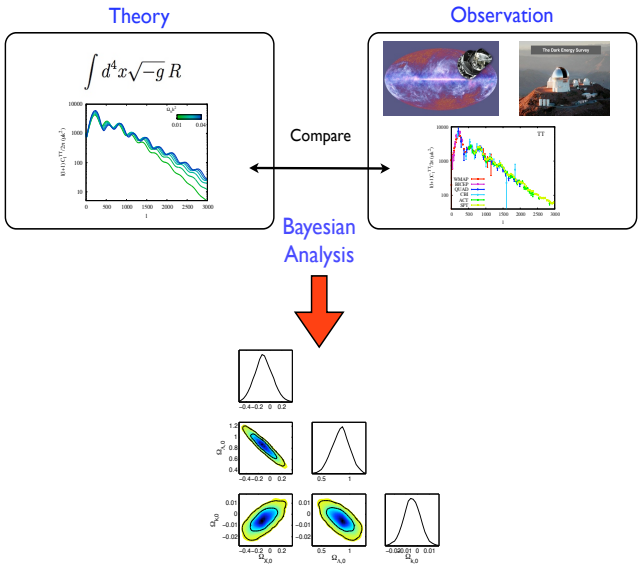
# Updated Cosmology

## with Python




**José-Alberto Vázquez**

ICF-UNAM / Kavli-Cambridge

In progress

August 12, 2017

# 1

# Bayesian Statistics

Bayesian statistics and MCMC (Markov Chain Monte Carlo) algorithms have found their place in the field of Cosmology. They have become important mathematical and numerical tools, especially in parameter estimation and model comparison. In this paper we review some of the fundamental concepts to understand Bayesian statistics, to then introduce the MCMC algorithms and samplers that allow us to perform the parameter inference procedure. We also provide a general description of the standard cosmological model, known as the ΛCDM model, along with several alternatives to it; and current datasets coming from astrophysical and cosmological observations. Finally, with the tools acquired we use a MCMC algorithm implemented in python -called SimpleMC- to test the cosmological models and find out the combination of parameters that best describes the universe.

## 1.1 Introduction

The beginning of the standard cosmology as it is known today emerged after 1920 when the Shapley-Curtis debate was carried out [**?** ]. This debate was held between the astronomers Harlow Shapley and Heber Curtis, resulting in a revolution for astronomy at that time by reaching an important conclusion: "The universe had a larger scale than the Milky Way". Several observations at that epoch established that the size and dynamics of the cosmos could be explained by Einstein's General Theory of Relativity. In its childhood, cosmology was a speculative science based only on a few data sets, and it was 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 increased enough to discriminate and rule out compelling

theories, being the BB model awarded as the most accepted theory. During the same decade, David Schramm heralded the "Golden Age of Cosmology" at a National Academy of Sciences colloquium [**?** ].

Once the new age of cosmological observations arrived with a large variety of data, it was necessary to confront the cosmological models with such data. This was usually done through statistics. It is important to notice that, since we have only one universe, we cannot rely on a frequentist interpretation of statistics (we are not able to create multiple universes and make a frequentist inference of our models). An alternative interpretation to help us in our task is Bayesian statistics. In Bayesian statistics the probability is interpreted as a "degree of belief" and it may be useful when repetitive processes are complicated to reproduce.

The main aim of this work is to provide an introduction of Bayesian parameter inference and its applications to cosmology. We assume the reader is familiarized with the basic concepts of statistics, but not necessarily with Bayesian statistics. Then, we provide a general introduction to this subject, enough to work out some examples. This review is written in a generic way so the reader interested on the parameter inference may apply the theory to any subject, in particular we put into practice the Bayesian concepts on the cosmology branch of physics.

The paper is organized as follows. We start in Section 1.2 by pointing out the main differences between the Bayesian and Frequentist approaches to statistics. Then, in Section 1.3 we explain the basic and necessary mathematical concepts in Bayesian statistics to perform the parameter estimation procedure for a given model. Once we have the mathematical background, we continue in Section 1.4 with some of the numerical resources available to simplify our task, such numerical tools may become important given the fact that, in general, it is not possible to derive analytical results, specially when a model contains several parameters that need to be confronted with data. We then provide an example of some of these methods and tools applied to the simple problem of fitting a straight line in Section 1.5. In Section **??** we present an introduction to cosmology and applications of the tools given in previous sections in cosmology, then in Section **??** the focus is in some of the codes available to perform this work. Then, in Section **??**, we apply these techniques to constrain the parameter space that describes the standard cosmological model, namely the ΛCDM model, and several alternatives to it. Finally, in Section **??** we present our conclusions.

## 1.2 Bayesian vs Frequentist statistics

Fundamentally, the main difference between Bayesian and Frequentist statistics is on the definition of probability. From a Frequentist point of view, probability has meaning in limiting cases of repeated measurements

$$P = \frac{n}{N}, \tag{1.1}$$

where $n$ denotes the number of successes and $N$ the total number of trials. Frequentist statistics defines probability as the limit for the number of independent trials going to infinity. 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**.

Here we introduce some key concepts to understand the consequences this difference entails; for an extended review see [**? ? ? ? ?** ]. Let $x$ be a random variable related to a particular event and $P(x)$ its corresponding probability distribution, for both cases the same rules of probabilities apply[1]:

$$P(x) \geq 0, \tag{1.2a}$$

$$\int_{-\infty}^{\infty} dx P(x) = 1. \tag{1.2b}$$

For *mutually exclusive* events we have

$$P(x_1 \cup x_2) = P(x_1) + P(x_2), \tag{1.2c}$$

but in general

$$P(x_1 \cup x_2) = P(x_1) + P(x_2) - P(x_1 \cap x_2).$$

These rules are summed up as follow. The first condition (1.2a) is necessary due to the probability of having an event is always positive. The second rule (1.2b) is a normalized relation, which tells us that we are certain to obtain one of the possible outcomes. Now, in the third point (1.2c) we have that the probability of obtaining an observation, from a set of mutually exclusive events, is given by the individual probabilities of each event. Finally, and in general,

[1] These rules are defined for a continuous variable; however, the corresponding discrete definition can be given immediately by replacing $\int dx \to \sum$.

if one event occurs *given* the occurrence of another then the probability that both $x_1$ and $x_2$ happen is equal to the probability of $x_1$ times the probability of $x_2$ given that $x_1$ has already happened

$$P(x_1 \cap x_2) = P(x_1)P(x_2|x_1). \tag{1.2d}$$

If two events $x_1$ and $x_2$ are mutually exclusive then

$$P(x_1 \cap x_2) = 0 = P(x_2 \cap x_1). \tag{1.3}$$

The rules of probability distributions must be fulfilled by both Frequentist and Bayesian statistics. However, there are some consequences derived by the fact these two scenarios have a different definition of probability, as we shall see.

### 1.2.1 Frequentist statistics

Any frequentist inferential procedure relies on three basic ingredients: the data, the 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 $P_0 : \hat{B} \to [0, 1]$ in the sample space $(x, \hat{B})$ representing the "true distribution of the data"

$$X \sim P_0.$$

Now there is the model. For Frequentist statistics the **model** $Q$ is a collection of probability measurements $P_\theta : \hat{B} \to [0, 1]$ in the sample space $(x, \hat{B})$. The distributions $P_\theta$ are called *model distributions*, with $\theta$ as the model parameters; in this approach $\theta$ 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 $\hat{P} : x \to Q$, representing our "best guess" $\hat{P} \in Q$ for $P_0$ based on the data $X$.

| 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 1.1:** Main differences between the Bayesian and Frequentist interpretations.

Hence, the Frequentist statistics is based on trying to answer the following questions: "what the data is trying to tell us about $P_0$?" or "considering the data, what can we say about the mean value of $P_0$?".

### 1.2.2 Bayesian statistics

In Bayesian statistics, data and model are two elements of the same space [**?** ], i.e. no formal distinction is made between measured quantities $X$ and parameters $\theta$. One may envisage the process of generating a measurement's outcome $Y = y$ as two draws, one draw for $\Theta$ (where $\Theta$ is a model with associated probabilities to the parameter $\theta$) to select a value of $\theta$ and a subsequent draw for $P_\theta$ to arrive at $X = x$. This perspective may seem rather absurd in view of the definitions for a Frequentist way of thinking, but in Bayesian statistics where probabilities are related to our own knowledge, it results natural to associate probability distributions to our parameters. In this way an element $P_\theta$ of the model is interpreted simply as the distribution of $X$ given the parameter value $\theta$, i.e. as the conditional distribution $X|\theta$.

### 1.2.3 Comparing both descriptions

Table 1.1 provides a short summary of the most important differences between the two statistics. To understand these differences let us review a typical example. Here we present an experiment and, since we are interested in comparing both descriptions, we show only the basic results from both points of view: Frequentist and Bayesian.

*Example.-* Let us assume we have a coin that has a probability $p$ to land as heads and a probability $1 - p$ to land as tails. In the process of trying to estimate $p$ (which must be $p = 0.5$

since we have only two possible states) we flip the coin 14 times, obtaining heads in 10 of the trials. 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 approach.* As mentioned previously, in Frequentist statistics probability is related to the frequency of events, then our best estimate for $p$ is $P(head) = p = \frac{\#\ of\ heads}{\#\ of\ events} = 10/14$. So, the probability of having 2 heads in a row is $P(2heads) = P(head)P(head) \simeq 0.51$.

- *Bayesian approach.* In Bayesian statistics $p$ is not a value, it is a random variable with its own distribution, and it must be defined by the existing evidence. In this example a good distribution for $p$ is a binomial distribution

$$P(D|p) = \binom{14}{10} p^{10}(1-p)^4, \tag{1.4}$$

  where $D$ is our data set (14 trials and 10 successes). Then, by considering a non-informative prior (beforehand we do not know anything about $p$) and averaging over all possible values of $p$ we have that the probability of having two heads is

$$P(2heads|D) = \frac{B(13,5)}{B(11,5)} = 0.485, \tag{1.5}$$

  where $B(x,y)$ is the beta function. This Bayesian example will be expanded in detail during the following section, but for now we just want to stress out that both approximations arrive at different results.

In the Frequentist approach, since we adopt the probability as a frequency of events (the probability of having a head was fixed by $p = 10/14$), hence the final result was obtained by only multiplying each of these probabilities (since we assume the events are independent of each other). On the other hand, in the Bayesian framework it was necessary to average over all possible values of $p$ in order to obtain a numerical value. 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 unfamiliar with Bayesian statistics, do not be scared of the last example. In the next section we review the basic concepts and get back to this example to use the new tools learned.

## 1.3 A first look at Bayesian statistics

Before we start with the applications of Bayesian statistics in cosmology it is necessary to understand the most important mathematical tools in the Bayesian procedure. In this section, we present an informal revision but encourage the reader to look for the formal treatment in the literature, cited in each section.

### 1.3.1 Bayes theorem, priors, posteriors and all that stuff

When anyone is interested on the Bayesian framework, there are several concepts to understand before presenting the results. 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 shown in Eqs. (1.2). From Eqn. (1.2d), without loss of generality, it must be fulfilled that $P(x_1 \cap x_2) = P(x_2 \cap x_1)$. In such case the following relation applies

$$P(x_2|x_1) = \frac{P(x_1|x_2)P(x_2)}{P(x_1)}. \tag{1.6}$$

As already mentioned, in the Bayesian framework data and model are part of the same space. Given a model (or hypothesis) $H$, considering $x_1 \to D$ as a set of data, and $x_2 \to \theta$ as the parameter vector of said hypothesis, we can rewrite the above equation as

$$P(\theta|D, H) = \frac{P(D|\theta, H)P(\theta|H)}{P(D|H)}. \tag{1.7}$$

This last relation is the so-called **Bayes theorem** and the most important tool in a Bayesian inference procedure. In this result, $P(\theta|D, H)$ is called the **posterior** probability of the model. $P(D|\theta, H) \equiv L(D|\theta, H)$ is called the **likelihood** and it will be our main focus in future sections. $P(\theta|H) \equiv \pi(\theta)$ is called the **prior** and expresses the knowledge about the model before acquiring the data. This prior can be fixed depending on either previous experiment results or the theory behind. $P(D|H) \equiv \mathcal{Z}$ is the evidence of the model, usually referred as the **Bayesian Evidence**. We notice that this evidence acts only as a normalizing factor, and is nothing more than the average of the likelihood over the prior

$$P(D|H) = \int d^N\theta P(D|\theta, H)P(\theta|H), \tag{1.8}$$

where $N$ is the dimensionality of the parameter space. This quantity is usually ignored, for practical reasons, when testing the parameter space of a unique model. Nevertheless, the Bayesian

| $\lvert\mathcal{B}_{0,1}\rvert$ | Odds | Probability | Strength |
|---|---|---|---|
| $< 1.0$ | $< 3:1$ | $< 0.750$ | Inconclusive |
| 1.0-2.5 | $\sim 12:1$ | 0.923 | Significant |
| 2.5-5.0 | $\sim 150:1$ | 0.993 | Strong |
| $> 5.0$ | $> 150:1$ | $> 0.993$ | Decisive |

**Table 1.2:** Jeffreys guideline scale for evaluating the strength of evidence when two models are compared.

evidence plays an important role for selecting the model that best "describes" the data, known as *model selection*. For convenience, the ratio of two evidences

$$K \equiv \frac{P(D|H_0)}{P(D|H_1)} = \frac{\int d^{N_0}\theta_0 \; P(D|\theta_0, H_0)P(\theta_0|H_0)}{\int d^{N_1}\theta_1 \; P(D|\theta_1, H_1)P(\theta_1|H_1)} = \frac{\mathcal{Z}_0}{\mathcal{Z}_1}, \tag{1.9}$$

or equivalently the difference in log evidence $\ln \mathcal{Z}_0 - \ln \mathcal{Z}_1$ if often termed as the **Bayes factor** $\mathcal{B}_{0,1}$:

$$\mathcal{B}_{0,1} = \ln \frac{\mathcal{Z}_0}{\mathcal{Z}_1}, \tag{1.10}$$

where $\theta_i$ is a parameter vector (with dimensionality $N_i$) for the hypothesis $H_i$ and $i = 0, 1$. In Eqn. (1.10), the quantity $\mathcal{B}_{0,1} = \ln K$ provides an idea on how well model 0 may fit the data when is compared to model 1. Jeffreys provided a suitable guideline scale on which we are able to make qualitative conclusions (see Table 1.2).

We can see that Bayes theorem has an enormous implication with respect to a statistical inferential point of view. In a typical scenario we collect some data and hope to interpret it with a given model, however, we usually do the opposite. That is, first we have a set of data and then we can confront a model considering the probability that our model fits the data. Bayes theorem provides a tool to relate both scenarios. Then, thanks to the Bayes theorem, in principle, we are able to select the model that best fits the data.

*Example.*- We go back to the example shown in the last section: the coin toss. We are interested in the probability of obtaining two heads in a row given the data $P(2heads|D)$ ($D =$ the previous 14 coin tosses acting as data). First of all let us assume that we have a model with parameter $p$ to define the probability of obtaining the two heads given our model $P(2heads|p)$. This parameter will have a probability distribution $P(p|D)$ depending on the data we already

have. Therefore the probability can be obtained by averaging over all the possible parameters with its corresponding density distribution

$$P(2heads|D) = \int_0^1 P(2heads|p)P(p|D)dp. \tag{1.11}$$

For simplicity we do not update $p$ between the two tosses and we assume that both are independent from each other. With this last assumption we have

$$P(2heads|p) = [P(head|p)]^2, \tag{1.12}$$

where $P(head|p)$ is the probability of obtaining a head given our model. We assume a simple description of $P(head|p)$ as

$$P(head|p) = p \quad \Rightarrow \quad P(2heads|p) = p^2. \tag{1.13}$$

On the other hand, notice that we do not know *a priori* the quantity $P(p|D)$ but $P(D|p)$ (i.e. we know the probability of obtaining a dataset by considering a model as correct). A good choice for experiments that have two possible results is a binomial distribution

$$P(x|p,n) = \binom{n}{x} p^x (1-p)^{n-x}, \tag{1.14}$$

with $n$ the number of trials (this case = 14) and $x$ the number of successes (here =10). Hence, we have an expression for $P(D|p)$ [Eqn. (1.4)]. Now we need to compute $P(p|D)$. Using the Bayes formula we have

$$P(p|D) = \frac{P(D|p)P(p)}{P(D)}. \tag{1.15}$$

A very convenient prior distribution for this scenario is the *beta distribution* $Beta(p;a,b)$[1] defined as

$$Beta(p;a,b) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} p^{a-1}(1-p)^{b-1}, \tag{1.16}$$

where $\Gamma$ is the gamma function. So

$$P(p) = Beta(p;a,b). \tag{1.17}$$

We are interested in the explicit form of $P(p|D)$ and in such case we need to compute $P(D)$. Plugging Eqn. (1.4) and Eqn. (1.17) into the integral of Eqn. (1.8) we have

$$P(D) = B(10+a, 4+b) \equiv \frac{\Gamma(10+a)\Gamma(4+b)}{\Gamma((10+a)+(4+b))}, \tag{1.18}$$

[1] It is chosen because it describes several statistical distributions, in particular the normal distribution defined as the non-informative one.

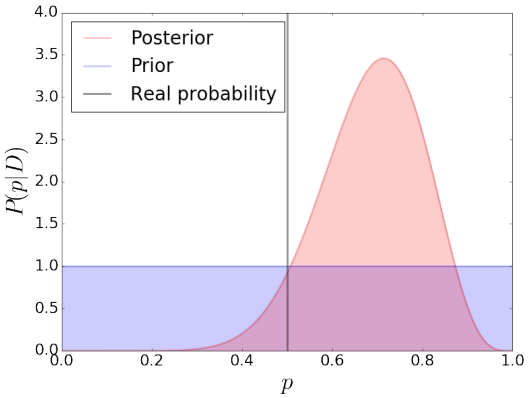



**Figure 1.1:** The coin example: blue figure displays the prior distribution $P(p)$ which is updated, using the data, to get the posterior distribution $P(p|D)$, (red). The vertical black line corresponds to the real value, $p = 0.5$.

and therefore

$$P(p|D) = \frac{p^{10+a-1}(1-p)^{4+b-1}}{B(10+a, 4+b)}. \tag{1.19}$$

Now we need to know the values of $a$ and $b$. If we assume that we know nothing about $p$, then we can assume the prior as an uniform distribution, this means $a = b = 1$. Notice from Fig. 1.1 that our posterior result (Red figure) described by Eqn. (1.19) does not exactly agree with the real value of $p$ (black dashed vertical line). We would expect the posterior distribution be centered at $p = 0.5$ with a very narrow distribution. Nevertheless this value is recovered by increasing the experimental data.

Finally, solving the integral in Eqn. (1.11) using (1.13) and (1.19) we arrive at the result obtained in the previous section

$$P(2heads|D) = \frac{B(13,5)}{B(11,5)} = 0.485. \tag{1.20}$$

### 1.3.2 Updating the probability distribution

As seen in the coin example, we weren't able to get the real value of $p$ because the lack of enough data. If we want to be closer, we would have to keep flipping the coin until the amount of data becomes sufficient. Let us continue with the example: suppose that after throwing the coin 100 times we obtain, let's say, 56 heads, while after throwing it 500 times we obtain 246 heads. Then, we expect to obtain a thinner distribution with center close to $p = 0.5$ (see Fig.

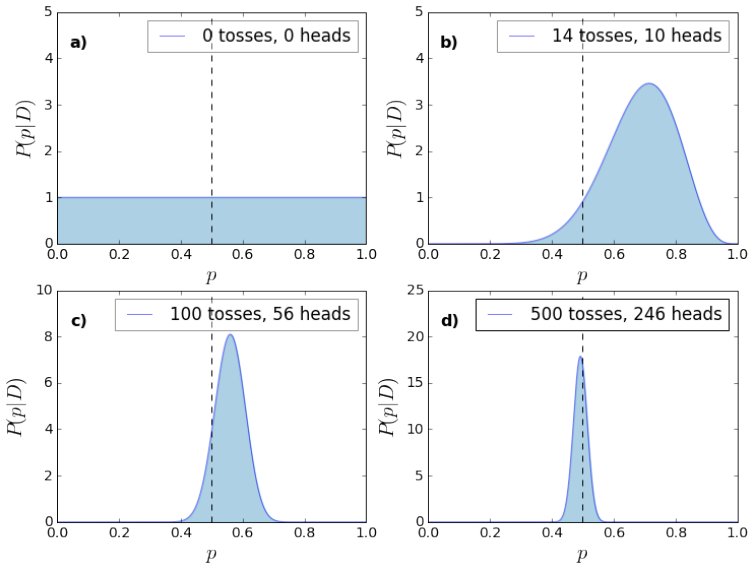



**Figure 1.2:** Posterior distributions $P(p|D)$, when the data is increased. Notice that while we continue increasing the experimental results, the posterior distribution starts to be more localized near by the real value $p = 0.5$.

1.2). Given this, it is clear that in order to confront a parameter model and be more accurate about the most probable (or "real") value, it is necessary to increase the amount of data (and the precision) in any experiment. That is, if we take into account the 500 tosses – with 246 heads – the previous result is updated to $P(2heads|D) = 0.249$, much closer to the real value.

Then, we have some model parameters that have to be confronted with different sets of data. This can be done in two alternative ways: (a) by considering the sum of all datasets we have; or (b) by taking each data set as the new data, but our prior information updated by the previous information. The important point in Bayesian statistics is that it is indeed equivalent to choose any of these two possibilities. In the coin toss example it means that it is identical to start with the prior given in Fig. 1.2-a and then by considering the 500 datapoints we can arrive at the posterior in Fig. 1.2-d, or similarly start with the posterior shown in Fig. 1.2-c as our prior and consider only the last 400 datapoints to obtain the same posterior, displayed in Fig. 1.2-d.

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

some prior information $I$ [**?** ]

$$P(\theta|DI,H) = \frac{P(\theta|I,H)P(DI|\theta,H)}{P(D|I,H)}, \tag{1.21}$$

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

$$P(\theta|D'I',H) = \frac{P(\theta|I,H)P(DD'I|\theta,H)}{P(DD'|I,H)} = P(\theta|[DD']I,H), \tag{1.22}$$

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

### 1.3.3 About the Likelihood

We mentioned that the Bayesian evidence is usually set apart when doing any inference procedure in the parameter space of a single model. Then, without loss of generality, we can fix it to $P(D|H) = 1$. If we ignore the prior[1] we can identify the posterior with the likelihood $P(\theta|D,H) \propto L(D|\theta,H)$ and thus, by maximizing it, we can find the most probable set of parameters for a model given the data. However, having ignored $P(D|H)$ and the prior, we are not able to provide an absolute probability for a given model, but only relative probabilities. On the other hand, 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 best-fit value, or the maximum likelihood $L_{max}$. Then, we can say that some parameters are acceptable if the likelihood ratio

$$\Lambda = -2\ln\left[\frac{L(D|\theta,H)}{L_{max}}\right], \tag{1.23}$$

is bigger than a given value.

Let us assume we have a Gaussian posterior distribution, which is single-peaked. We consider that $\hat{\theta}$ is the **mean** of the distribution

$$\hat{\theta} = \int d\theta\theta P(\theta|D,H). \tag{1.24}$$

If our model is well-specified and the expectation value of $\hat{\theta}$ corresponds to the real or most probable value $\theta_0$, we have

$$\langle\hat{\theta}\rangle = \theta_0, \tag{1.25}$$

[1] It is expected that the real value of any given parameter for a large enough dataset is independent of the prior.

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

$$\ln L(D|\theta) = \ln L(D|\theta_0) + \frac{1}{2}(\theta_i - \theta_{0i})\frac{\partial^2 \ln L}{\partial\theta_i \partial\theta_j}(\theta_j - \theta_{0j}) + ..., \tag{1.26}$$

where $\theta_0$ corresponds to the parameter vector of the real model. In this manner, we have that the likelihood can be expressed as a multi-variable likelihood given by

$$L(D|\theta) = L(D|\theta_0)\exp\left[-\frac{1}{2}(\theta_i - \theta_{0i})H_{ij}(\theta_j - \theta_{0j})\right], \tag{1.27}$$

where

$$H_{ij} = -\frac{\partial^2 \ln L}{\partial\theta_i \partial\theta_j}, \tag{1.28}$$

is called the **Hessian matrix** and it controls whether the estimates of $\theta_i$ and $\theta_j$ are correlated. 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. 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. On the other hand, if the data is not normally distributed we can resort to the central limit theorem. In this way, the central limit theorem tell us that the resulting distribution will be best approximated by a multi-variate Gaussian distribution [**?** ].

### 1.3.4 Letting aside the priors

In this section we present an argument for letting aside the prior in the parameter estimation. For this, we follow the example given in [**?** ]. In this example there are two people, A and B, that are interested in the measurement of a given physical quantity $\theta$. A and B have different prior beliefs regarding the possible value of $\theta$. 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 assume they are described by two Gaussian distributions with mean $\mu_i$ and variance $\Sigma_i^2$. Now, A and B make a measurement of $\theta$ together using an apparatus subject to a Gaussian noise with known variance $\sigma$. They obtain the value $\theta_0 = m_1$. Therefore they can write their likelihoods for $\theta$ as

$$L(D|\theta, HI) = L_0 \exp\left[-\frac{1}{2}\frac{(\theta - m_1)^2}{\sigma^2}\right]. \tag{1.29}$$

By using the Bayes formula, the posterior of the model A (and B) becomes

$$P(\theta|m_1) = \frac{L(m_1|\theta I_i)P(\theta|I_i)}{P(m_1|I_i)}, \tag{1.30}$$

where we have skipped writing explicitly the hypothesis $H$ and used the notation given in Eqn. (1.21). Then, the posterior of A and B are (again) Gaussian with mean

$$\hat{\mu}_i = \frac{m_1 + (\sigma/\Sigma_i)^2\mu_i}{1 + (\sigma/\Sigma_i)^2}, \tag{1.31}$$

and variance

$$\tau_i^2 = \frac{\sigma^2}{1 + (\sigma/\Sigma_i)^2}, \quad (i = A, B). \tag{1.32}$$

Thus, if the likelihood is more informative than the prior i.e. $(\sigma/\Sigma_i) \ll 1$ the posterior mean of A (and B) will converge towards the measured value, $m_1$. As more data are obtained one can simply replace the value of $m_1$ in the above equation by the mean $\langle m \rangle$ and $\sigma^2$ by $\sigma^2/N$. Then, we can see that the initial prior $\mu_i$ of A and B will progressively be overridden by the data. This process is illustrated in Figure 1.3 where the green (red) curve corresponds to the probability distribution of $\theta$ for person A (B) and the blue curve corresponds to their likelihood.

### 1.3.5 Chi-square and goodness of fit

We mentioned the main aim of parameter estimation is to maximize the likelihood in order to obtain the most probable set of model parameters, given the data. If we consider the Gaussian approximation given in Eqn. (1.27) we can see the likelihood will be maximum if the quantity

$$\chi^2 \equiv (\theta_i - \theta_{0i})H_{ij}(\theta_j - \theta_{0j}), \tag{1.33}$$

is minimum. The quantity $\chi^2$ is usually called **chi-square** and is related to the Gaussian likelihood via $L = L_0 e^{-\chi^2/2}$. Then, we can say that maximizing the Gaussian likelihood is equivalent to minimizing the chi-square. However, as we mentioned before, there are some circumstances where the likelihood cannot be described by a Gaussian distribution, in these cases the chi-square and the likelihood are no longer equivalent.

The probability distribution for different values of $\chi^2$ around its minimum, is given by the $\chi^2$ 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 $\chi^2$ exceeds by chance a value $\hat{\chi}$ for the correct model. This probability is given

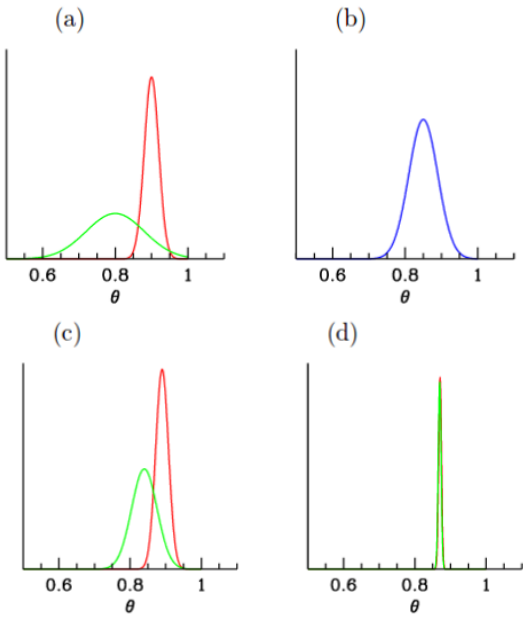



**Figure 1.3:** Converging views in Bayesian inference (taken from [**?** ]). A and B have different priors $P(\theta|I_i)$ for a value $\theta$ (panel (a)). Then, they observe one datum with an apparatus subject to a Gaussian noise and they obtained a likelihood $L(\theta; HI)$ (panel (b)), after which their posteriors $P(\theta|m_1)$ are obtained (panel (c)). Then, after observing 100 data, it can be seen how both posteriors are practically indistinguishable (panel (d)).

by $Q(v, \hat{\chi}) = 1 - \Gamma(v/2, \hat{\chi}/2)$ [**?** ], where $\Gamma$ is the incomplete Gamma function. Then, the probability that the observed $\chi^2$ (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 $Q$ for the best-fit values (minimum chi-square) we can have a measure of the goodness of 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 error measurements are not normally distributed.

On the other hand, if $Q$ is too large there are some reasons to believe that:

- Errors have been overestimated.

| | | $\Delta\chi^2$ | | |
|---|---|---|---|---|
| $\sigma$ | $p$ | $M=1$ | $M=2$ | $M=3$ |
| 1 | 68.3% | 1.00 | 2.30 | 3.53 |
| 2 | 95.4% | 4.00 | 6.17 | 8.02 |
| 3 | 99.73% | 9.00 | 11.8 | 14.20 |

**Table 1.3:** $\Delta\chi^2$ for the conventional 68.3%, 95.4% and 99.73% as a function of the number of parameters ($M$) for the joint confidence level.

- Data are correlated or non-independent.
- The distribution is non-Gaussian.

### 1.3.6 Contour plots and confidence regions

Once the best fit parameters are obtained we would like to know the confidence regions where values could be considered good candidates for our model. The most logical election is to take values inside a compact region around the best fit value. Then, a natural choice are regions with constant $\chi^2$ boundaries. When the $\chi^2$ possesses more than one minimum, it is said that we have non-connected confidence regions, and for multi-variate Gaussian distributions (as the likelihood approximation in Eqn. (1.27)) these are ellipsoidal regions. In this section we exemplify how to calculate the confidence regions, following [**?** ].

We consider a little perturbation from the best fit of chi-square $\Delta\chi^2 = \chi^2 - \chi^2_{best}$. Then we use the properties of $\chi^2$ distribution to define confidence regions for variations on $\chi^2$ to its minimum. In Table 1.3 we see the typical 68.3%, 95.4% and 99.73% confidence levels as a function of number of parameters $M$ for the joint confidence level. For Gaussian distributions (as likelihood) these correspond to the conventional 1, 2 and 3 $\sigma$ confidence levels. As an example we plot in Figure 1.4 the corresponding confidence regions associated to the coin example.

The general recipe to compute constant $\chi^2$ confidence regions is as follows: after finding the best fit by minimizing $\chi^2$ (or maximizing the likelihood) and checking that $Q$ is acceptable for the best parameters, 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:

$$Q(n-M, min(\chi^2)+\Delta\chi^2) = p. \tag{1.34}$$

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

### 1.3.7 Marginalization

It is clear that a model may (in general) depend on more than one parameter. However, some of these parameters $\theta_i$ may be of less interest. For example, they may correspond to nuisance parameters like calibration factors, or it may be the case that we are interested in only one of the parameter constraints rather than the joint of two or more of them simultaneously. Then we **marginalize** over the uninteresting parameters by

$$P(\theta_1, ..., \theta_j, H|D) = \int d\theta_{j+1}...d\theta_m P(\theta, H|D), \tag{1.35}$$

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

### 1.3.8 Fisher Matrix

Once we have a dataset it is important to know the accuracy for which we can estimate parameters. Fisher suggested a way 70 years ago [**?** ]. In this section we review the main results of his work.

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

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

$$F_{ij} = -\left\langle \frac{\partial^2 \mathcal{L}}{\partial\theta_i \partial\theta_j} \right\rangle, \tag{1.36}$$

where

$$\mathcal{L} = \ln L. \tag{1.37}$$

It is clear that $F = \langle H \rangle$, where the average is made with observational data.

As we can see from Eqn. (1.2c), for independent data sets the complete likelihood is the product of the likelihoods, and the Fisher matrix is the sum of individual Fisher matrices.

A pedagogical and easy case is having one-parameter $\theta_i$ with a Gaussian likelihood. In this scenario

$$\Delta\mathcal{L} = \frac{1}{2}F_{ii}(\theta_i - \theta_{0i})^2, \tag{1.38}$$

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

$$\sigma_{ij}^2 \geq (F^{-1})_{ij}. \tag{1.39}$$

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

$$\sigma_{\theta_i} \geq (F^{-1})_{ii}^{1/2}. \tag{1.40}$$

The beauty of the Fisher matrix approach is that there is a simple prescription for setting it up by only knowing the model and measurement uncertainties, and under the assumption of a Gaussian likelihood the Fisher matrix is the inverse of the covariance matrix. So, all we have to do is set up the Fisher matrix and then invert it to obtain the covariance matrix (that is, the uncertainties on the model parameters). In addition, its fast calculations also enables one to explore different experimental setups and optimize the experiment.

The main point of the Fisher matrix formalism is to predict how well the experiment will be able to constrain the parameters, of a given model, before doing the experiment and perhaps even without simulating it 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 inequality in Eqn. (1.39) is called the Kramer-Rao inequality. One can see that the Fisher information matrix represents a lower bound of the errors. Only when the likelihood is normally distributed, the inequality is transformed into an equality. However as we saw in Sec. 1.3.3 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.

#### 1.3.8.1 Constructing Fisher Matrices: A simple description

Let us construct Fisher matrices in a simple way. Suppose we have a model that depends of $N$ parameters $\theta_1, \theta_2, ..., \theta_N$. We consider $M$ observables $f_1, f_2, ..., f_M$ each one related to the model

parameters by some equation $f_i = f_i(\theta_1, \theta_2, ..., \theta_N)$. Then the elements of the Fisher matrix can be computed as

$$F_{ij} = \sum_k \frac{1}{\sigma_k^2} \frac{\partial f_k}{\partial \theta_i} \frac{\partial f_k}{\partial \theta_j}, \tag{1.41}$$

where $\sigma_k$ are the errors associated to each observable and we have considered them Gaussianly distributed.

Here, instead of taking the real data values (which could be unknown) it is possible to recreate the data with a fiducial model. The errors associated to the mock data can be taken as the expected experimental errors, and then be possible to calculate the above expression.

To complement the subject, there is also the **Figure of Merit** used by the Dark Energy Task Force (DETF) [**?** ] which is defined as the reciprocal of the area in the plane enclosing the 95% confidence limit of two parameters. The larger the figure of merit the greater accuracy one has measuring said parameters. As an example let us take a look at Figure **??** and right panel of Figure **??**, the area of the error ellipse with only Hubble Data (HD) is clearly bigger than the error ellipse using HD plus several data sets. Then, for this case the figure of merit would be bigger than with only HD data since its area is smaller, making it more accurate for measuring the parameters $\Omega_m$ and $h$. The DETF figure of merit can also be used to see how different experiments break degeneracies. It can also be used to predict accuracy in future experiments (experimental design).

### 1.3.9 Importance Sampling

We call **Importance Sampling** (IS) to different techniques of determining properties of a distribution by drawing samples from another one. The main request of this idea, is that the distribution one samples from should be representative of the distribution of interest (for a larger number of samples). In such case, we should infer different quantities out of it. In this section we review the basic concepts necessary to understand the IS, following [**?** ].

Suppose we are interested in computing the expectation value $\mu_f = E_p[f(X)]$, where $f(X)$ 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 $\mu_f$ as

$$\mu_f = \int f(x)p(x)dx = \int f(x)\frac{p(x)}{q(x)}q(x)dx = E_q[f(X)w(x)], \tag{1.42}$$

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

$$\hat{\mu}_f = \frac{1}{m} \sum_{j=1}^{m} w(x^{(j)}) f(x^{(j)}). \tag{1.43}$$

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

$$\hat{\mu}_f = \frac{\sum_{j=1}^{m} w(x^{(j)}) f(x^{(j)})}{\sum_{j=1}^{m} w(x^{(j)})}. \tag{1.44}$$

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

Another useful quantity to compute in Bayesian analysis is the ratio between evidences for two different models

$$\frac{P'(D)}{P(D)} = E\left[\frac{P'(\theta, D)}{P(\theta, D)}\right]_{P(\theta|D)} \simeq \frac{1}{N} \sum_{n=1}^{N} \frac{P'(D|\theta_n) P'(\theta_n)}{P(D|\theta_n) P(\theta_n)}, \tag{1.45}$$

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.

### 1.3.10 Combining datasets: Hyperparameter method

Suppose we are dealing with multiple datasets $\{D_1, ..., D_N\}$, coming from a collection of different surveys $\{S_1, ..., S_N\}$. Sometimes it is difficult to know, *a priori*, if all our data are consistent with each other, or whether there could be one or more that are likely to be erroneous. If we were sure that all datasets are consistent, then it should be enough to update the probability as seen in Sec. 1.3.2 in order to calculate the new posterior distribution for the parameters we are interested in. However, since there is usually an uncertainty about this, a way to know how useful a data may be is by introducing the **hyperparameter method**. This method was initially introduced by [**? ?** ] in order to perform a joint estimation of cosmological parameters from combined datasets. This method may be used as long as every survey is independent from each other. In this section we review the main steps necessary to understand the hyperparameter method. If the reader is interested in a more extended explanation, we

encourage to consult [**? ?** ].

The main feature of this process is the introduction of a new set of "hyperparameters" $\alpha$ in the Bayesian procedure to allow extra freedom in the parameter estimation. These hyperparameters are equivalent to nuisance parameters in the sense that we need to marginalize over them in order to recover the posterior distribution, i.e.

$$P(\theta|D,H) = \frac{1}{P(D|H)} \int P(\theta|\alpha,H)P(\alpha|D,H)d\alpha, \tag{1.46}$$

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

$$P(\theta|D,H) = \frac{P(\theta,H)}{P(D|H)} \left[ \prod_{k=1}^{N} \int P(D_k|\theta,\alpha_k,H)P(\alpha_k)d\alpha_k \right]. \tag{1.47}$$

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

We add these $\alpha_k$ in order to weight every dataset and take away the data that does not seem to be consistent with other ones. Then, we would like to know whether the data supports the introduction of hyperparameters or not. A way to address this point is given by the Bayesian evidence $K$ defined in Eqn. (1.9). If we consider a Gaussian likelihood with maximum entropy prior, and assuming that in average the hyperparameters' weight are unity, we can rewrite the marginalized likelihood function $L(D|\theta,H_1)$ for model $H_1$ as

$$P(D|\theta,H_1) = \prod_{k=1}^{N} \frac{2\Gamma(\frac{n_k}{2}+1)}{\pi^{n_k/2}|V_k|^{1/2}} (\chi_k^2+2)^{-(\frac{n_k}{2}+1)}, \tag{1.48}$$

obtaining an explicit functional form for $K$, given by

$$K = \prod_{k=1}^{N} \frac{2^{n_k/2+1}\Gamma(n_k/2+1)}{\chi_k^2+2} e^{-\chi_k^2/2}. \tag{1.49}$$

Here, $\chi^2_k$ is given by (1.33) for every dataset and $n_k$ is the number of points contained in $D_k$. In equation (1.48) $V_k$ is the covariance matrix for the $k$-data. Suppose we have two models, one with hyperparameters, called $H_1$, and a second one without them, called $H_0$. The Bayesian evidence $P(D|H_i)$ is the key quantity for making a comparison between two different models. In fact, by using the Bayes factor $K$ from Eqn. (1.49) we can estimate the necessity to introduce the hyperparameters to our model using the criteria given in Table 1.2. 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 (1.45).

## 1.4 Numerical tools

In typical scenarios it results very difficult to compute the posterior distribution analytically. For these cases the numerical tools available play an important role during the parameter estimation task. There exist several options to carry out this work, nevertheless in this section we focus only on the Markov Chain Monte Carlo (MCMC) with the Metropolis Hastings algorithm (MHA). Additionally, in this section we present some useful details we take into account to make more efficient our computation.

### 1.4.1 MCMC techniques for parameter inference

The purpose of a MCMC algorithm is to build up a sequence of points (called "**chain**") in a parameter space in order to evaluate the posterior of Eqn. (1.7). In this section we review the basic results for this procedure in a simplistic way, but for curious readers it is recommendable to check [**? ? ? ?** ] for the Markov chain theory.

A **Monte Carlo** simulation is assigned to algorithms that use random number generators to approximate a specific quantity. On the other hand, a sequence $X_1, X_2, ...$ of elements of some set is a **Markov Chain** if the conditional distribution of $X_{n+1}$ given $X_1, ..., X_n$ depends only on $X_n$. In other words, a Markov Chain is a process where we can compute subsequent steps based only in the information given at the present. An important property of a Markov Chain is that it converges to a stationary state where successive elements of the chain are samples from the target distribution, in our case it converges to the posterior $P(\theta|D, H)$. In this way we can estimate all the usual quantities of interest out of it (mean, variance, etc).

The combination of both procedures is called a **MCMC**. The number of points required to get good estimates in MCMCs 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

$$p(\theta|D,H) \simeq \frac{1}{N}\sum_{i=1}^{N}\delta(\theta-\theta_i), \tag{1.50}$$

being $N$ the number of points in the chain. Then, the posterior mean is computed as

$$\langle\theta\rangle = \int d\theta\theta P(\theta,H|D) \simeq \frac{1}{N}\sum_{i=1}^{N}\theta_i, \tag{1.51}$$

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

$$\langle f(\theta)\rangle \simeq \frac{1}{N}\sum_{i=1}^{N} f(\theta_i). \tag{1.52}$$

As mentioned before, in a Markov Chain it is necessary to generate a new point $\theta_{i+1}$ from the present point $\theta_i$. However, as it is expected, we need a criteria for accepting (or refusing) this new point depending on whether it turns out to be better for our model or not. 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 all of it. The simplest algorithm that contains all this information in its methodology is known as the Metropolis-Hastings algorithm.

#### 1.4.1.1 Metropolis-Hastings algorithm

In the **Metropolis-Hastings algorithm** [**? ?** ] it is necessary to start from a random initial point $\theta_i$, with an associated posterior probability $p_i = p(\theta_i|D,H)$. We need to propose a candidate $\theta_c$ by drawing from a **proposal distribution** $q(\theta_i,\theta_c)$ used as a generator of new random steps. Then, the probability of acceptance the new point is given by

$$p(acceptance) = min\left[1, \frac{p_c q(\theta_c,\theta_i)}{p_i q(\theta_i,\theta_c)}\right]. \tag{1.53}$$

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

$$p(acceptance) = min\left[1, \frac{p_c}{p_i}\right]. \tag{1.54}$$

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

1. Choose a random initial condition $\theta_i$ in the 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-Hastings 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.

#### 1.4.1.2 A first example of parameter inference

In order to exemplify the numerical tools learned in this section, let us go back to the coin toss example seen in Sec. 1.3.1. Since our main interest is that the reader understands the basic procedure given in this section, let us try to estimate the value of $p$ (or region of values for $p$) that best matches our data (hence, we assume only the 14 times that the coin was thrown). To calculate the posterior distribution (1.17) we use the MHA.

As mentioned before, we consider a likelihood given by a binomial distribution (1.4) and a normal distributed prior (1.16) ($a = b = 1$). As our first "guess" for $p$ we consider $p_i = 0.1$. We generate a new candidate $p_c$ as $p_c = p_{cu} + G(p_{cu}, \hat{\sigma})$, where $G(p_{cu}, \hat{\sigma})$ is our proposed Gaussian distribution centered at $p_{cu}$ with variance $\hat{\sigma} = 0.1$; $p_{cu}$ is the current value of $p$, for our first step is $p_{cu} = p_i$. Then, we introduce the MHA in a Python code, as can be seen in Appendix **??**. Our final result, (shown in Fig. 1.4), is a posterior distribution that matches very well with the results calculated analytically (shown in Figure 1.1). Numerically we obtained $p = 0.695^{+0.123}_{-0.107}$, where the upper and lower values for $p$ correspond to the $1\sigma$ standard deviation. Notice that we have plotted the width of our $1\sigma$, $2\sigma$ and $3\sigma$ confidence regions in the same figure.
To complement the example we also show in the right panel of Figure 1.4 the Markov Chain generated by our code where we have considered 5000 steps in our chain. It is easy to see that the chain oscillates with a large amplitude around a middle value. This amplitude is expected because we do not have enough data to constrain more accurately the value of $p$.

Remark: In appendix **??** we include the MCMC algorithm using an explicit code for the MCMC process. However, in Python there are some modules that can simplify this task. For example, PyMC3 [**?** ] 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 by applying the tools already learned.

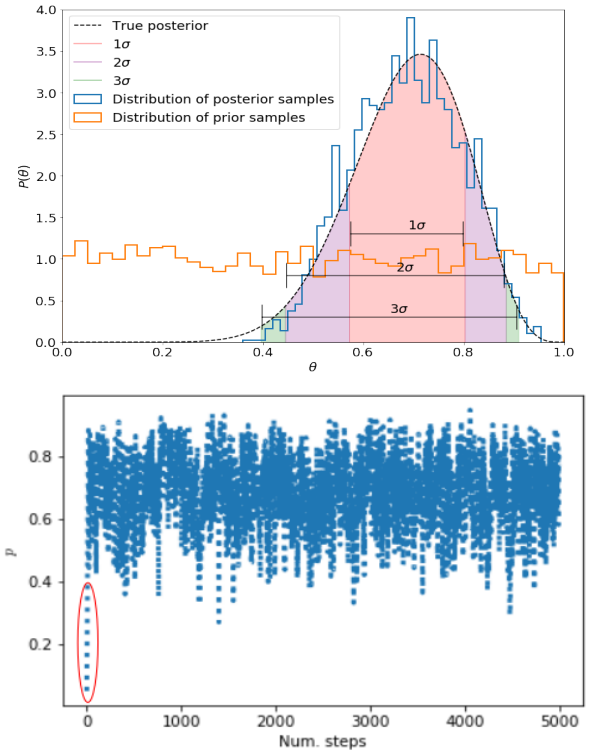



**Figure 1.4:** Left panel: 1D posterior distribution for our example. We plot the prior distribution (red), true posterior (dashed-black) and the posterior calculated by the MHA (blue). We plot 1,2 and $3\sigma$ confidence regions for the estimation of $p$. Right panel: associated Markov chain. We use $p_i = 0.1$ as our first "guess" for $p$.

#### 1.4.1.3 Convergence test

It is clear that we need a test to know when our chains have converged. We need to verify that the points in the chain are not converging to a "false convergent point" or to a local maximum point. In this sense, we need that our algorithm takes into account this possible difficulty. The simplest way (the informal way) to know if our chain is converging to a global maximum is by running several chains starting with different initial proposals for the parameters we are interested in. Then, if we see by naked eye, that all the chains seem to converge into a single region of the possible value for our parameter, we may say 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 (region) of $p$ that we found is a stationary value. In Figure 1.5 we plot 5 different Markov chains with initial "guess" conditions $p = 0.2, 0.3, 0.5, 0.7, 0.9$. As we

expected from the analytical result, after several steps all the chains seem to concentrate near by the same value.

The convergence method used above is very informal and we would like to have a better way to ensure that our result is correct. The usual test is the *Gelman-Rubin* convergence criterion [**? ?** ]. That is, by starting with $M$ chains with very different initial points and $N$ points per chain, if $\theta_i^j$ 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

$$\langle \theta^j \rangle = \frac{1}{N} \sum_{i=1}^{N} \theta_i^j, \tag{1.55}$$

and the mean of all the chains

$$\langle \theta \rangle = \frac{1}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} \theta_i^j. \tag{1.56}$$

Then, the chain-to-chain variance $B$ is

$$B = \frac{1}{M-1} \sum_{j=1}^{M} (\langle \theta^j \rangle - \langle \theta \rangle)^2, \tag{1.57}$$

and the average variance of each chain is

$$W = \frac{1}{M(N-1)} \sum_{i=1}^{N} \sum_{j=1}^{M} (\theta_i^j - \langle \theta^j \rangle)^2. \tag{1.58}$$

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

$$\hat{R} = \frac{\frac{N-1}{N} W + B(1 + \frac{1}{M})}{W}, \tag{1.59}$$

which is the ratio of the two estimates, approaches unity. A typical convergence criteria is when $0.97 < \hat{R} < 1.03$.

#### 1.4.1.4 Some useful details

**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) $\hat{\sigma} = 0.1$. This value was taken because we initially explored, by hand, different values for $\hat{\sigma}$ and we select the quickest that approaches the analytic posterior distribution of $p$. However, 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 very long to explore the target distribution which implies that the algorithm will be very inefficient. As we can see in Figure 1.6 (left panel), considering

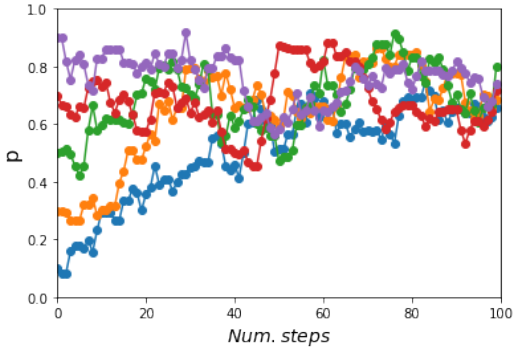



**Figure 1.5:** Multiple MCMC. We use five Markov Chains to estimate the convergence.

an initial step $p_i = 0.6$ and a variance for the proposal distribution $\hat{\sigma} = 0.002$, the number of points are not enough for the system to move to its "real" posterior distribution. On the other hand, if the scale of $q$ is too large, the chain gets stuck and it does not jump very frequently (right panel of the figure with $\hat{\sigma} = 0.8$) so we will have different "peaks" in our posterior.

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. This process can be computed a couple of times before running the "real" MCMC.

**The burn-in.** It is important to notice that at the beginning of the chain we will have a region of points outside the stationary region (points inside the ellipse in the right panel of Figure 1.4). This early part of the chain (called "burn-in") must be ignored, this means that the dependence on the starting point must be lost. Thus, it is important to have a reliable convergence test.

**Thinning.** 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 of 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 [**?** ]. Thinning the chains can be useful in other kind of circumstances, 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

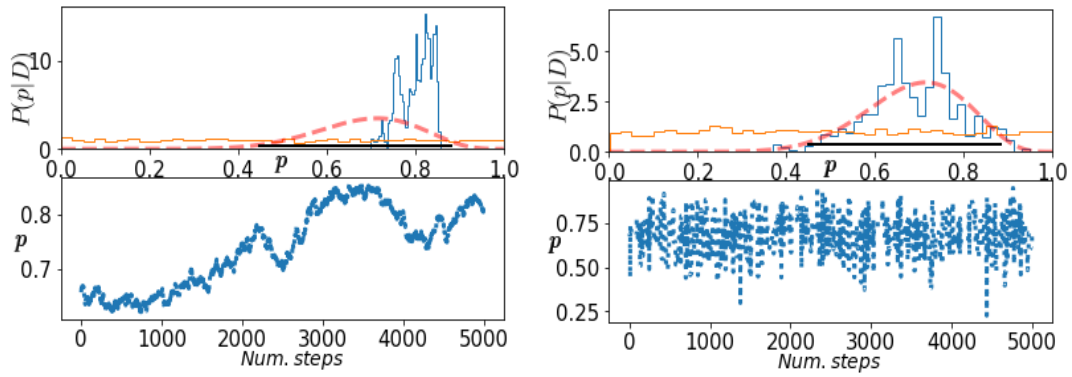



**Figure 1.6:** Two Markov Chains considering different variance for our Gaussian proposal distribution. Left panel corresponds to $\hat{\sigma} = 0.002$, while right panel corresponds to $\hat{\sigma} = 0.8$.

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 autocorrelation *lag* $k$ is defined as the correlation between every sample and the sample $k$ steps before. It can be quantified as [**?** **?** ]

$$\rho_k = \frac{Cov(X_t, X_{t+k})}{\sqrt{Var(X_t)Var(X_{t+k})}} = \frac{E[(X_t - X)(X_{t+k} - X)]}{\sqrt{E[(X_t - X)^2]E[(X_{t+k} - X)^2]}}, \tag{1.60}$$

where $X_t$ is the $t$-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).

### More samplers

**Gibbs sampling.** The basic idea of the Gibbs sampling algorithm [**?** ] is to split the multidimensional $\theta$ into blocks and sample each block separately, conditional on the most recent values of the other blocks. It basically breaks a high-dimensional problem into low-dimensional problems.

The algorithm reads as follows:

1. $\theta$ consists of $k$ blocks $\theta_1, ..., \theta_k$. Then, at step $i$
2. Draw $\theta_1^{i+1}$ from $p(\theta_1|\theta_2^i, ..., \theta_k^i)$
3. Draw $\theta_2^{i+1}$ from $p(\theta_2|\theta_1^{i+1}, \theta_3^i, ..., \theta_k^i)$

4. ...

5. Draw $\theta_k^{i+1}$ from $p(\theta_k|\theta_1^{i+1}, \theta_2^{i+1}, ..., \theta_{k-1}^{i+1})$

6. Repeat the above steps for the wished iterations with $i \to i+1$.

The distribution $p(\theta_1|\theta_2, ..., \theta_k) = \frac{p(\theta_1,...,\theta_k)}{p(\theta_2,...,\theta_k)}$ is known as the *full conditional distribution of* $\theta_1$. This algorithm is a special case of MHA where the proposal is always accepted.

**Metropolis Coupled Markov Chain Monte Carlo ($MC^3$).** It is easy to see that it could be a little problematic if our likelihood has local maxima. The $MC^3$ is a modification of the standard MCMC algorithm that consists of running several Markov Chains in parallel to explore the target distribution for different "temperatures". This simplifies the way we sample our parameter space and help us to avoid this local maxima. Here we exemplify the basic idea of this algorithm, however if you are interested in a more extensive explanation, or a modification to make the temperature of the chains dynamical, please consult the reference [**?** ].

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

$$P(\theta, T|D, H) \propto L(\theta, D)^{1/T} P(\theta, H), \tag{1.61}$$

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 $T_1 < T_2 < ... < T_N$, usually taken with a geometrically distributed division, with $T_1 = 1$. The coldest chain $T_1$ samples the posterior distribution more accurately and behaves as a typical MCMC. Then, we define this chain as the main chain. The rest of the chains are running such 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

$$A_{i,j} = min\left\{\left(\frac{L(\theta_i)}{L(\theta_j)}\right)^{1/T_j - 1/T_i}, 1\right\}. \tag{1.62}$$

In this way, if a swap is accepted, chains $i$ and $j$ must exchange their current position in the parameter space, then chain $i$ has to be in position $\theta_j$ and chain $j$ has to move to position $\theta_i$.

We can see that, since the hottest chain $T_{max}$ can access easier to all the modes of $P(\theta, H, T_{max}|D)$, 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.

**Affine Invariant MCMC Ensemble Sampler.** The main property of this algorithm relies on its invariance under affine transformations. Let's consider a highly anisotropic density

$$p(x_1, x_2) \propto \exp\left(\frac{-(x_1 - x_2)^2}{2\epsilon} - \frac{(x_1 + x_2)^2}{2}\right), \tag{1.63}$$

which is difficult to calculate for small $\epsilon$. But by making the affine transformation

$$y_1 = \frac{x_1 - x_2}{\sqrt{\epsilon}}, \quad y_2 = x_1 + x_2, \tag{1.64}$$

we can rewrite the anisotropic density into the easier problem

$$p(y_1, y_2) \propto \exp\left(\frac{-(y_1^2 + y_2^2)}{2}\right). \tag{1.65}$$

A MCMC sampler has the form $X(t+1) = R(X(t), \psi(t), p)$, where $X(t)$ is the sample after $t$ iterations, $R$ is the sampler algorithm, $\psi$ is the sequence of independent identically distributed random variables and $p$ is the density. A sampler is said to be affine invariant if, for any affine transformation $Ax + b$,

$$R(AX(t) + b, \psi(t), p_{A,b}) = AR(X(t), \psi(t), p) + b. \tag{1.66}$$

There are already several algorithms that are affine invariant, one of the easiest is known as the *stretch move* [**?** ]. An algorithm fully implemented in Python under the name **EMCEE** [**?** ] is also affine invariant, and there are also some other algorithms that can be found in [**?** ].

**Even more samplers.** The generation of the elements in a Markov chain is probabilistic by construction and it depends on the algorithm we are working with. The MHA is the easiest algorithm used in Bayesian inference. However, there are several algorithms that can help us to fulfill our mission. For instance, some of the most popular and effective ones, are the Hamiltoninan Monte Carlo (see e.g. [**? ?** ]) or the Adaptative Metropolis-Hastings (AMH) (see e.g. [**?** ]).

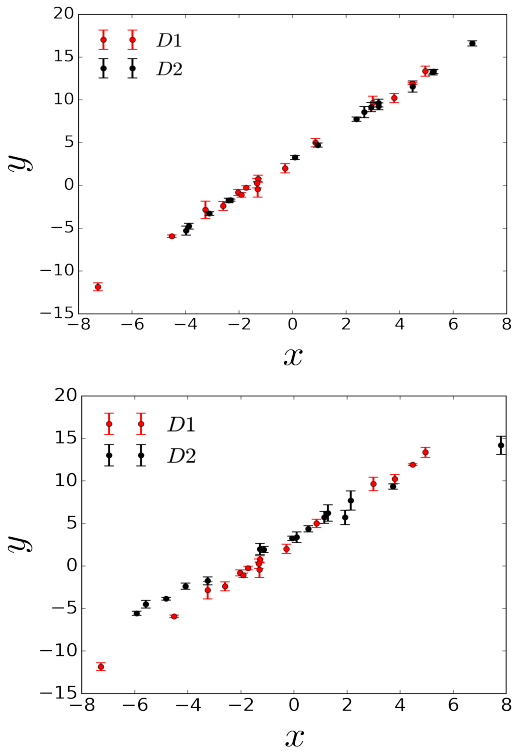



**Figure 1.7:** Datasets $D_1$ and $D_2$ measured by our straight-line theory. Case 1 (left) and case 2 (right).

## 1.5 Fitting a straight-line

In this section we apply the tools learned so far to the simplest example: fitting a straight-line. That is, we assume that we have a certain theory where our measurements should follow a straight line. Then, in order to apply our techniques, we simulate several datasets along this line. One of the principal topics we want to analyse is the hyperparameter method and how it works, so we will apply our analysis to two different cases (Figure 1.7):

1. Consider two datasets taken from the same straight-line but with different errors.

2. Consider two datasets but now we simulate both of them from different straight-lines and different errors.

In our analysis we used the PyMC3 module implemented in Python. Our complete code can be downloaded from the git repository [**?** ]. This code is simple to use and can be modified easily for any model to be tested. We recommend to use the file called "new model" where the reader can find a blank project. Here the data and model can be added up and, by running all the

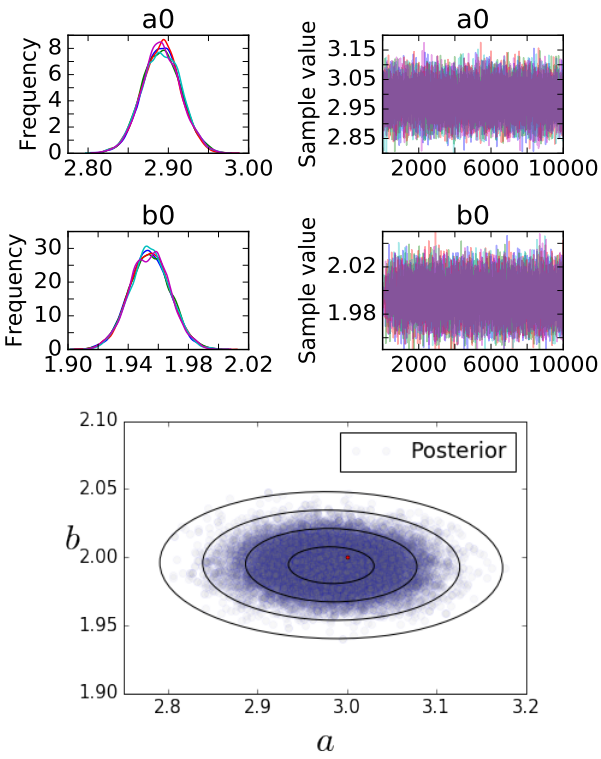



**Figure 1.8:** Left panel: 1D marginalized posterior distributions for our samples and the Markov chains for model $H_0$. Right panel: 2D marginalized posterior distributions along with 1-4 confidence regions for our parameters for model $H_0$. The red point corresponds to the true value.

notebook, obtain all the analysis we present in this section. One can find as well several notes that will help in programming the model with PyMC3, even if the model contains functions that are not defined in PyMC3.

### 1.5.1 Case 1

In this example we start by considering that our measurements for a given theory (a straight-line $y = a + bx$) are given by the data shown in left panel of Figure 1.7. These two datasets, D1 and D2, were generated from the line $y = 3 + 2x$, adding a gaussian error to each point. For D1 we add an error with a standard deviation $\sigma_1 = 0.3$, while for D2 we use $\sigma_2 = 0.2$. Then, we would like to estimate 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.

**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 some limits for $a$ and $b$ (we can see them by eye in our data). Then we consider the flat priors

$$a \propto U[0,5] \quad \text{and} \quad b \propto U[0,3], \tag{1.67}$$

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

From equation (1.27) we can write our likelihood as

$$L(D;line) \propto \exp\left[-\sum_d \frac{(y_d-y)^2}{2\sigma_d^2}\right], \tag{1.68}$$

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

We use the MHA to generate our MCMC. In our analysis we ran 5 chains with 10,000 steps for each one. We ran each chain with a temperature $T = 2$ and we thinned them every 50 steps. The results we obtained correspond to $a = 2.982 \pm 0.047$ and $b = 1.994 \pm 0.013$, and their posterior distributions are plotted in Figure 1.8. Notice that there are some regions where the frequency of events in our sample is increased. So we can say that such parameter regions seem to more likely match the data. Additionally we compute the Gelman-Rubin criterion for each variable in order to verify that our results converged, i.e. for a is 1.000017 and for b is 1.000291. We see that this number is very close to 1, so our convergence criterion is fulfilled. Right panel of Figure 1.8 displays the $1-4$ $\sigma$ confidence regions. We also add a point in red to show the real value for our parameters. The real value for $a$ and $b$ are inside of the curve corresponding to one standard deviation of our estimations in the inferential method.

We continue with the autocorrelation plots. As we mentioned, we need these plots to be small as $k$ increases in order to consider that our analysis is converging. We see in Figure 1.9 such plots and notice that our convergence criteria is fulfilled. Then, in Case 1 we can see that the model $H_0$ looks to be a very good estimation procedure.

**With hyperparameters. Model $H_1$.**

Now let us consider the Hyperparameter method. In this case our likelihood can be written as Eq. (1.48). Similarly to the last procedure, we compute the posterior with flat priors and using 5 chains with 10,000 steps for each one, and check for autocorrelations. Our results are as

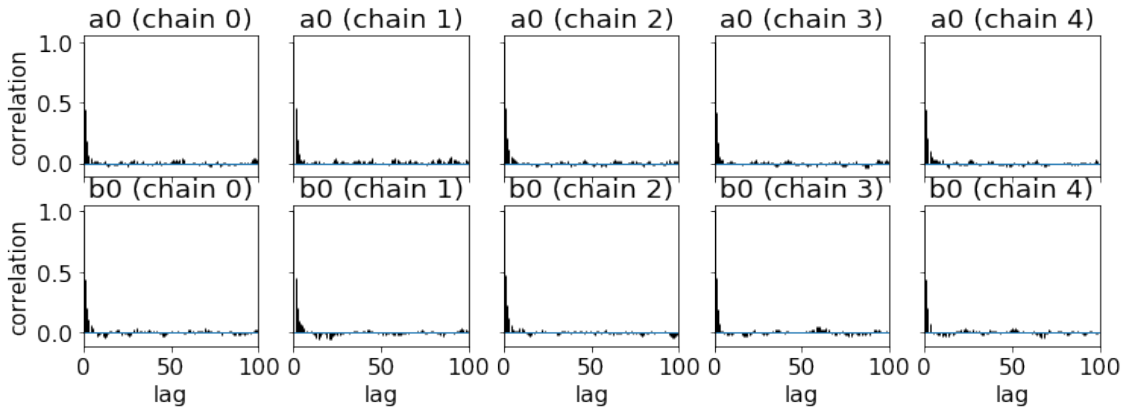



**Figure 1.9:** Autocorrelation plots for model $H_0$.

followed: $a = 2.97 \pm 0.038$ with Gelman-Rubin of 1.000113 and $b = 1.995 \pm 0.010$ with Gelman-Rubin 1.000155. Comparing both procedures we observe they provide similar results. In fact, the confidence regions for both approximations, Fig. 1.8 and left panel of Figure 1.10, are similar as well. So, which 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 we compute the evidence ratio $K$ between both models. We obtained from Eqn. (1.49)

$$K = 3. \tag{1.69}$$

Then, comparing with Table 1.2 we can say that the evidence for $H_1$ to be better than $H_0$ is weak. In such case it should be equally better to work with $H_0$ as to $H_1$, as we explained before.

Finally, in order to exemplify our results, let us plot in the right panel of Figure 1.10 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.

### 1.5.2 Case 2

Here we consider that we have the same theory for the straight-line but different measurements. The data points are given in the right panel of Figure 1.7. These correspond to our dataset $D_1$ and $D_2$, but now 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 with each other. Let us make again a parameter estimation for the parameters $a$ and $b$ and look for

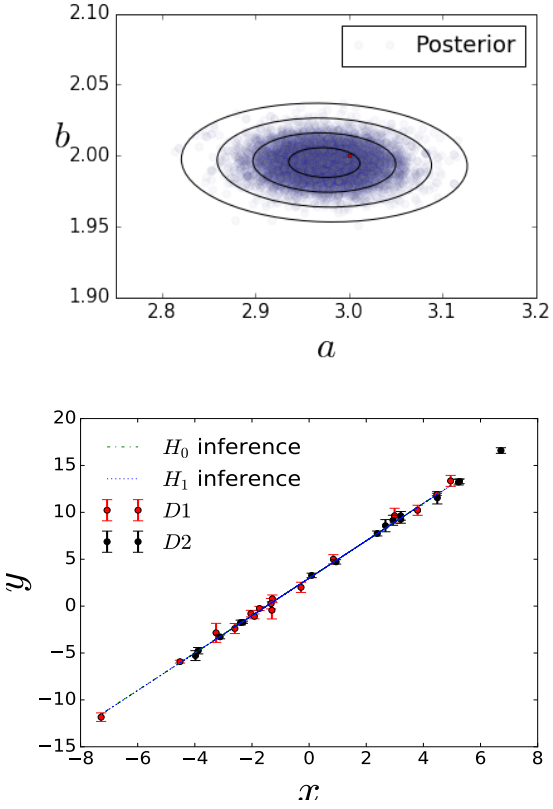



**Figure 1.10:** Left panel: confidence regions for the parameters in model $H_1$. Right panel: the best-fit for the straight-lines inferred by the data.

the differences in both procedures.

**Without hyperparameters. Model $H_0$.**

We follow 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 are the following: $a = 3.528 \pm 0.056$ and $b = 1.795 \pm 0.014$. Then we plotted our $1 - 4\sigma$ confidence regions in left panel of Figure 1.11. It is easy to see that our estimation differs so much from the real parameters in our datasets (red points). Of course this is because we are trying to fit a model with non auto-consistent datasets and therefore we arrive at incorrect results. Now, let us see what happens in the hyperparameters procedure.

**With hyperparameters. Model $H_1$.**

In the top right panel of Figure 1.11 we plotted our posterior distribution. We see immediately that both approximations are very different. 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 local maximum regions near the real values for our datasets (red dots). For this example we do not calculate the typical mean and standard deviation for our results.

As the last example we compare both methods. 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

$$K = 37, \tag{1.70}$$

which means that we have a very strong evidence that $H_1$ is better that $H_0$.
Finally, we can plot the straight-line inferred by model $H_0$ and the two inferred by model $H_1$. Considering parameters inside the two regions in the top right panel of Figure (1.11) we obtain the bottom panel of Figure 1.11.

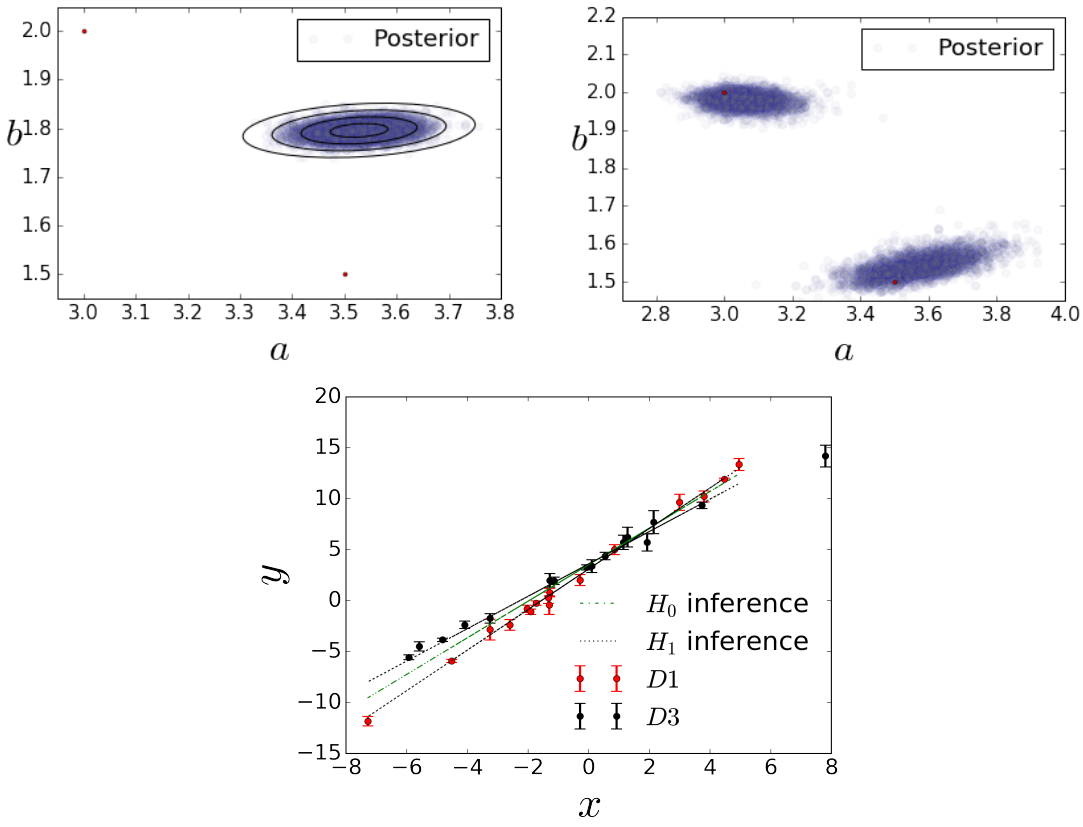



**Figure 1.11:** Top left panel: confidence regions for the parameters in model $H_0$. Top right panel: confidence regions for the parameters in model $H_1$. Bottom panel: Best-fit values for the straight-lines for Case 2 inferred by our with data.

# Bibliography