotted 5 different Markov chains with initial “guess” condition $p = 0.2, 0.3, 0.5, 0.7, 0.9$. As we expected from the analytical result, all the chains seem to concentrate near the same value.

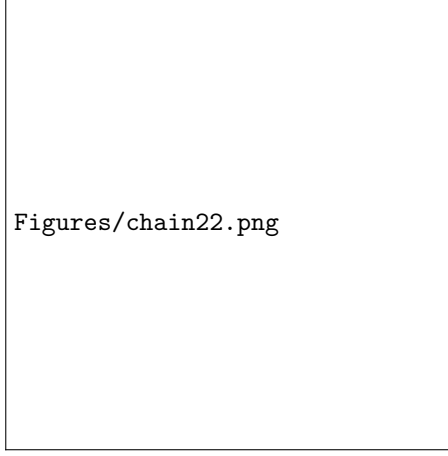


FIG. 6: Multiple MCMC. We calculate 5 Markov chains to estimate convergence of our chains.

The convergence method above is very informal and we would like to have a better way to ensure that our result is correct. The classical test used for this is the *Gelman-Rubin* (1992) convergence criterion. This is (following [14], [3]) by starting with M chains with very different initial points and N points per chain, if θ_i^j is a point in the parameter space of position i and belonging to the chain j , we need to compute the mean of each chain

$$\langle \theta^j \rangle = \frac{1}{N} \sum_{i=1}^N \theta_i^j \quad (50)$$

and the mean of all the chains

$$\langle \theta \rangle = \frac{1}{NM} \sum_{i=1}^N \sum_{j=1}^M \theta_i^j. \quad (51)$$

Then the chain-to-chain variance B is

$$B = \frac{1}{M-1} \sum_{j=1}^M (\langle \theta^j \rangle - \langle \theta \rangle)^2 \quad (52)$$

and the average variance of each chain is

$$W = \frac{1}{M(N-1)} \sum_{i=1}^N \sum_{j=1}^M (\theta_i^j - \langle \theta^j \rangle)^2 \quad (53)$$

If our chains converge, W and B/N must agree. In fact we say that the chain converges when the quantity

$$\hat{R} = \frac{\frac{N-1}{N}W + B(1 + \frac{1}{M})}{W}, \quad (54)$$

which is the ratio of the two estimates, approach to unity. A typical convergence criteria is when $\hat{R} < 1.03$.

Some useful details

About the proposal distribution. The choice of a proposal distribution q is crucial for the efficient exploration of the posterior. In our example we used a Gaussian-like distribution with a variance (step) $\sigma = 0.1$. This value was taken because we explored, by hand, different values for σ and we took the one that looked to approach more quickly to the

real main value of p . If the scale of q is too small compared to the scale of the target (in the sense that the typical jump is small), then the chain may take a very long time to explore the target distribution (left side of the plot) which imply that the algorithm will be very inefficient. As we can see in figure 7 (left side), considering a prior for $p = 0.8$, the number of points that we plot are not enough for the system to move to its “real” posterior distribution. On the other side, if the scale of q is too large, the chain gets stuck and it does not jump very frequently (right side of the figure) which implies that we will have different “picks” in our posterior distribution.

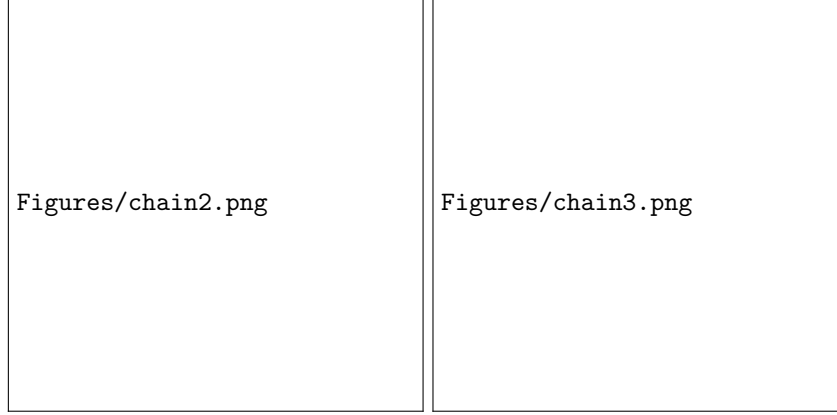


FIG. 7: Two Markov chains considering different variance for our Gaussian proposal distribution. Left side corresponds to $\sigma = 0.003$, while right side corresponds to $\sigma = 0.8$.

In order to fix this issue in a more efficient way, it is recommendable to run an exploratory MCMC, compute the covariance matrix from the samples, and then re-run with this covariance matrix as the covariance of a multivariate Gaussian proposal distribution. Of course, this process can be computed a couple of times before running the “real” MCMC.

About the burn-in. It is important to notice that when we start a chain we will have a region of points outside the stationary region where the chain converges (in our chain we could consider those points as the ones inside the ellipse in figure 5). This early part of the chain (called “burn-in”) must be ignored, this means that the dependence on the starting point must be lost. For it, it is important to have a convergence test which can help us to know when the chain has converged.

Thinning the chains.- There are several bayesian statisticians that usually thin their MCMC, this means that they do not prefer to save every step given by the MCMC; instead, they prefer to save a new step each time n steps have taken place. An obvious consequence that follows by thinning the chains is that the amount of autocorrelation is reduced. However, as long as the chains are thinned, the precision for the estimated parameters is reduced [37]. Thinning the chains can be useful in other kind of circumstances like, for example, if we have limitations in memory. Notice that thinning a chain does not yield incorrect results; it yields correct results but less efficient than using the full chains.

Autocorrelation probes.- A complementary way to look for convergence in a MCMC estimation is by looking for the autocorrelation between the samples. The *lag k* autocorrelation is defined as the correlation between every sample and the sample k steps before. It can be quantified as [36]

$$\rho_k = \frac{Cov(X_t, X_{t+k})}{\sqrt{Var(X_t)Var(X_{t+k})}} = \frac{E[(X_t - X)(X_{t+k} - X)]}{\sqrt{E[(X_t - X)^2]E[(X_{t+k} - X)^2]}} \quad (55)$$

where X_i is the i -th sample and X is the mean of the samples. This autocorrelation should become smaller as long as k increases (this means that samples start to become independent).

Metropolis Coupled Markov Chain Monte Carlo (MC^3)(No estoy seguro si dejarlo o no).- It is easy to see that it could be a little problematic if our likelihoods possess local maxima. The MC^3 is a modification of the standard MCMC algorithm that consists in running several Markov Chains in parallel to explore the target distribution for different “temperatures”, and then simplify the way we sample our parameter space and help us to avoid this local maxima. In this little section we exemplify the basic idea of this algorithm following [38]. If you are interested in

a more extensive explanation of this algorithm, or a modification to make the temperature of the chains dynamical, please consult reference [38].

We consider a tempering version of the posterior distribution $P(\theta, H, T|D)$

$$P(\theta, H, T|D) \propto L(\vec{\theta}, D)^{1/T} P(\theta, H) \quad (56)$$

where L is the likelihood and $P(\theta, H)$ the prior. Notice that, for higher T , individual peaks of L become flatter, making the distribution easier to sample with a MCMC algorithm. Now, we have to run N chains with different temperatures assigned in a ladder $T_1 < T_2 < \dots < T_N$, usually taken with a geometrically distributed division, with $T_1 = 1$. The coldest chain T_1 is the one that samples the posterior distribution more accurately and behaves as a typical MCMC. Then, we define this chain as the main chain. The other chains are running in such a way that they can cross local maximum likelihoods easier and transport this information to our main chain.

The chains explore independently the landscape for a certain number of generations. Then, in a pre-determined interval, the chains are allowed to “swap” its actual position with a probability

$$A_{i,j} = \min \left\{ \left(\frac{L(\theta_i)}{L(\theta_j)} \right)^{1/T_j - 1/T_i}, 1 \right\}. \quad (57)$$

In this way, if a swap is accepted, chains i and j must exchange their current position between them in the parameter space, then chain i has to be on position θ_j and chain j has to move to position θ_i .

We can see that, since the hottest chain T_{max} can access to all the modes of $P(\theta, H, T_{max}|D)$ easier, then it can propagate its position to colder chains, to be precise, it can propagate its position to the coldest chain $T = 1$. At the same time, the position of colder chains can be propagated to hotter chains, allowing them to explore the entire prior volume.

(Monte Carlo Sequential Importance Sampling (MCSIS) ‘¿sería bueno agregarlo?’)


More samples.-The generation of the elements in a Markov chain is probabilistic by construction and it depends on the algorithm that we are working with. The MHA is the easiest algorithm used in bayesian inference. However, there are several other algorithms that can help us fulfilling our mission. For instance, some of the most popular and effective ones, apart of the MHA, are the Gibbs sampling (GB) (see e.g. [16, 17]), the Hamiltonian Monte Carlo (see e.g. [18, 19]) or the Adaptative Metropolis-Hastings (AMH) (see e.g. [20]).

A first complete session work: Adjusting a straight-line

In this section we apply everything we have learned until now to the simplest example: adjusting a straight-line. This is, we assume that we have a certain theory where our measurements should be in a straight line. Then, in order to apply our techniques, we simulate several datasets along a given line. One of the principal topics that we want to analyse is the hyperparameter method and how it works, so we will apply our analysis for two different examples: first, we consider that we have 2 datasets taken from the same straight-line but with different errors; while in the second case we consider that we have two datasets but now we simulate both of them from different straight-lines and different errors. In our analysis we used the PyMC3 module [red] implemented in Python. Our complete code can be seen in ref. [ref]. This code is so simple to use and can be modified very easily if a new model would be tested. We recommend the archive called “new model” where the reader can find a blank project where the data and model can be put and, by running all the notebook, obtain all the analysis that we will see in this section. One can find too in the same archive several notes that will help in programming the model with PyMC3.

Case 1

In this example we start by considering that our measurements for a given theory where a straight-line $y = a + bx$ is the one that is expected to be given by the data shown in figure 8. We call this case *Case 1*. This two datasets, D1 and D2, were generated from the line $y = 3 + 2x$, adding a gaussian error to each data in our datasets. For D1 we add an error with a standard deviation $\sigma_1 = 0.3$, while for D2 we add $\sigma_2 = 0.2$. Then, what we would like to estimate are the parameters of the model, i. e. a and b . We will analyse this data with and without the hyperparameter method and discuss in detail our results.



Figures/data_1.png

FIG. 8: Datasets D_1 and D_2 measured by our straight-line theory.

Case without hyperparameters. Model H_0 .- Before we make a Bayesian estimation, it is necessary to specify our priors. As we have seen, a good prior is a non informative one. Suppose we only know the boundary limits for a and b (we can see them by eye in our data). Then we consider the flat priors

$$a \propto U[0, 5] \quad \text{and} \quad b \propto U[0, 3] \quad (58)$$

where $U[\alpha, \beta]$ are uniform distributions with lower limit α and upper limit β .

If now we consider that there are values for a and b for which our data fits better, then from [eq] we can write our likelihood as

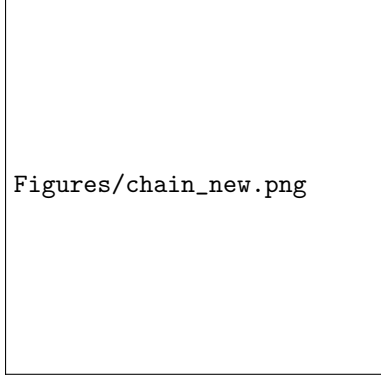
$$L(D; line) \propto \exp \left[- \sum_d \frac{(y_d - y)^2}{2\sigma_d^2} \right] \quad (59)$$

where y_d is our data taken from the dataset $D = D_1 + D_2$ and σ_d its errors.

Now we can generate our MCMC with the MHA. In our analysis we ran 6 chains with 10,000 steps for each one. We ran each chain with a temperature $T = 2$ and we thinned them every 50 steps. We can see our results in table IV and figure 9. Notice that there are some regions where the frequency of events in our sample is increased. So we can say that such parameter regions look to be more likely to match with the data. Additionally we compute the Gelman-Rubin criterion for each variable in order to verify that our results converged. We see from table IV that this number is very similar to 1, so our convergence criterion is fulfilled.

	Mean	Std. Dev.	Gelman-Rubin
a	2.982407	0.047978	1.000200
b	1.994251	0.013490	1.000352

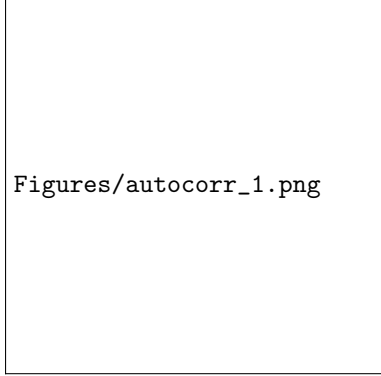
TABLE IV: Our means obtained in the Bayesian estimation for model H_0 . We also calculated the Gelman-Rubin criterion for each parameter.



Figures/chain_new.png

FIG. 9: Results for our sample in the Markov chains for model H_0 .

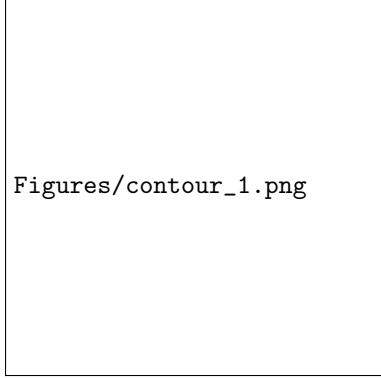
Now we need to continue with the autocorrelation plots. As we mentioned, we need that these plots are small as k increases in order to consider that our analysis is converging. We see in figure 10 such plots and notice that our convergence criteria is fulfilled.



Figures/autocorr_1.png

FIG. 10: Autocorrelation plots for model H_0 .

Finally in figure 11 we show the typical $1 - 4\sigma$ confidence regions. We also plot in red the real value for our parameters. The real value for a and b are inside the curve corresponding to 1 standard deviation of our estimations by our inferential method. Then, in Case 1 we can see that the model H_0 looks to be a very good estimation procedure.



Figures/contour_1.png

FIG. 11: Confidence regions for our parameters for model H_0 .

Case with hyperparameters. Model H_1 . - Now it is time to prove our h. Hyperparameter method. In this case our likelihood can be written from [ref] as

$$L(D; \theta, H_1) = \prod_{k=1}^N \frac{2\Gamma(\frac{n_k}{2} + 1)}{\pi^{n_k/2} |V_k|^{1/2}} \left[\frac{(y_d - y)^2}{2\sigma_k^2} + 2 \right]^{-\left(\frac{n_k}{2} + 1\right)} \quad (60)$$

Now, same as the last procedure, we compute the posterior with our flat priors and using 6 chains with 10,000 steps for each one. Our results and autocorrelation plots can be seen in table V and figure 12. Comparing tables IV and

V we can notice that both procedures look very similar. In fact, the confidence regions for both approximations, fig. 11 and 13, are similar as well. So, what method is better? We could say that the method with hyperparameters is as good as the one without them, but in order to be sure of it we need to compute the ratio K between both models. We obtained from [eq]

$$K = 3 \tag{61}$$

Then, comparing with table [ref] we can say that the evidence for H_1 to be better than H_0 is weak. In such a case it should better to work with H_0 as we explained before.

	Mean	Std. Dev.	Gelman-Rubin
a	2.974059	0.038583	1.000229
b	1.995189	0.010611	1.000044

TABLE V: Our means obtained in the Bayesian estimation for model H_1 . We also calculated the Gelman-Rubin criterion for each parameter.

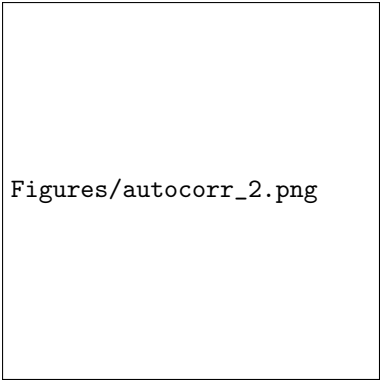


FIG. 12: Autocorrelation plots for model H_1 .

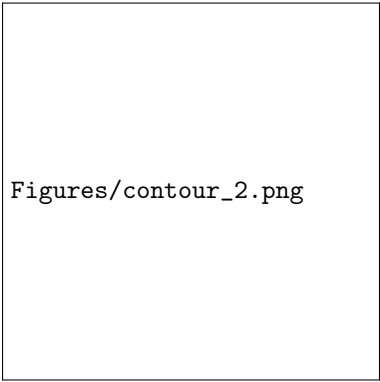
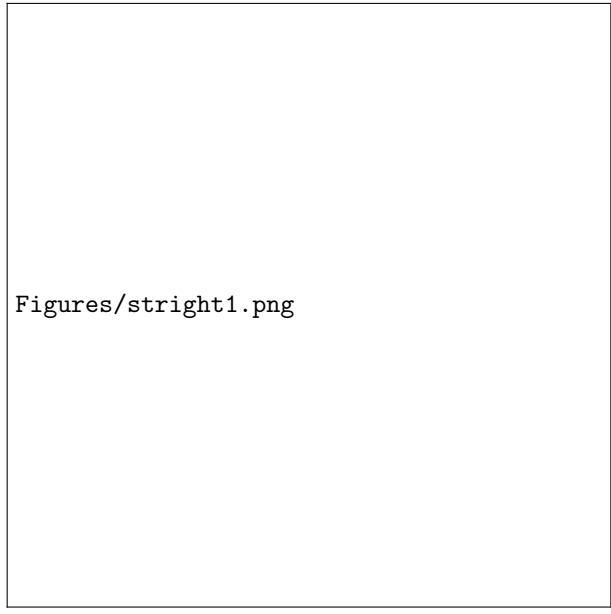


FIG. 13: Confidence regions for the parameters in model H_1 .

Finally, in order to exemplify our results, let us plot in figure 14 our data with the straight-line inferred by the mean parameters of both models. As we expected our estimation fits well the data for both cases.




Figures/stright1.png

FIG. 14: Our stright-lines inferred by our procedures confronted with the data.

Case 2

Now we consider that we have a new set of data and the same theory for the straight-line, but suppose that our measurements is kind of distinct. Suppose that we measure the data given in figure 15, this data corresponds to considering our dataset D_1 and changing D_2 by 16 new points generated around the line $y = 3.5 + 1.5x$ with a Gaussian noise and standard deviation $\sigma = 0.5$. So, our datasets are not auto-consistent between them. Let us make again a parameter estimation for our parameters and look for the differences in both procedures.



Figures/data_2.png

FIG. 15: Datasets D_1 and D_2 measured by our straight-line theory.

Case without hyperparameters. Model H_0 .- We followed the same procedure as in Case 1. We computed our posterior and verified that our results converged with the help of the Gelman-Rubin criterion and the autocorrelation plots. Our results can be seen in table VI. Then we plotted our $1 - 4\sigma$ confidence regions in figure 16. It's easy to see that our estimation differs so much from the real parameters in our datasets. Of course this is because we are trying to fit a model with non auto-consistent datasets and then we arrive to incorrect results. Now, let us see what

happens in the hyperparameters procedure.

	Mean	Std. Dev	Gelman-Rubin
a	3.528359	0.056511	1.000153
b	1.795464	0.014116	1.000393

TABLE VI: Means obtained in the Bayesian estimation for model H_0 . We also calculated the Gelman-Rubin criterion for each parameter.

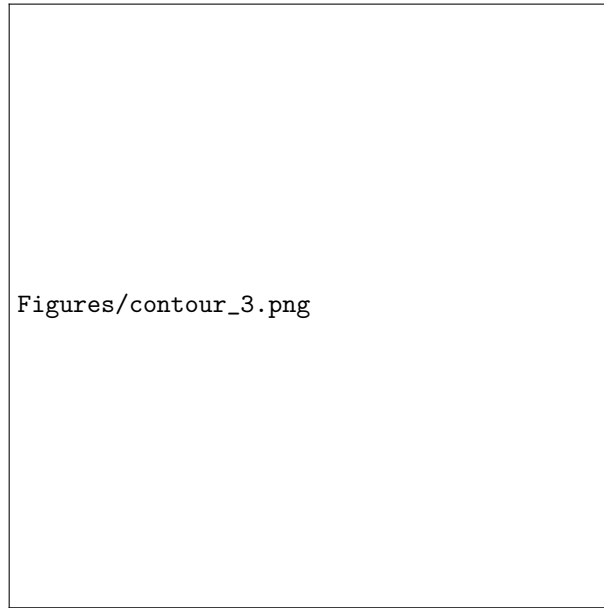


FIG. 16: Confidence regions for the parameters in model H_0 .

Case with hyperparameters. Model H_1 .- We can see in table VII the results of our estimation. In figure 17 we plotted our posterior. What we can see immediately is that both approximations are very different in our posterior. While for model H_0 we obtained a single region far away of the real values of our data, for model H_1 we obtained two locally maximum regions near the real values for our datasets.

	Mean	Std. Dev.	Gelman-Rubin
a	3.528359	0.056511	1.000153
b	1.795464	0.014116	1.000393

TABLE VII: Means obtained in the Bayesian estimation for model H_1 . We also calculated the Gelman-Rubin criterion for each parameter.

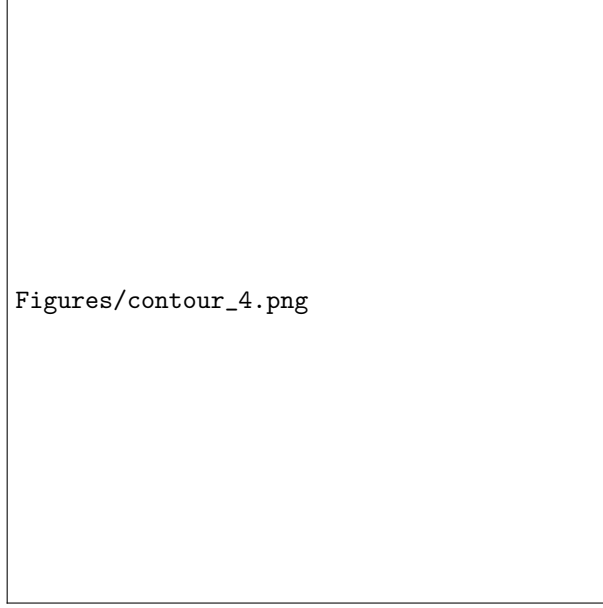


FIG. 17: Confidence regions for the parameters in model H_1 .

Finally, we only need to say which method is better. Given the fact that we know *a priori* the real values of our parameters for this example, we could immediately say that the method with hyperparameters is a better approximation than the case without them. However, we confirm this assumption by calculating the ratio K between both models. We obtain

$$K = 37 \quad (62)$$

which means that we have a very strong evidence that H_1 is better than H_0 .

BAYESIAN STATISTIC AND COSMOLOGY

In this section we will present the necessary topics of Cosmology in order to understand the application of Bayesian Statistics. We will use natural units.

The metric.

To study the Universe we consider that it is homogeneous and isotropic at large scales. This is known as **Cosmological Principle**. Isotropic means that the galaxies distribution doesn't depend on the direction, and homogeneous means that it is independent of the position. In addition, we will use the formalism of General Relativity. It studies the interaction between the geometry and matter contained in the space-time. The curvature of the space-time produces physical effects on the matter it contains, these effects are associated with a gravitational field. On the other hand, the curvature is related to the matter contained by an energy-momentum tensor. The above can be summarized by saying that matter tells space-time how to curve and, in turn, geometry tells matter how to move. We can write the above in the Einstein equation:

$$G_{\mu\nu} = 8\pi G T_{\mu\nu}. \quad (63)$$

Where $G_{\mu\nu}$ is the Einstein tensor (geometry of the space-time), $T_{\mu\nu}$ is the energy-momentum tensor (matter contained in the Universe) and G is the gravitational constant [?], [?].

The distance between two points in a curved space-time can be measured using:

$$ds^2 = g_{\mu\nu} dx^\mu dx^\nu. \quad (64)$$

Where $g_{\mu\nu}$ is the metric tensor which contains all the information about the structure of the space-time. The values that the indices μ and ν can take depend on the dimensions of the space-time.

To study the Universe we use the Friedmann-Lemaître-Robertson-Walker metric (**FLRW**). This describes a homogeneous, isotropic and expanding Universe.

$$ds^2 = dt^2 - a^2(t)\gamma_{ij}dx^i dx^j, \quad (65)$$

where

$$\gamma_{ij} \equiv \delta_{ij} + \kappa \frac{x_i x_j}{1 - \kappa (x_k x^k)}. \quad (66)$$

In equation (66), κ describes the curvature. In (65) a is the scale factor which depends on time. By convention we set $a(t_0) \equiv 1$ today [?].

Friedmann and continuity equations.

With the Cosmological Principle, the energy-momentum tensor describes a perfect fluid [?]

$$T_{\mu\nu} = (\rho + P)U_\mu U_\nu - P g_{\mu\nu}. \quad (67)$$

Where ρ is the energy density, P is the fluid pressure and U^μ is the 4-velocity relative to the observer. Using equations (63) and (67) we can deduce Friedmann and continuity equations. This last equation is given by

$$\dot{\rho} + 3\frac{\dot{a}}{a}(\rho + P) = 0. \quad (68)$$

Equation (68) implies energy conservation. On the other hand, Friedmann equations describe the expansion of the Universe. These are

$$H^2 = \left(\frac{\dot{a}}{a}\right)^2 = \frac{8\pi G}{3} \sum_i \rho_i - \frac{k}{a^2}, \quad (69)$$

$$\frac{\ddot{a}}{a} = -\frac{4\pi G}{3} \sum_i (\rho_i + 3P_i). \quad (70)$$

Where $H \equiv \frac{\dot{a}}{a}$ is the Hubble parameter. Using the dimensionless parameters ³

$$\Omega_i \equiv \frac{\rho_i}{\rho_{crit}}, \quad (71)$$

$$\rho_{crit} = \frac{3H_0^2}{8\pi G} \quad (72)$$

we can rewrite (69) as

$$\frac{H^2}{H_0^2} = \Omega_r a^{-4} + \Omega_m a^{-3} + \Omega_k a^{-2} + \Omega_\Lambda. \quad (73)$$

Where Ω_r is the radiation dimensionless density parameter, Ω_m corresponds to the matter, Ω_k with curvature and Ω_Λ corresponds to Cosmological Constant. H_0 is the Hubble parameter value today.

³ ρ_{crit} is the condition to have a flat Universe.

Content of the Universe.

- **Matter:** It has no pressure and its energy density takes the form $\rho \propto a^{-3}$. Matter can be baryons (ordinary matter) or dark matter. This dark matter is proposed to explain a lot of observations like the dynamics of the galaxies in Coma cluster or the rotation curves of galaxies. This kind of matter only interacts gravitationally with the rest of the Universe.
- **Radiation:** It is everything that meets the relation $P = \frac{1}{3}\rho$. This implies a density of the form $\rho \propto a^{-4}$. We consider photons and neutrinos as radiation.
- **Dark Energy:** We don't know what it is but we propose it in order to explain the accelerating expansion of the Universe. Dark energy can be vacuum energy or cosmological constant.

We can describe the above using the following state equation

$$\omega = \frac{P}{\rho}. \quad (74)$$

(74) is called barotropic equation because the density only depends on pressure.

Component	ω
Matter	0
Radiation	$\frac{1}{3}$
Cosmological constant	-1

Cosmological observables and parameters.

To explain the Universe we use the dimensionless density parameter of each component (Ω_i), the Hubble constant H_0 and the optical depth to scattering τ . Although there are more parameters that can be used, these can be derived from the previous ones.

In the beginning of this section we mentioned that the FLRW metric describes a Universe with some curvature but the cosmological observations indicate a flat Universe. So the curvature density parameter is zero $\Omega_k = 0$.

- **Hubble Constant:** It is the slope between the recessional velocity and the proper distance from the galaxy to the observe.

A first example in parameter inference for Cosmology

The simplest way to understand how all these concepts can be useful in cosmology is applying them to an example. We consider the typical example in cosmology for parameter inference, which is, we estimate the value of the Hubble parameter H_0 at our present time and the density of matter in the Universe Ω_m , considering a Λ CDM cosmology. In this section we present the results of a complete work session. We wrote our own Python code using the PyMC Python's module [ref]. For interested readers, the code can be seen in appendix [B]. Notice that this can be used not only in cosmology but also in whatever model that you most prefer; what you need to do in order to prove a new model is only specify it in "pm.model()".

What is the model and the theory?

The standard Λ CDM cosmology considers a flat Universe which contains around $\sim 31\%$ of ordinary matter plus dark matter and $\sim 68\%$ of dark energy. In this Λ CDM model it is consider that the matter component of the universe

follows an equation of state given by $p = 0$, while the dark energy is a cosmological constant, i.e. $p = -\rho$. Considering this components, the Universe's dynamics would be given by the Friedmann equation

$$H^2 \equiv \left(\frac{\dot{a}}{a}\right)^2 = H_0^2[\Omega_m(1+z)^3 + \Omega_\Lambda] \quad (75)$$

and the acceleration equation⁴. Here H is the Hubble parameter, H_0 corresponds with the Hubble parameter at present, Ω_m and Ω_Λ is the matter and dark energy densities at our epoch and follows the constraining condition $\Omega_m + \Omega_\Lambda = 1$, and z is the redshift which is associated with a time parameter; $z = 0$ is at present. Notice that thanks to the constraining condition we can rewrite $\Omega_\Lambda = 1 - \Omega_m$ and then we can reduce by one the number of parameters in the model.

The observables and the data

It is possible to measure $H(z)$ by using what it is known as the *cosmic chronometer* approach [34]. We use the data reported in [35] as our data for our estimation. A plot of them can be seen in figure 18.



FIG. 18: Multiple MCMC. We calculate 5 Markov chains to estimate convergence of our chains.

Inferring the free parameters of the model

Now, giving a model and a set of data, we are ready to apply what we have learned until now. First of all, notice that the only free parameters in our model are Ω_m (or Ω_Λ) and H_0 . We suppose that we don't know anything about our free parameters, in such case a good prior for them is a Uniform distribution. However, in order to simplify our life we consider that we know something about the limit values for both parameters, say: $\Omega_m \in [0.1, 1]$ and $H_0 \in [10, 100]$. In this way we have as our priors

$$\Omega_m \sim U[0.1, 1] \quad (76a)$$

$$H_0 \sim U[10, 100] \quad (76b)$$

⁴ We do not show this equation because it is not necessary for our estimation and we do not want to confuse the reader.

What is next?

Una vez de que ya entedimos los procedimientos para la inferencia de parámetros, hablar de que el siguiente paso es saber que hay varios códigos que ya hacen todo lo que vimos (tanto la cosmología, como la estadística) y que es mejor aprender a moverles que andar haciendo nuestro propio cdigo (quizás)

Statistical codes

Once our cosmological model is established we need a statistical code which can help us to estimate the free parameters of our model. A first idea could be continuing programing our own MCMC code, but, as it is expected, while the number of free parameters of our model increases, it is more challenging to construct an efficient code. Fortunately there are several MCMC codes free to download on-line that can make this homework taking as our theory the cosmological codes showed above. In this section we review the most common of them.

Monte Python.- Monte Python is a Monte Carlo code for Cosmological Parameter extraction that can be downloaded in [30]. It contains likelihood codes of most recent experiments, and interfaces with the Boltzmann code Class for computing the cosmological observables.

The code contains several sampling methods available: Metropolis-Hastings, Nested Sampling (through MultiNest), EMCEE (through CosmoHammer) and Importance Sampling. If you are interested to work with this parameter inference code you can get help in [29] and [31].

CosmoMC.- CosmoMC (to download [32]) is a fortran 2003 MCMC engine for exploring cosmological parameter space. It contains Monte Carlo samples and inportance sampling. It containg likelihoods of most recent experiments, and interfaces with CAMB.

SimpleMC.- SimpleMC is a MCMC code for cosmological parameter estimation where only expansion history matters. It was written by Ane Slosar and Jose Vazquez and can be downloaded on [33]

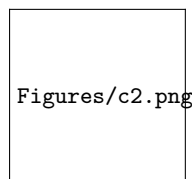
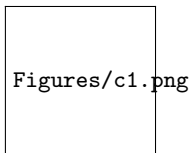
Some examples

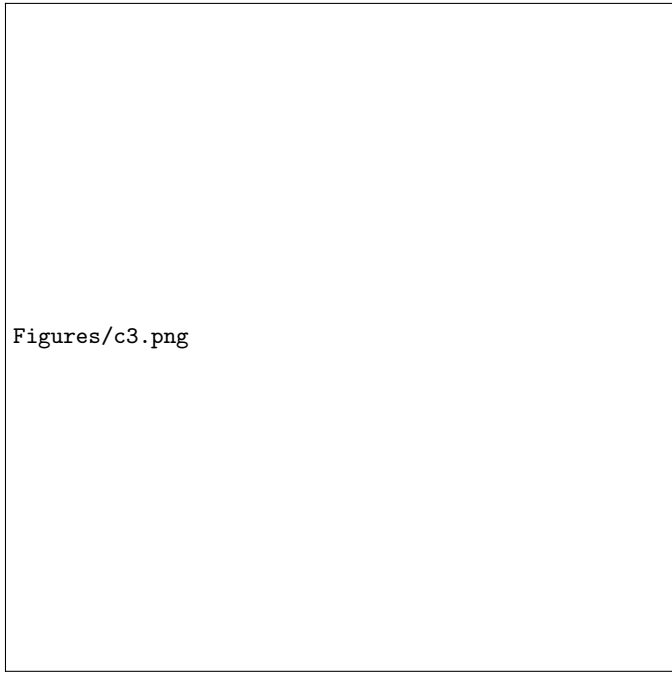
Examples of cosmology

Conclusions

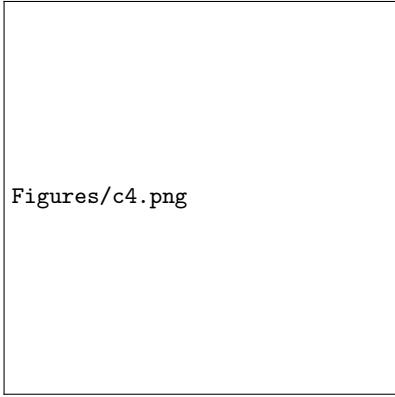
A. A simple MCMC python code

Here we show our MCMC written in Python. This code is very simple and its propose is to help the reader to understand how to programing a MCMC code. However, if you are interested in more sophisticated algorithms you can see the PyMC module of python [ref]. We wrote our code using the jupyter notebook [ref] which is a excelent editor when we have a program no much extense.





Figures/c3.png



Figures/c4.png

* epadilla@fis.cinvestav.mx

† vetovazquez@hotmail.com

‡ ltellez@fis.cinvestav.mx

- [1] The Shapley - Curtis Debate in 1920, [https : //apod.nasa.gov/diamondjubilee/debate1920.html](https://apod.nasa.gov/diamondjubilee/debate1920.html), visited on December 2017
- [2] B.J.K. Kleijin; Bayesian statistic, lecture notes 2015
- [3] Alan Heavens; Statistical techniques in cosmology; May 2010
- [4] Roberto Trotta; Bayes in the sky: Bayesian inference and model selection in cosmology; March, 2008
- [5] Licia Verde; Statistical methods in cosmology; Nov, 2009
- [6] Roberto Trotta; Bayes Methods in Cosmology; Jan, 2017
- [7] Fisher R.A. (1935) *J. Roy. Stat. Soc.* **98**, 39
- [8] Numerical Recipes
- [9] anner, M. (1993) Tools for Statistical Inference, Method for Exploration of Posterior Distributions and Likelihood Functions.
- [10] Gilks, W., Richardson, S. and Spiegelhalter, D. (1996) Markov Chain Monte Carlo in Practice.
- [11] Gelman, A., Carlin, J., Stern, H and Rubin, D. (1995) Bayesian Data Analysis.
- [12] oss, Sheldon, (1989) Introduction to Probability models 4th Edit.
- [13] Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., Teller, E.: Equation of state calculations by fast computing machines. *J. Chem. Phys.* 21 , 10871092 (1953)

- [14] Verde L., *astro-ph/0712.3028* (2007)- <http://www.behind-the-enemy-lines.com/2008/01/are-you-bayesian-or-frequentist-or.html>
- [15] 218.163.109.230 et al. (20042014); *Observational cosmology - 30h course*.
- [16] A. Smith and G. Roberts, *J. R. Statist. Soc. B* 55 323 (1993).
- [17] Ilker Yildirim, Bayesian Inference: Gibbs Sampling, August 2012
- [18] K. M. Hanson, Markov Chain Monte Carlo posterior sampling with the Hamiltonian method, in M. Sonka and K. M. Hanson eds, *Medical Imaging: Image Processing Vol. 4322*, Proc. SPIE, pp. 456467.
- [19] Radford M. Neal, MCMC using Hamiltonian dynamics, arXiv:1206.1901v1 [stat.CO]
- [20] Surya T. Tokdar and Robert E. Kass, Importance Sampling: A review, DOI: 10.1002/wics.56
- [21] K. N. Abazajian, K. Arnold, J. Austerlmann, B. A. Ben-son, C. Bischoff, J. Bock, J. R. Bond, J. Borrill, E. Calabrese, J. E. Carlstrom, et al., *ArXiv e-prints* (2013), 1309.5383.
- [22] https://lambda.gsfc.nasa.gov/toolbox/tb_cmbfast_v.cfm
- [23] <http://adsabs.harvard.edu/abs/2010ascl.soft07004D>
- [24] <http://camb.info/>
- [25] <http://camb.info/readme.html>
- [26] <http://cosmocoffee.info/viewforum.php?f=11>
- [27] <http://cosmologist.info/notes/CAMB.pdf>
- [28] <http://class-code.net/>
- [29] Zumalacarregui Miguel; *CLASS, hi class and Monte Python basics IFT School on Cosmology Tools*; March 11, 17.
- [30] <http://baudren.github.io/montepython.html>
- [31] Benjamin Audren, *Monte Python documentation, Release 2.2.0*; October 21, 2015
- [32] <http://cosmologist.info/cosmomc/>
- [33] <https://github.com/ja-vazquez/SimpleMC>
- [34] R. Jimenez and A. Loeb, Constraining Cosmological Parameters Based on Relative Galaxy Ages, *ApJ*, vol. 573, pp. 3742, July 2002.
- [35] Michele Moresco, Raul Jimenez, Licia Verdec, Andrea Cimatti, Lucia Pozzetti, Claudia Maraston, *Constraining the time evolution of dark energy, curvature and neutrino properties with cosmic chronometers*, *Journal of Cosmology and Astroparticle Physics*, (2016), arXiv:1604.00183v1 [astro-ph.CO]
- [36] Model checking diagnostic, PyMC 2.3.6 documentation, link: <https://pymc-devs.github.io/pymc/modelchecking.html>
- [37] William A. Link and Mitchell J. Eaton, On thinning of chains in MCMC, *Methods in ecology and evolution*,
- [38] W. D. Voudsen, W. M. Farr and I. Mandel, *Dynamic temperature selection for parallel-tempering in Markov chain Monte Carlo simulations*, *MNRAS* (January 11, 2016) Vol. 455 1919-1937, arXiv:1501.05823 [astro-ph.IM]
- [39] Lahav O., Bridle S. L., Hobson M. P., Lasenby A. N., & Sodre L., 2000, *MN-RAS*, 315, 45
- [40] Hobson M. P., Bridle S. L., & Lahav O., 2002, *MNRAS*, 335, 377 (HBL)
- [41] Jeffreys H., *The Theory of Probability*, Oxford University Press, 1961.
- [42] Surya T Tokdar and Robert E Kass, Importance sampling: A Review, DOI: 10.1002/wics.56