Bayesian inference and gravitational wave observations

Walter Del Pozzo

August 21, 2015

Abstract

Basic concepts of Bayesian inference and relevant applications for GW observations will be presented, with emphasis on binary systems. The mini-course will include key conceptual aspects of the numerical techniques used to apply these methods to general problems in Bayesian inference, with specific examples for GW observations and software libraries used in the actual LIGO/Virgo analysis. The exercise sessions will be a hands-on session to design a simple stochastic sampler to measure the masses of binary compact objects and an actual analysis of LIGO/Virgo data with the standard LIGO library analysis tools to locate a source in the sky and provide information about the relevant astrophysical parameters.

1 Basic concepts

We are so used to the way our brain processes information that we rarely stop to wonder about its mechanisms. For instance, imagine a situation in which a policeman sees a gentleman running with a purse in his hand. He decides immediately that the gentleman is dishonest and starts chasing him. There could be many perfectly legitimate and reasonable explainations for this man running with a purse in his hand, maybe his wife forgot it home ans she is at the train station and he needs to get there in time, however the policeman implicitly deems any explaination other than "he stole the purse" much less *probable*. What is the process that leads the policeman to his *inference* about some observation? What are its principles?

Scientific inference is dealing with the same exact proble as the policeman is: given some data which consist of some physical effect and superimposed noise, what do we learn about the physical effect of interest? When do we decide that a given physical effect is real or that a given explanation is "correct"?

1.1 Fundamentals of logic: propositions

A logical proposition is any sentence which can be either true or false. Examples of logical propositions are

Table 1: Truth table for the propositions $A, B, \bar{A}, \bar{B}, AB, A+B$.

A	B	$\bar{A}$	$\bar{B}$	AB	$A+B$
0	0	1	1	0	0
1	0	0	1	0	1
0	1	1	0	0	1
1	1	0	1	1	1

- x is greater than 10;
- the sky is cloudy today.

We are going to indicate logical propositions with capital letters $A, B, C, \ldots$. For instance we might indicate:

$$A \equiv \text{“the total mass of this binary black hole system is } 22M_\odot''. \qquad (1)$$

We will indicate the denial of a proposition with $\bar{A} \equiv A$ is false. Any admissible proposition can only assume two values, true or false.

It is possible to construct more complex propositions by introducing operations between propositions:

- the *conjunction* or *logical product* AB which asserts that both A and B are true;
- the *disjunction* or *logical sum* $A+B$ which asserts that either A or B are true.

We already introduced the *negation* of a proposition $\bar{A}$.

Given two propositions, how do we estabilish if they are equivalent? This can be done by constructing a truth table. Table 2 gives an example truth table for the proposition $A + \bar{B}$.

Boolean algebra has some very useful properties and identities:

1. *Idempotence* $AA = A$, $A + A = A$
2. *Commutativity* $AB = BA$, $A + B = B + A$
3. *Associativity* $A(BC) = (AB)C$, $A + (B + C) = (A + B) + C$
4. *Distributivity* $A(B + C) = AB + AC$, $A + (BC) = (A + B)(A + C)$

which can be used to prove many useful, and non-trivial, identities.

1.2 Deductive inference

The greek philosopher Aristotle set the rules for deductive reasoning based on the *strong syllogism*:

1. major premise: if A is true, then B is true

Table 2: Truth table for the propositions $A, B, AB, A = AB$.

A	B	AB	$A = AB$
0	0	0	1
1	0	0	0
0	1	0	1
1	1	1	1

  - minor premise: A is true
  - conclusion: B is true.

2. major premise: if A is true, then B is true
    - minor premise: A is false
    - conclusion: B is false.

In Boolean terms, the strong syllogism is expressed as:

$$A = AB \tag{2}$$

which is known as the *implication* and indicated as $A \implies B$. An example of deductive reasoning:

- all black holes in binary systems are spinning
- the event detected is a binary black hole
- therefore both black holes were spinning.

Exercise 2: verify that any false proposition implies all proposition. **Solution**: Let's compute the truth table for the implication: So $A \implies B$ is false only when A is true and B is false. On one hand, when both A and B are true, $A = AB$ is true, so in logic *every true statement implies every true statement*. On the other hand, if A is false then $A \implies B$ and $A \implies \bar{B}$ are both true, regardless of the truth value of B. Therefore, *a false proposition implies every proposition*.

1.3 Weak syllogism

In real life application, we very rarely have enough information to reason deductively and apply the strong syllogism and we have to fall back to the *weak* syllogism:

1. if A is true, then B is true
    - B is true
    - A becomes more plausible.
2. if A is true, then B is true

- A is false
- B becomes less plausible.

An example:

- all black holes in binary systems are spinning
- both stars in the detected event were spinning
- therefore the fact that both stars were black holes becomes more plausible.

1.4 Desiderata of Bayesian probability theory

The foundations of Bayesian probability theory as extended logic can be found already in the work of James Bernoulli, Rev. Thomas Bayes and Pierre Simon Laplace. Unfortunately none of them laid out the rules and context to found the theory on solid bases. Consequently, Bayesian thoery was largely replaced by the so-called "frequentist" approach. Notably, Bayesian theory was kept alive by the work of Sir Harold Jeffreys. During the 20th century, the work of G. Polya, R.T. Cox and E.T.Jaynes laid down the solid foundations that the theory required. Their efforts culminated in a set of "desiderata" for a consistent theory of extended logic which lay directly to the product and sum rules that we will find later. They are called desiderata rather than axiom because they do not assert that anything is true *per se*, but list a set of desirable goals for the theory. The desiderata are:

1. Degrees of plausibility are represented by real numbers
2. Plausibility must be in qualitative agreement with rationality and common sense. This means that when new information supporting the truth of a proposition is available, the degree of plausibility should increase in a continous and monotonic way up to the limit of deductive logic.
3. Consistency:
    - if a conclusion can be reached in more than one way, every possible way must lead to the same result;
    - the theory must account for all relevant information available
    - equivalent states of knowledge must lead to the same degree of plausibility assignment.

In literature, plausibilities are usually indicated as $(A|B)$ which reads "the plausibility of A given B". The probability is then introduced as a map between plausibilities and the set of real numbers between $[0, 1]$.

Since Bayesian probability deals with logical propositions as *information*, it follows that in this theory of probability, there is no absolute probability, all probabilities are conditional on the information at hand. Therefore, even in the same settings different information lead to different probability assignments. For instance, two gamblers are betting on the outcome of a coin toss. They are given two different background informations:

1. I_1: the coin is perfectly fair;
2. I_2: the coin is biased, with the "head" side more probable than the tail.

Before placing their bets, the two gamblers assign probabilities to the two outcomes. Gambler 1 believes the coin is perfectly fair, therefore he assigns equal probabilities to either outcomes by maximising its uncertainty. Gambler 2, on the other hand, knows that "head" is more probable, therefore he will decide to assign some higher probability to the head outcome.

1.5 Probability theory as extended logic

One can show [1, 2] that the above desiderata lead to a unique formulation of probability theory in which the basic rules are the *product* and the *sum* rules. These two rules are all that is needed to manipulate logic propositions and perform calculations with them.

1.6 The basic rules

1.6.1 The product rule

Given three statements A, B and C, the product rule asserts that:

$$p(AB|C) = p(A|BC)p(B|C) \tag{3}$$

and the commutativity of boolean conjunction ensures that

$$p(AB|C) = p(BA|C) = p(B|AC)p(A|C)\,. \tag{4}$$

In the case in which the two propositions A and B are *independent*, e.g. the truth value of A does not depend on B, Eq. (3) reduces to

$$p(AB|C) = p(A|C)p(B|C)\,. \tag{5}$$

For example, consider the case in which two different facilities observe the same event, e.g. the two LIGO detectors observe the same gravitational wave signal. Consider the propositions

- D_H:"LIGO Hanford observed a GW signal at a given time t"
- D_L:"LIGO Livingstone observed a GW signal at a given time $t + \Delta t$"

with Δt equal to the light travel time between Hanford and Livingstone. The propositions D_H and D_H are clearly *logically* independent, the fact that Hanford observed a GW signal does not influence the observation made in Livingstone. At the same time D_H and D_H are *not* causally independent, our understanding of the physics implies that the GW signal observed in both detectors must be the same.

1.6.2 The sum rule

Given three statements A, B and C, the sum rule asserts that:

$$p(A+B|C) = p(A|C) + p(B|C) - p(AB|C)\,. \tag{6}$$

In the case in which the propositions A and B are *logically disjoint* or *mutually exclusive*, e.g. they cannot be true at the same, it reduces to

$$p(A+B|C) = p(A|C) + p(B|C)\,. \tag{7}$$

Examples of mutually exclusive propositions are

- $x \in [0,1]$ and $x \in [2,3]$
- “the GW event was a binary neutron star system” and “the GW event was a binary black hole”.

Furthermore, if $B = \bar{A}$, then A and $\bar{A}$ are also *exhaustive*, therefore

$$p(A|C) + p(\bar{A}|C) = 1\,. \tag{8}$$

1.7 Bayes’ Theorem

We are now in the position of giving a proof to Bayes’ theorem, which is the main (and only) tool available to process information in our theory of probability. Consider three propositions A, B and C and the product rule given in Eq. (3). Thanks to the commutative property, we noted that also Eq. (4), so putting the two together we get:

$$p(AB|C) = p(A|BC)p(B|C) \tag{9}$$
$$p(AB|C) = p(B|AC)p(A|C) \tag{10}$$

therefore

$$p(A|BC) = \frac{p(A|C)p(B|AC)}{p(B|C)}\,. \tag{11}$$

Eq (11) is known as *Bayes’ theorem* and it is the rule according to which the plausibility of propositions change according to new information. In the form given in Eq (11), this character is not very evident. In the context of scientific inference Bayes’ theorem is typically written in terms of *hypotheses* and *data*. Let’s define the following propositions:

- I: the prior information available;
- D: a proposition representing some data we are collecting;
- H_i: a proposition assserting the truth of some hypothesis (or model) we are interested in.

Let's write Bayes' theorem in terms of the propositions defined above:

$$p(H_i|DI) = \frac{p(H_i|I)p(D|H_iI)}{p(D|I)} \tag{12}$$

where each term is given a special name:

- $p(H_i|DI)$ is the posterior probability for H_i;
- $p(H_i|I)$ is the prior probability for H_i;
- $p(D|H_iI)$ is the likelihood function for the data D given H_i. Sometimes this quantity is also referred to as *sampling probability* for D;
- $p(D|I) = \sum_i p(H_i|I)p(D|H_iI)$ is (for the moment) a normalisation factor to ensure that $\sum_i p(H_i|DI) = 1$.

We are going to briefly discuss each of the terms on the right hand side of Eq. (12) later.

1.7.1 Discrete and continuos parameters

Bayes' theorem assigns probabilities to sets of competing hypotheses. This set of hypotheses is sometimes defined as the *hypothesis space*. The hypothesis space can either be a discrete space or a continuous space, depending on the nature of the problem. For instance, when analysing data from an interferometer we are interested in understanding whether the component stars in the observed binary system where spinning or not. In this case we are interested in a discrete space made of the following propositions:

1. H_1: body 1 is spinning;
2. H_2: body 2 is spinning;

and, ultimately, their compound proposition:

$$H = H_1 + H_2\,. \tag{13}$$

Exercise 3: it is a very useful procedure the *reduction in disjunctive normal form.* Reduce the proposition H in disjunctive normal form and then compute its posterior. **Solution**: the disjunctive normal form for H is

$$H = H_1 + H_2 = H_1H_2 + H_1\bar{H}_2 + \bar{H}_1H_1 \tag{14}$$

where all propositions on the right hand side are mutually exclusive. The posterior for H given some data D can then be written as:

$$p(H|DI) = p(H_1H_2|DI) + p(H_1\bar{H}_2|DI) + p(\bar{H}_1H_2|DI) \tag{15}$$

Assume that we know (from our prior information I) that both stars are spinning with spins $\vec{s}_1, \vec{s}_2$. In this case $H = H_1H_2$. For simplicity assume that we are interested only in the magnitude of the spins s_1, s_2 rather that in their orientations. The hypothesis space in this case is a continuous space, some suitably chosen subset of $\mathbb{R} \times \mathbb{R}$. The posterior for s_1, s_2 given some data D is given by $p(s_1 s_2|HDI)ds_1 ds_2$ which is to be interpreted as "the probability that $s_1 \in [s_1, s_1 + ds_1$ and $s_2 \in [s_2, s_2 + ds_2]$, given the data D and H and I." Furthermore, $p(s_1 s_2|HDI)$ is called the *joint* probability density distribution for s_1 and s_2 given the data D and H and I. The probability distribution for s_1 and s_2 is obtained by integration:

$$p(s_1 s_2|HDI) = \int_0^{s_1} ds_1' \int_0^{s_2} ds_2' p(s_1' s_2'|HDI) \tag{16}$$

1.7.2 Marginalisation

Imagine that, somehow, we calculated the joint probability $p(s_1 s_2|HDI)$, but we are not interested in the value of s_2. The posterior for s_1 can be obtaining by *marginalising* over s_2:

$$p(s_1|HDI) = \int \mathrm{d}s_2 \, p(s_1 s_2|HDI) \, . \tag{17}$$

For discrete variables x, y, the integral in Eq. (17) is to be replaced by a sum:

$$p(x|I) = \sum_i p(xy_i|I) \tag{18}$$

1.7.3 The prior probability

The prior probability describes the state of knowledge of an observer *before* having observed the data D. Note that before here is not to be intended in a temporal or causal sense, but only in a logical sense: what is known about the hypothesis H_i in the absence of the data D.

Indifference principle: The simplest way of assigning probabilities dates back to Laplace and it is sometimes referred to as the *indifference principle*: "if among the possible outcomes, there is no reason to prefer any of them over any other, then all outcomes should be equally probable." The indifference principle appeals to common sense, as stated in desideratum 2. Indeed, this is what the gambler with information I_1 at the end of Section 1.4 is appealing to when assigning equal probability to "head" or "tail" outcomes.

Invariance arguments: Assume that we want to estimate the standard deviation σ of a Gaussian distribution, ad we know, by definition, that $\sigma > 0$. Which prior correctly represents our knowledge? We are dealing here with what is called a *scale* parameter. In this case, we can obtain a functional form for $p(\sigma|I)$ by the following argument; we want or probability distribution to be invariant under a scale transformation:

$$p(\sigma|I)\mathrm{d}\sigma = p(\sigma'|I)\mathrm{d}\sigma' \tag{19}$$

and $\sigma' = k\sigma$:

$$p(\sigma|I)\mathrm{d}\sigma = kp(k\sigma|I)\mathrm{d}\sigma \tag{20}$$

which is a functional equation with solution

$$p(\sigma|I) \propto \sigma^{-1} \tag{21}$$

which is an example of a *Jeffreys prior*. It is important to note that $p(\sigma|I) \propto \sigma^{-1}$ implies $p(\log\sigma|I) \propto \mathit{constant}$, therefore we are imposing a uniform prior over the order of magnitude of σ; if no information is available, *any* size of the error is equally probable. The arguments briefly presented here are formalised in [1].

Maximum entropy: A formal principle to assign probabilities is the *maximum entropy principle*. Shannon, in his seminal paper about information theory, introduced the concept of *information entropy* as the measure of uncertainty associated to a given probability distribution. [**?**] and others show that the least informative distribution that obeys some given constraints is the distribution that maximises the information entropy. The information entropy is defined as:

$$H(p) = -\sum_i p_i \log(p_i/m_j) \tag{22}$$

where the m_j are eventual prior probabilities. Note that, if we generalise to the continuum, the entropy becomes

$$H(p) = -\int \mathrm{d}x p(x)\log(p(x)/m(x)) \tag{23}$$

and $m(x)$ would be the Lebesgue measure which ensure invariance of the entropy under changes of variables.

Exercise: Consider the following probability distributions:

$$p_1 \equiv \frac{1}{2}, \frac{1}{2} \tag{24}$$

$$p_2 \equiv \frac{1}{4}, \frac{3}{4} \tag{25}$$

which one is more uncertain? **Solution**: the entropy for p_1 is ~ 0.69, while the entropy for p_2 is ~ 0.56. Therefore p_1 contains the least information.

We will not dwell in the general details of obtaining maximum entropy distributions, but we will examine a few useful cases. Full treatment can be found in [1, 2].

- *Uniform distribution*: assume that the only constraint the probability distribution we sought has to obey is $\sum_{j=1}^{M} p_j = 1$. We want then to find the p_j that obeys the aforementioned constraint and maximising (23). We can do so by using Lagrange multipliers:

$$\mathrm{d}\left[-\sum_j p_j \log(p_j/m_j) - \lambda(\sum_i p_i - 1)\right] = 0 \tag{26}$$

$$\mathrm{d}\left[-\sum_j p_j \log(p_j) + \sum_j p_j \log(m_j) - \lambda(\sum_i p_i - 1)\right] = 0 \tag{27}$$

$$\sum_j \left[-\log(p_j) - p_j\frac{\partial \log(p_j)}{\partial p_j} + \log(m_j) - \lambda\frac{\partial p_j}{\partial p_j}\right]\mathrm{d}p_j = 0 \tag{28}$$

$$\sum(-\log(p_j/m_j) - 1 - \lambda)\mathrm{d}p_j = 0 \tag{29}$$

Thus we have $\forall j$:

$$(-\log(p_j/m_j) - 1 - \lambda = 0 \tag{30}$$

$$\implies p_j = m_j e^{-(\lambda+1)} \tag{31}$$

The constraint $\sum_j p_j = 1$ requires:

$$e^{-(\lambda+1)}\sum_j m_j = 1 = e^{-(\lambda+1)} \implies \lambda = -1 \tag{32}$$

thus $p_j = m_j$. If we have no prior information, then we can invoke the indifferent principle and set $m_j = 1/M$ and therefore p_j is a uniform distribution.

- *Gaussian distribution*: assume that the measure m_j, the prior for p_j has the following form:

$$m_j = \begin{cases} 1/(x_M - x_m) & \text{if } x_m < x_j < x_M \\ 0, \quad \text{otherwise} & \end{cases}. \tag{33}$$

We impose the following constraints:

$$\sum_j p_j = 1 \tag{34}$$

$$\sum_j (x_j - \mu)^2 p_j = \sigma^2 \tag{35}$$

We are looking for the p_j that maximises

$$H(p_j) = -\sum_j p_j \log(p_j) \tag{36}$$

since m_j is a constant. We are looking for a solution to

$$\mathrm{d}\left[-\sum_j p_j \log p_j - \lambda\left(\sum_j p_j - 1\right) - \omega\left(\sum_j (x_j-\mu)^2 p_j - \sigma^2\right)\right] = 0 \tag{37}$$

which leads to

$$\sum_j \left[-\log p_j - 1 - \lambda - \omega(x_j-\mu)^2\right] \mathrm{d}p_j = 0\,. \tag{38}$$

The solution to Eq. (38) is

$$p_j = e^{-\lambda_0} e^{-\omega(x_j-\mu)^2} \quad \text{with} \quad \lambda_0 \equiv 1+\lambda\,. \tag{39}$$

To get the value of the multipliers, we are going to generalise to the continuum. The solution (39), generalises to

$$p(x) = e^{-\lambda_0} e^{-\omega(x-\mu)^2}\,. \tag{40}$$

Let's now impose the normalisation constraint:

$$\int_{x_m}^{x_M} \mathrm{d}x\, p(x) = 1 = e^{-\lambda_0} \int_{x_m}^{x_M} \mathrm{d}x\, e^{-\omega(x-\mu)^2} \tag{41}$$

$$\lambda_0 = \log\left[\frac{\sqrt{\pi}}{2\sqrt{\omega}}\right] + \log\left[\mathrm{erf}(\sqrt{\omega}(x_M-\mu)) - \mathrm{erf}(\sqrt{\omega}(\mu-x_m))\right] \tag{42}$$

If we take $\sqrt{\omega}(x_M-\mu) >> 1$ and $\sqrt{\omega}(\mu-x_m) << 1$, then

$$\mathrm{erf}(\sqrt{\omega}(x_M-\mu)) \to 1 \tag{43}$$

$$\mathrm{erf}(\sqrt{\omega}(\mu-x_m)) \to -1\,, \tag{44}$$

thus, we get

$$\lambda_0 = \log\sqrt{\frac{\pi}{\omega}} \tag{45}$$

which we can substitute in the second constraint to get:

$$\int \mathrm{d}x (x-\mu)^2 \sqrt{\frac{\omega}{\pi}} e^{-\omega(x-\mu)^2} = \sigma^2 \tag{46}$$

$$\sqrt{\frac{\omega}{\pi}} \int \mathrm{d}x (x-\mu)^2 e^{-\omega(x-\mu)^2} = \sqrt{\frac{\omega}{\pi}} \int \mathrm{d}y\, y^2 e^{-\omega y^2} \tag{47}$$

$$\omega = \frac{1}{2\sigma^2} \tag{48}$$

which we then substitute back in Eq. (45) and finally we obtain

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}\,. \tag{49}$$

The Gaussian distribution is the maximum uncertainty distribution for a fixed variance.

1.7.4 The likelihood function

The likelihood function describes the probability of observing the data D assuming that the hypothesis H_i and the prior information I are true. The likelihood function therefore represents the predictions of the hypothesis H. Models in general depend on some parameters θ in which case the likelihood function is written as:

- $p(D|\theta HI) \equiv$"the probability of the data D, given that model H and information I are true *and* the value of the parameters is θ" .

The likelihood function is therefore a function of the model parameters θ. [1] We will follow Section 4.8 in [2]. Define for convenience the following propositions:

- $D = D_1 \dots D_N$: datum $d_i \in [d_i, d_i + \mathrm{d}d_i]$;
- $H = X_1 \dots X_N$: datum d_i is in the range x_i and $x_i + \mathrm{d}x_i$;
- $E = E_1 \dots E_N$: the error value on datum d_i is $e_i \in [e_i, e_i + \mathrm{d}e_i]$.

We can write:

$$d_i = x_i + e_i \,. \tag{50}$$

We can write the probability distribution for the X_i as

$$p(X_i|\theta HI) = f(x_i) \tag{51}$$

and, similarly for the E_i:

$$p(E_i|\theta HI) = g(e_i) \,. \tag{52}$$

Our purpose is to compute $p(D_i|\theta HI)$. We can do that by considering the joint distribution of D_i, E_i, X_i and then marginalising:

$$\begin{aligned} p(D_i|\theta HI) &= \int\int \mathrm{d}X_i \mathrm{d}E_i p(D_i X_i E_i|\theta HI) \\ &= \int\int \mathrm{d}X_i \mathrm{d}E_i p(D_i|X_i E_i \theta HI) p(X_i|\theta HI) p(E_i|\theta HI) \end{aligned} \tag{53}$$

where we assumed the propositions X_i and E_i to be independent. Since $d_i = x_i + e_i$ we have:

$$p(D_i|X_i E_i \theta HI) = \delta(d_i - x_i - e_i) \tag{54}$$

[1] We remind the reader that every time we write $p(x|I)$, x is to be interpreted as the logical proposition:

$$x : x \in [x, x + dx] \,.$$

thus Eq. (53) becomes:

$$\begin{aligned} p(D_i|\theta HI) &= \int \mathrm{d}x_i f(x_i) \int \mathrm{d}e_i g(e_i) \delta(d_i - x_i - e_i) \\ &= \int \mathrm{d}x_i f(x_i) g(d_i - x_i) \end{aligned} \tag{55}$$

We are going to evaluate Eq. (55) for the case of deterministic and probabilistic models.

Deterministic models: in the deterministic case, there is no uncertainty over the predictions of the model H. Given a predicting function $m(x_i;\theta)$,

$$f(x_i) = \delta(x_i - m(x_i;\theta)) \tag{56}$$

and Eq. (55) reduces to

$$p(D_i|\theta HI) = g(d_i - m(x_i;\theta)) \tag{57}$$

and if all errors are independent:

$$p(D|\theta HI) = \prod_{i=1}^{N} g(d_i - m(x_i;\theta))\,. \tag{58}$$

For deterministic models, the likelihood of any datum is simply given by the product of the probabilities of the errors. Thus for any given model, it is the errors distribution that determines the likelihood.

Probabilistic models: In probabilistic models the predictions are uncertain, either because the model prediction includes some statistical noise component or because the independent variable is not known exactly. Since we are not going to touch on this subject, the interested reader is referred to Section 4.8.2 of [2].

1.8 Model selection

A vast branch of scientific inference deals with the following question:

given some data D, some prior information I and two (or more) competing models H_1 and H_2, which one is better explaining the data?

Bayesian inference naturally answers the above question: let's write the posterior probability for both models:

$$p(H_1|DI) = p(H_1|I)\frac{p(D|H_1 I)}{p(D|I)} \tag{59}$$

$$p(H_2|DI) = p(H_2|I)\frac{p(D|H_2 I)}{p(D|I)} \tag{60}$$

with $p(D|I) = \sum_j p(H_j|I)p(D|H_j I)$. Unless the set of models considered is exhaustive $p(H_1|I) + p(H_2|I) = 1$ we are unable to compute the normalisation constant $p(D|I)$ and therefore unable to compute the two posterior probabilities. However, we can circumvent this problem by taking the ratio between the two posterior probabilities:

$$O_{1,2} \equiv \frac{p(H_1|I)}{p(H_2|I)} \frac{p(D|H_1 I)}{p(D|H_2 I)} \,. \tag{61}$$

The quantity $O_{1,2}$ is called the *odds ratio* which is given by the product ot the *prior* odds $p(H_1|I)/p(H_2|I)$ and the ratio of the marginal likelihoods, or *evidences*, $p(D|H_1 I)/p(D|H_2 I)$. This last quantity is sometimes referred to as *Bayes' factor*. If models H_1 and H_2 depend on some parameters θ and λ, the marginal likelihoods are given by:

$$p(D|H_1 I) = \int \mathrm{d}\theta p(\theta|H_1 I) p(D|\theta H_1 I) \tag{62}$$

$$p(D|H_2 I) = \int \mathrm{d}\lambda p(\lambda|H_2 I) p(D|\lambda H_2 I) \,. \tag{63}$$

1.9 Exercises

Exercise 1: we are going to apply Bayes' theorem to simple real world case. Define the following propositions:

- H: I have a disease;
- D: I take some empirical test and it scored positive;
- I: I am equally likely to have the disease or not, $p(H|I) = p(\bar{H}|I) = 1/2$;
- the probability that the test is accurate $p(D|HI) = x$, thus $p(D|\bar{H}I) = 1 - x$.

Calculate the probability that I have the disease. **Solution**: we write Bayes' theorem:

$$p(H|DI) = \frac{p(H|I)p(D|HI)}{p(H|I)p(D|HI) + p(\bar{H}|I)p(D|\bar{H}I)} \tag{64}$$

and substituting:

$$p(H|DI) = \frac{x/2}{x/2 + (1-x)/2} = x \,. \tag{65}$$

Imagine that after some research, I discover that the disease has an incidence on the general population of f. In other words, my new information is:

- I': the incidence of the disease is f thus $p(H|I) = f$ and $p(\bar{H}|I) = 1 - f$.

The posterior for H now becomes:

$$p(H|DI') = \frac{fx}{fx + (1-f)(1-x)} \,. \tag{66}$$

Table 3: Data for the fit.

x_i	y_i
0.0	1.57129490689
0.125	0.914016426729
0.25	2.1243353749
0.375	2.10805830428
0.5	1.66384432878
0.625	1.95268182775
0.75	2.43112267387
0.875	2.40144721746
1.0	3.40462164356

We can generalise the posterior above to N independent tests:

$$p(H|D_1 \dots D_N I') = \frac{f x^N}{f x^N + (1-f)(1-x)^N} \,. \tag{67}$$

If $x = 0.9$ and $f = 10^{-5}$, how many tests do I need to take to be sure at 99.5% to have the disease? We are going to solve this using `python`:

```
import numpy as np
import matplotlib.pyplot as plt

def p(f,x,n):
    num = f*x**n
    den = f*x**n+(1.0-f)*(1.0-x)**n
    return num/den

number_of_tests = range(1,11)
ps = np.array([p(0.00001,0.9,n) for n in number_of_tests])
closest_index = np.abs(ps-0.995).argmin()
print "number␣of␣tests␣needed␣is␣",number_of_tests[closest_index]
plt.plot(number_of_tests,ps)
plt.xlabel("number␣of␣tests")
plt.ylabel("probability␣of␣having␣the␣disease")
plt.show()
```

For our parameters, the answer is 8.

Exercise 2: we are given a set of data plus error bars, Table 3. We try to explain the observed data both using a linear and a quadratic laws. Which one is favored by the data?

Solution: define the following propositions:

- H_1: the datum $y_i = a x_i + b$;
- H_2: the datum $y_i = a x_i^2 + b x_i + c$;

- *I*: the models H_1 and H_2 are equally probable and the distribution of uncertainties is Gaussian with standard deviation σ;
- *D*: the data $d_1, \dots, d_n$.

As usual, we write our data as

$$d_i = y_i + e_i \tag{68}$$

and the distribution of e_i is known to be a Gaussian.

Linear law: the parameters in our model are a and b, the slope and intercept of the "fitting" line. As usual, we write Bayes' theorem:

$$p(ab|DH_1I) = p(ab|H_1I)\frac{p(D|abH_1I)}{\int \mathrm{d}a\mathrm{d}b p(ab|H_1I)p(D|abH_1I)} . \tag{69}$$

We have seen that in deterministic models, the likelihood is defined by the uncertainty probability distribution, therefore:

$$p(D|abH_1I) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{1}{2}\sum_i \frac{(d_i - ax_i - b)^2}{\sigma^2}\right] . \tag{70}$$

We need to specify prior distributions for a and b. We are going to chose independent priors so that $p(ab|H_1I) = p(a|H_1I)p(b|H_1I)$ and set them to be uniform between some a_{min}, a_{max} and b_{min}, b_{max}:

$$p(a|H_1I) = \frac{1}{a_{max} - a_{min}} \tag{71}$$

$$p(b|H_1I) = \frac{1}{b_{max} - b_{min}} \tag{72}$$

$$\tag{73}$$

from which we get the joint posteriors as:

$$p(ab|DH_1I) \propto \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{1}{2}\sum_i \frac{(d_i - ax_i - b)^2}{\sigma^2}\right] . \tag{74}$$

We are, however, interested in the evidence for model H_1

$$p(D|H_1I) = \int_{a_{min}}^{a_{max}} \mathrm{d}a \int_{b_{min}}^{b_{max}} \mathrm{d}b \exp\left[-\frac{1}{2}\sum_i \frac{(d_i - ax_i - b)^2}{\sigma^2}\right] . \tag{75}$$

The integral in Eq. (75) can be computed analytically by completing the square and assuming that the limits of integration are so large to be approximately ±∞. However, we are going to compute it numerically using python, see later section.

Quadratic law: the parameters in our model are a, b and c. Let's write Bayes' theorem:

$$p(abc|DH_1I) = p(abc|H_1I)\frac{p(D|abcH_1I)}{\int \mathrm{d}a\mathrm{d}b\mathrm{d}c p(abc|H_1I)p(D|abcH_1I)} \, . \tag{76}$$

As before, the likelihood is defined by the uncertainty probability distribution:

$$p(D|abcH_1I) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left[-\frac{1}{2}\sum_i \frac{(d_i - ax_i^2 - bx_i - c)^2}{\sigma^2}\right] \, . \tag{77}$$

Choosing uniform prior distributions for a, b and c, we obtain the posterior:

$$p(abc|DH_1I) \propto \frac{1}{\sqrt{2\pi}\sigma}\exp\left[-\frac{1}{2}\sum_i \frac{(d_i - ax_i^2 - bx_i - c)^2}{\sigma^2}\right] \, . \tag{78}$$

We are, however, interested in the evidence for model H_1

$$p(D|H_1I) = \int_{a_{min}}^{a_{max}} \mathrm{d}a \int_{b_{min}}^{b_{max}} \mathrm{d}b \int_{c_{min}}^{c_{max}} \mathrm{d}c \exp\left[-\frac{1}{2}\sum_i \frac{(d_i - ax_i^2 - bx_i - c)^2}{\sigma^2}\right] \, . \tag{79}$$

The integral in Eq. (79) could also be computed analytically under the same approximation as for Eq.(75). However, we are going to compute it numerically using python.

```
import numpy as np
y = np.array([ 0.62165013,   1.02361924,   1.51161683,   0.78429014,
0.76852557,   1.3840013,
        2.98271159,   2.62484856,   2.51702153])
x = np.linspace(0,1,9)
from scipy.integrate import dblquad,tplquad

 sig = 0.5

def linear(x,a,b):
    return a*x+b

def quadratic(x,a,b,c):
    return a*x**2+b*x+c

def likelihood_lin(a,b,x,y):
    num = (y-linear(x,a,b))**2
    return np.exp(-0.5*np.sum(num)/sig**2)

def likelihood_quad(a,b,c,x,y):
    num = (y-quadratic(x,a,b,c))**2
    return np.exp(-0.5*np.sum(num)/sig**2)

a_min = -10.0
a_max = 10.0
b_min = -10.0
b_max = 10.0
c_min = -10.0
c_max = 10.0
result_lin = dblquad(likelihood_lin,
                     b_min, b_max,
                     lambda x:a_min,
                     lambda x:a_max,
                     args=(x,y))
print ”marginal likelihood linear model = ”,result_lin[0]
result_quad = tplquad(likelihood_quad,
                      a_min, a_max,
                      lambda y:b_min, lambda y:b_max,
                      lambda x,y:c_min,
                      lambda x,y:c_max,
                      args=(x,y))
print ”marginal likelihood quadratic model = ”,result_quad[0]
print ”Bayes factor = ”,result_lin[0]/result_quad[0]
```

the odds ratio between H_1 and H_2 for prior odds equal to 1, as specified by the information I is:

$$O_{1,2} \simeq 0.14\,. \tag{80}$$

A Bayesian verification of the Central Limit theorem: We are going to give now a Bayesian demonstration of a fundamental theorem in probability theory, the Central Limit Theorem (CLT). The CLT states:

- given a set of n independent variables that are identically distributed with unknown probability distribution having finite mean μ and finite variance σ^2, then the sample average has a distribution with mean μ and variance σ^2/n that tends to a Gaussian distribution for $n \to \infty$.

. Incidentally, this might be to origin of the name “Normal” which is also used to indicate the Gaussian distribution.

Consider this problem:

- I: a widget is made of 2 components
- Y: the widget has a lenght $\in [y, y + \mathrm{d}y]$
- X_1: the first component has a lenght $\in [x_1, x_1 + \mathrm{d}x_1]$
- X_2: the widget has a lenght $\in [x_2, x_2 + \mathrm{d}x_2]$

We know that

$$p(X_1|I) = f_1(x_1) \tag{81}$$

$$p(X_2|I) = f_2(x_2) \tag{82}$$

and we want to calculate $p(Y|I)$. Consider the joint probability for Y, X_1, X_2:

$$p(Y|I) = \int\int \mathrm{d}X_1 \mathrm{d}X_2 p(Y X_1 X_2|I) = \int\int \mathrm{d}X_1 \mathrm{d}X_2 p(Y|X_1 X_2 I) p(X_1|I) p(X_2|I) \tag{83}$$

from the definition of the problem, we have:

$$p(Y|X_1, X_2 I) = \delta(y - x_1 - x_2) \tag{84}$$

so

$$p(Y|I) = \int\int \mathrm{d}x_1 \mathrm{d}x_2 f_1(x_1) f_2(x_2) \delta(y - x_1 - x_2) = \int \mathrm{d}x_1 f_1(x_1) f_2(y - x_1) \tag{85}$$

which is a convolution integral. We can extend the treatment to the case in which the widget is made of three components introducing

- Z: the widget has a lenght $\in [z, z + \mathrm{d}z]$.

$$p(Z|I) = \int\int\int \mathrm{d}X_1 \mathrm{d}X_2 \mathrm{d}X_3 p(Z X_1 X_2 X_3|I) \tag{86}$$

$$= \int\int \mathrm{d}Y \mathrm{d}X_3 p(X_3|I) p(Z|Y X_3|I) p(Y|I) \tag{87}$$

$$\implies p(Z|I) = \int dy f(y) f_3(z - y) \tag{88}$$

Exercise : write a small script to reproduce Figure 1. **Solution :**

```
import numpy as np
from scipy.signal import fftconvolve
import matplotlib.pyplot as plt

def constant(x):
    ret = np.zeros(len(x))
    for i in xrange(len(x)):
        if -1<x[i]<1:
            ret[i]=1.
    return ret

def normal(x,mu,sigma):
    return np.exp(-0.5*((x-mu)/sigma)**2)/np.sqrt(2.*np.pi*sigma*sigma)
```

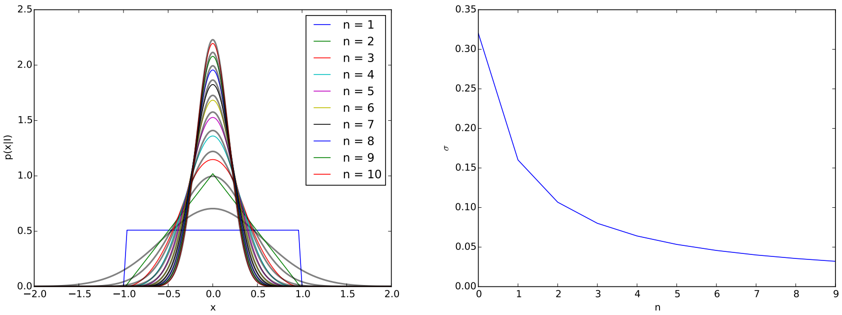

Figure 1: Left: probability distribution obtained from the convolution of $n \in [2, 10]$ uniform distributions. In black the Gaussian distributions having same mean and variances as the resulting distribution from the convolution. Already after 4 convolutions, the Gaussian is a very good approximation to the actual distribution. Right: scaling of the variance as a function of n. The variance follows the expectation of $1/n$.

```
delta = 0.1
x = np.linspace(-5,5,256)#np.arange(10)*delta-3.0
delta = np.diff(x)[0]
c = constant(x)
c/=(c*delta).sum()
plt.plot(x,c,label=”n = 1”)
mu = []
var =[]
mu.append( np.sum(x*c*np.diff(x)[0]))
var.append( np.sqrt(np.sum(c*np.diff(x)[0]*(x-mu[-1])**2)))
plt.plot(x,normal(x,mu[-1],var[-1]),
          color='k',linewidth = 2.0,
          alpha = 0.5)
for j in xrange(1,10):
    c = np.convolve(c,constant(x),'full')
    xp = np.linspace(-5,5,len(c))
    c/=(c*np.diff(xp)[0]).sum()
    plt.plot(xp,c,label=”n = %d”%(j+1))
    mu.append( np.sum(xp*c*np.diff(xp)[0]))
    var.append( np.sqrt(np.sum(c*np.diff(xp)[0]*(xp-mu[-1])**2)))
    plt.plot(xp,normal(xp,mu[-1],var[-1]),
              color='k',linewidth = 2.0, alpha = 0.5)
plt.legend()
plt.xlabel('x')
plt.ylabel('p(x|I)')
plt.xlim(-2,2)
plt.savefig('clt.pdf',bbox_inches='tight')
plt.clf()
plt.plot(np.array(var)**2)
plt.xlabel('n')
plt.ylabel(r'$\sigma$')
plt.savefig('clt_sigma.pdf',bbox_inches='tight')
```

2 Numerical methods

The recent years emergence and success of Bayesian methods has been fueled also by the advances in computational techniques. The computation of posteriors and evidences requires the evaluation of multi-dimensional integrals which are untreatable without special care. As we will see in Section 3, the analysis of gravitational wave signals from the coalescence of compact binary systems requires integrating functions in 9 to 15 dimensions for general relativity models that include a minimal amount of physics. The dimensionality in more advanced models, e.g. including the effect of matter or putative violations of general relativity, can increase substantially. These problems can be tackled and solved using Monte Carlo techniques, and in particular Markhov Chain Monte Carlo (MCMC) methods such as the Metropolis-Hastings algorithm.

2.1 Metropolis-Hastings algorithm

Assume we can write the joint posterior density for a set of parameters x, $p(x|DI)$ and we are interested in computing the expectation value of some function $f(x)$ over $p(x|DI)$. The expectation value is defined as

$$E[f(x)] =< f(x) >= \int_V \mathrm{d}x f(x) p(x|DI) \equiv \int_V \mathrm{d}x g(x) \tag{89}$$

where V is the volume of the parameter space defined by x. For instance, in the one dimensional case

- mean: $\mu = \int \mathrm{d}x x p(x|DI)$;
- variance: $\sigma^2 = \int \mathrm{d}x (x-\mu)^2 p(x|DI)$.

When the dimensionality of the parameter space is large, computing the integrals necessary for expectation values is an extremely challenging task. This is the subject matter of Monte Carlo integration. In its most basic variant, the procedure is to pick n random points uniformly distributed in the volume V

estimate

$$\int \mathrm{d}x g(x) \approx V\times < g(x) > \pm V \times \sqrt{\frac{< g^2(x) > - < g(x) >^2}{n}} \tag{90}$$

$$< g(x) > = \frac{1}{n}\sum_j g(x_j) \tag{91}$$

$$< g^2(x) > = \frac{1}{n}\sum_j g^2(x_j)\,. \tag{92}$$

It is clear that the naive Monte Carlo procedure outlined above is bound to fail for large dimensional spaces. First of all, the error in the integral decreases only as $1/n$, but most importantly the efficiency of the algorithm decreases exponentionally with the dimensionality. Thus, the key to the solution of this problem is to be able to produce samples from the *target* density $p(x|DI)$ in an efficient way.

The idea of MCMC algorithms is to replace the uniform sampling in the volume V with a random walk in V in such a way that the walk moves across V following $p(x|DI)$. The random walk is achieved via some *transition* probability $q(y|x_t)$ that governs whether a given move is accepted or not. In a nutshell the Metropolis-Hastings algorithm is:

- initialise x_0 randomly in V;
- while $t < N$:
  - generate y from $q(y|x_t)$;
  - compute the *acceptance probability* $a(x_t, y)$

    $$a(x_t, y) = \min\left(1, \frac{p(x|DI)}{p(x|DI)}\frac{q(x_t|y)}{q(y|x_t)}\right); \tag{93}$$

  - sample u from a uniform distribution $\in [0, 1]$;
  - if $u \leq a(x_t, y)$ then $x_{t+1} = y$;
  - else $x_{t+1} = x_t$ and $t = t + 1$

Exercise 3: write a Metropolis-Hastings algorithm in `python` to generate samples from standard Gaussian distribution.

Solution:

```
import numpy as np
import matplotlib.pyplot as plt

def standard_normal(z):
    return np.exp(-z*z/2.)/np.sqrt(2*np.pi)

n = 10000
```

```
alpha = 1
x = np.random.uniform(-1,1)
samples = []
samples.append(x)
# generate n random updates choosing a uniform transition probability
# between -alpha and alpha
updates = np.random.uniform(-alpha,alpha,size=n)
for i in xrange(1,n):
    y = x + updates[i]
    #acceptance probability, the transition probability simplifies
    aprob = min([1.,standard_normal(y)/standard_normal(x)])
    u = np.random.uniform(0,1)
    if u < aprob:
        x = y
        samples.append(x)

#plotting the results:
#theoretical curve
x = np.linspace(-3,3,100)
y = standard_normal(x)
myfig = plt.figure(1)
ax = myfig.add_subplot(211)
ax.set_title('Metropolis-Hastings')
ax.plot(samples)
ax = myfig.add_subplot(212)

ax.hist(samples, bins=30,normed=1)
ax.plot(x,y,'r')
ax.set_ylabel('Frequency')
ax.set_xlabel('x')
ax.legend(('PDF','Samples'))
plt.show()
```

3 Gravitational waves data analysis

The problem of measuring the parameters of a gravitational wave signal from the coalescence of a compact binary system can be summerised as follows. Under the hypothesis that there is a signal embedded in the noise, the detector data stream $d(t)$ is

$$d(t) = n(t) + h(t;\theta) \tag{94}$$

where $n(t)$ is the noise time series and $h(t,\theta)$ is the gravitational wave signal which depends on a set of parameters θ, properly coupled to the detector tensor

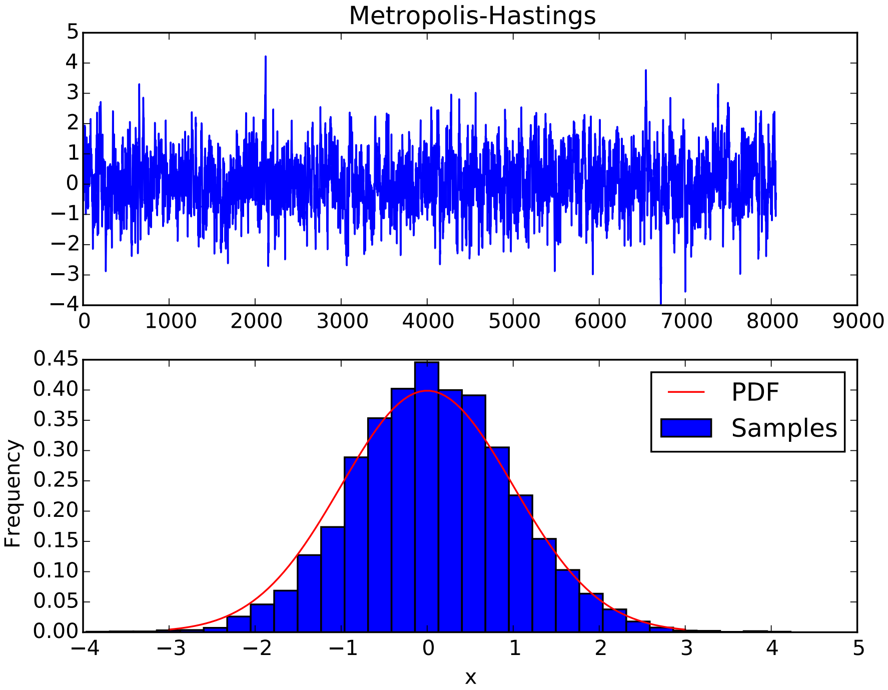

Figure 2: Top: Markhov chain. Bottom: 5000 samples from the target distribution, in red the true posterior.

in the frame of the detector:

$$h(t,\theta) = F_+ h_+(t,\theta) + F_\times h_\times(t,\theta) \tag{95}$$

and the functions F_+ and $F_\times$ are the *antenna pattern* functions. Our purpose is to compute the joint posterior distribution for θ under the assumptions made by model H. As usual, we write Bayes' theorem:

$$p(\theta|DHI) = p(\theta|HI)\frac{p(D|\theta HI)}{\int \mathrm{d}\theta p(\theta|HI)p(D|\theta HI)}\,. \tag{96}$$

The usual assumption is that the waveform predicted by model H is known exactly, therefore we are under the conditions described in Section 1.7.4 and the likelihood for the data D is uniquely set by the distribution of the noise.

3.1 The noise model

The output of a complex system as a ground based laser interferometer is a very complex function of all its components. At any given instant, what the sensors register is a superposition of many independent sources of noise: for instance thermal noise from the mirror, seismic noise, thermal noise from the suspensions of the mirror, laser frequency noise, laser shot noise, etc. However, we are not interested in the details of how each process is contributing to the output, all we care about is understanding the statistical properties of this incoherent superposition and understand the average output as well as the fluctuations around it. In this way, one is able to flag some extreme output, such as the presence of a coherent gravitational wave, as an extremely unlikely noise event. Thus, we are looking for a probability distribution $p(n|I)$ which maximises our uncertainty. We have already noted in Section **??** that given some contraints, the distribution that maximise our state of uncertainty must have maximal entropy. Therefore, what constaints can we put on $p(n|I)$?

The following treatment follows closely [3]. Consider now a time series $n(t)$ and its samples $n_1, \ldots, n_k$ taken at equally spaced times t_i, $t_i = i\Delta t$. The time series n_i can be equivalently expressed in Fourier series:

$$n_i = \frac{1}{\sqrt{k\Delta t}}\sum_{j=0}^{k/2} a_j\cos(2\pi f_j t_i) + b_j \sin(2\pi f_j t_i) \quad \text{with} \quad f_j = \frac{j}{N\Delta t} = j\Delta f \tag{97}$$

where, by definition $b_0 = b_{k/2} = 0$ and we assumed k to be even. The number of non-zero elements in the n_is and in the a_j and b_js is the same, and the coefficients are obtained by a discrete Fourier transform:

$$a_j = \frac{2}{k\Delta t}\sum_i n_i \cos(2\pi f_j t_i) \tag{98}$$

$$b_j = \frac{2}{k\Delta t}\sum_i n_i \sin(2\pi f_j t_i)\,. \tag{99}$$

The trigonometric functions in Eq. (97) are an orthonormal basis in the sample space. Moreover, the expression in (97) could be written in terms of an amplitude and a phase rather than in terms of the two amplitudes a_j and b_j:

$$n_i = \frac{1}{\sqrt{k\Delta t}} \sum_{j=0}^{k/2} \lambda_j \sin(2\pi f_j t_i + \phi_j) \tag{100}$$

and:

$$\lambda_j = \sqrt{a_j^2 + b_j^2} \tag{101}$$

$$\phi_j = \begin{cases} \arctan(b_j/a_j) & \text{if } a_j > 0 \\ \arctan(b_j/a_j) \pm \pi & \text{if } a_j < 0 \end{cases} \tag{102}$$

Define now the function of the Fourier frequencies of the time series $s(f_j)$ as

$$s(f_j) = a_j^2 + b_j^2 = \frac{\Delta t}{k} |\tilde{n}(f)|^2 \tag{103}$$

which is the discrete analog of the *one-sided power spectral density.* The function $\tilde{n}(f)$ is the discrete Fourier transform of $n(t)$.[2] The variables a_j^2, b_j^2, as well as n_i $\forall i, j$, are unknown. They are therefore described by some (unknown) probability distribution. If we assume that $p(n_i|I)$ $\forall i$ is a zero mean distribution, thanks to Eq. (103), necessarily the a_j^2, b_j^2 and $s(f_j)$ are described by some zero mean distribution. If we compute the expectation value[3] of Eq. (103) over the unknown distribution $p(n_i|I)$, we find that

$$E[s(f_j)] = E[a_j^2 + b_j^2] = E[a_j^2] + E[b_j^2] = \sigma_{a_j}^2 + \sigma_{b_j}^2 \equiv \sigma_j^2 \,. \tag{107}$$

The expectation value of the (one sided) power spectral density is equal to the variance of the Fourier components of the noise. If we assume that σ_j^2 is known and that it is the only constraint that our probability distribution needs to obey, the maximum entropy principle, see Section **??**, dictates that the probability

[2] We use a definition of the discrete Fourier transform (DFT) of a function $h(t)$ as:

$$\tilde{h}(f) = \sum_{j=0}^{N-1} h(j\Delta t) \exp -2\pi\rangle j\Delta t f \tag{104}$$

and of the inverse DFT:

$$h(t) = \frac{1}{N} \sum_{j=0}^{N-1} \tilde{h}(f_j) \exp 2\pi\rangle f_j t \tag{105}$$

[3] We remind the reader that the expectation value of a function g(x) over some distribution p(x) is defined as

$$\int \mathrm{d}x g(x) p(x) \equiv E[g(x)] \,. \tag{106}$$

, see Eq. (89)

distribution for the noise $\tilde{n}_j$ is a Gaussian distribution with variance equal to σ_j:

$$p(n_j|I) = \frac{1}{\sqrt{2\pi}\sigma_j} \exp\left[-\frac{1}{2}\frac{\tilde{n}_j^2}{\sigma_j^2}\right] . \tag{108}$$

If we assume that the Fourier components $a_j, b_j \ \forall j$ are statistically independent, which is true for stationary processes, we can finally write the joint probability for all noise samples as:

$$p(\tilde{n}_1 \dots \tilde{n}_k|I) = \frac{1}{(2\pi)^{k/2}\prod_{j=1}^k \sigma_j} \exp\left[-\frac{1}{2}\sum_{j=1}^k \frac{\tilde{n}_j^2}{\sigma_j^2}\right] . \tag{109}$$

Thanks to the linearity of the Fourier transform, Eq. (94) holds also in the frequency domain:

$$\tilde{d}(f) = \tilde{n}(f) + \tilde{h}(f, \theta) \tag{110}$$

We are therefore in the position of writing the likelihood for the data $\tilde{d}_1 \dots \tilde{d}_k$:

$$p(\tilde{d}_1 \dots \tilde{d}_k|\theta H I) = \frac{1}{(2\pi)^{k/2}\prod_{j=1}^k \sigma_j} \exp\left[-\frac{1}{2}\sum_{j=1}^k \frac{(\tilde{d}_j - \tilde{h}_j(\theta))^2}{\sigma_j^2}\right] \tag{111}$$

and, as a function or the power spectral density:

$$p(\tilde{d}_1 \dots \tilde{d}_k|\theta H I) = \exp\left[-2\Delta f \sum_{j=1}^k \frac{(\tilde{d}_j - \tilde{h}_j(\theta))^2}{S(f_j)}\right] \tag{112}$$

where the normalisation constant, which is irrelevant, has been dropped.

3.2 The signal model

Before we can compute the posterior for θ, we need to specify what our model for $\tilde{h}(f;\theta)$ is and the prior probability on the parameters θ. We are going to start by considering a frequency domain analytical model for $\tilde{h}(f;\theta)$, the inspiral only non-spinning `TaylorF2` model [4]:

$$\tilde{h}(f) = A(f)e^{i\Psi(f)} \tag{113}$$

$$A(f) \propto \frac{\mathcal{M}^{5/6} f^{-7/6}}{D_L} \tag{114}$$

$$\Psi(f) = 2\pi f t_c - \phi_c - \frac{\pi}{4} + \sum_{j=0}^{7}\left[\psi_j + \psi_j^{(l)} \ln f\right] f^{(j-5)/3}, \tag{115}$$

and the *post-Newtonian* coefficients ψ_j are given functions of the chirp mass $\mathcal{M}$ and q, which are defined in terms of the component masses m_1 and m_2 as:

$$\mathcal{M} = \frac{(m_1 m_2)^{3/5}}{m_1 + m_2)^{1/5}} \tag{116}$$

$$q = \frac{m_2}{m_1} \,. \tag{117}$$

The set of parameters θ that are necessary to fully describe our gravitational wave signal is:

- extrinsic parameters: sky position coordinates α and δ, luminosity distance D_L, polarisation angle ψ and inclination angle ι, time at coalescence t_c and phase at coalescence ϕ_c;
- intrinsic parameters: chirp mass $\mathcal{M}$ and mass ratio q.

We have seen that the quantity that enters the detector is *not* the gravitational wave signal in (113), but

$$\tilde{h}(f, \theta) = F_+ \tilde{h}_+(f) + F_\times \tilde{h}_\times(f) \tag{118}$$

where the functions F_+ and $F_\times$ are the antenna pattern functions (see the notes from Sturani).

3.2.1 Priors

Let's define prior probabilities for the parameters of interest for the `TaylorF2` model. The joint prior distribution for all parameters of interest factorises as

$$\begin{aligned} &p(\mathcal{M}\eta t_c \phi_c \iota \alpha \delta D_L \psi | HI) = \\ &\qquad = p(t_c|HI) p(\phi_c|HI) p(\iota\psi|HI) p(\alpha\delta D_L|HI) p(\mathcal{M}\eta|HI) \,. \end{aligned} \tag{119}$$

Prior on time and phase at coalescence t_c and ϕ_c: since there is not reason to prefer any specific values of t_c and ϕ_c, $p(t_c|HI)$ and $p(\phi_c|HI)$ are set by the indifference principle:

$$p(t_c|HI) = \frac{1}{t_{max} - t_{min}} \tag{120}$$

$$p(\phi_c|HI) = \frac{1}{2\pi} \,. \tag{121}$$

For practical reasons, however, the range of allowed t_cs is chosen to be a predetermined width around the putative time of detection.

Prior on right ascension α, declination δ and luminosity distance D_L: $p(\alpha\delta D_L|HI)$ is set by the requirement that the number density of sources in the Universe is constant[4]. The total number of sources N within a given volume is given by:

$$\int_0^{V_{max}} n(V)\mathrm{d}V = N\,. \tag{122}$$

where $n(V) \equiv \frac{\mathrm{d}N}{\mathrm{d}V}$. $n(V) = n_0$ constant because of homogeneity. Thus, the probability of finding a source within $[V, V + dV]$ is proportional only to dV. Expanding the volume element in spherical coordinates:

$$dV = D_L^2 \cos(\delta)\mathrm{d}D_L\mathrm{d}\delta\mathrm{d}\alpha \tag{123}$$

finally

$$p(\alpha\delta D_L|HI) \propto D_L^2 \cos(\delta)\,. \tag{124}$$

Prior on inclination ι and polarisation ψ: the angles ι and ψ define the orientation of the orbital plane of the binary with respect of the line of sight. We can repeat a similar argument as for the volume element and obtain

$$p(\iota\psi|HI) \propto \cos(\iota)\,. \tag{125}$$

Prior on chirp mass $\mathcal{M}$ and mass ratio q: $p(\mathcal{M}q|HI)$ is set by requiring that all masses for the two stars in the binary system are equally likely. In other words

$$p(m_1 m_2|HI) \propto 1\,. \tag{126}$$

We can transform into $\mathcal{M}$ and q as follows; we know that all probability distributions are positive definite and add up to one:

$$\int \mathrm{d}m_1\mathrm{d}m_2 p(m_1 m_2|HI) = \int \mathrm{d}\mathcal{M}\mathrm{d}q p(\mathcal{M}\mathrm{d}q|HI) = 1 \tag{127}$$

therefore, the integrals are monotonic functions of their integrands. This implies that the integrands themselves must be equal:

$$\mathrm{d}m_1\mathrm{d}m_2 p(m_1 m_2|HI) = \mathrm{d}\mathcal{M}\mathrm{d}q p(\mathcal{M}\mathrm{d}q|HI)\,. \tag{128}$$

The required probability distribution $p(\mathcal{M}\mathrm{d}q|HI)$ is then equal to

$$p(\mathcal{M}q|HI) = p(m_1 m_2|HI)||J(m_1, m_2; \mathcal{M}, q)||\,. \tag{129}$$

where $J(m_1, m_2; \mathcal{M}, q)$ is the Jacobian matrix for the transformation

$$m_1, m_2 \rightarrow \mathcal{M}(m_1, m_2), q(m_1, m_2)\,. \tag{130}$$

[4] In this section we are going to neglect the fact that our Universe is not Euclidian but rather described by a Friedmann-Robertson-Walker-LeMaitre metric. For sufficiently small redshifts this is not a bad approximation.

In particular, given the definitions in (116), we have

$$m_1(\mathcal{M}, q) = \mathcal{M}(1+q)^{1/5} q^{-3/5} \tag{131}$$

$$m_2(\mathcal{M}, q) = \mathcal{M}(1+q)^{1/5} q^{2/5} \tag{132}$$

after taking derivatives and some algebra, we find:

$$\frac{\partial m_1}{\partial \mathcal{M}} = (1+q)^{1/5} q^{-3/5} \tag{133}$$

$$\frac{\partial m_1}{\partial q} = \frac{\mathcal{M}}{5} \left[q^{-3/5}(1+q)^{-4/5} - 3(1+q)^{1/5} q^{-8/5})) \right] . \tag{134}$$

The derivatives of m_2 are equal to the ones of m_1. Finally, the Jacobian, and therefore our density for $\mathcal{M}$ and q is given by

$$||J|| = p(\mathcal{M}q|I) = |\frac{\partial m_1}{\partial \mathcal{M}} \frac{\partial m_2}{\partial \mathcal{M}} - \frac{\partial m_1}{\partial q} \frac{\partial m_2}{\partial q}| \tag{135}$$

4 Hand on examples

We are now in the position of taking over some examples. Let's start by generating a sample waveform using the `lalsimulation` library:

```
import lalsimulation as lalsim
import numpy as np
import matplotlib.pyplot as plt

srate = 4096
T = 32
df = 1./srate
m1 = 5.0
m2 = 4.0
f_isco = 1.0/((6.**1.5) *np.pi*(m1+m2)*lalsim.lal.MTSUN_SI)
m1 *=lalsim.lal.MSUN_SI
m2 *=lalsim.lal.MSUN_SI
distance = 130.9717e6 * lalsim.lal.PC_SI
f_low=40.0
f_ref=100.0
wave_flags = None
non_GR_params = None
spin1x=0.0
spin1y=-0.0
spin1z=0.0
spin2x=0.0
spin2y=0.0
spin2z=-0.0
amp_order=0
```

```
phase_order=7
iota=np.pi/3.
approx = lalsim.TaylorF2
phase=2.725808
hp,hc = lalsim.SimInspiralChooseFDWaveform(phase,
                                           df,
                                           m1, m2,
                                           spin1x, spin1y, spin1z,
                                           spin2x, spin2y, spin2z,
                                           f_low, f_isco, f_ref,
                                           distance,
                                           iota,
                                           0.0, 0.0,
                                           wave_flags, non_GR_params,
                                           amp_order, phase_order,
                                           approx)
h1 = np.trim_zeros(hp.data.data)
freq = np.linspace(f_low,f_isco,len(h1))
plt.plot(freq,h1,label=”TaylorF2”,alpha=0.5)

plt.xlabel(”frequency”)
plt.ylabel(”h(f)”)
plt.show()
```

Let’s then look at what a typical noise stream looks like; load the file “chirp-gaussian-noise.txt” and plot its contents:

```
import numpy as np
import matplotlib.pyplot as plt

data = np.loadtxt('chirp_gaussian_noise.txt')
fig = plt.figure()
plt.plot(data[:,0],data[:,1],label=”real”)
plt.plot(data[:,0],data[:,2],label=”imaginary”)
plt.xlabel(”$frequency$␣$[Hz]$”)
plt.ylabel(”$\sqrt{Hz^{-1}}$”)
plt.legend()
plt.show()
```

4.1 Measuring the physical parameters of a source

We are now going to use built-in `lal` functions to compute likelihood and estimate the masses of a binary black hole:

```
from pylab import *
import lalsimulation as lalsim
import lal
```

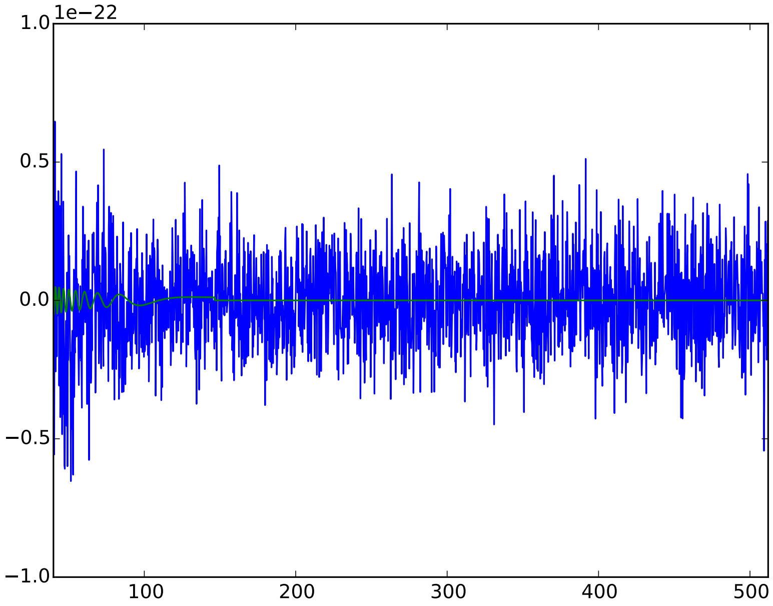

Figure 3: Simulated frequency domain noise stream with a simulated GW signal giving an SNR ~ 10 superimposed.

```
from lal.lal import StrainUnit
from lal.lal import CreateCOMPLEX16FrequencySeries,DimensionlessUnit
from lal.lal import LIGOTimeGPS
from pylal import antenna as ant

def likelihood(m1,m2):
    srate=1024.
    seglen=16.0
    length=srate*seglen
    deltaT=1/srate
    deltaF = 1.0 / (length* deltaT)
    f_max = srate/2.
    f_ref = 100.0
    REAL8time=900000000
    GPStime=LIGOTimeGPS(REAL8time)
    M1=m1
    M2=m2
    D=3e2
    m1=M1*lal.MSUN_SI
    m2=M2*lal.MSUN_SI
    phiRef=0.0

    f_min = 40.0
    s1x = 0.0
    s1y = 0.0
    s1z = 0.0
    s2x = 0.0
    s2y = 0.0
    s2z = 0.0

    r=D*lal.PC_SI*1.0e6
    iota=np.pi/3.0

    lambda1=0
    lambda2=0
    waveFlags=None
    nonGRparams=None

    injapproximant=lalsim.TaylorF2
    amplitudeO=int(0)
    phaseO=4

    ra=0.0
    dec=0.0
    psi=0.0
    segStart=100000000
```

```
strainF= CreateCOMPLEX16FrequencySeries(”strainF”,segStart,
                                        0.0,
                                        deltaF,
                                        DimensionlessUnit,
                                        int(length/2. +1));
[plus,cross]=lalsim.SimInspiralChooseFDWaveform(phiRef,
                                                deltaF,
                                                m1,
                                                m2,
                                                s1x,
                                                s1y,
                                                s1z,
                                                s2x,
                                                s2y,
                                                s2z,
                                                f_min,
                                                f_max,
                                                f_ref,
                                                r,
                                                iota,
                                                lambda1,
                                                lambda2,
                                                waveFlags,
                                                nonGRparams,
                                                amplitudeO,
                                                phaseO,
                                                injapproximant)
ifos=[’H1’]
for ifo in ifos:
    (fp,fc,fa,qv)=ant.response(REAL8time,ra,dec,iota,psi,’radians’,ifo)

for k in np.arange(strainF.data.length):
    if k<plus.data.length:
        strainF.data.data[k]=((fp*plus.data.data[k]+fc*cross.data.data[k]))
    else:
        strainF.data.data[k]=0.0
# copy in the dictionary
inj_strains=np.array([strainF.data.data[k] for k in arange(int(strainF.data.
frequency = np.array([strainF.f0+ k*strainF.deltaF for k in np.arange(int(st
N = len(frequency)
chisq = np.zeros(N)
for i,fi in enumerate(frequency):
    if fi>f_min:
        chisq[i] = (2.0/(deltaT*N))*(np.real(strainF.data.data[i])*
                                     data[i,2]+
                                     np.imag(strainF.data.data[i])*
```

```
                                         data[i,1])
                                        /(1.35e-50*(0.5*AdvLIGOPSD(fi)))
    return np.sum(chisq)

def AdvLIGOPSD(f):
    x = f/215.;
    x2 = x*x;
    f10=f/10.0;
    f50=f/50.0;
    f100=f/100.0;
    f200=f/200.0;
    f300=f/300.0;
    f1000=f/1000.0;
    f2000=f/2000.0;
    x1=f10**30.
    x2=f50*f50*f50*f50*f50*f50;
    psd = f*((60000.0/x1)+5.0/x2+
             1.07*pow(f100,-3.25)+
             3.7*pow(f200,-1.25)+
             0.9*pow(f300,-0.08)+
             0.85*pow(f1000,0.8)+
             0.35*f2000*f2000*f2000);
    return  psd

data = np.loadtxt('chirp_gaussian_noise.txt')
n = 10000
alpha = 0.1
m1,m2 = np.random.uniform(7.5,8.5),np.random.uniform(6.5,7.5)
samples = []
if m2 > m1:
    tmp = m1
    m1 = m2
    m2 = tmp
samples.append([m1,m2])
# generate n random updates choosing a uniform transition probability
# between -alpha and alpha
updates = [np.random.uniform(-alpha,alpha,size=n),
           np.random.uniform(-alpha,alpha,size=n)]
logl0 = likelihood(m1,m2)
for i in xrange(1,n):
    m1_p,m2_p = m1 + updates[0][i],m2 + updates[1][i]
    if m2_p > m1_p:
        tmp = m1_p
        m1_p = m2_p
        m2_p = tmp
    if 7.5<m1_p<8.5 and 6.5<m2_p<7.5:
        #acceptance probability, the transition probability simplifies
        loglnew = likelihood(m1_p,m2_p)
        aprob = loglnew-logl0
        u = np.log(np.random.uniform(0,1))
        if u < aprob:
            m1,m2 = m1_p,m2_p
            samples.append([m1_p,m2_p])
            print "n:",i,"logL␣:",logl0,"-->",loglnew,"m1:",m1,"m2:",m2
            logl0 = loglnew
#plotting the results:
#theoretical curve
nsamps = len(samples)
print "acceptance␣=",len(samples)/float(n)
samples = np.array(samples)
x1 = np.linspace(3,7,100)
myfig = plt.figure(1)
ax = myfig.add_subplot(311)
ax.set_title('Metropolis-Hastings')
ax.plot(samples[nsamps/2:,0])
ax.plot(samples[nsamps/2:,1])
ax = myfig.add_subplot(312)

ax.hist(samples[:,0], bins=30,normed=1)
ax.set_ylabel('Frequency')
ax.set_xlabel('m1')
ax.axvline(8.0,color='r')
ax = myfig.add_subplot(313)

ax.hist(samples[:,1], bins=30,normed=1)
ax.set_ylabel('Frequency')
ax.set_xlabel('m2')
ax.axvline(7.0,color='r')
plt.savefig('mcmc_chirp.pdf',bbox_inches='tight')
```

5 Hierarchical modeling

In this section we are to briefly introduce the concept of hierarchical modeling which is used to infere population parameters from the observation of a set of single events. For instance, consider the case in which the observed events $D = d_1 \dots d_n$ are sampled for a given population but each event parameters do not depend on the parameters of the population. Define λ the parameters of the population and $\theta_1, \dots, n$ the parameters describing each single event. We are interested in

$$p(\lambda|DHI) = p(\lambda|HI)\frac{p(D|\lambda HI)}{p(D|HI)}\,. \tag{136}$$

However, we never observe directly the parameters λ, but only the parameters relative to the single event. Thus, we can extend the conversation to $\theta_1, \ldots, \theta_n$ and then marginalise them away, to obtain:

$$p(\lambda|DHI) = p(\lambda|HI)\frac{\int \mathrm{d}\theta_1 \ldots \mathrm{d}\theta_n p(\theta_1 \ldots \theta_n|\lambda HI)p(D|\theta_1 \ldots \theta_n \lambda HI)}{p(D|HI)}\,. \tag{137}$$

Let's concentrate on the integrand. If we assume that the events are statistically independent, we can write

$$p(\theta_1 \ldots \theta_n|\lambda HI)p(D|\theta_1 \ldots \theta_n \lambda HI) = \prod_i p(\theta_i|\lambda HI)p(d_i|\theta_i \lambda HI) \tag{138}$$

therefore, we obtain the posterior for λ as

$$p(\lambda|DHI) = p(\lambda|HI)\frac{\int \mathrm{d}\theta_1 \ldots \mathrm{d}\theta_n \prod_i p(\theta_i|\lambda HI)p(d_i|\theta_i \lambda HI)}{p(D|HI)}\,. \tag{139}$$

The parameters λ are sometimes called *hyper parameters*. However, the above nomenclature can be misleading since it can lead to the thought that there is something special to them which distinguishes them from the standard concept of parameter.

5.1 A worked example: measuring the cosmological parameters from GW observations

We have seen that the luminosity distance D_L is directly measurable from GW signals. This property guarantees that GW are self-calibrating sources. Thanks to this property, and in analogy to standard candles as supernovae type IA, they are sometimes deemed as *standard sires*. The key ingredients for the construction of an Hubble diagram are the measurements of D_L and of the redshift z since, in a Friedmann-Robertson-Walker-Lemáitre cosmology they are related via the luminosity distance-redshift relation: $D_L \equiv D_L(z, \Omega)$, where Ω are the set of cosmological parameters. Unfortunately, GW in general cannot provide a measurement of the redshift which has to be obtained independently, for instance via spectoscopy on the host galaxy or on the eventual optical counterpart. In what follows, we are going to assume that such a measurement is somehow available. Let's specify Eq. (139) to our specific case by writing it down for just one event. Since the only relevant paramenters are D_L and z, we can perform most of the integrals and get

$$p(\Omega|DHI) = p(\Omega|HI)\frac{\int \mathrm{d}D_L \mathrm{d}z p(D_L z|\Omega HI)p(d|D_L z \Omega HI)}{p(D|HI)}\,. \tag{140}$$

Further conditioning, we get to

$$p(\Omega|DHI) = p(\Omega|HI)\frac{\int \mathrm{d}D_L \mathrm{d}z p(D_L|z\Omega HI)p(z|\Omega HI)p(d|D_L z\Omega HI)}{p(D|HI)} \,. \tag{141}$$

Once we specify a given cosmological model, e.g. FRWL, z and Ω uniquely determine D_L, thus

$$p(D_L|z\Omega HI) = \delta(D_L - D_L(z,\Omega)) \tag{142}$$

and the posterior (141) becomes:

$$p(\Omega|DHI) = p(\Omega|HI)\frac{\int \mathrm{d}z p(z|\Omega HI)p(d|D_L(z,\Omega)z\Omega HI)}{p(D|HI)} \,. \tag{143}$$

5.1.1 With electromagnetic counterparts

In case of a unique EM identification, the prior for z is particularly simple:

$$p(z|\Omega HI) = \delta(z - z_t) \tag{144}$$

therefore, the posterior for Ω is

$$p(\Omega|DHI) = p(\Omega|HI)\frac{p(d|D_L(z_t,\Omega)z_t\Omega HI)}{p(D|HI)} \,. \tag{145}$$

References

[1] Jaynes, E. T., Probability Theory: The Logic of Science, Cambridge University Press, 2003

[2] Gregory, P., Bayesian Logical Data Analysis for the Physical Sciences, Cambridge University Press, 2010

[3] Roever, C. et al, http://arxiv.org/pdf/0804.3853v6.pdf

[4] Buonanno, A., Iyer, B. R., Ochsner, E., Pan, Y., & Sathyaprakash, B. S. 2009, Phys. Rev. D, 80, 084043