Argelander-
Institut
für
Astronomie

Rheinische Friedrich-Wilhelms-Universität Bonn

Statistical methods for astrophysics and cosmology

Author: Robert Reischke

Semester: Summer Term 2026

Last updated: July 15, 2026

Contents

1 Why statistics matters and how to use these notes

Progress in the physical sciences is achieved by developing models of the real (mostly inanimate) world that can reproduce and predict specific data or measurements. Successful models constitute physical theories that can be updated or replaced if new data become available. Physical science can be seen as the science of measurement. This is not meant in a quantum-theoretical manner[1], but rather to highlight that the physical sciences strive to be as honest as possible about our ability to state truths or our degree of belief about the world and the corresponding physical theories. We therefore need rules for propagating our degree of belief in our measurements to the underlying theories. Statistics is thus the bridge between observations and theory, and it provides rules for updating our beliefs about the underlying theories. This "circle" is displayed below in Figure 1.1. Let us give some astrophysical context to this. Suppose

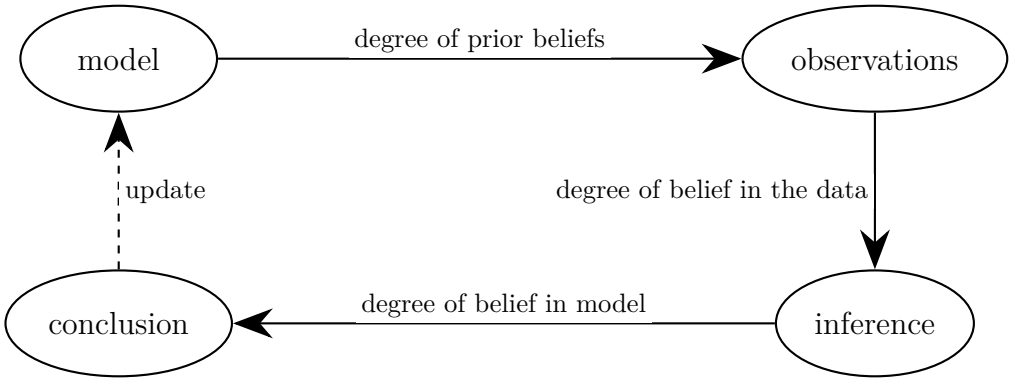


Figure 1.1: *The circle of the logic of science. One starts from a model, compares it to observations, infers certain aspects from those observations and concludes some (statistical) truth from this inference. This conclusion might then be used to update the model.*

you are a theorist working on particle dark matter models, and you want to confront your model with annihilation signals of that dark matter candidate in astrophysical systems. Since the annihilation signal scales as the squared number density of dark matter particles, you aim to observe it in cosmologically high-density environments, such as clusters of galaxies. The model itself must fulfil certain theoretical conditions, for example, dictated by symmetry considerations. Furthermore, you want to include information from other data, such as the Cosmic Microwave Background, which already tells you the amount of dark matter you can expect relative to ordinary matter in the Universe. This means that you want to impose a degree of prior beliefs on your model's parameter space, which is both derived from theory and other experiments. Next, you use or make observations of galaxy clusters and compare them with your model; you therefore need to introduce a notion or a measure of the degree of belief in your data. By respecting both degrees of belief, you are able to formulate a final degree of belief in the possible configurations of your model by inferring its parameters. From this, you can draw conclusions about the model itself, which can then be used to update prior information for upcoming experiments. You can even take a birds-eye view and repeat this circle for another family of models with different parameters and ask the question whether one model is more likely to present the data than another.

[1] It is interesting though that the real world seems to be, at its most fundamental level to be very concerned about what a measurement is and how it interacts with the physical system.

All the arrows displayed in Figure 1.1 here would be true or false statements in classical propositional logic. In the real world, however, where we have only finite data, these mappings can only be represented by something which we might call a degree of belief or certainty in the conviction that an assertion is indeed true. Crucially, a logical assertion of the form X then Y is only that: logical. There is no causality implied. This is also what statistics deals with, logical deductions, such that if Y is true, we are more certain that X is true as well.

The last point is particularly important for astrophysics and especially cosmology. While disciplines like particle physics can always generate new data, we are given a single Universe to measure. Of course, astrophysical processes can be observed in many equivalent but different systems to generate independent realisations, but if we are, for example, to observe the distribution of galaxies in the Universe, it is clear that we only have access to a limited amount of information. One could therefore argue that *frequentist* interpretation of statistics, that is, interpreting the probability of an event by the frequency at which it occurs, becomes somewhat limited as it, at its heart, requires independent repetitions of the same experiment.

From the discussion above, it is clear that we want to treat statistics as a means of updating our (prior) knowledge as new data becomes available. That being said, we will simultaneously introduce *frequentist* tools where necessary and useful. The structure of these lecture notes is as follows: We begin with basic notions of probability theory, such as the Kolmogorov axioms and conditional probability, in Section 2. Having established this notion, we will talk about what we can learn about probability distributions from samples of the underlying distribution in Section 3. We will then tackle the problem of parameter estimation in Section 4 in a purely *frequentist* manner and expand it to a *Bayesian* framework in Section 5. Section 6 will teach us how to create samples from a probability distribution with Monte Carlo techniques and how they are used in parameter estimation. At this point, we will be well-equipped to handle most standard inference problems in practice. We will then discuss machine learning and artificial intelligence are linked to and applied in classical statistics in Section 7. As an example, we discuss increasingly popular methods of simulation-based inference in Section 8. If time permits, we will discuss some links between probability theory and information theory.

Before starting, you will have noticed that the outer edge of the lecture notes has quite a bit of white space. It sometimes happens that I say something smart during the lecture that is not explicitly written in the lecture notes. You are therefore very encouraged to take notes, as I do believe that active note-taking helps understanding. Sometimes I include excessive mathematical details in boxes labelled “mathematical background”. Those can be skipped if one is not interested in the very details and only in the practical applications. The lecture notes will also be accompanied by `python` notebooks for further reading and experimentation, allowing you to create plots yourself. You can find them on GitHub. I thus expect that you have a basic understanding of programming with `python` or any real programming language. Lastly, please note that I will update the manuscript as the term progresses, as this is the first time I am giving the lecture in this setting, meaning I will upload the chapters as they are completed.

2 Basics of probability theory

As discussed in the introduction, probability is the mathematical language for quantifying uncertainty, or, more generally, our degree of belief in a given proposition (e.g., most of the matter in the Universe interacts only weakly with light, i.e., it is dark). In this chapter, we lay the mathematical foundation for probability theory, which the rest of the lecture will build upon.

2.1 Set theory recap

Probability theory is built on the language of sets (as is physics, of course). Here, we briefly recall the essential definitions and operations. Let us define a set:

Definition 2.1: Set

A set A is a collection of objects, called elements. We write $x \in A$ if x belongs to A, and $x \notin A$ otherwise. The empty set $\emptyset$ contains no elements.

Definition 2.2: Subset

A is a subset of B, written $A \subseteq B$, if every element of A is also an element of B. $A = B$ if and only if $A \subseteq B$ and $B \subseteq A$.

With these definitions, we introduce standard operations that allow the manipulation of sets to form new sets. For this, let $A \subseteq \Omega$ and $B \subseteq \Omega$ be sets. In particular, we define the following operations:

1. Union: $A \cup B = \{x \in \Omega : x \in A \text{ or } x \in B\}$
2. Intersection: $A \cap B = \{x \in \Omega : x \in A \text{ and } x \in B\}$
3. Complement: $A^{\mathrm{c}} = \{x \in \Omega : x \notin A\}$
4. Difference: $A \setminus B = A \cap B^{\mathrm{c}}$

These are shown in Figure 2.1 as Venn diagrams, where the shaded region is always the result of the depicted operation.
With the notion defined so far, we call sets A and B disjoint (or mutually exclusive) if $A \cap B = \emptyset$. The complement (lower left) Venn diagram already shows an intuitive consequence of the definitions so far:

$$(A \cup B)^{\mathrm{c}} = A^{\mathrm{c}} \cap B^{\mathrm{c}}, \qquad (A \cap B)^{\mathrm{c}} = A^{\mathrm{c}} \cup B^{\mathrm{c}}. \tag{2.1}$$

Those are known as De Morgan's laws. One can also make this more formal: Let $x \in (A \cup B)^{\mathrm{c}}$. Then $x \notin A \cup B$, which means $x \notin A$ and $x \notin B$, i.e. $x \in A^{\mathrm{c}}$ and $x \in B^{\mathrm{c}}$. The reverse inclusion is analogous, and the second law follows by symmetry. This proof generalises to a countable number of sets A_i with $i \in I$, where I is an index set.

Definition 2.3: Partition

A partition of Ω is a collection $\{A_i\}_{i \in I}$ of non-empty, pairwise disjoint subsets, i.e. $A_i \cap A_j = \emptyset$, $\forall i, j \in I$, such that $\bigcup_{i \in I} A_i = \Omega$.

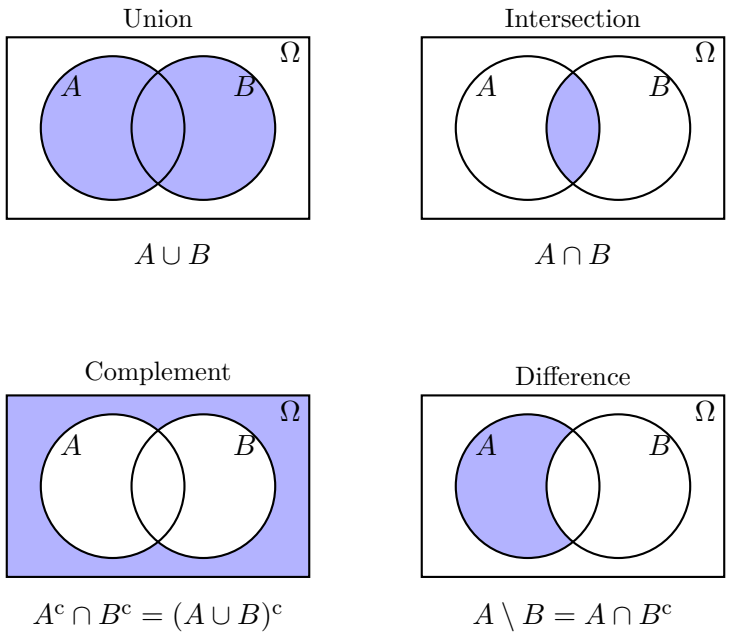


Figure 2.1: *Venn diagrams for different operations on sets. The complement shows De Morgan's law.*

Mathematical background: Naive vs. axiomatic set theory

Here, we worked in naive set theory. This means that sets are treated as simple collections of objects, and we reason about them informally. This is entirely sufficient for our purposes. However, naive set theory is logically inconsistent. Its central flaw is unrestricted comprehension: the assumption that any property $\varphi(x)$ defines a legitimate set $\{x : \varphi(x)\}$. This, however, leads to a paradox: let

$$R = \{x : x \notin x\}.$$

Then $R \in R \iff R \notin R$, a contradiction. The modern remedy is axiomatic set theory, most commonly the Zermelo–Fraenkel system with the Axiom of Choice (ZFC), which constrains set formation to avoid such paradoxes.

2.2 Sample spaces and events

Any experiment is subject to some degree of randomness, making it impossible to predict the exact outcome of the measurement. This can have several origins; there can be systematic errors or statistical fluctuations in the measurement device, leading to different outcomes; the physical system under consideration might be itself only a single realisation out of a population of objects (this is particularly important in astrophysics and cosmology); some physical processes, such as radioactive decay, are naturally random. It therefore suffices to say that any measurement or observation will be an outcome of (possibly infinitely many) possibilities.

Definition 2.4: Sample space

The sample (or outcome) space Ω is the set of all possible outcomes of a random experiment. Its elements $\omega \in \Omega$ are called sample points (or outcomes).

Some examples of the sample space for different experiments are:

- Tossing a coin: $\Omega = \{H, T\}$.
- Rolling a die: $\Omega = \{1, 2, 3, 4, 5, 6\}$.
- Measuring the mean number of galaxies in the Universe in a patch of the sky: $\Omega = [0, N_{\text{gal}}]$.

Let us examine the last example in more detail. What creates the randomness here? We do know that the Universe is isotropic and homogeneous on large scales; thus, for an infinitely large patch, there should not be any randomness involved. However, if the patch has a finite size, we will find only finitely many galaxies within it. If all patches under consideration have a total of N_{gal} galaxies in them, the number in each patch must be bounded by 0 and N_{gal}. However, one patch might be in a region of the sky with fewer galaxies because it is a void, whereas another patch might cover a galaxy cluster, so the number density is higher. We do not know the initial conditions in the early Universe from which those galaxies formed, so we cannot determine them before performing the experiment. Additionally, we can imagine a situation in which the seeing of the telescope (if we are not lucky enough to be in space) differs across the different patches (or pointings) of the telescope. Consequently, galaxies will fall below the telescope's detection threshold, and the observed number of galaxies will fluctuate. A proper statistical analysis needs to account for these effects.

Definition 2.5: Event

An event is a subset $A \subseteq \Omega$. The collection of all events is called the event space $\mathcal{F}$.

From this definition, you can think of the event space as the power set of the outcome space (at least for finite spaces) $\mathcal{P}(\Omega)$. Thus, an event could be $A = \{1, 3, 4\}$ in the case of die rolling.

2.3 The σ-algebra and measurable spaces

Our goal is to define a probability measure for all possible events of the sample space Ω, i.e. the event space $\mathcal{F}$. For finite spaces, i.e. for which the cardinality of the set $|\Omega| < \infty$ this can be easily done since $\mathcal{P}(\Omega)$ has 2^n elements, and one can define a probability measure on all of them without contradiction. The situation changes drastically for infinite, in particular uncountable, sample spaces.

Mathematical background: The Vitali set

Suppose $\Omega = \mathbb{R}$ or $\Omega = [0, 1]$. Here, naive attempts to assign a "uniform" probability to every subset of $\mathbb{R}$ run into set-theoretic obstructions. The classic example is the construction of a Vitali set: assuming the Axiom of Choice, one can exhibit a subset of $[0, 1]$ that cannot be assigned any consistent length (measure). More precisely, there is no function

$$\mu : \mathcal{P}(\mathbb{R}) \longrightarrow [0, \infty]$$

that simultaneously satisfies translation invariance, countable additivity, and $\mu([0, 1]) = 1$. The resolution is to restrict the domain of the measure to a carefully chosen family of well-behaved sets; the σ-algebra. The σ here signals closure under countable (but possibly infinite) operations.

Definition 2.6: σ-algebra

Let Ω be a non-empty set. A collection $\mathcal{F} \subseteq \mathcal{P}(\Omega)$ is called a σ-algebra (or σ-field) on Ω if it satisfies the following three axioms:

1. $\Omega \in \mathcal{F}$.
2. If $A \in \mathcal{F}$, then $A^c := \Omega \setminus A \in \mathcal{F}$.
3. If $A_1, A_2, \ldots \in \mathcal{F}$, then $\bigcup_{n=1}^{\infty} A_n \in \mathcal{F}$.

The pair $(\Omega, \mathcal{F})$ is called a measurable space.

From this definition, we can immediately conclude several corollaries: $\emptyset \in \mathcal{F}$ (because of 1. and 2.); $\mathcal{F}$ is closed under countable intersections:

$$\bigcap_{n=1}^{\infty} A_n \in \mathcal{F}, \tag{2.2}$$

and under set differences:

$$A \setminus B = A \cap B^c \in \mathcal{F}. \tag{2.3}$$

It is an instructive exercise to quickly show the previous two. For any Ω, any σ-algebra $\mathcal{F}$ satisfies:

$$\mathcal{F}_{\min} = \{\emptyset, \Omega\} \subseteq \mathcal{F} \subseteq \mathcal{F}_{\max} = \mathcal{P}(\Omega). \tag{2.4}$$

Mathematical background: Borel σ-algebra

The most important σ-algebra in probability theory is the Borel σ-algebra on $\mathbb{R}$, defined as

$$\mathcal{B}(\mathbb{R}) \;:=\; \sigma(\{(a,b) : a < b,\ a, b \in \mathbb{R}\}),$$

i.e., the σ-algebra generated by all open intervals (equivalently, by all open sets in the standard topology).

2.4 The Kolmogorov axioms

With the definitions and structure introduced so far, we are now in a position to put probability theory on a rigorous measure-theoretic foundation. This was first done by A . Kolmogorov in 1933.

Definition 2.7: Kolmogorov axioms

Let $(\Omega, \mathcal{F})$ be a measurable space. A probability measure is a function $\mathbb{P} : \mathcal{F} \to [0, 1]$ satisfying:

1. $\mathbb{P}(A) \geq 0$ for all $A \in \mathcal{F}$.
2. $\mathbb{P}(\Omega) = 1$.
3. Countable Additivity (σ-additivity): For every countable collection of pairwise disjoint events $A_1, A_2, \ldots \in \mathcal{F}$:

$$\mathbb{P}\left(\bigcup_{n=1}^{\infty} A_n\right) = \sum_{n=1}^{\infty} \mathbb{P}(A_n).$$

The triple $(\Omega, \mathcal{F}, \mathbb{P})$ is called a probability space.

Note that the axioms do not tell us anything about how to assign probabilities to events; they only state the properties they have to satisfy. This also leaves a lot of room for interpretation of how we are to understand $\mathbb{P}(A)$. The standard frequentist interpretation is to assign the frequency at which A is true as $\mathbb{P}(A)$. For a fair coin flip, $\mathbb{P}(H) = 1/2$ would mean that as the number of trials tends to infinity, one would get heads half the time. This highlights a potential limitation of the frequentist notion, since such a function exists only in an idealised world. On the other hand, the Bayesian school describes degrees of belief in that A is true. We will discuss those differences more in Section 5. For now, it is important to notice that both schools fundamentally agree on the Kolmogorov axioms.

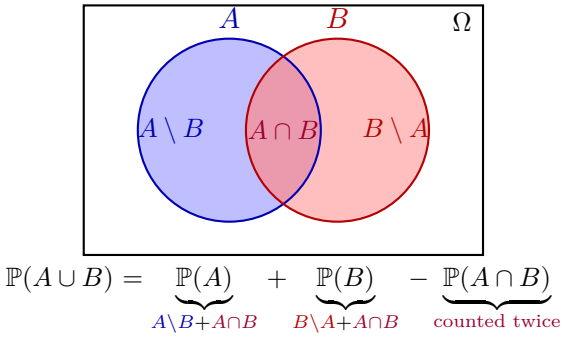
From the axioms, we can immediately deduce a couple of useful properties:

$$\begin{aligned}
&\mathbb{P}(\emptyset) = 0, \\
&A \subset B \Rightarrow \mathbb{P}(A) \leq \mathbb{P}(B), \\
&0 \leq \mathbb{P}(A) \leq 1, \\
&\mathbb{P}(A^c) = 1 - \mathbb{P}(A), \\
&A \cap B = \emptyset \Rightarrow \mathbb{P}(A \cup B) = \mathbb{P}(A) + \mathbb{P}(B)\ .
\end{aligned} \tag{2.5}$$

Lastly, we can derive that for any events $A, B \in \mathcal{F}$:

$$\mathbb{P}(A \cup B) = \mathbb{P}(A) + \mathbb{P}(B) - \mathbb{P}(A \cap B). \tag{2.6}$$

This can be shown formally, but is graphically very obvious:



A good example is two fair coins, each with probability $1/2$ of flipping heads. The probability $\mathbb{P}(H_1 \cup H_2)$, i.e. that one of the coins flips heads, is according to Equation (2.6): $\mathbb{P}(H_1 \cup H_2) = 1/2 + 1/2 - 1/4 = 3/4$.

2.5 Conditional probability and independence

We now define the probability of an event A if another event B has happened. This conditional probability is denoted as $\mathbb{P}(A|B)$ and reads "A given B". A classic example is having a disease and its associated symptoms. Say A is that you have measles, while B is that you have dots on your skin. In this case, $\mathbb{P}(B|A) = 1$ because you will have dots if you have measles. The converse, however, is $\mathbb{P}(A|B) \leq 1$.

Definition 2.8: Conditional probability

Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space and $A, B \in \mathcal{F}$ with $\mathbb{P}(B) > 0$. The conditional probability of A given B is:

$$\mathbb{P}(A \mid B) = \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(B)}.$$

The best way to understand this definition is that, in principle, every probability is restricted to the underlying sample space. However, because of $\mathbb{P}(\Omega) = 1$ and $A \cap \Omega = A$, we have $\mathbb{P}(A|\Omega) = \mathbb{P}(A)$. Therefore, we can view the conditional probability in such a way that by restricting to B it becomes the new Ω. It should also be noted that for fixed B with $\mathbb{P}(B) > 0$, the function $A \mapsto \mathbb{P}(A \mid B)$ is itself a probability measure on $(\Omega, \mathcal{F})$.

From the definition of the conditional probability, we can immediately follow the multiplication rule for probability as

$$\mathbb{P}(A \cap B) = \mathbb{P}(A|B)P(B). \tag{2.7}$$

Definition 2.9: Independence of two events

Events A and B are independent, often written $A \perp B$, if:

$$\mathbb{P}(A \cap B) = \mathbb{P}(A)\,\mathbb{P}(B).$$

If $\mathbb{P}(B) > 0$, independence is equivalent to $\mathbb{P}(A \mid B) = \mathbb{P}(A)$: knowing that B occurred gives no information about A. This is exactly how we would intuitively define independence. If we have more than two events, we have to generalise this notion:

Definition 2.10: Mutual independence

A collection of events $\{A_i\}_{i \in I}$ is mutually independent if for every finite subset $J \subseteq I$:

$$\mathbb{P}\left(\bigcap_{j \in J} A_j\right) = \prod_{j \in J} \mathbb{P}(A_j).$$

Note that pairwise independence does not imply mutual independence. Consider, as an example, the toss of two fair coins, with $A = \{\text{first is H}\}$, $B = \{\text{second is H}\}$, $C = \{\text{exactly one H}\}$. Then A, B, C are pairwise independent but $\mathbb{P}(A \cap B \cap C) = 0 \neq \frac{1}{8} = \mathbb{P}(A)\mathbb{P}(B)\mathbb{P}(C)$.

Lastly, we need the law of total probability: Let $\{B_i\}_{i=1}^n$ be a partition of Ω with $\mathbb{P}(B_i) > 0$ for all i. Then for any event $A \in \mathcal{F}$ we can first write:

$$\mathbb{P}(A) = \sum_{i=1}^{n} P(A \cap B_i), \tag{2.8}$$

using conditional probabilities and thus the multiplication rule, Equation (2.7), we can then rewrite this as:

$$\mathbb{P}(A) = \sum_{i=1}^{n} \mathbb{P}(A \mid B_i)\,\mathbb{P}(B_i). \tag{2.9}$$

2.6 Random variables and their distributions

So far, we have discussed probabilities on rather abstract events, particularly subsets of the sample space Ω. Already tossing a coin 50 times yields a sample space of cardinality 2^{50}, and working with these subsets is cumbersome. We therefore require a mechanism that enables computation and translates abstract outcomes into numerical values while preserving the probabilistic structure.

A random variable $X : \Omega \to \mathbb{R}$ assigns each outcome to a measurable space. Intuitively, we can think of this function as assigning a real number to each outcome

of an experiment. This makes it easy to define functions on the outcome space, since we can simply define them on the corresponding random variable. We will not go into further mathematical details here, but you can find it in the box below.

Mathematical background: random variable

Definition 2.1: Random Variable

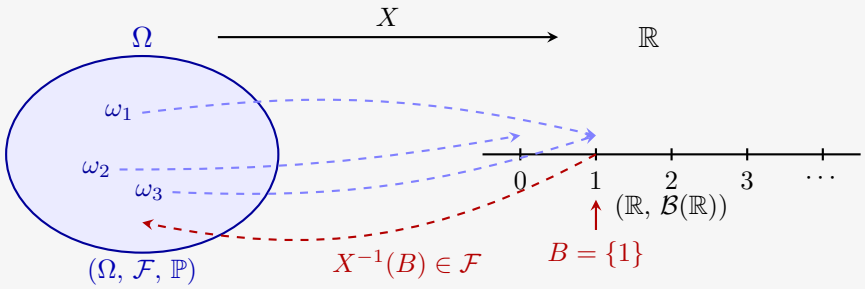
Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space. A function $X : \Omega \to \mathbb{R}$ is a random variable if it is $(\mathcal{F}, \mathcal{B}(\mathbb{R}))$-measurable, meaning:

$$X^{-1}(B) = \{\omega \in \Omega : X(\omega) \in B\} \in \mathcal{F} \quad \text{for all } B \in \mathcal{B}(\mathbb{R}).$$

Equivalently: X is measurable if and only if

$$\{\omega \in \Omega : X(\omega) \leq x\} \in \mathcal{F} \quad \text{for all } x \in \mathbb{R}.$$

This ensures that for any Borel set $B \subseteq \mathbb{R}$, the pre-image $X^{-1}(B)$ is an event in $\mathcal{F}$ to which $\mathbb{P}$ can assign a probability. Without it, asking "what is the probability that $X \in B$?" would be undefined. The diagram below illustrates the structure:



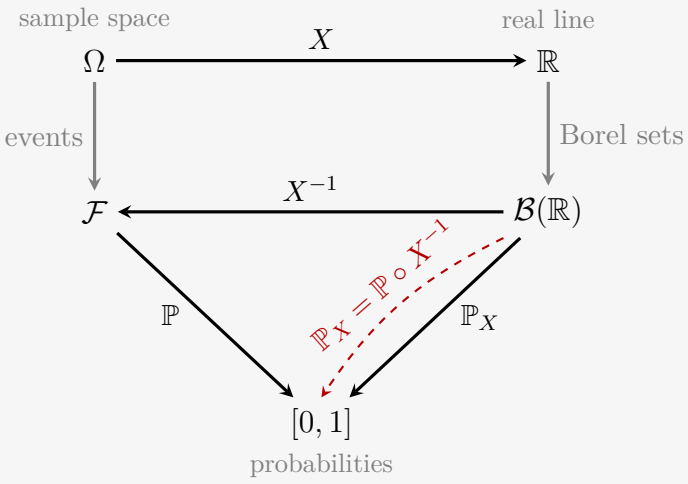
Definition 2.2: Induced distribution

The distribution of X is the probability measure $\mathbb{P}_X$ on $(\mathbb{R}, \mathcal{B}(\mathbb{R}))$ defined by:

$$\mathbb{P}_X(B) = \mathbb{P}(X \in B) = \mathbb{P}(X^{-1}(B)), \quad B \in \mathcal{B}(\mathbb{R}).$$

The distribution $\mathbb{P}_X$ contains all probabilistic information about X.

In the end, $\mathbb{P}_X$ is simply the composition which maps from the measurable space $(\mathbb{R}, \mathcal{B}(\mathbb{R}))$ back to $\mathcal{F}$ and then to the probabilities via $\mathbb{P}$.



With the procedure introduced so far, we are able to deal with experiments with both discrete outcomes (where we do not have to worry about all the problems of introducing a measure on Ω) and continuous (particularly uncountable) outcomes, e.g. what is the temperature, what is the mass of the Higgs boson, or how much matter is there in the Universe. Since the σ-algebra allows for a proper notion of open sets, all the machinery of continuity or differentiability applies.

Definition 2.11: Cumulative distribution function

The cumulative distribution function (CDF) of a random variable X is:

$$F_X(x) = \mathbb{P}(X \leq x), \quad x \in \mathbb{R}.$$

The CDF yields the probability that X takes on any value smaller than or equal to x. It has a couple of important properties

$$x \leq y \Rightarrow F_X(x) \leq F_X(y) \qquad \text{(non-decreasing)} \tag{2.10}$$

and

$$\lim_{x \to -\infty} F_X(x) = 0 \quad \text{and} \quad \lim_{x \to \infty} F_X(x) = 1. \tag{2.11}$$

Definition 2.12: Discrete random variable

X is discrete if it takes values in a countable set $\mathcal{X} = \{x_1, x_2, \ldots\} \subseteq \mathbb{R}$. Its distribution is completely described by the probability mass function (PMF):

$$p_X(x) = \mathbb{P}(X = x) \geq 0, \quad \sum_{x \in \mathcal{X}} p_X(x) = 1.$$

The corresponding CDF is

$$F_X(x) = P(X \leq x) = \sum_{x_i \leq x} P(X = x_i).$$

Definition 2.13: Continuous random variable

A random variable X is (absolutely) continuous if there exists a non-negative function $f_X : \mathbb{R} \to [0, \infty)$, the probability density function (PDF), such that:

$$F_X(x) = \int_{-\infty}^{x} f_X(t)\,\mathrm{d}t, \quad x \in \mathbb{R},$$

or equivalently, $\mathbb{P}(X \in B) = \int_B f_X(t)\,\mathrm{d}t \;\forall\; B \in \mathcal{B}(\mathbb{R})$. The PDF satisfies

$$\int_{-\infty}^{\infty} f_X(t)\,\mathrm{d}t = 1.$$

For a continuous random variable, $\mathbb{P}(X = x) = 0$ for every $x \in \mathbb{R}$. This may seem weird; every outcome has probability zero, yet some outcome must occur. The resolution is that probability zero does not mean impossible; it means the event occupies zero weight in the continuum. Probabilities are assigned to intervals, not points. Recall that a probability measure is not defined on individual outcomes, but on events, that is, on measurable sets belonging to a σ-algebra. On the real line, the relevant σ-algebra is generated by intervals, and intervals are the true building blocks of continuous probability. A single point is not an interval. When we spread probability continuously and uniformly across an interval, each individual point

receives an infinitesimally small share, and in the limit, that is exactly zero. In Figure 2.2, we show an example of a PDF of a continuous random variable.

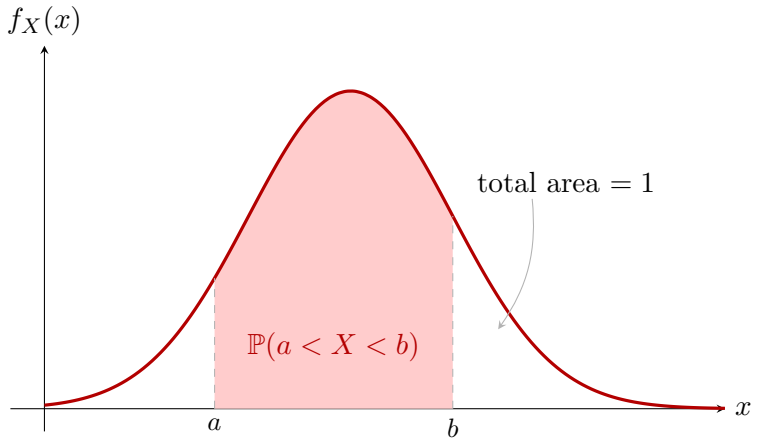


Figure 2.2: *PDF of a continuous random variable. The total area under the curve is unity, since the PDF must be normalised. Probabilities are calculated in finite intervals.*

2.7 Multivariate distributions and marginalisation

Everything we have done for a single random variable $X : \Omega \to \mathbb{R}$ extends in a straightforward manner to a random vector $\mathbf{X} = (X_1, \ldots, X_n) : \Omega \to \mathbb{R}^n$. The sample space of values is now $\mathbb{R}^n$, and the role of $\mathcal{B}(\mathbb{R})$ is played by the product Borel σ-algebra $\mathcal{B}(\mathbb{R}^n)$.

Definition 2.14: Joint CDF

The joint cumulative distribution function of n random variables $(X_1, \ldots, X_n)$ is:

$$F_{X_1,\ldots,X_n}(x_1, \ldots, x_n) = \mathbb{P}(X_1 \leq x_1, \ \ldots, \ X_n \leq x_n), \quad (x_1, \ldots, x_n) \in \mathbb{R}^n.$$

It satisfies the same structural properties as in one dimension.

Definition 2.15: Joint PDF

Continuous random variables $(X_1, \ldots, X_n)$ have a joint PDF $f_{X_1,\ldots,X_n} : \mathbb{R}^n \to [0, \infty)$ such that

$$\mathbb{P}((X_1, \ldots, X_n) \in B) = \int_B f_{X_1,\ldots,X_n}(x_1, \ldots, x_n) \, \mathrm{d}x_1 \cdots \mathrm{d}x_n, \quad B \in \mathcal{B}(\mathbb{R}^n).$$

The joint CDF and PDF are related by:

$$F_{X_1,\ldots,X_n}(x_1, \ldots, x_n) = \int_{-\infty}^{x_1} \cdots \int_{-\infty}^{x_n} f(t_1, \ldots, t_n) \, \mathrm{d}t_1 \cdots \mathrm{d}t_n.$$

In the discrete case, the random variables $(X_1, \ldots, X_n)$ have a joint PMF

$$p(x_1, \ldots, x_n) = \mathbb{P}(X_1 = x_1, \ldots, X_n = x_n) \geq 0.$$

There are two ways to move from a joint distribution back to a distribution of a single (or simply fewer) variable(s). One is marginalisation: that is, integrate (or sum) out all the variables you do not care about. The second option is to condition

a joint PDF on a specific value, which we have already done in the definition of the conditional probability.

Definition 2.16: Marginal PDF

For simplicity, let us first assume a bivariate joint distribution of (X, Y), the marginal distributions of X and Y individually are:

$$f_X(x) = \int_{-\infty}^{\infty} f_{X,Y}(x, y)\,\mathrm{d}y,$$
$$f_Y(y) = \int_{-\infty}^{\infty} f_{X,Y}(x, y)\,\mathrm{d}x.$$

In n dimensions, the marginal of $(X_1, \ldots, X_k)$ from the joint of $(X_1, \ldots, X_n)$ is obtained by integrating out $x_{k+1}, \ldots, x_n$:

$$f_{X_1,\ldots,X_k}(x_1, \ldots, x_k) = \int_{\mathbb{R}^{n-k}} f_{X_1,\ldots,X_n}(x_1, \ldots, x_n)\,\mathrm{d}x_{k+1} \cdots \mathrm{d}x_n.$$

Figure 2.3 shows the procedure of marginalisation for a bivariate Gaussian distribution. As described above, this generalises trivially to multivariate distributions.

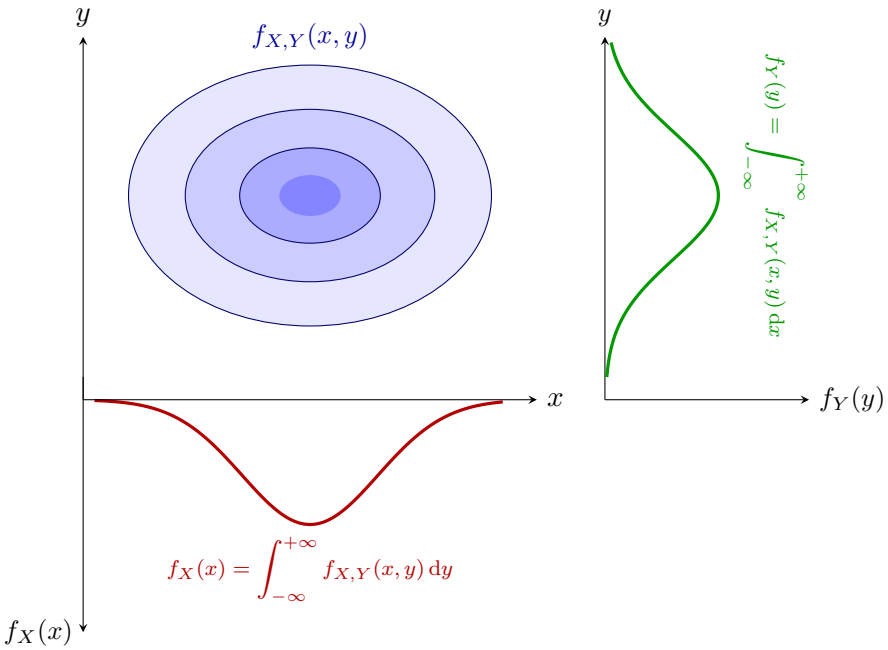


Figure 2.3: *Marginalisation: the joint PDF $f_{X,Y}(x, y)$ (centre, shown as blue contours) is integrated over the y domain to produce the marginal $f_X(x)$ (bottom), and over the x domain to produce the marginal $f_Y(y)$ (right).*

From now on, we will most often write only the continuous case. For the discrete case, simply replace the integral by discrete sums over the corresponding random variables.

Definition 2.17: Conditional PDF

iven a joint PDF $f_{X,Y}$, the conditional PDF of Y given $X = x$ if $f_X(x) > 0$ is:

$$f_{Y|X}(y \mid x) = \frac{f_{X,Y}(x, y)}{f_X(x)}.$$

Note that this is simply the continuous analogue of $\mathbb{P}(A \mid B) = \mathbb{P}(A \cap B)/\mathbb{P}(B)$. The law of total probability generalises to:

$$f_Y(y) = \int_{-\infty}^{\infty} f_{Y|X}(y \mid x)\, f_X(x)\, \mathrm{d}x.$$

With these definitions, we can, as before with discrete events, define random variables $X_1, ..., X_n$ to be independent if

$$P(X_1 \leq x_1 ... X_n \leq x_n) = P(X_1 \leq x_1) ... P(X_n \leq x_n) \tag{2.12}$$

for all real numbers $x_1, ..., x_n$. Note: Compare this with the independence of events: here the relation needs to hold for all x_i, i.e., you can let x_i go to infinity and still obtain independence of all subsets. Written differently, we can also stipulate that two continuous random variables X and Y are independent if and only if:

$$f_{X,Y}(x, y) = f_X(x)\, f_Y(y) \quad \text{for almost all } (x, y) \in \mathbb{R}^2. \tag{2.13}$$

Here, almost all means, except possibly on a set of measure zero, that the set has no weight for the purposes of integration. Familiar examples are individual points and finite collections of points. The reason is that, almost everywhere, the natural qualifier for PDFs is that they are not really pointwise objects.

You can find a `python` notebook with some examples for continuous and discrete PDFs on GitHub.

3 Estimates from samples

In the last chapter, we introduced random variables, their underlying properties, and how to describe probability distribution functions using the PDF $f_X(x)$ and the CDF $F_X(x)$ for both discrete and continuous random variables. From now on, we will often drop the function X in the index and always assume the Borel σ-algebra unless stated otherwise. With this knowledge, we would like to answer two questions. First, can we summarise the distribution of a random variable into a (possibly finite?) set of numbers that contains information about said random variable? Second: In reality, we are not given the distribution $f(x)$ but rather a finite number of samples from the distribution. The question, therefore, becomes what we can learn from this finite sample about $f(x)$?

3.1 Moments of a distribution

As indicated earlier, we will, from now on, simply drop the random variable and explicitly assume it, so that we can work directly with the observed values. We define the **expectation** (or population mean) of a random variable X as:

$$\mu \equiv \mu_X \equiv \langle X \rangle \equiv \mathbb{E}[X] := \begin{cases} \int x\, f(x)\, \mathrm{d}x = \int x\, \mathrm{d}F(x) & \text{continuous} \\ \sum_x x\, f(x) & \text{discrete.} \end{cases} \tag{3.1}$$

For the discrete case, the sum is over all possible outcomes. Since both the sum and the integral are linear operations, the expectation is itself linear. Throughout this script, we will use the notation above interchangeably. Note that this definition is a property of the distribution, not of any particular realisation. However, intuitively $\mathbb{E}$ is the average value of X over infinitely many independent random samples of the distribution. The expectation of X can be generalised to any function h of the random variable:

$$\mathbb{E}(h) = \int h(x)\, f(x)\, \mathrm{d}x\ . \tag{3.2}$$

For the special case of $h(x) = 1$, this simply returns the normalisation condition of the PDF and has to be fulfilled thanks to the Kolmogorov axioms for $f(x)$ to qualify as a PDF in the first place. Figure 3.1 shows the expectation value of a random

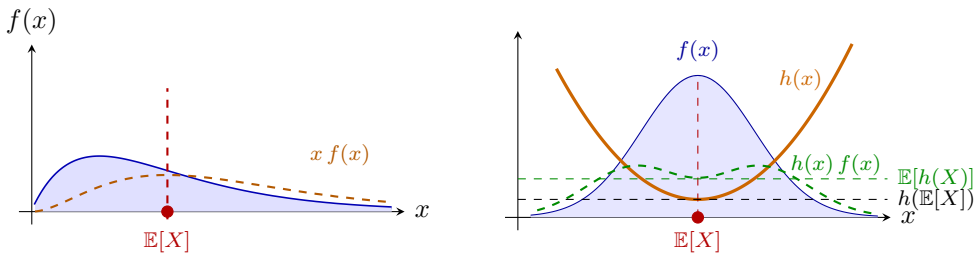


Figure 3.1: *Left: expectation value of an asymmetric PDF (blue), which can be interpreted as the centre of mass of the PDF and is marked with red. Right: expectation value of a function $h(x)$.*

variable as well as that of a function h. An important aspect to notice here is that the following holds:

$$\mathbb{E}(h(X)) \geq h(\mathbb{E}(X)), \tag{3.3}$$

for any convex function $h(x)$. This is the Jensen inequality. A special case of functions for which the expectation value can be computed is that of monomials. i.e.

x^k as they form a basis for analytical functions. We call the expectation of the k-th monomial the k-th **moment**:

$$\mu_k \equiv \mathbb{E}(X^k) = \int x^k \, f(x) \, \mathrm{d}x. \tag{3.4}$$

The moments themselves can also be defined with respect to a constant. Often we will use the first moment $\mathbb{E}(X)$ as the centre so that

$$\mu_k^{\mathrm{c}} \equiv \mathbb{E}((X-\mu)^k) = \int (x-\mu)^k \, f(x) \, \mathrm{d}x, \tag{3.5}$$

these are called the **central moments**. Below, we summarise the interpretation of the first three central moments $k > 1$:

- $k = 2$ is the variance, $\sigma^2 = \mathbb{E}[(X-\mu)^2] = \mathbb{E}[X^2] - \mu^2$, measuring spread around the mean.
- $k = 3$ is the skewness, $\mathbb{E}[(X-\mu)^3]$, related to the asymmetry of the distribution.
- $k = 4$ is the kurtosis, $\mathbb{E}[(X-\mu)^4]$, which relates to the heaviness of the tails of the distribution.

In Figure 3.2, we show how these three moments influence the shape of the PDF. If we specify all (countably many but infinitely many) moments, the PDF is uniquely determined.

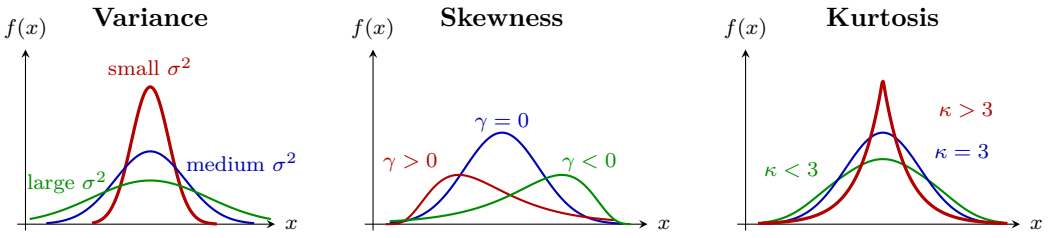


Figure 3.2: *Effect of the second, third and fourth central moments on the PDF shape. Variance (left): larger σ^2 flattens and widens the bell, all curves share the same mean μ. Skewness (centre): $\gamma = 0$ is symmetric (blue); positive/negative skewness pulls the long tail right/left. Kurtosis (right): mesokurtic ($\kappa = 3$, blue) is the normal; leptokurtic ($\kappa > 3$, red) has a sharper peak and heavier tails; platykurtic ($\kappa < 3$, green) is flatter with lighter tails.*

The whole shebang can be naturally generalised to random vectors. Let us assume a multivariate PDF in d dimensions $f(x, \ldots, x_d)$. We can collect the d values of the random variable in a d-dimensional vector $\boldsymbol{x} = (x_1, \ldots, x_d)^\top$ with components x_i. The mean of that random vector $\boldsymbol{X}$ is then simply

$$\boldsymbol{\mu} \equiv \mathbb{E}(\boldsymbol{X}) = \begin{pmatrix} \mathbb{E}(X_1) \\ \vdots \\ \mathbb{E}(X_d) \end{pmatrix} = (\mu_i)_{i=1}^d, \qquad \mu_i = \mathbb{E}(X_i). \tag{3.6}$$

Linearity of expectation extends immediately:

$$\mathbb{E}[\mathbf{A}X + \boldsymbol{b}] = \mathbf{A}\boldsymbol{\mu} + \boldsymbol{b} \tag{3.7}$$

for any matrix $\mathbf{A}$ and vector $\boldsymbol{b}$. The variance generalises to the covariance matrix:

$$\boldsymbol{\Sigma} = \mathrm{Cov}(\boldsymbol{X}) = \mathbb{E}\Big[(\boldsymbol{X} - \boldsymbol{\mu})(\boldsymbol{X} - \boldsymbol{\mu})^\top\Big], \tag{3.8}$$

with components:

$$\Sigma_{ij} = \mathrm{Cov}(X_i, X_j) = \mathbb{E}[(X_i - \mu_i)(X_j - \mu_j)] = \mathbb{E}[X_i X_j] - \mu_i \mu_j. \tag{3.9}$$

The diagonal entries $\Sigma_{ii} = \mathrm{Var}(X_i) = \sigma_i^2$ are the individual variances. The covariance matrix is always symmetric ($\Sigma_{ij} = \Sigma_{ji}$) and positive semi-definite. Often, one defines the Pearson correlation coefficient $\rho_{ij} \in [-1, 1]$, which indicates the strength of the correlation, ranging from perfectly anti-correlated to fully correlated. It is defined as

$$\rho_{ij} = \frac{\Sigma_{ij}}{\sigma_i \sigma_j} \,. \tag{3.10}$$

Generalisation for arbitrary moments

For a random vector $\boldsymbol{X}$, the mixed moment of order $(k_1, \ldots, k_d)$ with $k_i \in \mathbb{N}_0$ is:

$$\mathbb{E}\left[\prod_{i=1}^{d} X_i^{k_i}\right] = \mathbb{E}\left[X_1^{k_1} X_2^{k_2} \cdots X_d^{k_d}\right] = \left(\prod_{i=1}^{d} \int \mathrm{d}x_i \, x_i^{k_i}\right) f(x_1, \ldots, x_d).$$

The total order of the moment is $|\mathbf{k}| = k_1 + \cdots + k_d$. The univariate k-th moment $\mathbb{E}[X^k]$ is the special case $d = 1$. The covariance $\Sigma_{ij} = \mathbb{E}[X_i X_j] - \mu_i \mu_j$ involves the mixed moment of order $|\mathbf{k}| = 2$ with $k_i = k_j = 1$ and all other $k_l = 0$.

Given a PDF (discrete or continuous), all of these operations are easily executable using the definition of the expectation from Equation (3.1).

Let us specify the multivariate Gaussian distribution in terms of its first two moments, the mean and the covariance matrix:

$$f(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^d |\boldsymbol{\Sigma}|}} \mathrm{e}^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\boldsymbol{x}-\boldsymbol{\mu})}, \tag{3.11}$$

where $\boldsymbol{x}$ is the vector of random variables and $\boldsymbol{\mu}$ is the vector of their means. The covariance matrix is given as

$$\boldsymbol{\Sigma} = \begin{pmatrix} \langle (X_1 - \mu_1)(X_1 - \mu_1) \rangle & \cdots & \langle (X_1 - \mu_1)(X_d - \mu_d) \rangle \\ \vdots & \ddots & \vdots \\ \langle (X_d - \mu_d)(X_1 - \mu_1) \rangle & \cdots & \langle (X_d - \mu_d)(X_d - \mu_d) \rangle \end{pmatrix} \tag{3.12}$$

and its determinant is denoted as $|\boldsymbol{\Sigma}|$. For a bivariate Gaussian $(X_1, X_2)^\top \sim \mathcal{N}(\mathbf{0}, \Sigma)$ with unit marginal variances and correlation $\rho = \Sigma_{12}$, the level sets of the joint PDF are ellipses. When $\rho = 0$ the ellipses are circles (independence); positive ρ tilts them along the diagonal $x_1 = x_2$; negative ρ tilts them along $x_1 = -x_2$. This is shown in Figure 3.3 for a negative and positive correlation. For zero correlation, you could look at Figure 2.3. A quick remark on notation is in order; we will write $X \sim f(\boldsymbol{\theta})$ to indicate that X follows, or is sampled from, the distribution f which in turn depends on certain parameters $\boldsymbol{\theta}$. If we make the relation explicit, however, as in Equation (3.11), we will also treat the independent variable as a function argument. So if we want to write that the one-dimension variable X follows a Gaussian distribution with mean μ and variance σ^2 we write: $X \sim \mathcal{N}(\mu, \sigma^2)$ but we will write its PDF explicitly as $f_X(x) = \mathcal{N}(x; \mu, \sigma^2)$.

Let us close this discussion by considering the following example: Let $Y = \sum_i X_i$, and we would like to determine the variance of Y, i.e. $\sigma(Y)$. We know that the

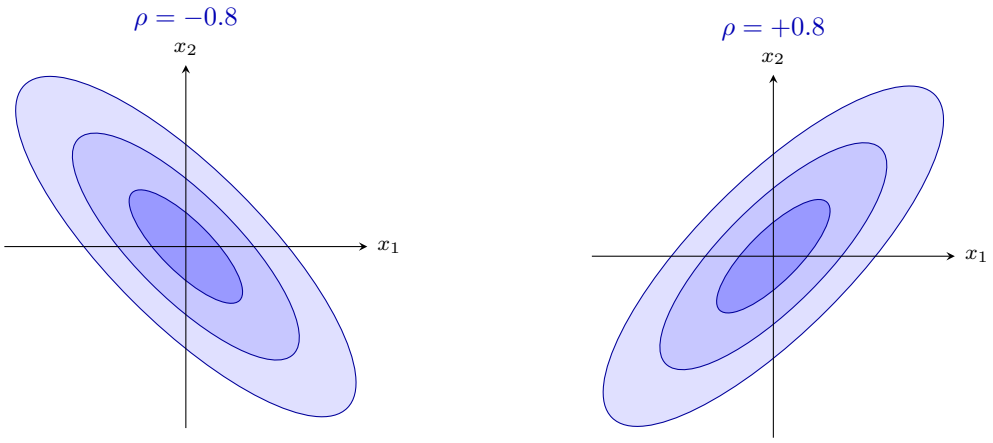


Figure 3.3: *Contour plots of a bivariate Gaussian. Left ($\rho = -0.8$): ellipses tilted towards $x_2 = -x_1$; knowing X_1 is large makes X_2 likely small. Right ($\rho = +0.8$): ellipses tilted towards $x_2 = x_1$; the two components tend to move together.*

expectation value of Y is given by $\mu_Y = \sum_i \mu_i$ due to linearity. The variance can thus be written as:

$$\begin{aligned}\sigma^2(Y) &= \mathbb{E}\left[\left(\sum_i X_i - \sum_i \mu_i\right)^2\right] \\ &= \sum_i \mathbb{E}\left[(X_i - \mu_i)^2\right] + \sum_{i,j,i\neq j} \mathbb{E}[(X_i - \mu_i)(X_j - \mu_j)] \\ &= \sum_i \sigma^2(X_i) + \sum_{i,j,i\neq j} \mathrm{Cov}(X_i, X_j).\end{aligned}$$

On GitHub, you can find an example for the sum of two random variables and how the variance of the sum changes with the degree of correlation.

Moment generating function

The moment generating function (MGF) makes the PDF's analyticity via a Taylor series evident. It is given by

$$M_X(t) = \mathbb{E}\left[e^{tX}\right], \quad t \in \mathbb{R},$$

defined for all t in an open interval around 0 where the expectation is finite. We can now expand the exponential into a power series:

$$e^{tX} = 1 + tX + \frac{t^2X^2}{2!} + \frac{t^3X^3}{3!} + \cdots.$$

And apply the expectation value on both sides of the equation:

$$M_X(t) = 1 + t\,\mathbb{E}[X] + \frac{t^2}{2!}\mathbb{E}[X^2] + \frac{t^3}{3!}\mathbb{E}[X^3] + \cdots = \sum_{k=0}^{\infty} \frac{\mathbb{E}[X^k]}{k!}\,t^k.$$

The k-th moment therefore appears as the coefficient of $t^k/k!$ in this series, and can be extracted by differentiating:

$$\mathbb{E}[X^k] = M_X^{(k)}(0) = \frac{\mathrm{d}^k}{\mathrm{d}t^k} M_X(t)\bigg|_{t=0}.$$

In other words, the moments are the coefficients of the Taylor expansion of the MGF, which is where it gets its name from. It has two further important properties: First, if $M_X(t)$ is finite in an open neighbourhood of zero, it uniquely determines the distribution of X: two random variables with the same MGF have the same distribution. Second, for independent X and Y:

$$M_{X+Y}(t) = \mathbb{E}\left[e^{t(X+Y)}\right] = \mathbb{E}\left[e^{tX}\right]\mathbb{E}\left[e^{tY}\right] = M_X(t)\,M_Y(t),$$

so the MGF of a sum of independent variables is the product of the individual MGFs.
If $X \sim \mathcal{N}(\mu, \sigma^2)$, completing the square in the exponent gives:

$$M_X(t) = \exp\left(\mu t + \tfrac{1}{2}\sigma^2 t^2\right).$$

Differentiating once: $M'_X(0) = \mu$ (the mean). Differentiating twice: $M''_X(0) = \mu^2 + \sigma^2$ (the second moment), so $\mathrm{Var}(X) = \mu^2 + \sigma^2 - \mu^2 = \sigma^2$, as expected.

3.2 Point estimates

Let us start with the PDF $f(x)$ of a random variable X. We call the collection $\{X_1, \ldots, X_n\}$ an independent and identically distributed (iid) random sample of size n and the observed values $x_1, \ldots, x_n$ are a single realisation of that sample.

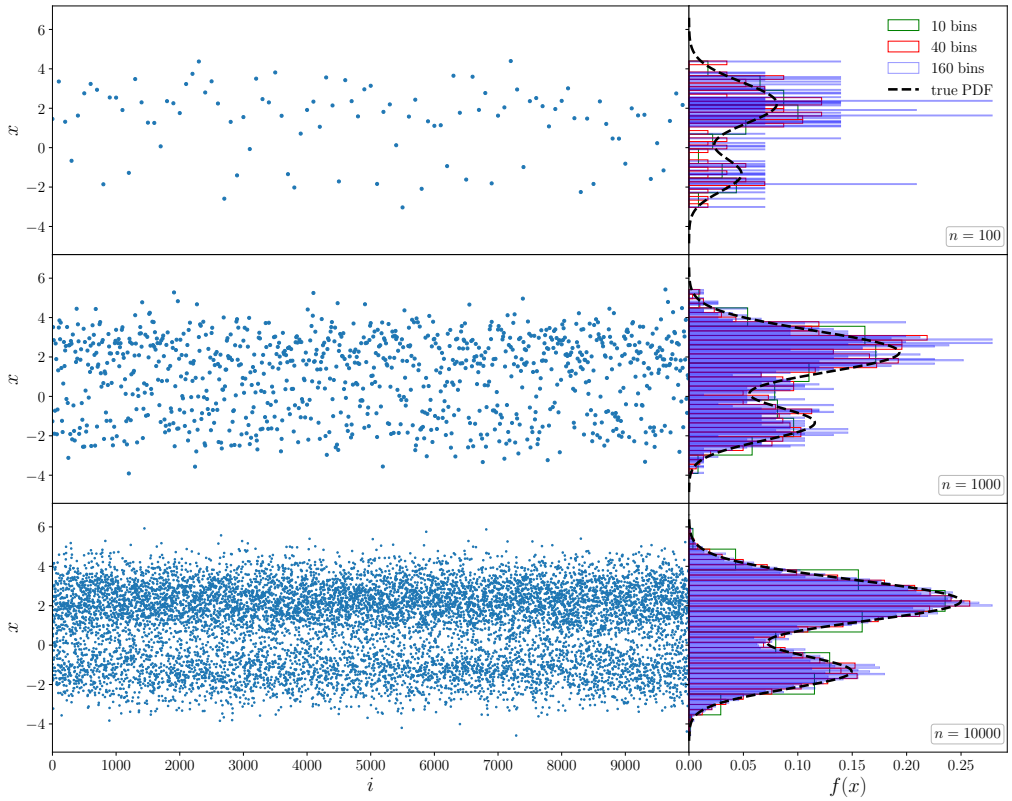


Figure 3.4: *$n = 100$, 1 000 and 10 000 samples from a probability distribution $f(x)$ shown in dashed black as the true PDF. On the left, the data, i.e., the x_i, are shown. Right: histograms of that data with different numbers of bins in x. All histograms are normalised.*

To illustrate the problem more clearly, the right panel of Figure 3.4 shows a

PDF $f(x)$ as a black dashed line. As discussed, in reality, we do not have access to this function, but rather repeated realisations of that sample. Let us assume an experiment in which the data are generated by a random process that follows $f(x)$. We repeat this experiment $n = 100$, 1000, 10000 times and plot the results as blue dots in the left panel. From top to bottom, we are adding more and more data points. A first step in addressing the problem of point estimation is to construct a histogram, i.e., the relative frequency or occurrence of measurements in x. To do so, we have to divide x into finite intervals since it is continuous. The result of this exercise is shown in the right panel of Figure 3.4 where we bin x into 10, 40 and 160 linearly spaced bins respectively in each row. This already shows that having access to a sufficiently large ($n \to \infty$) sample of a PDF allows us to infer the underlying distribution. Often, it suffices to learn a finite number of properties of the PDF, such as its mean and variance. How can we estimate these quantities directly from the data? Note that this is also crucial even if you know the functional form of the PDF, for example, a Gaussian, to infer the actual values of the corresponding parameter characterising the distribution.

3.3 Estimators

Let us assume we have a PDF $f(x|\boldsymbol{\theta})$ which depends on a set of parameters bundled into the vector $\boldsymbol{\theta}$. Note that we explicitly state that we are dealing with a PDF for given values of the parameters $\boldsymbol{\theta}$. Given n iids of x (remember, related to n observations), we would like to estimate the parameters of the distribution. The estimator is defined as:

$$\hat{\boldsymbol{\theta}}_n \coloneqq \hat{\boldsymbol{\theta}}(X_1, \ldots, X_n). \tag{3.13}$$

This means that $\hat{\boldsymbol{\theta}}$ is defined as a function of the random variables of the underlying population. It is computed for, or rather from, a particular realisation of the random variables. Crucially, $\hat{\boldsymbol{\theta}}_n$ is a random variable itself as it inherits this property from the X_i. Thus, we can apply all the procedures introduced so far to the estimator as well.

While this makes formal sense, it is very general and does not specify anything about the estimator itself. Let us therefore define a few properties that our estimator $\hat{\boldsymbol{\theta}}$ should fulfil to be a good estimator. The estimator should be **unbiased**, that is, the bias $\boldsymbol{b}$ vanishes:

$$\boldsymbol{b} \coloneqq \mathbb{E}[\hat{\boldsymbol{\theta}}] - \boldsymbol{\theta} = 0. \tag{3.14}$$

We call estimators asymptotically unbiased if

$$\lim_{n\to\infty} \boldsymbol{b}_n = 0. \tag{3.15}$$

We can also define the mean squared error (MSE):

$$\mathrm{MSE}(\hat{\boldsymbol{\theta}}) = \mathbb{E}\Big[(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})^2\Big] = b^2 + \sigma^2. \tag{3.16}$$

Note that this is evidently not the same as the variance σ^2 which was defined as the second central moment of the distribution. It only agrees with the variance for an unbiased estimator. If we were to minimise the MSE for our estimator, it is clear that there is a trade-off between bias and variance. This dilemma is shown in Figure 3.5. The optimal case is obviously the case with low bias and low variance. If this is not achievable, however, we would always prefer a low bias and larger variance to a low variance at the cost of a large bias.

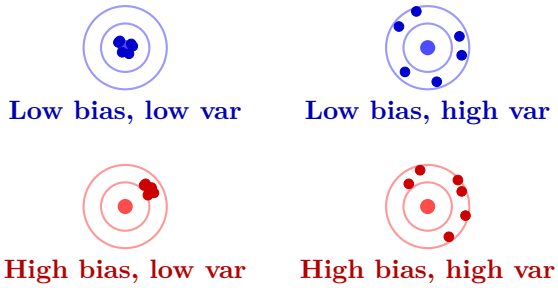


Figure 3.5: *Each dot is one estimate from a different sample, and the bullseye is the true parameter $\boldsymbol{\theta}$. Low bias means estimates are centred on the target; low variance means they are tightly clustered. The MSE combines both sources of error.*

3.4 Consistent estimators

An estimator that converges to the true parameter as $n \to \infty$ is called consistent. More formally, a sequence of estimators as defined in Equation (3.13) is weakly consistent for a parameter $\boldsymbol{\theta}$ if it converges in probability to $\boldsymbol{\theta}$:

$$\hat{\boldsymbol{\theta}}_n \xrightarrow{\mathbb{P}} \boldsymbol{\theta} \quad \text{as } n \to \infty, \tag{3.17}$$

that is, for every $\varepsilon > 0$:

$$\mathbb{P}\Big(|\hat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}| \geq \varepsilon\Big) \to 0 \quad \text{as } n \to \infty. \tag{3.18}$$

It is strongly consistent if:

$$\mathbb{P}(\lim_{n\to\infty} \hat{\boldsymbol{\theta}}_n = \boldsymbol{\theta}) = 1. \tag{3.19}$$

Note that this is a stronger statement than asymptotic unbiasedness, as you need to assume that the variance vanishes as $n \to \infty$ additionally. Consistency, however, is a minimal requirement for a useful estimator. An estimator that does not converge to the truth, even with unlimited data, is fundamentally flawed, regardless of how well it behaves at any fixed sample size. One should keep in mind, however, that consistency alone says nothing about the speed of convergence or the behaviour for finite n.

3.5 Example: sample mean

Let us again assume that we have n iid random variables sampled from an unknown distribution $f(x)$ with true mean μ. We define the estimator of the mean as the sample mean:

$$\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} X_i. \tag{3.20}$$

Let us first calculate the bias:

$$\mathbb{E}(\hat{\mu}) - \mu = \mathbb{E}\left(\frac{1}{n}\sum_{i=1}^{n} X_i\right) - \mu = \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}(X_i) = 0. \tag{3.21}$$

Thus, the sample mean is an unbiased estimator of the true population mean. The next question, however, is how precise this estimator is, i.e., what is its scatter (variance)? We can calculate this by inserting the unbiasedness already:

$$\sigma^2_{\hat{\mu}} = \mathbb{E}\left(\frac{1}{n}\sum_{i=1}^{n} X_i - \mu\right)^2 = \frac{1}{n^2}\sum_{i=1}^{n}\mathbb{E}\left[(X_i - \mu)^2\right] = \frac{n}{n^2}\sigma^2 = \frac{\sigma^2}{n}. \tag{3.22}$$

Hence, if we have a random variable with variance σ^2, the variance of the mean is reduced by a factor of $1/n$. So the intuitive rule-of-thumb: "more data equals more precision" is indeed true.

3.6 The central limit theorem

If the n random variables are Gaussian, the estimator of the mean is simply a sum of Gaussian random variables and hence itself a Gaussian random variable, so we already know that:

$$p(\hat{\mu}|X_1, ..., X_n) = \mathcal{N}\left(\hat{\mu}, \mu, \frac{\sigma}{\sqrt{n}}\right). \tag{3.23}$$

For arbitrary distributions, this does not hold. However, the central limit theorem comes to the rescue: given an arbitrary distribution $f(x)$ with mean μ and finite standard deviation $\sigma < \infty$, the mean of n iid values x_i drawn from $f(x)$ will approach a Gaussian distribution $\mathcal{N}(\mu, \sigma/\sqrt{n})$ as $n \to \infty$.

Cumulants and the proof of the central limit theorem

Moments are intuitive; however, the moments of a sum of independent random variables do not combine in a simple way. Cumulants fix this, so that the k-th cumulant of a sum of independent variables equals the sum of the individual k-th cumulants.
For the MGF, products occur; it is therefore natural to consider the logarithm of the MGF. Where the MGF expresses the distribution as an exponential of a power series in t, the cumulant generating function (CGF) $K_X(t)$ expresses the log-probability structure. Particularly, for independent X and Y:

$$K_{X+Y}(t) = \log M_{X+Y}(t) = \log M_X(t) + \log M_Y(t) = K_X(t) + K_Y(t).$$

The coefficients of the Taylor series are exactly the cumulants. The **k-th cumulant** κ_k is thus the coefficient of $t^k/k!$ in the Taylor expansion of $K_X(t)$ around $t = 0$:

$$K_X(t) = \sum_{k=1}^{\infty} \kappa_k \frac{t^k}{k!} \qquad \Longleftrightarrow \qquad \kappa_k = \frac{\mathrm{d}^k}{\mathrm{d}t^k} K_X(t)\bigg|_{t=0}.$$

It follows directly that if X and Y are independent, then for all $k \geq 1$:

$$\kappa_k(X + Y) = \kappa_k(X) + \kappa_k(Y).$$

This follows immediately from $K_{X+Y}(t) = K_X(t) + K_Y(t)$, which holds because independence gives $M_{X+Y}(t) = M_X(t)M_Y(t)$ and log converts products to sums.

To prove the central limit theorem, we show the case $\mu = 0$, $\sigma = 1$ without loss of generality. Let $Y_i = (X_i - \mu)/\sigma$ so that $\mathbb{E}[Y_i] = 0$, $\mathbb{E}[Y_i^2] = 1$, and

$$S_n = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} Y_i.$$

We wish to show $\kappa_k(S_n) \to \kappa_k(\mathcal{N}(0, 1))$ for every k, i.e. $\kappa_1(S_n) \to 0$, $\kappa_2(S_n) \to 1$, and $\kappa_k(S_n) \to 0$ for all $k \geq 3$.

Since $Y_1, \ldots, Y_n$ are independent, additivity yields:

$$\kappa_k\left(\sum_{i=1}^{n} Y_i\right) = \sum_{i=1}^{n} \kappa_k(Y_i) = n\,\kappa_k(Y),$$

where $\kappa_k(Y)$ denotes the common k-th cumulant of each Y_i. Next, we calculate the CGF of S_n:

$$\kappa_k(S_n) = \kappa_k\left(\frac{1}{\sqrt{n}}\sum_{i=1}^{n} Y_i\right) = \frac{1}{(\sqrt{n})^k}\cdot n\,\kappa_k(Y) = n^{1-k/2}\,\kappa_k(Y).$$

Lastly, we take the limit of all the cumulants order by order:

- $k = 1$: $\kappa_1(S_n) = n^{1/2}\,\kappa_1(Y) = 0$, since $\mathbb{E}[Y] = 0$.
- $k = 2$: $\kappa_2(S_n) = n^0\,\kappa_2(Y) = 1$, since $\sigma_Y^2 = 1$.
- $k \geq 3$: $\kappa_k(S_n) = n^{1-k/2}\,\kappa_k(Y) \to 0$, because $1 - k/2 < 0$ for $k \geq 3$.

Thus, as $n \to \infty$, the cumulants of S_n converge to those of $\mathcal{N}(0,1)$: mean 0, variance 1, and all higher cumulants zero. Therefore, S_n approaches $\mathcal{N}(0,1)$ as $n \to \infty$.

Figure 3.6 shows the convergence towards the CLT for a sum of $n = 1$, 2 4 and 32 iid random variables. Already for $n = 4$, one can see that the distribution of the sum is well approximated by a Gaussian.

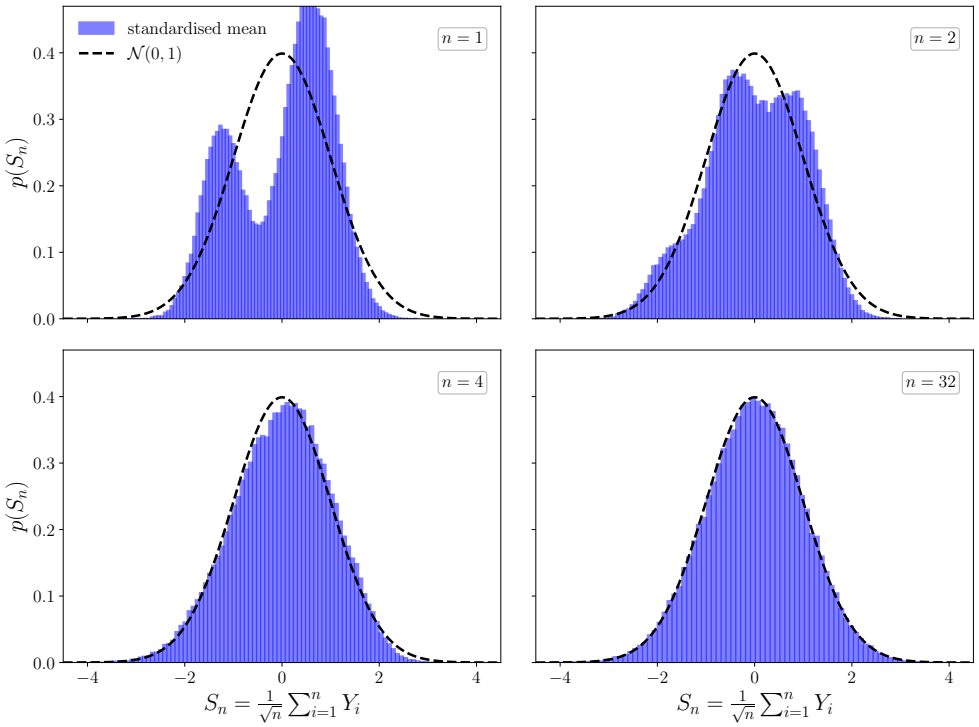


Figure 3.6: *Convergence to the central limit theorem of a sum of iid random variables distributed in the same way as in Figure 3.4. You can find the script to generate the plot on GitHub.*

3.7 Example: sample variance

The second example we consider is estimating the variance of a given PDF. We will see that this can be very subtle, and one has to distinguish between two cases.

Let us first assume that the underlying PDF $f(x)$ has a known mean μ. We can

define the estimator for the variance σ^2 as

$$\hat{\sigma}^2 \coloneqq \frac{1}{n}\sum_{i=1}^{n}(X_i - \mu)^2, \tag{3.24}$$

i.e. we define our estimator using the known mean of the distribution. We can rewrite this slightly

$$\hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n} X_i^2 - \frac{2}{n}\mu\sum_{i=1}^{n} X_i + \frac{1}{n}\sum_{i=1}^{n}\mu^2 = \frac{1}{n}\sum_{i=1}^{n} X_i^2 - 2\mu\hat{\mu} + \mu^2 \tag{3.25}$$

and again apply the expectation value:

$$\mathbb{E}(\hat{\sigma}^2) = \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}\left(X_i^2 - 2\mu\hat{\mu} + \mu^2\right)\right] = \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}(X_i^2) - \mu^2 = \frac{1}{n}\sum_{i=1}^{n}(\mu^2 + \sigma^2) - \mu^2 = \sigma^2. \tag{3.26}$$

Therefore, if we know the mean of the distribution, the sample variance is an unbiased estimator for the population variance.

We can now play the same game in the case where the mean of the underlying distribution is unknown. Let us first naively use the same estimator as before but replace the known mean μ with the estimated mean $\hat{\mu}$:

$$\hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n}(X_i - \hat{\mu})^2. \tag{3.27}$$

Its expectation value is now given by:

$$\begin{aligned}
\mathbb{E}(\hat{\sigma}^2) &= \mathbb{E}\left(\frac{1}{n}\sum_{i=1}^{n}(X_i - \hat{\mu})^2\right) \\
&= \mathbb{E}\left(\frac{1}{n}\sum_{i=1}^{n} X_i^2 - \hat{\mu}^2\right) \\
&= \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}(X_i^2) - \mathbb{E}(\hat{\mu}^2) \\
&= \frac{1}{n}\sum_{i=1}^{n}(\mu^2 + \sigma^2) - \left(\mu^2 + \frac{\sigma^2}{n}\right) = \sigma^2 - \frac{\sigma^2}{n} = \frac{n-1}{n}\sigma^2.
\end{aligned} \tag{3.28}$$

Thus, if the mean is unknown, this estimator remains asymptotically unbiased. However, for finite n there is a bias. There is a simple fix, though, which we can read off from the above result, an unbiased estimator of the variance with unknown mean:

$$\hat{\sigma}^2 \coloneqq \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \hat{\mu})^2. \tag{3.29}$$

4 Maximum likelihood and inference

In Section 3, we introduced the concept of an estimator and discussed desirable properties such as unbiasedness and consistency. Recall, however, that we postulated the estimators for the sample mean and variance. In other words, it was left completely open how to construct the estimator systematically. Maximum likelihood estimation (MLE) provides exactly such a systematic approach, and we will see that the maximum likelihood estimator (also MLE) has excellent asymptotic properties and reproduces results from the previous section. The building block is the likelihood function, which will also be essential in Bayesian inference (Section 5).

4.1 The likelihood function

Suppose we observe n iid realisations $x_1, \dots, x_n$ of a random variable $X \sim f(x|\boldsymbol{\theta})$, where $\boldsymbol{\theta}$ again denotes the parameters of the distribution. Note that each x_i can, in principle, be a random vector, but f is still a real number. Because the observations are independent, their joint PDF factorises:

$$f(x_1, \dots, x_n|\boldsymbol{\theta}) = \prod_{i=1}^{n} f(x_i|\boldsymbol{\theta}). \tag{4.1}$$

This is simply the joint PDF from Section 2, which we interpreted as the distribution of the data at fixed parameters. For example, you can think of this as a product of many Gaussians, with all of them having a fixed but unknown mean and variance. Let us make a crucial conceptual shift and interpret Equation (4.1) not as a function of the data, $\{x_i\}$ but instead as a function of $\boldsymbol{\theta}$ for fixed, observed data. We call

$$\mathcal{L}(\boldsymbol{\theta}) \equiv \mathcal{L}(\boldsymbol{\theta}|x_1, \dots, x_n) \coloneqq \prod_{i=1}^{n} f(x_i|\boldsymbol{\theta}), \tag{4.2}$$

the *likelihood function*. For discrete distributions, f is replaced by the PMF, but the definition is otherwise identical. It is important to note that $\mathcal{L}(\boldsymbol{\theta})$ is, at this stage, not a probability distribution over $\boldsymbol{\theta}$, it is simply a real-valued function over the parameter space which measures how compatible fixed observed data is with each value of the parameters. In particular

$$\int \mathcal{L}(\boldsymbol{\theta})\, \mathrm{d}\boldsymbol{\theta} \neq 1, \tag{4.3}$$

in general, it therefore does not qualify as a probability distribution. Note that $\mathrm{d}\boldsymbol{\theta}$ denotes the volume element of the parameter space. So for a d-dimensional Euclidean parameter space $\mathrm{d}\boldsymbol{\theta} = \mathrm{d}^d\theta$. The intuitive motivation is now to find the parameters $\hat{\boldsymbol{\theta}}$ that maximise Equation (4.2), as these will best describe the given data.

In practice, it is inconvenient to work directly with the likelihood function: as n grows large, the product of very small numbers can cause underflow errors very quickly. Recall that the smallest representable standard floating-point numbers are around 10^{-38} for 32-bit floats and 10^{-308} for 64-bit doubles. Furthermore, products are mathematically just more difficult to handle than sums. Since the logarithm is a strictly monotonically increasing function, maximising $\mathcal{L}(\boldsymbol{\theta})$ is equivalent to maximising its logarithm. Additionally, it will translate products into sums. We thus define the *log-likelihood*

$$L(\boldsymbol{\theta}) \coloneqq \ln \mathcal{L}(\boldsymbol{\theta}) = \sum_{i=1}^{n} \ln f(x_i|\boldsymbol{\theta}) = \sum_{i=1}^{n} L_i(\boldsymbol{\theta}). \tag{4.4}$$

From this point onward, we will work primarily with $L(\boldsymbol{\theta})$ for convenience, as discussed. However, there are also some fundamental reasons why this is a good choice, as we will see in **??**.

4.2 Maximum likelihood estimation

After having defined the log-likelihood, we can define the MLE, $\hat{\boldsymbol{\theta}}_{\mathrm{MLE}}$, as the value of $\boldsymbol{\theta}$ that maximises the likelihood (equivalently, the log-likelihood):

$$\hat{\boldsymbol{\theta}}_{\mathrm{MLE}} \coloneqq \arg\max_{\boldsymbol{\theta}\in\Theta} \; L(\boldsymbol{\theta}) = \arg\max_{\boldsymbol{\theta}\in\Theta} \sum_{i=1}^{n} \ln f(x_i|\boldsymbol{\theta}), \tag{4.5}$$

where Θ is the parameter space, so in the case of a simple Gaussian, you can think of it as $\mathbb{R}^2$. It can, however, house as many parameters as you like. Typical parameter spaces in astrophysics and cosmology contain upwards of twenty parameters.

Provided that $L(\boldsymbol{\theta})$ is differentiable, the MLE can formally be found by solving the following set of equations:

$$\nabla_{\boldsymbol{\theta}} L\big|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}_{\mathrm{MLE}}} = 0, \tag{4.6}$$

subject to verification that the stationary point is indeed a maximum. Throughout, we will often use index notation from now on: the components of the operator $\nabla_{\boldsymbol{\theta}}$ in a coordinate basis of the parameter vector space are:

$$\frac{\partial}{\partial\theta_\alpha} \equiv \partial^\alpha. \tag{4.7}$$

Moving forward, we will use Greek letters to denote the components of the parameter vector $\boldsymbol{\theta}$. We will also use the sum convention, so that repeated indices appearing as both sub- and superscripts are summed over. We can evaluate the vector-valued function on the left-hand side of Equation (4.6) at an arbitrary $\boldsymbol{\theta}$, the *score function* with components:

$$s^\alpha(\boldsymbol{\theta}) \coloneqq \partial^\alpha L(\boldsymbol{\theta}) = \sum_{i=1}^{n} \partial^\alpha L_i(\boldsymbol{\theta}). \tag{4.8}$$

Let us take the expectation of the score function at the true parameters $\boldsymbol{\theta}_0$:

$$\begin{aligned}
\langle s^\alpha(\boldsymbol{\theta}_0)\rangle &= \sum_{i=1}^{n} \langle \partial^\alpha L_i(\boldsymbol{\theta}_0)\rangle \\
&= \sum_{i=1}^{n} \int f(x_i|\boldsymbol{\theta}_0)\, \partial^\alpha L_i(\boldsymbol{\theta}_0)\, \mathrm{d}x_i \\
&= \sum_{i=1}^{n} \int \partial^\alpha f(x_i|\boldsymbol{\theta}_0)\, \mathrm{d}x_i \\
&= \sum_{i=1}^{n} \partial^\alpha \int f(x_i|\boldsymbol{\theta}_0)\, \mathrm{d}x_i \\
&= 0.
\end{aligned} \tag{4.9}$$

Here, we assumed regularity conditions (smoothness of the log-likelihood, identifiability of the model, and finite information), which ensure that the estimator is well-behaved, allowing us to interchange differentiation and integration. It is worth recalling what is random in the expressions above, because the notation can be compact. Once the data $x_1, \ldots, x_n$ have been observed, the log-likelihood is a deterministic function of θ. There is no randomness in it; it lives entirely in the data,

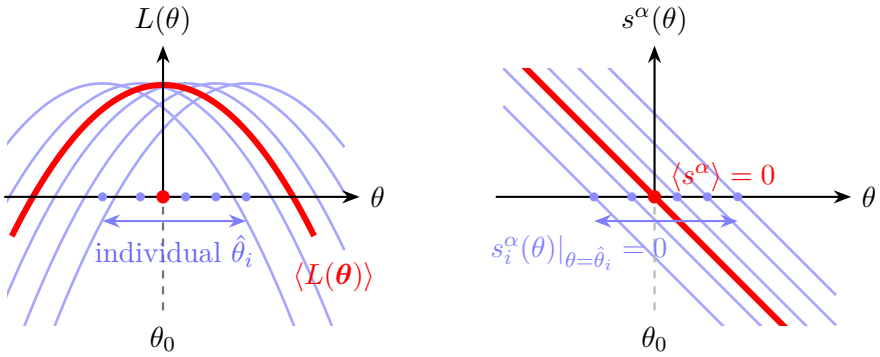


Figure 4.1: *Left: log-likelihood for different realisations of the data in light blue, each giving an individual MLE $\hat{\boldsymbol{\theta}}_i$. Taking the expectation over many realisations of the data returns the true value $\boldsymbol{\theta}_0$. Right: the score function for different realisations, where it vanishes for different values of θ. Only upon taking the expectation, it vanishes at the true parameters $\boldsymbol{\theta}_0$.*

and enters only when we think about what $L(\boldsymbol{\theta})$ would look like under a different draw from the same experiment. This is depicted on the left side of Figure 4.1, where the log-likelihood $L(\boldsymbol{\theta})$ is shown for different realisations of the data, and each realisation yields a different curve and a different MLE. In that sense, before the data are observed, $L(\boldsymbol{\theta})$ is a random function of $\boldsymbol{\theta}$ as it inherits the randomness of $X_1, \ldots, X_n$. The same is true for the score: evaluated at a fixed $\boldsymbol{\theta}$, it is a sum of n iid random variables. This is why we can apply results such as the central limit theorem to it.

The expectation is taken over repeated draws of the data from the true model $f(x|\boldsymbol{\theta}_0)$, and the score vanishes upon this averaging procedure. For any single dataset, however, $s(\boldsymbol{\theta}_0)$, will generically differ from zero, because the observed sample mean will not coincide exactly with the true mean. This finite-sample fluctuation of the score at θ_0 is what Figure 4.1 shows: each individual score line crosses zero at a different position.

4.3 Examples

Let us consider two examples of the MLE to illustrate a few points more clearly. In particular, let us rederive the results from Section 3, which were, at that stage, rather postulated.

4.3.1 Gaussian with unknown mean

Let $X_1, \ldots, X_n \sim \mathcal{N}(\mu, \sigma^2)$ with σ^2 known. We want to estimate the population mean μ. The log-likelihood is therefore only a function of a single parameter:

$$L(\mu) = -\frac{n}{2}\ln(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i - \mu)^2. \tag{4.10}$$

Setting the score to zero,

$$\begin{aligned} \frac{\partial L}{\partial \mu} &= \frac{1}{\sigma^2}\sum_{i=1}^{n}(x_i - \mu) = 0 \\ \Rightarrow\ \hat{\mu}_{\mathrm{MLE}} &= \frac{1}{n}\sum_{i=1}^{n} x_i = \hat{\mu}\ . \end{aligned} \tag{4.11}$$

In conclusion, the MLE for the mean is simply the sample mean, as in Section 3, confirming that it is the natural estimator for the mean.

4.3.2 Gaussian with unknown mean and variance

Now, let both μ and σ^2 be unknown, so $\boldsymbol{\theta} = (\mu, \sigma^2)^\top$. The log-likelihood is still the same as in Equation (4.10) but treated as a function of μ and σ^2. Setting the partial derivative of with respect to μ to zero, of course, yields again the sample mean. For σ^2 we find, however:

$$\begin{aligned} \frac{\partial L}{\partial(\sigma^2)} &= -\frac{n}{2\sigma^2} + \frac{1}{2(\sigma^2)^2}\sum_{i=1}^{n}(x_i - \mu)^2 = 0 \\ &\Rightarrow \sigma^2\frac{n}{2} = \frac{1}{2}\sum_{i=1}^{n}(x_i - \mu)^2 \\ &\Rightarrow \hat{\sigma}^2_{\text{MLE}} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{\mu})^2. \end{aligned} \tag{4.12}$$

Crucially, this is the estimator with the factor $1/n$ and not the unbiased $1/(n-1)$ estimator from the previous section, Equation (3.29). The MLE is therefore biased for σ^2, although it is asymptotically unbiased. This holds in general: the MLE is not guaranteed to be unbiased in finite samples.

4.4 Asymptotic properties of the MLE

The MLE's power lies in its behaviour as $n \to \infty$. Under standard regularity conditions, the following three properties hold:

i) **Consistency**: The MLE converges in probability to the true parameter:

$$\hat{\boldsymbol{\theta}}_{\text{MLE}} \xrightarrow{\mathbb{P}} \boldsymbol{\theta}_0 \quad \text{as } n \to \infty. \tag{4.13}$$

Intuitively, the log-likelihood is a sum of n iid terms, each with expectation $\langle \ln f(X|\boldsymbol{\theta})\rangle$. By the law of large numbers, the average converges to its expectation, which is uniquely maximised at the true $\boldsymbol{\theta}_0$.

ii) **Asymptotic normality**: The standardised MLE converges in distribution (written as "$\xrightarrow{\text{d}}$") to a Gaussian:

$$\sqrt{n}\,(\hat{\boldsymbol{\theta}}_{\text{MLE}} - \boldsymbol{\theta}_0) \xrightarrow{\text{d}} \mathcal{N}(0,\ \boldsymbol{\Sigma}(\boldsymbol{\theta}_0))\,. \tag{4.14}$$

The $\sqrt{n}$ scaling is familiar from the central limit theorem. We will discuss what determines that Gaussian's covariance, $\boldsymbol{\Sigma}(\boldsymbol{\theta}_0)$, in Sections 4.5 and 4.6.

iii) **Asymptotic efficiency**: The MLE achieves the minimum asymptotic variance: no consistent estimator can have a smaller asymptotic variance, making the MLE asymptotically efficient.

4.5 Fisher information and the variance of the score

We have established that the MLE is asymptotically unbiased. The next question we would like to answer is the uncertainty of the estimator. In particular, we would like to know the variance of $\boldsymbol{\theta}$ and thus the precision with which $\boldsymbol{\theta}$ can be estimated

from the data. To understand this, let us expand the log-likelihood around the true parameters $\boldsymbol{\theta}_0$:

$$L(\boldsymbol{\theta}) = L(\boldsymbol{\theta})\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_0} + \partial_\alpha L(\boldsymbol{\theta})\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_0} (\theta^\alpha - \theta_0^\alpha) + \frac{1}{2}\underbrace{\partial_\alpha\partial_\beta L(\boldsymbol{\theta})\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_0}}_{\equiv R_{\alpha\beta}(\boldsymbol{\theta}_0)} (\theta^\alpha - \theta_0^\alpha)(\theta^\beta - \theta_0^\beta) + \dots . \tag{4.15}$$

The first term is the log-likelihood evaluated at the true parameter values. This is a random constant shifting the curve up or down, depending on the specific realisation of the data. It, however, does not affect the position of the maximum. Second, we have the score evaluated at the true parameters. Note that this only vanishes at the MLE by construction or upon expectation over the data. The last term is the curvature, $R_{\alpha\beta}$, of the log-likelihood at the truth. Because we are at (or near) a maximum, the second derivatives (with respect to the same parameter) are negative. The magnitude of the curvature controls how sharply peaked the log-likelihood is, and therefore how tightly the data constrain $\boldsymbol{\theta}$. All terms are illustrated on the left-hand side of Figure 4.2. Setting the derivative of Equation (4.15) to zero and

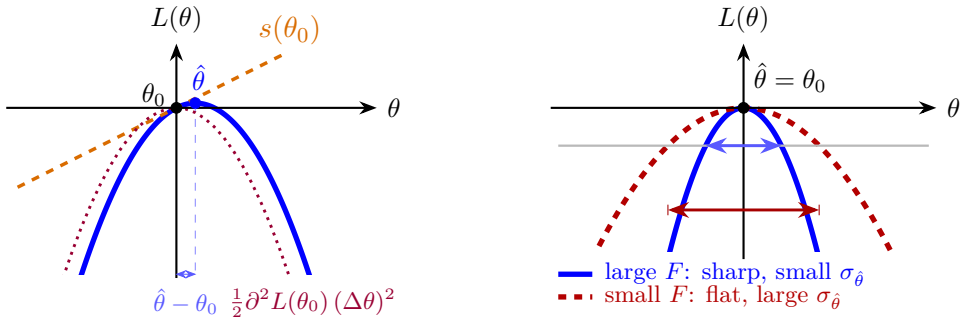


Figure 4.2: *Left: Content of the Taylor expansion, Equation (4.15) for a single log-likelihood realisation, i.e. for a single set of data. The blue curve is $L(\boldsymbol{\theta})$, the orange dashed line is the tangent at $\boldsymbol{\theta}_0$, whose slope is the score $s(\theta_0)$ (which does not vanish because we did not take the expectation), and the purple dotted curve is the quadratic curvature term from the third term in the Taylor expansion. The MLE $\hat{\theta}$ is displaced from θ_0 by the ratio of the score to the curvature. Right: Two log-likelihoods with the same MLE $\hat{\theta} = \theta_0$ but different Fisher information. A larger $\mathcal{F}$ (blue, solid) produces a sharper peak. The uncertainty is read off as the width of the log-likelihood at the $\Delta\ell = -\frac{1}{2}$ level (grey horizontal line).*

solving for $\hat{\boldsymbol{\theta}}$ gives:

$$0 = s_\alpha(\boldsymbol{\theta}_0) + R_{\alpha\beta}(\boldsymbol{\theta}_0)(\hat{\theta}^\beta_{\text{MLE}} - \theta_0^\beta) \;\Rightarrow\; \hat{\theta}^\alpha_{\text{MLE}} - \theta_0^\alpha = -(R^{-1}(\boldsymbol{\theta}_0))^{\alpha\beta} s_\beta(\boldsymbol{\theta}_0) \tag{4.16}$$

This equation contains all that is needed to understand the MLE. The numerator is a sum of n iid random variables with vanishing mean, so, by the CLT, it fluctuates $\propto 1/\sqrt{n}$. The denominator is itself a sample average of the Hessian of the log-likelihoods, i.e. its curvature. Technically, these are, of course, elements of vectors and matrices, respectively.

To proceed, let us examine the curvature more closely by starting with the same exercise as in Equation (4.9), but this time differentiating the normalisation condition

of $f(x|\boldsymbol{\theta}_0) \equiv f$ twice:

$$\begin{aligned} 0 &= \int \partial_\alpha \partial_\beta f(x|\boldsymbol{\theta}_0)\, \mathrm{d}x \\ &= \int \partial_\alpha \partial_\beta \exp\left[\ln(f)\right]\, \mathrm{d}x \\ &= \int \partial_\alpha \left[f\, \partial_\beta \ln f\right]\, \mathrm{d}x \\ &= \int \left[\partial_\alpha f\, \partial_\beta \ln f + f\, \partial_\alpha \partial_\beta \ln f\right]\, \mathrm{d}x \\ &= \int \left[f\, \partial_\alpha \ln f\, \partial_\beta \ln f + f\, \partial_\alpha \partial_\beta \ln f\right]\, \mathrm{d}x\,. \end{aligned} \tag{4.17}$$

In other words, we arrive at the following important result

$$\langle \partial_\alpha \partial_\beta L \rangle = -\langle \partial_\alpha L\, \partial_\beta L \rangle\ . \tag{4.18}$$

Since the expectation value of the score vanishes, Equation (4.9), the negative curvature is exactly the variance of the score. We will call this quantity the *Fisher information matrix*:

$$F_{\alpha\beta}(\boldsymbol{\theta}) \coloneqq \langle \partial_\alpha L(\boldsymbol{\theta})\, \partial_\beta L(\boldsymbol{\theta}) \rangle = -\langle \partial_\alpha \partial_\beta L(\boldsymbol{\theta}) \rangle\ . \tag{4.19}$$

Equation (4.17) was defined for a single realisation of the iid. However, it simply generalises to n iid by the linearity of the derivative, expectation value and log-likelihood. For n iid, the Fisher information matrix is:

$$F^{(n)}_{\alpha\beta}(\boldsymbol{\theta}) = \sum_{i=1}^{n} F^{(i)}_{\alpha\beta}(\boldsymbol{\theta}) = n F_{\alpha\beta}(\boldsymbol{\theta})\ , \tag{4.20}$$

since all individual Fisher matrices are the same, as they are population averages. The expression is also very intuitive: independent information should be additive.

Let us go back to Equation (4.16): both R and s are in principle random variables. We aim to replace them with their respective population limits, i.e., the expectations. However, replacing the score with its expectation yields no new information, since we already showed that it converges to zero; all we would be left with is $\hat{\theta}^\beta_{\mathrm{MLE}} - \theta^\beta_0 \approx 0$, which tells us nothing new. It is the statistical fluctuation of the numerator that we would like to characterise. Let us instead look at the denominator. At the true value:

$$\begin{aligned} -\frac{1}{n} R_{\alpha\beta}(\boldsymbol{\theta}_0) = -\frac{1}{n} \sum_{i=1}^{n} \partial_\alpha \partial_\beta \ln f(x_i|\boldsymbol{\theta}_0) &\approx -\left\langle R_{\alpha\beta} \right\rangle (\boldsymbol{\theta}_0)(1 + \mathcal{O}(1/\sqrt{n})) \\ &= F_{\alpha\beta}(\boldsymbol{\theta}_0)(1 + \mathcal{O}(1/\sqrt{n})), \end{aligned} \tag{4.21}$$

because the negative curvature itself is the sample mean and therefore its fluctuation follows the law of large numbers as shown in Equation (3.22). Going back into Equation (4.16) we find by approximating the denominator:

$$\hat{\theta}^\beta_{\mathrm{MLE}} - \theta^\beta_0 \approx (F^{-1}(\boldsymbol{\theta}_0))^{\alpha\beta}\, \frac{s_\alpha(\boldsymbol{\theta}_0)}{n}\, (1 - \mathcal{O}(1/\sqrt{n}))\,. \tag{4.22}$$

Therefore, we conclude that the scatter of the MLE is given by:

$$\hat{\theta}^\beta_{\mathrm{MLE}} - \theta^\beta_0 \approx (F^{-1}(\boldsymbol{\theta}_0))^{\alpha\beta}\, \bar{s}_\alpha(\boldsymbol{\theta}_0)\ , \tag{4.23}$$

where we defined the average score $\bar{s}_\alpha = s_\alpha/n$. The finding is consistent with our earlier result: the Fisher matrix is the variance of the score, which defines the MLE.

Note that $\boldsymbol{F}$ is just a matrix and in itself does not contain any randomness. As a last step, let us calculate the variance of $\hat{\theta}^{\beta}_{\mathrm{MLE}}$:

$$\begin{aligned}
\sigma^2_{\hat{\theta}^{\beta}_{\mathrm{MLE}}} &= \left\langle (\hat{\theta}^{\beta}_{\mathrm{MLE}} - \theta^{\beta}_0)(\hat{\theta}^{\beta}_{\mathrm{MLE}} - \theta^{\beta}_0) \right\rangle \\
&= \frac{1}{n^2} \left\langle (F^{-1})^{\alpha\beta} s_\alpha \, (F^{-1})^{\gamma\beta} s_\gamma \right\rangle \\
&= \frac{1}{n^2} (F^{-1})^{\alpha\beta} (F^{-1})^{\gamma\beta} \left\langle s_\alpha \, s_\gamma \right\rangle \\
&= \frac{1}{n} (F^{-1})^{\alpha\beta} (F^{-1})^{\gamma\beta} F_{\alpha\gamma} \\
&= \frac{1}{n} (F^{-1})^{\beta\beta} \, ,
\end{aligned} \tag{4.24}$$

due to the scaling of $F_{\alpha\beta}$ for n iid, Equation (4.20). The precision of the MLE therefore scales linearly with both n and $\boldsymbol{F}$. A large Fisher information means a sharply curved log-likelihood, so the data strongly constrain $\boldsymbol{\theta}$ and the MLE is precise. This is shown on the right-hand side of Figure 4.2.

4.6 The Cramér–Rao bound

We have seen that the Fisher matrix sets the uncertainty of the MLE. In fact, it sets a fundamental limit on how precisely any unbiased estimator can recover $\boldsymbol{\theta}$. This is expressed by the *Cramér–Rao bound*. Let us formulate it first in one dimension, where it states that for any unbiased estimator $\hat{\theta}$ based on an iid sample of size n:

$$\sigma^2_{\hat{\theta}} \geq \frac{1}{nF(\theta)} \, . \tag{4.25}$$

Any estimator that saturates this inequality is called *efficient*. The proof is not difficult but can be somewhat lengthy and is presented below.

Proof of the Cramér–Rao bound

We prove the one-parameter version. Define the random variables

$$U = s(\theta) = \frac{\partial L(\theta)}{\partial \theta}, \qquad V = \hat{\theta} - \theta.$$

From the unbiasedness condition, $\langle \hat{\theta} \rangle = \theta$, we write:

$$\begin{aligned}
\mathrm{Cov}(U, V) &= \langle s(\theta)(\hat{\theta} - \theta) \rangle \\
&= \int f \, \partial_\theta \ln f \, (\hat{\theta} - \theta) \, \mathrm{d}x \\
&= \int \partial_\theta f \, (\hat{\theta} - \theta) \, \mathrm{d}x \\
&= \partial_\theta \int f \, (\hat{\theta} - \theta) \, \mathrm{d}x + 1 \\
&= \partial_\theta \langle \hat{\theta} \rangle - \int f \, \mathrm{d}x + 1 \\
&= \partial_\theta \langle \hat{\theta} \rangle = 1
\end{aligned}$$

To make progress, we have to prove the *Cauchy–Schwarz inequality*. Let us define the following random variable: $Z \coloneqq \alpha X + Y$, for $\alpha \in \mathbb{R}$, then:

$$\langle Z^2 \rangle = \alpha^2 \langle X^2 \rangle + 2\alpha \langle XY \rangle + \langle Y^2 \rangle \geq 0 \, .$$

This quadratic equation has roots at:

$$\alpha_{\pm} = \frac{b}{2a} \pm \sqrt{\frac{b^2}{4a^2} - \frac{c}{a}} \,,$$

thus for it to be non-negative, we must have $a > 0$, and there must not be two real roots, so:

$$b^2 < 4ac \quad \Rightarrow \quad \langle XY \rangle^2 \leq \langle Y^2 \rangle \langle X^2 \rangle$$

upon plugging back the definitions. This is the Cauchy–Schwarz inequality. Applying it to the result from above:

$$1 \;\leq\; \langle U^2 \rangle \langle V^2 \rangle = F(\theta) \langle (\hat{\theta} - \theta)^2 \rangle,$$

which rearranges to Equation (4.25) for n iid random variables.

The Cramér–Rao bound connects directly to our earlier discussion of the MSE. Because the MSE of an unbiased estimator equals its variance, Equation (3.16), no unbiased estimator can have an MSE smaller than $1/(n\,F(\boldsymbol{\theta}))$. The MLE asymptotically achieves this bound, making it asymptotically efficient.

If we are dealing with a vector of parameters, $\boldsymbol{\theta}$, instead, the Cramér–Rao bound becomes a statement about the covariance matrix instead:

$$\sigma^2_{\hat{\theta}_\alpha} \geq \frac{(\boldsymbol{F}^{-1})_{\alpha\alpha}}{n} \,. \tag{4.26}$$

Note that this means: take the inverse of the Fisher information matrix and then pick the α-th diagonal entry of that matrix. More formally one can write this as $\mathrm{Cov}(\hat{\boldsymbol{\theta}}) \succeq [\, n\, \boldsymbol{F}(\boldsymbol{\theta})\,]^{-1}$, meaning that the eigenvalues of the covariance matrix must satisfy this inequality. All of this has an important practical consequence: for large n, we can approximate Equation (4.14)

$$\hat{\boldsymbol{\theta}}_{\mathrm{MLE}} \sim \mathcal{N}\!\left(\hat{\boldsymbol{\theta}}_{\mathrm{MLE}},\; \frac{1}{n} \boldsymbol{F}^{-1}(\hat{\boldsymbol{\theta}}_{\mathrm{MLE}})\right) \,. \tag{4.27}$$

Practically, we evaluated the Fisher matrix at the MLE because $\boldsymbol{\theta}_0$ is unknown. Comparing this with Equation (3.11) already tells us that the inverse covariance matrix is the Fisher information matrix.

4.7 One-dimensional confidence intervals

Let us start with the one-dimensional *frequentist confidence intervals* in the following way: a $100(1-\alpha)\%$ confidence interval $[\hat{\theta}_\mathrm{l}, \hat{\theta}_\mathrm{u}]$ for a parameter θ_0 is a random interval (depending on the data) such that

$$\mathbb{P}\!\left(\hat{\theta}_\mathrm{l} \leq \theta_0 \leq \hat{\theta}_\mathrm{u}\right) = 1 - \alpha \,. \tag{4.28}$$

Note: it is the interval that is random, not θ_0. After observing the data, the true θ_0 either lies in the realised interval or it does not. The $100(1-\alpha)\%$ statement refers to the long-run coverage over repeated experiments.

This frequentist interpretation of uncertainty can be counterintuitive, as we have already noted in the introduction. It is unclear what is meant by "repeated experiments". For now, it is important to appreciate the following: a frequentist confidence interval controls long-run coverage. This is indicated in Figure 4.3, which shows the MLE and its 95% confidence interval for independent realisations of the

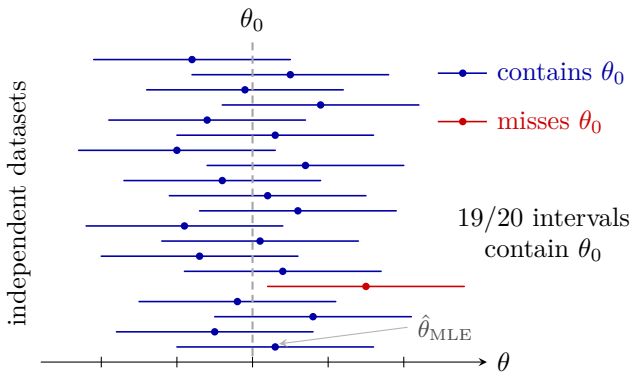


Figure 4.3: *Twenty 95% confidence intervals computed from independent datasets. The true parameter θ_0 (dashed) is fixed. Each interval either contains it or does not contain it. In the long run, 95% of such intervals will contain θ_0.*

data, i.e. repeated experiments. $\hat{\theta}_{\mathrm{MLE}}$ is a random number and fluctuates, so does the corresponding interval. In the case of the 95% interval, the frequentist interpretation states that the true parameter θ_0 must be covered 95% of the time. In the plot shown, this occurs 19 out of 20 times.

4.8 The χ^2 distribution

Before proceeding, let us consider a sum of standard Gaussian iid random variables $Z_1, \ldots, Z_k \sim \mathcal{N}(0,1)$. We can define the new random variable

$$Q = \sum_{i=1}^{k} Z_i^2, \tag{4.29}$$

which will follow a χ^2 distribution with k degrees of freedom written $Q \sim \chi^2_k$. Its probability density function is

$$f(x;\, k) = \frac{x^{k/2-1}\, e^{-x/2}}{2^{k/2}\, \Gamma(k/2)}, \qquad x > 0, \tag{4.30}$$

and $f(x;k) = 0$ for $x \leq 0$, where Γ denotes the Gamma function. The χ^2 distribution appears frequently in statistics because many estimators involve sums of squared deviations. The key properties of the χ^2 distribution are listed below:

i) **Mean and variance:** $\langle Q \rangle = k$ and $\mathrm{Var}(Q) = 2k$. Both increase linearly with the number of degrees of freedom.

ii) **Additivity:** If $Q_1 \sim \chi^2_k$ and $Q_2 \sim \chi^2_m$ are independent, then $Q_1 + Q_2 \sim \chi^2_{k+m}$.

iii) **Shape:** For $k = 1$ the distribution is maximally skewed with a singularity at the origin. For $k = 2$ it reduces to an exponential distribution. For $k \geq 3$ the density vanishes at zero, rises to a mode at $x = k - 2$, and decays exponentially. As $k \to \infty$, by the CLT applied to the definition Equation (4.29), $(Q - k)/\sqrt{2k} \xrightarrow{\mathrm{d}} \mathcal{N}(0,1)$, so the χ^2 distribution becomes increasingly Gaussian and symmetric.

Let us dwell on those points for a little bit: *i*) tells us that we would expect a quadratic difference of k for k iid. This statement takes a very intuitive meaning

when viewed as a single realisation of each random variable from an underlying Gaussian distribution. The expectation is that the data scatter within the central region of the individual PDFs, i.e., the 1σ intervals. In other words, upon averaging, each individual measurement will have the variance of the underlying Gaussian and therefore the scatter, $(x-\mu)^2$, is exactly the variance of the individual measurement as we already established in Section 3. Therefore, we expect that k measurements will have a scatter of k. The second point can be directly seen by simply splitting the sum in Equation (4.29) into two partial and independent sums. Lastly, *iii*) is shown in particular in Figure 4.4, where we show the χ^2 distribution for increasing numbers of k as a solid blue line. An overlaid dashed line indicates the corresponding normal distribution. More generally, any quadratic form in Gaussian random variables $\boldsymbol{x}$

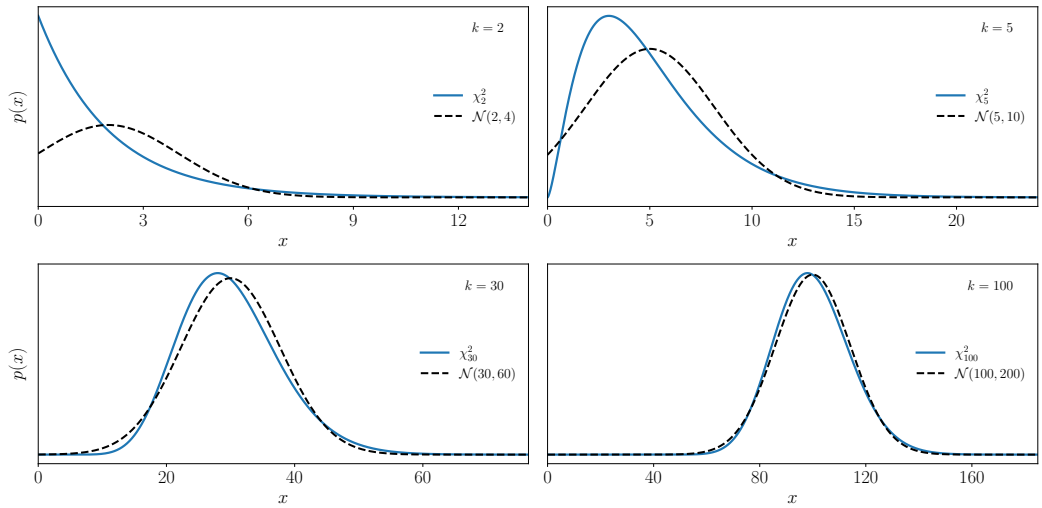


Figure 4.4: *PDF of the χ^2 distribution, χ^2_k, compared with its Gaussian approximation, $\mathcal{N}(k, 2k)$, for increasing degrees of freedom ($k = 2, 5, 30, 100$). The agreement improves with k. The notebook to generate this plot can be found on GitHub.*

$$Q := x_\alpha M^\alpha_\beta x^\beta \quad \alpha, \beta = 1, \ldots, d, \tag{4.31}$$

follows a (possibly non-central) χ^2 distribution with d degrees of freedom χ^2_d. This can be seen by first writing:

$$z^\alpha = A^\alpha_\beta x^\beta, \tag{4.32}$$

where A^α_β is the matrix root of $\boldsymbol{M}$ so that

$$A^\alpha_\beta A^\beta_\gamma = M^\alpha_\gamma \,. \tag{4.33}$$

If we now rewrite the quadratic as follows:

$$Q = x_\alpha M^\alpha_\beta x^\beta = A^\gamma_\alpha x_\gamma A^\alpha_\beta x^\beta = z_\alpha z^\alpha \,, \tag{4.34}$$

which is exactly the definition of Equation (4.29), showing that any general quadratic form will follow a χ^2 distribution with d degrees of freedom. Let us illustrate this result in Figure 4.5. We draw repeated samples from a six-dimensional multivariate Gaussian $\boldsymbol{x} \sim \mathcal{N}_6(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ and for each draw $\boldsymbol{x}$ compute the quadratic form $Q = (\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\boldsymbol{x}-\boldsymbol{\mu})$. Creating a histogram of Q over a large number of such draws yields the blue distribution, which agrees perfectly with the theoretical χ^2_6 prediction shown as the dashed line. Let us now introduce a nonlinear perturbation to the

samples: $x_i' = x_i + \epsilon(x_i^2 - 1)$, for $\epsilon = 0.05$ (orange) and $\epsilon = 0.10$ (green). The subtraction of unity ensures that the perturbation has zero mean for a standard Gaussian variate, so the population mean is preserved to leading order in ϵ. However, the perturbation introduces skewness and excess kurtosis: it is no longer a linear transformation of a Gaussian vector, and therefore $\boldsymbol{x}'$ is not itself Gaussian. Even a perturbation as small as $\epsilon = 0.05$ produces a visible distortion of the distribution of Q, concentrated primarily in the tails.

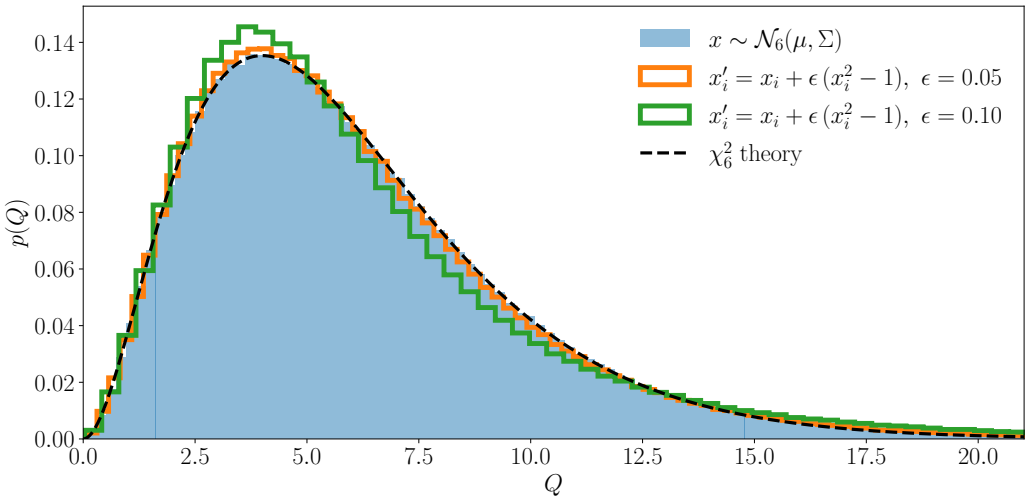


Figure 4.5: *Distribution of the quadratic form $Q = (\boldsymbol{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x} - \boldsymbol{\mu})$ under Gaussian and perturbed samples from a 6-dimensional multivariate normal. The baseline (blue histogram) shows perfect agreement with the theoretical χ^2_6 distribution. Two nonlinear perturbations of increasing strength, $x_i' = x_i + \epsilon(x_i^2 - 1)$ with $\epsilon = 0.05$ (orange) and $\epsilon = 0.10$ (green), demonstrate how small departures from Gaussianity induce visible deviations from the chi-square prediction, with larger shifts in the tail behaviour. The notebook to generate this plot can be found on GitHub.*

4.9 Goodness of fit and p-values

For Gaussian data with covariance matrix $\boldsymbol{C}$, maximising the log-likelihood is equivalent to minimising the quadratic form

$$\chi^2(\boldsymbol{\theta}) = (\boldsymbol{x} - \boldsymbol{\mu}(\boldsymbol{\theta}))^\top \boldsymbol{C}^{-1}(\boldsymbol{x} - \boldsymbol{\mu}(\boldsymbol{\theta})), \tag{4.35}$$

which reduces to Equation (4.29) for diagonal $\boldsymbol{C} = \mathrm{diag}(\sigma_1^2, \ldots, \sigma_n^2)$. Since the residuals $\boldsymbol{x} - \boldsymbol{\mu}(\hat{\boldsymbol{\theta}}_{\mathrm{MLE}})$ are Gaussian and Equation (4.34) shows that any quadratic form in Gaussian random variables follows a χ^2 distribution, fitting d parameters imposes d constraints and gives

$$\chi^2_{\mathrm{min}} \sim \chi^2_\nu, \qquad \nu = n - d. \tag{4.36}$$

The *p-value* is the probability of obtaining a value at least as large under the true model,

$$p = 1 - F_{\chi^2_\nu}(\chi^2_{\mathrm{min}}), \tag{4.37}$$

with small p indicating a poor fit. Note that p is not the probability that the model is correct. It is purely a statement about the data given to the model, and as such, a fundamentally frequentist concept: it quantifies how often one would obtain a fit at

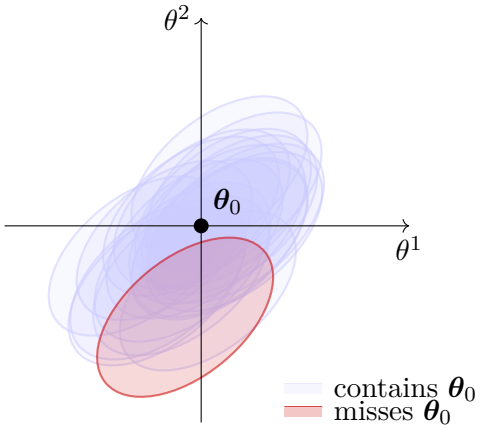


Figure 4.6: *Coverage in two dimensions.* 19/20 *of the ellipses contain* $\boldsymbol{\theta}_0$ *exactly as in the one-dimensional case in Figure 4.3.*

least this poor if the experiment were repeated many times under the true model. The Bayesian analogue, which instead assigns a probability directly to the model being correct, is discussed in Section 5. The *reduced chi-squared* $\chi^2_{\text{red}} = \chi^2_{\text{min}}/\nu$ has expectation 1 for a good fit, with $\chi^2_{\text{red}} \gg 1$ pointing to underfitting or underestimated uncertainties and $\chi^2_{\text{red}} \ll 1$ to overfitting or overestimated uncertainties.

4.10 Multi-parameter confidence regions

While the one-dimensional confidence interval of Equation (4.28) provides a clean picture for a single parameter, astrophysical and cosmological analyses involve tens or even hundreds of parameters simultaneously. Reporting d separate one-dimensional intervals is insufficient: it discards parameter correlations and can lead to inconsistencies. We now ask how to characterise uncertainty jointly across all parameters at once.

We know that the MLE converges to a Gaussian as shown in Equation (4.27) and that

$$Q(\boldsymbol{\theta}_0) = (\hat{\theta}_{\text{MLE}} - \theta_0)_\alpha n \boldsymbol{F}^\alpha_\beta(\boldsymbol{\theta}_0)(\hat{\theta}_{\text{MLE}} - \theta_0)^\beta \;\sim\; \chi^2_d. \tag{4.38}$$

Crucially, the distribution of $Q(\boldsymbol{\theta}_0)$ does not depend on the unknown value of $\boldsymbol{\theta}_0$: whatever the true parameters are, Q is always χ^2_d. Now, by definition of the CDF, the probability that χ^2_d falls below any threshold c is given by

$$\mathbb{P}\left(Q(\boldsymbol{\theta}_0) \leq c\right) = F_{\chi^2_d}(c). \tag{4.39}$$

We want this probability to equal exactly $1-\alpha$, so we choose c to be the $1-\alpha$ quantile of χ^2_d, i.e. the value $\chi^2_{d,1-\alpha}$ defined by $F_{\chi^2_d}(\chi^2_{d,1-\alpha}) = 1-\alpha$:

$$\mathbb{P}\left(Q(\boldsymbol{\theta}_0) \leq \chi^2_{d,1-\alpha}\right) = 1-\alpha. \tag{4.40}$$

Equation (4.40) is a statement about the random variable $\hat{\boldsymbol{\theta}}_{\text{MLE}}$, with $\boldsymbol{\theta}_0$ fixed. However, after observing the data the data, $\hat{\boldsymbol{\theta}}_{\text{MLE}}$ is fixed and $\boldsymbol{\theta}_0$ is the unknown. We can therefore read the inequality $Q(\boldsymbol{\theta}_0) \leq \chi^2_{d,1-\alpha}$ from the other direction: it holds if and only if $\boldsymbol{\theta}_0$ lies inside the set centred at $\hat{\boldsymbol{\theta}}_{\text{MLE}}$:

$$\mathcal{R}_\alpha \coloneqq \left\{\boldsymbol{\theta}_0 \in \Theta \;:\; (\hat{\theta}_{\text{MLE}} - \theta_0)_\gamma n \boldsymbol{F}^\gamma_\beta(\boldsymbol{\theta}_0)(\hat{\theta}_{\text{MLE}} - \theta_0)^\beta \;\leq\; \chi^2_{d,1-\alpha}\right\}. \tag{4.41}$$

Table 1: Example values for $\chi^2_{d,\,1-\alpha}$ defining the confidence region Equation (4.41) for coverage $1-\alpha$ and d free parameters. Equivalently, the log-likelihood must not drop below $\frac{1}{2}\chi^2_{d,\,1-\alpha}$ relative to its maximum, as in Equation (4.44). The row labels correspond to the 1σ, 2σ, 3σ conventions of the one-dimensional Gaussian.

Coverage $1-\alpha$	$d=1$	$d=2$	$d=3$	$d=4$	$d=5$
68.3%	1.00	2.30	3.53	4.72	5.89
95.4%	4.00	6.18	8.03	9.72	11.31
99.7%	9.00	11.83	14.16	16.25	18.21

Note that this is exactly in the spirit in which we introduced the likelihood in the first place: we treat the data as fixed rather than as a random variable. Combining with Equation (4.40) we see that:

$$\mathbb{P}\left(\boldsymbol{\theta}_0 \in \mathcal{R}_\alpha\right) = \mathbb{P}\left(Q(\boldsymbol{\theta}_0) \leq \chi^2_{d,1-\alpha}\right) = 1-\alpha. \tag{4.42}$$

The argument is now exactly as in the case of one-dimensional intervals: over repeated experiments, the random set $\mathcal{R}_\alpha$ contains the true parameter $\boldsymbol{\theta}_0$ with probability $1-\alpha$. Figure 4.6 shows this for a two-dimensional parameter space.

Let us understand the d-dimensional ellipsoid described by the set in Equation (4.41) a bit better by expanding the log-likelihood to second order around $\hat{\boldsymbol{\theta}}_{\mathrm{MLE}}$ and using the score vanishing at the MLE:

$$L(\boldsymbol{\theta}) \approx L(\hat{\boldsymbol{\theta}}_{\mathrm{MLE}}) - \frac{1}{2}(\boldsymbol{\theta}-\hat{\boldsymbol{\theta}})^\top F^{(n)}(\hat{\boldsymbol{\theta}}_{\mathrm{MLE}})\,(\boldsymbol{\theta}-\hat{\boldsymbol{\theta}}_{\mathrm{MLE}}). \tag{4.43}$$

Comparing with Equation (4.38), we see that $Q \approx -2\big[L(\boldsymbol{\theta}) - L(\hat{\boldsymbol{\theta}}_{\mathrm{MLE}})\big]$, so the confidence region becomes

$$\mathcal{R}_\alpha = \Big\{\boldsymbol{\theta}\in\Theta \;:\; L(\hat{\boldsymbol{\theta}}) - L(\boldsymbol{\theta}) \;\leq\; \tfrac{1}{2}\chi^2_{d,\,1-\alpha}\Big\}. \tag{4.44}$$

This is arguably the single most useful result in practical inference: the confidence region is the set of parameter values at which the log-likelihood has not dropped by more than $\frac{1}{2}\chi^2_{d,1-\alpha}$ relative to its maximum. In Table 1, we collect some values for the coverage levels and parameter dimensions.

A striking feature of Table 1 is that the threshold grows substantially with d. For a two-parameter problem, the 95.4% joint region requires a log-likelihood drop of $6.18/2 = 3.09$, not 2; the 2σ ellipse is thus larger than naive one-dimensional thinking suggests. Recall that Q is a sum of d independent squared standard normals. Since its expectation itself is d, the distribution therefore shifts to the right as d grows, and its quantile must follow, growing roughly as $d + O(\sqrt{d})$ for fixed α. In Figure 4.7, we develop some intuition for this effect, where we can see that both cases, which we would intuitively use moving from $d=1$ to $d=2$ (orange and blue), would underestimate the coverage.

4.11 From the Fisher matrix to the confidence ellipsoid

The shape of $\mathcal{R}_\alpha$ is encoded in the Fisher matrix. We can find the eigenbasis of $\boldsymbol{F}^{(n)}$ via $\boldsymbol{F}^{(n)} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^\top$, so that the columns of $\boldsymbol{U}$ are the eigenvectors and $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1,\ldots,\lambda_d)$ contains the eigenvalues on the diagonal. We can then align the coordinate system with principal axes $\boldsymbol{F}^{(n)}$ via the coordinate transformation:

$$\tilde{\boldsymbol{\theta}} = \boldsymbol{U}^\top(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_{\mathrm{MLE}}). \tag{4.45}$$

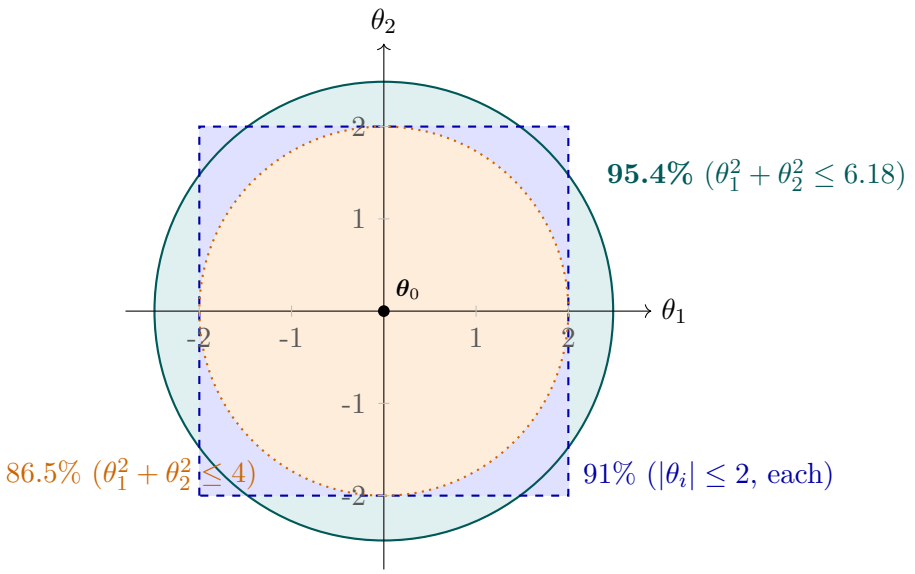


Figure 4.7: *Intuition behind the different scaling of $\chi^2_{d,1-\alpha}$ for $d = 2$ as reported in Table 1. You can find an additional plot on GitHub where you can see how the value of the χ^2 scales with the dimensionality.*

In these coordinates, Equation (4.41) becomes rather trivial:

$$\lambda_\gamma \, (\tilde{\theta}^\gamma)^2 \leq \chi^2_{d,\,1-\alpha} \,, \tag{4.46}$$

describing a standard axis-aligned ellipsoid with semi-axes

$$a_\gamma = \sqrt{\chi^2_{d,1-\alpha}/\lambda_\gamma} \,. \tag{4.47}$$

A large eigenvalue λ_γ means large information content and therefore that the data constrain the corresponding direction sharply (small semi-axis), while a small eigenvalue signals a poorly constrained combination of parameters (long semi-axis). The orientation of the eigenvectors determines which linear combinations of parameters are well or poorly determined.

The left panel of Figure 4.8 illustrates this geometry for $d = 2$: the eigenvectors of $\boldsymbol{F}^{-1}$ are drawn as arrows, with lengths proportional to the respective semi-axes. Correlation between parameters ($\rho \neq 0$) tilts the ellipse away from the coordinate axes; uncorrelated parameters produce an axis-aligned ellipse.

When presenting confidence intervals, we must be precise about which interval we are reporting. Let us first distinguish between a joint confidence region and a marginal confidence interval for a single parameter. Starting from the asymptotic Gaussian, Equation (4.27), the marginal distribution of any single component $\hat{\theta}^\alpha_{\mathrm{MLE}}$ is

$$\hat{\theta}^\alpha_{\mathrm{MLE}} \sim \mathcal{N}\left(\hat{\theta}^\alpha_{\mathrm{MLE}},\; \frac{(\boldsymbol{F}^{-1})_{\alpha\alpha}}{n}\right), \tag{4.48}$$

because the marginal of a multivariate Gaussian is itself Gaussian. The corresponding $100(1-\alpha)\%$ marginal confidence interval is

$$\hat{\theta}^\alpha_{\mathrm{MLE}} \;\pm\; z_{\alpha/2}\sqrt{\frac{(\boldsymbol{F}^{-1})_{\alpha\alpha}}{n}}\,, \tag{4.49}$$

where again $(\boldsymbol{F}^{-1})_{\alpha\alpha}$ is the α-th diagonal element of the *inverted* Fisher matrix; *not* $1/F_{\alpha\alpha}$. These two agree only when all off-diagonal elements of $\boldsymbol{F}$ vanish. In general,

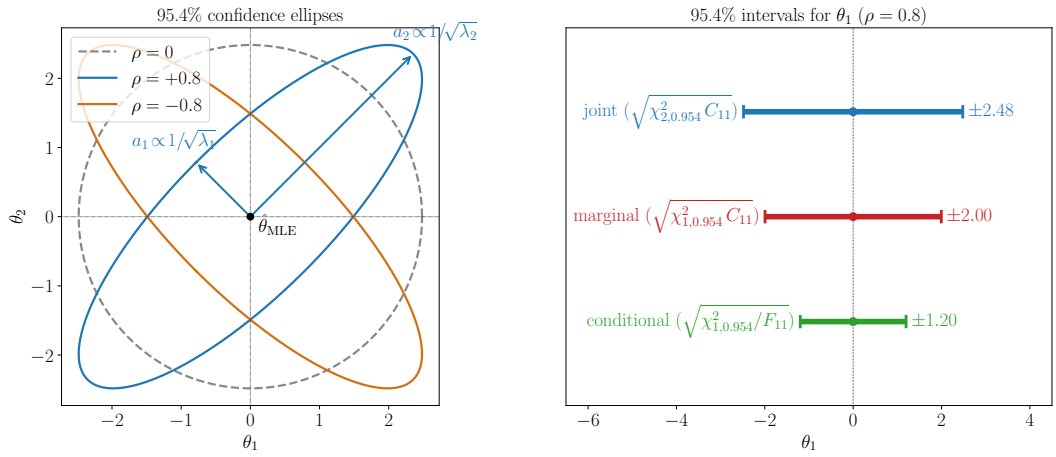


Figure 4.8: *Left:* 95.4% *confidence ellipses for two parameters* (θ^1, θ^2)*, centred on the MLE (black dot), for three values of the correlation (compare Figure 3.3). The arrows show the eigenvectors of* $\boldsymbol{F}^{-1}$ *for the* $\rho = 0.8$ *case, with lengths equal to the semi-axes* $a_i \propto 1/\sqrt{\lambda_i}$*. Right: comparison of three types of uncertainty intervals for* θ^1 *at* 95.4% *coverage, for* $\rho = 0.8$*. The joint projection (blue) is the projection of the full 2D ellipse onto the* θ^1 *axis and is the widest. The marginal interval (red) uses* $(F^{-1})_{11}$ *and the* χ^2_1 *threshold. It is narrower because it does not require all* θ^2 *values in the 2D ellipse to be covered. The conditional interval (green) fixes* $\theta^2 = \hat{\theta}^2_{\rm MLE}$ *and uses the smaller conditional variance* $(F_{11})^{-1}$*.*

the identity

$$(\boldsymbol{F}^{-1})_{\alpha\alpha} = \frac{1}{F_{\alpha\alpha} - \sum_{\beta\neq\alpha} F_{\alpha\beta}(\boldsymbol{F}^{-1})_{\beta\beta} F_{\beta\alpha}} \geq \frac{1}{F_{\alpha\alpha}} \tag{4.50}$$

makes it clear that marginalising over additional parameters always broadens the uncertainty on any one of them, with equality only if all parameters are truly independent.

Comparing with the projection of the joint ellipsoid onto the θ^α axis, the marginal interval, Equation (4.49), is narrower: the joint projection asks for which values of θ^α there exist other parameter values such that the full vector lies in the joint region, whereas the marginal already integrates out the other parameters. This hierarchy is illustrated in the right panel of Figure 4.8.

Summary: three types of confidence intervals

Given an MLE $\hat{\boldsymbol{\theta}}_{\rm MLE}$ and Fisher matrix $\boldsymbol{F}$ for d parameters, you can, in principle, present the following confidence intervals:

1. **Joint** $100(1-\alpha)\%$ region in d dimensions: the ellipsoid $Q(\boldsymbol{\theta}_0) \leq \chi^2_{d,1-\alpha}$, equivalently $L(\hat{\boldsymbol{\theta}}) - L(\boldsymbol{\theta}) \leq \frac{1}{2}\chi^2_{d,1-\alpha}$.
2. **Marginal** $100(1-\alpha)\%$ interval for θ^α: integrates out all other parameters; the width is set by $\sqrt{(\boldsymbol{F}^{-1})_{\alpha\alpha}/n}$ and the χ^2_1 threshold.
3. **Conditional** $100(1-\alpha)\%$ interval for θ^α given $\theta^\beta = \hat{\theta}^\beta$ $(\forall \beta \neq \alpha)$: uses the conditional variance $1/(nF_{\alpha\alpha})$, which is the narrowest of the three.

In astrophysical practice, the **marginal** interval is almost always what is reported.

4.12 Summary and some practical remarks

Figure 4.9 summarises the end-to-end MLE workflow, from model specification to visualisation, with pointers to the key results derived in this section.

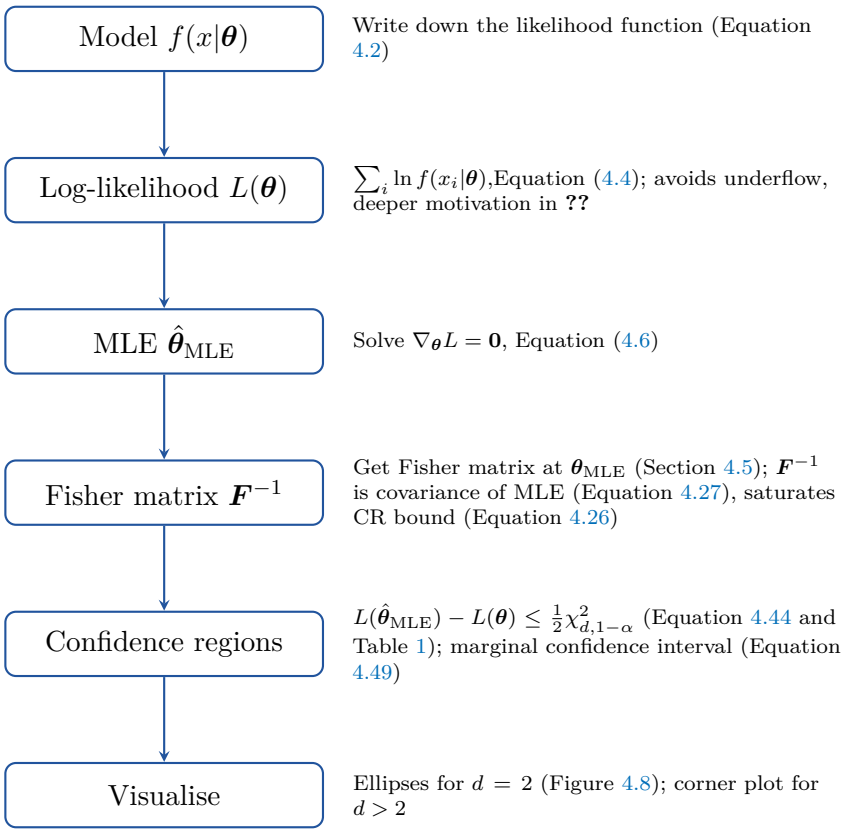


Figure 4.9: *End-to-end MLE workflow. Each step points to the equation, table, or section where the corresponding result is derived.*

We will close this section by listing some caveats, hints and remarks when applying the derived results to real problems:

1. **Corner plots**: For $d > 2$ parameters, the standard visualisation is the *corner plot* (triangle plot): a grid of panels showing all $\binom{d}{2}$ pairwise marginal 68% and 95% ellipses on the off-diagonal entries, and the marginal 1D distributions on the diagonal.

2. **Fisher forecasting**: An important application is *forecasting*: compute $\boldsymbol{F}$ from an assumed fiducial model before any data exists. This predicts the expected parameter uncertainties of a future experiment. The forecast marginal uncertainty on θ^α (Equation 4.49) is widely used to compare the constraining the power of next-generation experiments.

3. **Validity of the Gaussian approximation**: All results in this section rest on the asymptotic normality of the MLE (Equation 4.14). As illustrated in Figure 4.5, even small departures from Gaussianity distort the distribution of the quadratic form Q and hence the actual coverage of the confidence region. Before quoting confidence regions, it is good practice to verify the Gaussianity of the likelihood surface, for instance, by checking that the eigenvalues of the Hessian $-\partial_\alpha \partial_\beta L$ are stable across $\mathcal{R}_\alpha$.

4. **Beyond the Gaussian regime**: When the likelihood surface is strongly non-Gaussian, as is common for small data sets, highly correlated parameters, or non-linear models, the ellipsoidal approximation of Equation (4.44) breaks down. One then either maps out the profile likelihood numerically by maximising $L(\boldsymbol{\theta})$ over all nuisance parameters at each fixed value of interest, transitions to fully Bayesian posterior sampling (Sections 5 and 6), or employs simulation-based inference (Section 8).

5 Bayesian inference

In Section 4, we introduced the likelihood function $\mathcal{L}(\boldsymbol{\theta})$ as the central object of frequentist parameter inference and defined the maximum likelihood estimator as the value $\hat{\boldsymbol{\theta}}_{\mathrm{MLE}}$ that maximises its logarithm. We then established three asymptotic properties of the MLE: consistency, asymptotic normality, and asymptotic efficiency (the MLE saturates the Cramér–Rao bound). The Fisher matrix $\boldsymbol{F}$ plays an important role: its inverse sets the covariance of the MLE, and it defines the shape of the confidence regions, which are the sets of parameter values for which the log-likelihood has not dropped by more than $\frac{1}{2}\chi^2_{d,1-\alpha}$ relative to its maximum.

Powerful as this is, two structural limitations become especially important in astrophysical and cosmological practice. While external datasets can be incorporated by multiplying their likelihoods into a joint likelihood, and hard parameter bounds can be imposed by restricting the parameter space, the MLE offers no mechanism to encode soft, probabilistic prior beliefs. These are the kinds of degrees of belief that arise from theoretical constraints or earlier experiments for which the full likelihood is not available. A subtler but equally important issue concerns interpretation. A $100(1-\alpha)\%$ confidence interval is a statement about the following procedure: over many repetitions of the same experiment, it covers the true parameter with the stated probability. After observing the data, however, $\boldsymbol{\theta}_0$ either lies in the realised interval or it does not. In cosmology, where we measure a single Universe, the notion of long-run repetition is at best a metaphor; what one typically wants is a direct probability statement about $\boldsymbol{\theta}$ given the data, which the frequentist framework cannot provide.

Bayesian inference addresses both points directly. It treats $\boldsymbol{\theta}$ as a random variable, encodes existing knowledge in a prior distribution $\pi(\boldsymbol{\theta})$, and updates it with the data via Bayes' theorem to obtain the posterior $p(\boldsymbol{\theta} \mid \boldsymbol{x})$. We do not need anything new: Bayes' theorem follows from the conditional probability of Section 2, and the likelihood from Section 4 reappears unchanged. The interpretation, however, shifts from which parameter values are consistent with the data, to what is our degree of belief about $\boldsymbol{\theta}$ after seeing the data?

5.1 From conditional probability to Bayes' theorem

Recall from Section 2, that for two events A and B with $\mathbb{P}(B) > 0$, the conditional probability is

$$\mathbb{P}(A \mid B) = \mathbb{P}(A \cap B)/\mathbb{P}(B). \tag{5.1}$$

Applying this twice and using $\mathbb{P}(A \cap B) = \mathbb{P}(B \cap A)$, gives the multiplication rule in both directions, and solving for $\mathbb{P}(A \mid B)$ yields *Bayes' theorem*:

$$\mathbb{P}(A \mid B) = \frac{\mathbb{P}(B \mid A)\,\mathbb{P}(A)}{\mathbb{P}(B)}. \tag{5.2}$$

Note that we used only the definition of conditional probabilities, which is itself based on the Kolmogorov axioms. This is why Bayes' theorem is indeed a theorem and not an axiom! Hence, even if we apply Bayesian inference, we agree on the very basic axioms of probability theory, on which the frequentist school is also based. The only difference is the meaning of $\mathbb{P}$.

To map this into the language of parameter inference and connect it to our discussion in Section 4, we replace the abstract events by parameters $\boldsymbol{\theta} \in \Theta$ in a parameter space Θ and observed data $\boldsymbol{x} = (x_1, \ldots, x_n)$, and move to the continuous version of conditional probability from Section 2. The result is

$$p(\boldsymbol{\theta} \mid \boldsymbol{x}) = \frac{\mathcal{L}(\boldsymbol{x} \mid \boldsymbol{\theta})\,\pi(\boldsymbol{\theta})}{p(\boldsymbol{x})}. \tag{5.3}$$

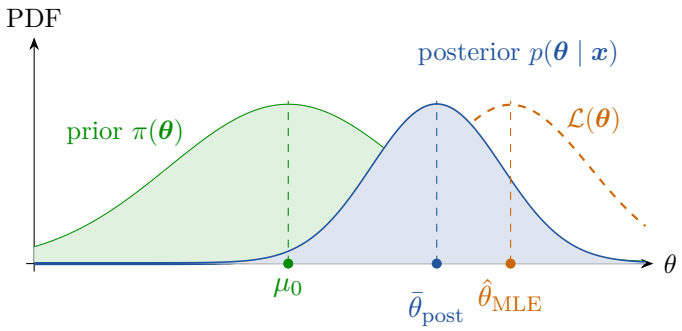


Figure 5.1: *The prior $\pi(\boldsymbol{\theta})$ (green) encodes belief before seeing data. The likelihood $\mathcal{L}(\boldsymbol{\theta})$ (orange, dashed) encodes information from the observations. The posterior $p(\boldsymbol{\theta} \mid \boldsymbol{x})$ (blue) is proportional to their product and is narrower than either factor.*

Figure 5.1 shows the different ingredients of Bayes' theorem and how they interact. We will provide a brief interpretation of each below, and discuss them in more detail in their own section.

i) The *prior* $\pi(\boldsymbol{\theta})$ (see Section 5.2 for more details) encodes everything known about $\boldsymbol{\theta}$ before observing any data. This includes hard physical constraints (a variance must be positive, a probability must lie in $[0, 1]$), results from previous experiments, or theoretical expectations from symmetry or dimensional arguments.

ii) The *likelihood* $\mathcal{L}(\boldsymbol{x} \mid \boldsymbol{\theta})$ is the same function defined in Equation (4.2): viewed as a function of the data at fixed $\boldsymbol{\theta}$ it is a joint PDF, but viewed as a function of $\boldsymbol{\theta}$ at fixed data it is not. In the Bayesian interpretation, it plays the role of an updating factor: it tilts the prior towards the parameter values that best explain the data we have observed. For n iid observations, $\mathcal{L}(\boldsymbol{\theta}) = \exp(L(\boldsymbol{\theta}))$ concentrates near $\hat{\boldsymbol{\theta}}_{\text{MLE}}$ as $n \to \infty$, which is why the posterior converges to the same limit regardless of the prior.

iii) The *evidence* $p(\boldsymbol{x})$ is the marginal likelihood of the data under the model: the probability of observing $\boldsymbol{x}$ averaged over all parameter values weighted by the prior. One can see this from the fact that the posterior needs to be normalised:

$$\int_{\Theta} p(\boldsymbol{\theta} \mid \boldsymbol{x}) \, \mathrm{d}\boldsymbol{\theta} = 1 = \frac{1}{p(\boldsymbol{x})} \int_{\Theta} \mathcal{L}(\boldsymbol{x} \mid \boldsymbol{\theta}) \, \pi(\boldsymbol{\theta}) \, \mathrm{d}\boldsymbol{\theta} \tag{5.4}$$

It is therefore also called the prior predictive distribution: before seeing data, it tells us what observations the model considers likely. Its role becomes central in model comparison (Section 5.6). The ratio of evidence for two models is the Bayes factor, which quantifies how much more strongly the data support one model over the other. The evidence integral is, in general, analytically intractable, which is one of the principal practical motivations for the Monte Carlo methods of Section 6.

iv) The *posterior* $p(\boldsymbol{\theta} \mid \boldsymbol{x})$ is the complete answer to the inference problem: a genuine probability distribution over Θ encoding everything the data and prior jointly say about $\boldsymbol{\theta}$. From it one extracts point estimates, marginal credible intervals, and joint credible regions, all of which will be discussed in Section 5.3. Since the evidence does not depend on $\boldsymbol{\theta}$, it is a normalisation constant within

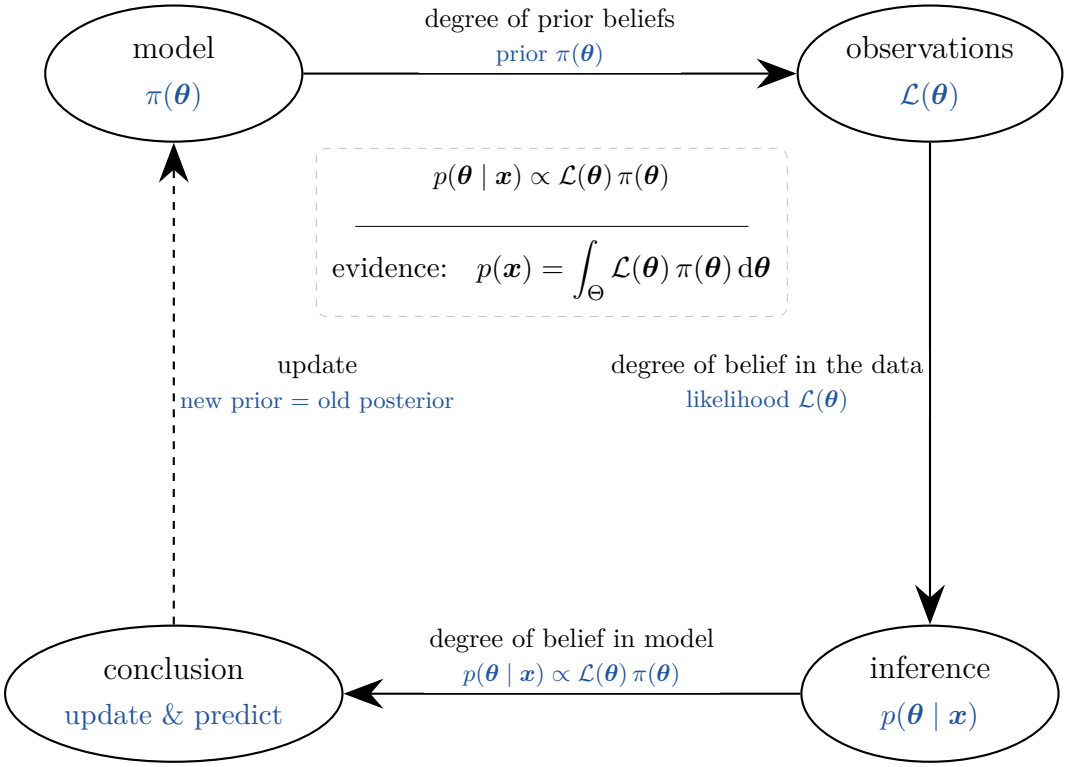


Figure 5.2: *The circle of the logic of science, now annotated with the Bayesian ingredients from Equation (5.3). Each ellipse carries both its role in the scientific process and the corresponding probabilistic object: the prior $\pi(\boldsymbol{\theta})$ encodes beliefs before data are seen, the likelihood $\mathcal{L}(\boldsymbol{\theta})$ encodes what the data say, and Bayes' theorem combines them into the posterior $p(\boldsymbol{\theta} \mid \boldsymbol{x})$. The dashed arrow is the sequential update of Equation (5.7): today's posterior is tomorrow's prior.*

a fixed model. Within a single model, i.e. all models with a shared parameter space Θ, we therefore most often work with the proportional form

$$p(\boldsymbol{\theta} \mid \boldsymbol{x}) \propto \mathcal{L}(\boldsymbol{x} \mid \boldsymbol{\theta})\ \pi(\boldsymbol{\theta}). \tag{5.5}$$

Taking the logarithm,

$$\ln p(\boldsymbol{\theta} \mid \boldsymbol{x}) = L(\boldsymbol{\theta}) + \ln \pi(\boldsymbol{\theta}) + \text{const}, \tag{5.6}$$

with $L(\boldsymbol{\theta})$ the log-likelihood from Equation (4.4). Maximising the log-posterior is therefore equivalent to maximising a penalised log-likelihood, with $\ln \pi(\boldsymbol{\theta})$ as the regularisation term. The point $\boldsymbol{\theta} \in \Theta$ where $p(\boldsymbol{\theta} \mid \boldsymbol{x})$ is maximal is the maximum a posteriori (MAP), $\hat{\boldsymbol{\theta}}_{\text{MAP}}$.

Crucially, the posterior of one inference step becomes the prior of the next: if a second independent dataset $\boldsymbol{x}^{(2)}$ is subsequently observed, $p(\boldsymbol{\theta} \mid \boldsymbol{x}^{(1)})$ enters Equation (5.7) as the new prior, and the analysis proceeds unchanged:

$$p(\boldsymbol{\theta} \mid \boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}) \propto \mathcal{L}(\boldsymbol{\theta} \mid \boldsymbol{x}^{(2)})\ p(\boldsymbol{\theta} \mid \boldsymbol{x}^{(1)}). \tag{5.7}$$

This is, of course, reminiscent of our circle of science discussion at the very beginning of the lecture (Figure 1.1). However, we are now in the position to fill out the details and draw the Bayesian expression in Figure 5.2 above.

5.2 Prior distributions

The prior $\pi(\boldsymbol{\theta})$ is the ingredient that distinguishes Bayesian from frequentist inference. Its choice expresses what we know about $\boldsymbol{\theta}$ before any data. Before examining it more closely, let us present some examples from astrophysics and cosmology.

- Many astrophysical quantities are positive by definition: fluxes, masses, line widths, etc. A Gaussian prior can have a non-negligible probability to have negative values whenever μ/σ is not large. A log-normal, $\ln\theta \sim \mathcal{N}(\mu_{\ln}, \sigma^2_{\ln})$, is more appropriate in such cases and also naturally handles parameters that span orders of magnitude. A related choice is the log-uniform (scale-invariant) prior $\pi(\theta) \propto 1/\theta$, $\theta \in [\theta_{\min}, \theta_{\max}]$, which assigns equal probability to each decade.

- In one of the exercises, you were supposed to use information from Big Bang Nucleosynthesis. The latter provides a constraint on the baryon density, $\Omega_{\rm b}h^2 = 0.0224 \pm 0.0002$, entirely independently of any CMB or large-scale structure measurement. This was effectively a Gaussian prior; you knew something before you looked at the data.

- For a binary star or planetary system whose orbital plane is randomly oriented with respect to the line of sight, the inclination $\imath \in [0, \pi/2]$, where $\imath = 0$ is face-on and $\imath = \pi/2$ is edge-on, is not uniformly distributed. Isotropy of orbital poles on the sphere gives a solid-angle element $\mathrm{d}\Omega \propto \sin\imath\,\mathrm{d}\imath$, so the physically correct prior is

  $$\pi(\imath) \propto \sin\imath,$$

  which strongly favours edge-on configurations. This is a case where the prior is uninformative in the sense that it encodes only geometry, yet it is far from flat. Using $\pi(\imath) \propto 1$ overweights face-on orbits and biases the inferred stellar and planetary radii.

- The radius R of a neutron star is constrained from below by nuclear saturation density arguments ($R \gtrsim 9\,\mathrm{km}$) and from above by causality — the speed of sound in dense matter cannot exceed c — giving $R \lesssim 16\,\mathrm{km}$. In addition, the observation of neutron stars with masses $M \gtrsim 2\,M_\odot$ rules out soft equations of state that would predict $R \lesssim 10\,\mathrm{km}$ for typical masses. The prior on R is therefore not flat but effectively truncated by these hard physical bounds. When gravitational-wave signals are used to infer R, this prior prevents the posterior from exploring equation-of-state models already excluded by nuclear physics.

Not all parameters are equally constrained by the data. Parameters where most of the information is contained in the prior and not in the likelihood are called prior-dominated. In Figure 5.1, this would mean that the green curve is much narrower than the orange dashed curve. From the Fisher matrix perspective of Section 4.5, this is the case in which $F_{\alpha\alpha}$ is small: the likelihood is nearly flat in that direction, and the data carry little information about θ^α. The prior then plays a decisive rather than merely regularising role, and different prior choices yield substantially different posteriors. A flat prior on a prior-dominated parameter is therefore not a neutral choice. Identifying prior-dominated parameters is thus an important diagnostic step: it signals that the experiment lacks the resolving power to constrain them.

It cannot be stressed enough that choosing the prior is a scientific decision and part of the model specification, not an afterthought. Since every posterior is implicitly conditional on $\pi(\boldsymbol{\theta})$, any result derived from it inherits that dependence. It

is therefore essential to report the prior explicitly alongside the results, with sufficient detail for the reader to assess how much of the conclusion is driven by the data and how much by prior assumptions. As a matter of good practice, a prior sensitivity analysis should accompany any Bayesian inference: repeat the analysis under a range of reasonable alternative priors and verify that the conclusions are robust.

From this discussion, it is clear (and will be shown formally in Equation 5.24) that the prior matters most when data are scarce (with respect to the model complexity); for large n it is washed out by the likelihood regardless of its form. So, how do we choose the prior?

5.2.1 Flat and uninformative priors

The simplest choice is a *flat* prior $\pi(\boldsymbol{\theta}) = \text{const}$. Since Equation (5.5) then gives $p(\boldsymbol{\theta} \mid \boldsymbol{x}) \propto \mathcal{L}(\boldsymbol{\theta})$, the posterior mode coincides exactly with the MLE:

$$\hat{\boldsymbol{\theta}}_{\text{MAP}} = \hat{\boldsymbol{\theta}}_{\text{MLE}} \qquad \text{(for flat priors)}. \tag{5.8}$$

Maximum likelihood estimation is therefore Bayesian MAP estimation with a flat prior, and the framework of Section 4 is the special case in which prior knowledge in a particular parametrisation is assumed absent. Why did we say *in a particular parametrisation*? However, a flat prior is not invariant under reparametrisation. For example, if θ is uniform on $[a, b] \subset \mathbb{R}$, then $\phi = \theta^2$ is not. This follows directly from the conservation of probability:

$$1 = \int p_\theta(\theta)\,\mathrm{d}\theta = \int p_\phi(\phi)\,\mathrm{d}\phi \tag{5.9}$$

and therefore

$$p_\phi(\phi) = p_\theta(\theta(\phi)) \left|\frac{\partial\theta}{\partial\phi}\right| = \frac{1}{b-a}\frac{1}{2\sqrt{\phi}}. \tag{5.10}$$

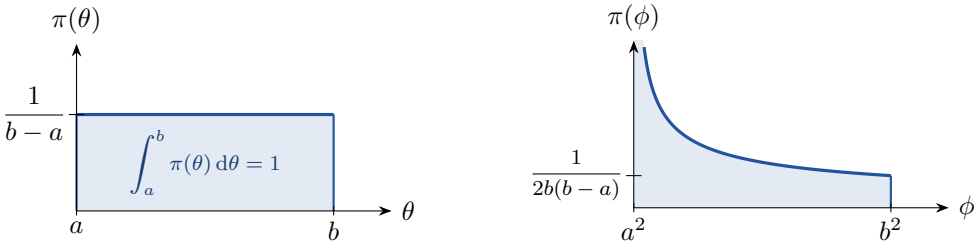


Figure 5.3: *A flat prior $\pi(\theta) = 1/(b-a)$ on $\theta \in [a, b]$ (left) does not induce a flat prior on $\phi = \theta^2 \in [a^2, b^2]$ (right). Changing variables gives $\pi(\phi) = 1/[2(b-a)\sqrt{\phi}]$, which diverges as $\phi \to a^2$ and decays monotonically to $1/[2b(b-a)]$ at $\phi = b^2$. Flatness is therefore not a coordinate-invariant property of a prior, which is the principal motivation for the Jeffreys prior, Equation (5.15): it is the unique prior that remains invariant under smooth reparametrisations of $\boldsymbol{\theta}$.*

5.2.2 The Jeffreys prior

The fact that the prior depends on the coordinate system, or in other words the chosen set of parameters is a bit unsatisfactory. Let us assume a general coordinate transformation $\boldsymbol{\theta}(\boldsymbol{\phi})$. The score transforms as:

$$\frac{\partial L(\boldsymbol{\phi})}{\partial \phi^\alpha} = \frac{\partial L(\boldsymbol{\theta}(\boldsymbol{\phi}))}{\partial \theta^\gamma}\frac{\partial \theta^\gamma}{\partial \phi^\alpha} = s_\gamma\, J^\gamma_\alpha\,, \tag{5.11}$$

with the Jacobian $\boldsymbol{J}$ of the coordinate transformation $\boldsymbol{\theta} \to \boldsymbol{\phi}$. The Fisher information matrix, Equation (4.17), therefore transforms as a rank-2 tensor (as the name already suggested)

$$\tilde{F}_{\alpha\beta}(\boldsymbol{\phi}) = J^{\gamma}_{\alpha} J^{\rho}_{\beta} F_{\gamma\rho}(\boldsymbol{\theta}) \,, \tag{5.12}$$

so that the determinant of the Fisher matrix transforms as

$$\det \tilde{\boldsymbol{F}}(\boldsymbol{\phi}) = \det \boldsymbol{F}(\boldsymbol{\theta}) (\det \boldsymbol{J})^2 . \tag{5.13}$$

However, we also know that

$$\pi(\boldsymbol{\phi}) = \pi(\boldsymbol{\theta}) \, |\det \boldsymbol{J}| \tag{5.14}$$

If we now define a prior

$$\pi_J(\boldsymbol{\theta}) \propto \sqrt{\det \boldsymbol{F}(\boldsymbol{\theta})}, \tag{5.15}$$

we see that

$$\pi(\boldsymbol{\phi}) \propto \sqrt{\det \boldsymbol{F}(\boldsymbol{\theta})} \, |\det \boldsymbol{J}| = \sqrt{\det \tilde{\boldsymbol{F}}(\boldsymbol{\phi})} \,. \tag{5.16}$$

Therefore, Equation (5.15) is invariant under smooth reparametrisations and is the canonical objective choice when no prior information is available; it is called the *Jeffreys prior*.

This reveals a fundamental geometric truth: $\sqrt{\det \boldsymbol{F}} \, \mathrm{d}\boldsymbol{\theta}$ is the Riemannian volume element of a statistical manifold. The factor $\det \boldsymbol{J}$ is exactly cancelled by the Jacobian from the measure transformation. Furthermore, the appearance of $\sqrt{\det \boldsymbol{F}(\boldsymbol{\theta})}$ in both the Jeffreys prior and the normalisation of a Gaussian is not a coincidence. The Jeffreys prior is therefore the uniform measure with respect to this canonical volume: it treats all parameter-space regions of equal information-geometric size as equally probable.

5.2.3 Informative priors

When proper prior knowledge is available one uses an *informative prior*. In cosmology, for example, the *Planck* satellite provides a tight measurement of the baryon density $\Omega_{\mathrm{b}} h^2$; a subsequent galaxy survey analysis can place a Gaussian prior centred on the Planck value with its quoted uncertainty. This is the quantitative realisation of the "update" arrow in Figure 5.2.

5.2.4 Proper and improper priors

A prior is *proper* if $\int_{\Theta} \pi(\boldsymbol{\theta}) \, \mathrm{d}\boldsymbol{\theta} < \infty$, so that it normalises to a genuine probability distribution. Flat priors on unbounded spaces are *improper*. They can still yield proper posteriors provided the likelihood is sufficiently peaked, which holds under the regularity conditions of Section 4.4. Improper priors can, however, produce improper posteriors in certain hierarchical models, and one should always verify this before computing posterior summaries.

5.3 Posterior summaries and credible intervals

The full posterior is the complete Bayesian answer to the inference problem. In practice, one compresses it into point estimates and intervals as before.

5.3.1 Point estimates

The two most common Bayesian point estimates are the MAP estimator,

$$\hat{\boldsymbol{\theta}}_{\mathrm{MAP}} \coloneqq \arg\max_{\boldsymbol{\theta}\in\Theta} \left[L(\boldsymbol{\theta}) + \ln \pi(\boldsymbol{\theta}) \right], \tag{5.17}$$

and the *posterior mean*,

$$\bar{\boldsymbol{\theta}} \coloneqq \langle \boldsymbol{\theta} \mid \boldsymbol{x} \rangle = \int_{\Theta} \boldsymbol{\theta}\, p(\boldsymbol{\theta} \mid \boldsymbol{x})\, \mathrm{d}\boldsymbol{\theta}. \tag{5.18}$$

For symmetric unimodal posteriors, the two coincide; for skewed ones, they differ, and in general, all the same arguments as in Section 3 apply since the posterior is a proper probability. All three estimates — MAP, posterior mean, and MLE — coincide when the prior is flat and the likelihood is Gaussian.

5.3.2 Credible intervals

The Bayesian analogue of a confidence interval is a *credible interval*. A $100(1-\alpha)\%$ credible interval $[\theta_{\mathrm{l}}, \theta_{\mathrm{u}}]$ satisfies

$$\mathbb{P}(\theta_{\mathrm{l}} \le \theta \le \theta_{\mathrm{u}} \mid \boldsymbol{x}) = \int_{\theta_{\mathrm{l}}}^{\theta_{\mathrm{u}}} p(\theta \mid \boldsymbol{x})\, \mathrm{d}\theta = 1 - \alpha. \tag{5.19}$$

This is the statement one typically wishes[2] a frequentist confidence interval made: θ lies in the interval with posterior probability $1-\alpha$. Recall from Equation (4.28) that the frequentist interval does not make this claim; it controls long-run coverage, nothing more.

There are two standard constructions. The *highest posterior density* (HPD) interval is the shortest interval satisfying Equation (5.19): all points inside have density $\ge p^\star$ and all points outside have density $\le p^\star$, for a threshold $p^\star$ chosen to achieve the required probability. The *equal-tails* interval simply cuts $\alpha/2$ from each tail. For symmetric posteriors, both constructions agree; for skewed ones, the HPD interval is strictly shorter and is the natural Bayesian counterpart of the likelihood-based region of Equation (4.44). We show the two constructions in Figure 5.4.

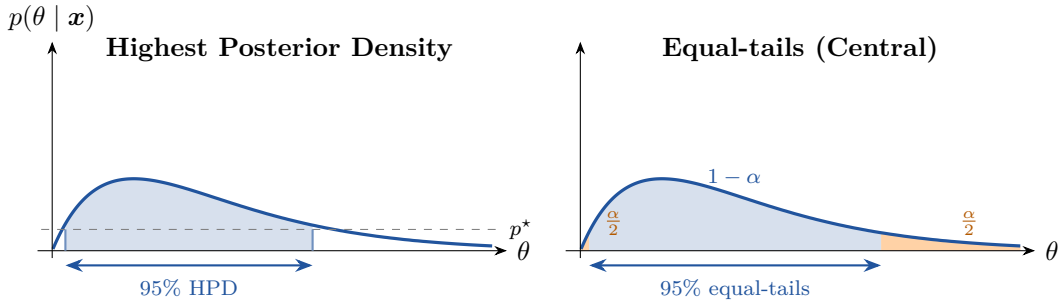


Figure 5.4: *Two constructions of a* 95% *credible interval for an asymmetric posterior. Left: the HPD interval (blue shading) is the shortest interval containing* 95% *of the posterior probability, with boundaries set by a density threshold $p^\star$ (grey dashed line). Right: the equal-tails interval (blue shading) cuts* 2.5% *from each tail (orange). For asymmetric posteriors the HPD interval is strictly shorter.*

In the multivariate case, the HPD region generalises to $\mathcal{C}_\alpha = \{\boldsymbol{\theta} : p(\boldsymbol{\theta} \mid \boldsymbol{x}) \ge p_\alpha\}$, chosen so that $\int_{\mathcal{C}_\alpha} p(\boldsymbol{\theta} \mid \boldsymbol{x})\, \mathrm{d}\boldsymbol{\theta} = 1-\alpha$. In the Gaussian limit this reduces to the ellipsoidal confidence region of Section 4.10, as we show next.

[2] At least I do.

5.4 The Laplace approximation of the posterior

For most realistic models, the posterior cannot be evaluated analytically and the methods of Section 6 are required. A powerful analytic approximation, however, connects the posterior directly back to the frequentist results of Section 4.

5.4.1 The Laplace approximation

Suppose the posterior is unimodal with a well-defined maximum at $\hat{\boldsymbol{\theta}}_{\mathrm{MAP}}$. As is customary by now, we expand the log-posterior to second order about this point, which gives

$$\ln p(\boldsymbol{\theta} \mid \boldsymbol{x}) \approx \ln p(\hat{\boldsymbol{\theta}}_{\mathrm{MAP}} \mid \boldsymbol{x}) - \frac{1}{2}(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_{\mathrm{MAP}})^{\top} \boldsymbol{H} (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_{\mathrm{MAP}}), \tag{5.20}$$

where the negative Hessian of the log-posterior at the MAP is

$$H_{\alpha\beta} \coloneqq -\partial_\alpha \partial_\beta \ln p(\boldsymbol{\theta} \mid \boldsymbol{x})\big|_{\hat{\boldsymbol{\theta}}_{\mathrm{MAP}}} = \underbrace{-\partial_\alpha \partial_\beta L}_{\to F^{(n)}_{\alpha\beta}} \; \underbrace{- \; \partial_\alpha \partial_\beta \ln \pi}_{\text{prior curvature}} \Bigg|_{\hat{\boldsymbol{\theta}}_{\mathrm{MAP}}}. \tag{5.21}$$

Exponentiating Equation (5.20) and recognising the quadratic form gives the *Laplace approximation*:

$$p(\boldsymbol{\theta} \mid \boldsymbol{x}) \approx \mathcal{N}\left(\boldsymbol{\theta}; \hat{\boldsymbol{\theta}}_{\mathrm{MAP}}, \boldsymbol{H}^{-1}\right). \tag{5.22}$$

For a flat prior, the prior curvature in Equation (5.21) vanishes, $\hat{\boldsymbol{\theta}}_{\mathrm{MAP}} = \hat{\boldsymbol{\theta}}_{\mathrm{MLE}}$, and the negative Hessian of the log-likelihood at the MLE is exactly $\boldsymbol{F}^{(n)}$ in expectation, so Equation (5.22) becomes

$$p(\boldsymbol{\theta} \mid \boldsymbol{x}) \approx \mathcal{N}\left(\boldsymbol{\theta}; \hat{\boldsymbol{\theta}}_{\mathrm{MLE}}, \frac{1}{n}\boldsymbol{F}^{-1}\big(\hat{\boldsymbol{\theta}}_{\mathrm{MLE}}\big)\right), \tag{5.23}$$

recovering the asymptotic distribution of the MLE from Equation (4.27). The inverse Fisher matrix $\boldsymbol{F}^{-1}/n$ is therefore simultaneously the posterior covariance (under a flat prior) and the frequentist sampling covariance of the MLE. Figure 5.5 illustrates the quality of this approximation for a non-Gaussian posterior.

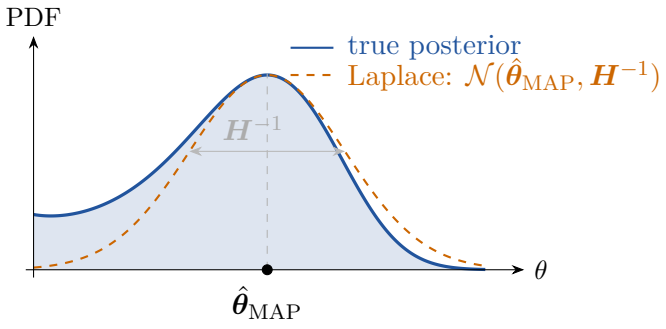


Figure 5.5: *The Laplace approximation (orange dashed) to a mildly non-Gaussian posterior (blue). Near the peak, the agreement is good, but the tail asymmetry is not captured.*

The Laplace approximation is a second-order approximation. The Bernstein–von Mises theorem establishes that it becomes asymptotically exact. If $\pi(\boldsymbol{\theta})$ is positive and continuous in a neighbourhood of the truth $\boldsymbol{\theta}_0$, and suppose the model satisfies

the regularity conditions of Section 4.4. Then, as $n \to \infty$ the posterior converges in total variation to

$$p(\boldsymbol{\theta} \mid \boldsymbol{x}) \xrightarrow{n\to\infty} \mathcal{N}\left(\hat{\boldsymbol{\theta}}_{\mathrm{MLE}}, \frac{1}{n}\boldsymbol{F}^{-1}(\boldsymbol{\theta}_0)\right), \tag{5.24}$$

regardless of the prior $\pi(\boldsymbol{\theta})$. Let us follow the connection between a Bayesian and a frequentist analysis from a single starting point. With a flat prior $\pi(\boldsymbol{\theta}) \propto \text{const}$, the posterior, Equation (5.5), reduces to $p(\boldsymbol{\theta} \mid \boldsymbol{x}) \propto \mathcal{L}(\boldsymbol{\theta})$, and its mode is exactly the MLE (Equation 5.8): the frequentist point estimate and the MAP coincide. Now, suppose we update sequentially, starting from this flat prior. Each new batch of data multiplies the current posterior by a new likelihood factor via Equation (5.7), and by the law of large numbers applied to the log-likelihood, the posterior concentrates near the truth at rate $1/\sqrt{n}$. The Bernstein–von Mises theorem (Equation 5.24) makes this precise: as $n \to \infty$ the posterior converges to $\mathcal{N}(\hat{\boldsymbol{\theta}}_{\mathrm{MLE}}, \boldsymbol{F}^{-1}/n)$ regardless of the starting prior, and the Bayesian credible intervals and the frequentist confidence intervals of Equation (4.49) agree numerically. They differ only in interpretation: the frequentist interval covers the true parameter in $100(1-\alpha)\%$ of repeated experiments, while the Bayesian interval contains the parameter with posterior probability $1-\alpha$ given the observed data. The deeper connection runs through the likelihood function itself, which is the common object of both schools. The frequentist maximises $L(\boldsymbol{\theta})$ and the Bayesian adds $\ln \pi(\boldsymbol{\theta})$ and normalises via the evidence. Whether $\boldsymbol{\theta}$ is a random variable or a fixed unknown only matters in the finite-data regime, where the prior shape influences the posterior and prior-dominated parameters can lead the two approaches to different conclusions. This suggests a pragmatic view: Bayesian inference is most valuable when data are scarce, and prior knowledge is substantial, that is, in the regime where the prior actively shapes the posterior and where the frequentist notion of long-run repetition may not even be well defined. The frequentist approach, on the other hand, is most natural when data are plentiful and prior information is absent. In practice, the choice between them is therefore often less a matter of philosophy than of the problem at hand.

5.5 Nuisance parameters and marginalisation

Realistic inference problems almost always involve parameters one does not care about, we call them *nuisance parameters*. Let $\boldsymbol{\theta} = (\boldsymbol{\psi}, \boldsymbol{\eta})$ with $\boldsymbol{\psi}$ the parameters of interest and $\boldsymbol{\eta}$ the nuisances. The Bayesian treatment is clean: integrate (marginalise) over $\boldsymbol{\eta}$ as in Section 2,

$$p(\boldsymbol{\psi} \mid \boldsymbol{x}) = \int p(\boldsymbol{\psi}, \boldsymbol{\eta} \mid \boldsymbol{x})\, \mathrm{d}\boldsymbol{\eta}. \tag{5.25}$$

All uncertainty about $\boldsymbol{\eta}$ is automatically propagated into $p(\boldsymbol{\psi} \mid \boldsymbol{x})$ without any further treatment. The frequentist alternative is the *profile likelihood*, which maximises over $\boldsymbol{\eta}$ at each fixed $\boldsymbol{\psi}$:

$$\mathcal{L}_{\mathrm{prof}}(\boldsymbol{\psi}) = \max_{\boldsymbol{\eta}} \mathcal{L}(\boldsymbol{\psi}, \boldsymbol{\eta}). \tag{5.26}$$

Profiling is a point optimisation: it conditions on the MLE of $\boldsymbol{\eta}$, whereas Bayesian marginalisation integrates over all values of $\boldsymbol{\eta}$ weighted by the posterior. The two give numerically similar results in the asymptotic regime, but can differ substantially otherwise.

If the joint posterior of $(\boldsymbol{\psi}, \boldsymbol{\eta})$ is a Gaussian with mean $(\hat{\boldsymbol{\psi}}, \hat{\boldsymbol{\eta}})$ and covariance

$$\boldsymbol{C} = \begin{pmatrix} \boldsymbol{C}_{\psi\psi} & \boldsymbol{C}_{\psi\eta} \\ \boldsymbol{C}_{\eta\psi} & \boldsymbol{C}_{\eta\eta} \end{pmatrix}, \tag{5.27}$$

the marginal posterior of $\boldsymbol{\psi}$ from Equation (5.25) is $p(\boldsymbol{\psi} \mid \boldsymbol{x}) = \mathcal{N}(\boldsymbol{\psi}; \hat{\boldsymbol{\psi}}_{\mathrm{MAP}}, \boldsymbol{C}_{\psi\psi})$. Under a flat prior $\boldsymbol{C} = [\boldsymbol{F}^{(n)}]^{-1}$, so $\boldsymbol{C}_{\psi\psi} = (\boldsymbol{F}^{-1})_{\psi\psi}$. This is exactly the marginal variance of Equation (4.49) and is strictly larger than the conditional variance $1/F_{\psi\psi}$ by the identity Equation (4.50): marginalising always broadens the uncertainty on the parameters of interest. Bayesian marginalisation and the frequentist marginal confidence interval therefore coincide in the Gaussian regime.

5.6 Bayesian model comparison

So far, we have asked what the parameter values are within a fixed model. A deeper question is which of two competing models is better supported by the data. In Section 4.9 this was addressed by the p-value and reduced χ^2. Bayesian inference provides a different approach via the evidence.

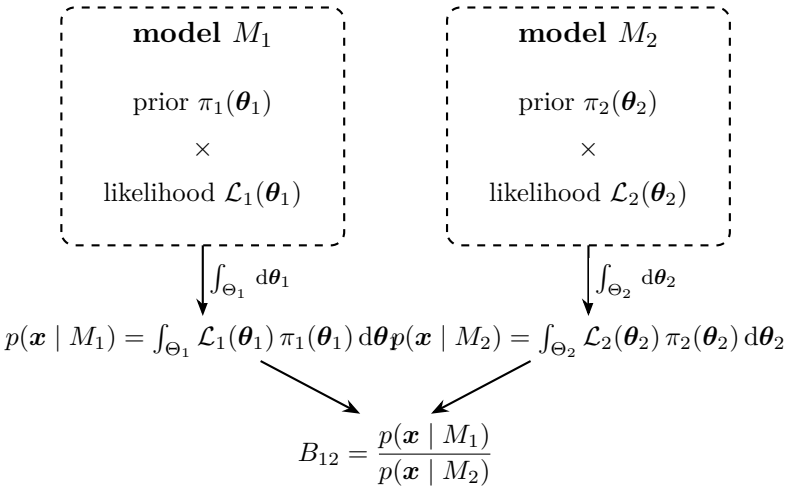


Figure 5.6: *Bayesian model comparison via the Bayes factor. The parameter spaces Θ_1 and Θ_2 are in general entirely distinct: the two models may differ in the number and identity of their parameters, not merely in their prior or likelihood choices. Each model yields a marginal likelihood (evidence) by integrating the prior-weighted likelihood over its own parameter space. The Bayes factor B_{12} is the ratio of these evidences; models with unnecessary parameters are automatically penalised because the prior $\pi_i(\boldsymbol{\theta}_i)$ must spread its unit mass over a larger Θ_i, reducing the evidence unless the extra parameters genuinely improve the fit (Occam's razor).*

For two models M_1 and M_2, the *Bayes factor* is the ratio of their marginal likelihoods (evidence) as illustrated in Figure 5.6:

$$B_{12} \coloneqq \frac{p(\boldsymbol{x} \mid M_1)}{p(\boldsymbol{x} \mid M_2)} = \frac{\int \mathcal{L}(\boldsymbol{\theta}_1 \mid M_1)\,\pi(\boldsymbol{\theta}_1 \mid M_1)\,\mathrm{d}\boldsymbol{\theta}_1}{\int \mathcal{L}(\boldsymbol{\theta}_2 \mid M_2)\,\pi(\boldsymbol{\theta}_2 \mid M_2)\,\mathrm{d}\boldsymbol{\theta}_2}. \tag{5.28}$$

With equal prior model probabilities, the Bayes factor equals the posterior odds ratio $\mathbb{P}(M_1 \mid \boldsymbol{x})/\mathbb{P}(M_2 \mid \boldsymbol{x}) = B_{12}$. A value $B_{12} > 1$ favours M_1. Table 2 gives the conventional choices, which are in themselves as arbitrary as the p-values.

The Bayes factor automatically penalises model complexity. Consider two nested models where M_2 has more parameters. At the likelihood maximum, M_2 fits the data at least as well as M_1. However, the evidence integral also integrates over the parameter volume that is *not* supported by the data: if the extra parameters of M_2 do not genuinely improve the fit, its evidence is smaller and M_1 is preferred. Simpler models that explain the data equally well are automatically favoured.

Table 2: The Jeffreys scale for interpreting Bayes factors.

$\ln B_{12}$	B_{12}	Evidence for M_1
< 1	< 2.7	Negligible
1–3	3–20	Positive
3–5	20–150	Strong
> 5	> 150	Very strong

Mathematical background: Bayesian information criterion

Applying the Laplace method to the evidence integral gives

$$\ln p(\boldsymbol{x} \mid M) \approx L(\hat{\boldsymbol{\theta}}_{\mathrm{MAP}}) + \ln \pi(\hat{\boldsymbol{\theta}}_{\mathrm{MAP}}) + \tfrac{d}{2} \ln(2\pi) - \frac{1}{2} \ln |\boldsymbol{H}|.$$

For n iid observations $|\boldsymbol{H}| \sim n^d |\boldsymbol{F}|$, so $-\frac{1}{2} \ln |\boldsymbol{H}| \approx -\frac{d}{2} \ln n + \text{const}$. Dropping constant and prior terms yields the *Bayesian Information Criterion*,

$$\mathrm{BIC} := -2L(\hat{\boldsymbol{\theta}}_{\mathrm{MLE}}) + d \ln n,$$

which approximates $-2 \ln p(\boldsymbol{x} \mid M)$. Minimising the BIC over competing models approximates Bayesian model selection. The penalty $d \ln n$ makes the complexity cost explicit: each additional parameter costs $\ln n$ units of log-likelihood.

Note that the Bayes factor is sensitive to the prior on the parameters even for large n, in contrast to parameter estimation within a fixed model where the Bernstein–von Mises theorem guarantees prior irrelevance.

5.7 On choosing the likelihood

We have talked extensively about the prior distribution in Section 5.2 and written likelihoods as functions of model parameters. However, we have never really thought about where they come from.

In Section 4 the likelihood was built as a product over n independent and identically distributed draws $x_1, \ldots, x_n$, Equation (4.2). That construction is tailored to repeated measurements, and the asymptotic machinery that follows from it: consistency, the $\sqrt{n}$ normality of Equation (4.14), asymptotic efficiency. In many applications the situation is the reverse: we have only a single realisation of the process, one dataset rather than a large ensemble of independent repetitions. The product over draws then collapses to a single factor. What we have access to is a single data vector $\boldsymbol{d} \in \mathbb{R}^N$, a set of N measured numbers and one realisation of the process under study. We never observe that process directly, only this single draw from it, so we treat d as a random variable governed by some sampling distribution. The likelihood is that distribution, evaluated at the data we actually obtained,

$$\mathcal{L} = p(\boldsymbol{d}) \,. \tag{5.29}$$

At this stage nothing has been assumed about its shape: Equation (5.29) is a completely general distribution, carrying a mean, a covariance, and in general non-trivial higher moments as well.

Since the two numbers n and N are easily confused let us summarise their difference

- n, the number of independent repetitions of the whole experiment, i.e. the iid index $x_1, \ldots, x_n$ of Equation (4.2). This is the count that runs to infinity in the asymptotic theory of Section 4.4; the $\sqrt{n}$ of Equation (4.14) and the $1/n$ in the marginal error are this n. In many problems it is small, frequently $n = 1$.
- N: the dimension of a single data vector $\boldsymbol{d}$. Its entries are generally correlated, as recorded by the off-diagonal terms of $\boldsymbol{C}$, so they are not N iid draws. Enlarging N is not the same as repeating the experiment and does not, by itself, provide the asymptotic results of Section 4.4.

The most important moment of that distribution is its mean, the value about which the data are centred, followed by the covariance, which records how the entries scatter and correlation

$$\boldsymbol{\mu} = \langle \boldsymbol{d} \rangle \,, \qquad \boldsymbol{C} = \left\langle (\boldsymbol{d} - \boldsymbol{\mu})(\boldsymbol{d} - \boldsymbol{\mu})^{\top} \right\rangle . \tag{5.30}$$

Still, by quoting this number, we have not chosen a distribution, they are simply properties of the likelihood. However, reducing it to $\boldsymbol{\mu}$ and $\boldsymbol{C}$ alone is a choice we will motivate in Section 5.7.2, not an identity. Read literally, Equation (5.29) is a statement about the data alone. Everything model-dependent enters only once we say what determines $\boldsymbol{\mu}$, $\boldsymbol{C}$, and, more generally, the shape of the distribution itself.

5.7.1 Folding in the model: the Bayesian view

The model enters through a single step: a theory predicts the mean of the data as a function of parameters $\boldsymbol{\theta}$,

$$\boldsymbol{\mu} \;\longrightarrow\; \boldsymbol{\mu}(\boldsymbol{\theta}) \,, \tag{5.31}$$

and, where the model is richer, the covariance and higher moments may carry parameter dependence as well. Substituting Equation (5.31) into Equation (5.29) turns the likelihood from a statement about the data alone into a function of the parameters,

$$\mathcal{L}(\boldsymbol{\theta}) \;=\; p(\boldsymbol{d} \mid \boldsymbol{\theta}) \,, \tag{5.32}$$

its shape still unspecified. It is illustrative to make the layered structure explicit. The data are generated by descending a chain:

$$\boldsymbol{\theta} \xrightarrow{\text{theory}} \boldsymbol{\mu}(\boldsymbol{\theta}) \xrightarrow{\text{noise}} \boldsymbol{d} \,. \tag{5.33}$$

Each arrow is a conditional distribution, and the joint distribution of all quantities in the problem factorises along the chain. Let us introduce a latent signal $\boldsymbol{s}$, model prediction, that the data are a noisy measurement of. The generative model factorises into conditionals, one per level,

$$p(\boldsymbol{d}, \boldsymbol{s} \mid \boldsymbol{\theta}) = \underbrace{p(\boldsymbol{d} \mid \boldsymbol{s})}_{\text{noise}} \, \underbrace{p(\boldsymbol{s} \mid \boldsymbol{\theta})}_{\text{signal}} . \tag{5.34}$$

Each level depends only on the one directly above it; this conditional independence is the entire content of the model. The likelihood used in practice is recovered by marginalising over the latent signal,

$$p(\boldsymbol{d} \mid \boldsymbol{\theta}) = \int \mathrm{d}\boldsymbol{s} \; p(\boldsymbol{d} \mid \boldsymbol{s}) \, p(\boldsymbol{s} \mid \boldsymbol{\theta}) \,. \tag{5.35}$$

The parameters set the distribution of a latent signal $\boldsymbol{s}$, and the measurement process turns that signal into a noisy data vector $\boldsymbol{d}$. The likelihood $p(\boldsymbol{d} \mid \boldsymbol{\theta})$ of Equation (5.32) is then nothing but Equation (5.35) with the latent signal already integrated out. Often, one uses:

$$p(\boldsymbol{s} \mid \boldsymbol{\theta}) = \delta^{\mathrm{D}}(\boldsymbol{s} - \boldsymbol{\mu}(\boldsymbol{\theta})), \tag{5.36}$$

which simply replaces the latent variable by the mean model prediction $\boldsymbol{\mu}(\boldsymbol{\theta})$.

5.7.2 Why are likelihoods so often Gaussian?

So far we have left the shape of the likelihood open, specifying only a mean and a covariance. The overwhelmingly common next step is to keep only those two moments and assume the data are Gaussian about the mean. There are three arguments that push towards a Gaussian likelihood

1. The CLT: Measured quantities are very often themselves sums or averages over many underlying contributions: a sample mean over many observations, a binned estimate over the points in a bin, a count of many independent events. Writing M for the number of contributions averaged within a single measured quantity, the leading non-Gaussianity, the skewness, falls as $M^{-1/2}$, so in the large-M regime the Gaussian is a limit rather than an assumption.

2. Maximum entropy: If the mean and covariance of Equation (5.30) are the only properties we are prepared to specify, the most conservative density consistent with them is the one that maximises the differential entropy subject to those two constraints and that density is uniquely the multivariate Gaussian (see **??** for more information). Any alternative with the same first two moments necessarily smuggles in unstated information about higher-order moments.

3. Summary statistics are frequently linear, or locally linearisable, functions of more fundamental data, and linear maps both preserve Gaussianity exactly and drive non-Gaussian inputs toward it (again, through the CLT).

In Section 4, the CLT acted on the score, a sum of n iid terms over repeated experiments, and produced the asymptotic normality of the estimator $\hat{\boldsymbol{\theta}}$, Equations (4.14) and (4.16). Here the theorem acts inside a single measurement, over the M contributions averaged to form it, and produces the normality of the data. In other words, the first governs the scatter of $\hat{\boldsymbol{\theta}}$ across hypothetical repetitions of the experiment, the second the scatter of $\boldsymbol{d}$ within the one experiment we actually have.

These arguments justify the Gaussian as a default, not as a law, so it is important to realise when it breaks: When only a few contributions are averaged (M small) the sampling distribution can visibly be skewed, closer to a χ^2 with few degrees of freedom than to a Gaussian. Strictly positive or bounded quantities, such as variances or counts, cannot be Gaussian near their boundary. Heavy tails from outliers or rare events violate the finite-variance premise of the CLT. Finally, when $\boldsymbol{C}$ is itself estimated from a finite sample rather than known, it is a random object and propagating that uncertainty replaces the naive Gaussian by a multivariate t-distribution.

5.8 Summary and some practical remarks

As before, we want to close with some practical remarks for Bayesian inference:

1. **Prior sensitivity**: Always check how sensitive the posterior is to the prior, especially with little data. A simple test is to repeat the analysis with a broader prior and verify that conclusions change only modestly. If they do not, more data or stronger theoretical motivation for the prior are needed.

2. **MCMC is the workhorse**: For most astrophysical and cosmological models the posterior cannot be evaluated analytically. Monte Carlo methods are required and will be discussed in Section 6.

3. **The Laplace approximation as a first check**: Before running Monte Carlo simulations, the Laplace approximation Equation (5.22) provides a fast consistency check. If the MAP and MLE differ substantially, or if the Hessian at the MAP differs strongly from the Fisher matrix, the posterior is likely non-Gaussian and the frequentist confidence ellipses of Section 4.10 should be treated with caution.

4. **Marginalisation vs. profiling**: Bayesian marginalisation (Equation 5.25) and the frequentist profile likelihood differ in finite samples. Marginalisation is generally more useful if prior information on parameters is available, since it propagates their full uncertainty rather than conditioning on a point estimate.

5. **Evidence computation**: The evidence integral is substantially harder to evaluate than the posterior peak. Nested sampling algorithms produce both posterior samples and a direct estimate of $\ln p(\boldsymbol{x} \mid M)$, and are widely used for model selection in astrophysics.

As in the MLE section, we summarise the overall workflow for Bayesian inference in Figure 5.7. The main difference to MLE workflow of Figure 4.9 lies in the interpretation of the results.

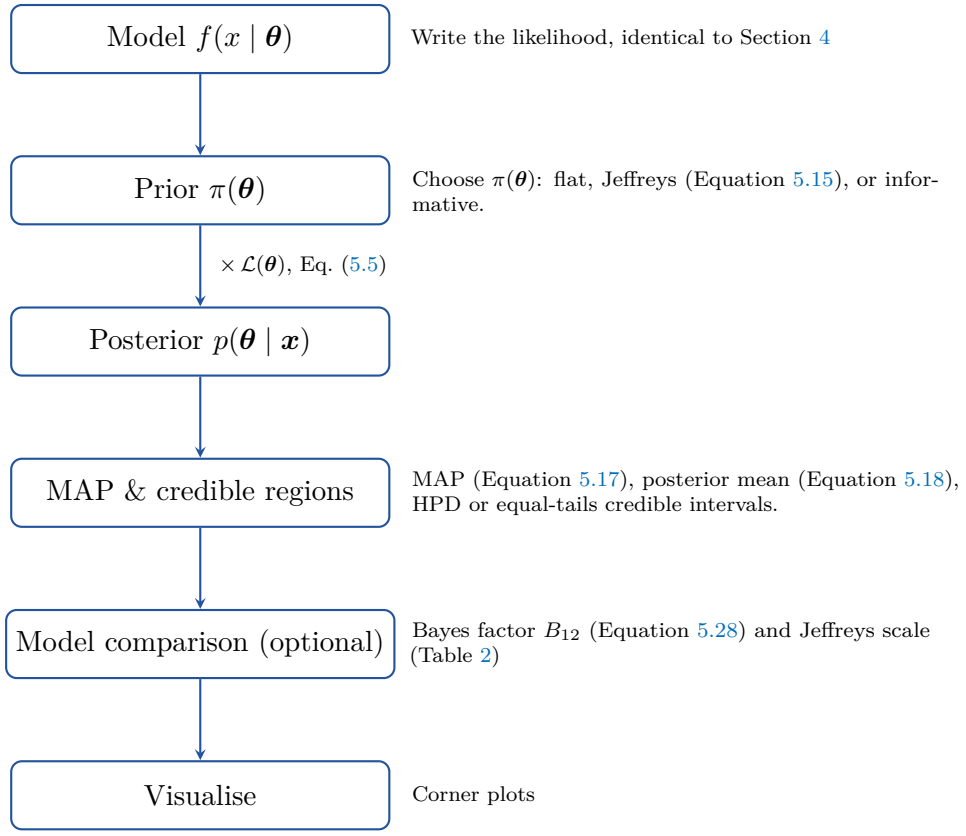


Figure 5.7: *End-to-end Bayesian inference workflow. Each step points to the equation or section where the corresponding result is derived, mirroring Figure 4.9.*

6 Sampling and Monte Carlo methods

In Section 5, we saw that the final answer to any Bayesian inference problem is the posterior $p(\boldsymbol{\theta} \mid \boldsymbol{x}) \propto \mathcal{L}(\boldsymbol{x} \mid \boldsymbol{\theta})\,\pi(\boldsymbol{\theta})$. Since the posterior is itself a distribution, we can describe it by point estimates or credible intervals. Furthermore, we saw that the evidence can be used for model comparison in Section 5.6. What all these operations have in common is that they are integrals over the parameter space or its regions. Quite generally, we want to evaluate:

$$\langle h \rangle = \int_\Theta h(\boldsymbol{\theta})\, p(\boldsymbol{\theta} \mid \boldsymbol{x})\, \mathrm{d}\boldsymbol{\theta}, \qquad p(\boldsymbol{x}) = \int_\Theta \mathcal{L}(\boldsymbol{x} \mid \boldsymbol{\theta})\, \pi(\boldsymbol{\theta})\, \mathrm{d}\boldsymbol{\theta}. \tag{6.1}$$

Outside the Gaussian (Laplace) regime of Section 5.4, neither can be done analytically. Real inference problems, however, are almost always not in the Gaussian regime, and hence we need tools to evaluate the kinds of integrals shown above. The central idea is extremely simple: instead of evaluating the integrals directly, we draw samples $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N$ from the posterior and replace integrals by averages over those samples. Most of this section, therefore, is concerned with how to generate such samples when we cannot even normalise the distribution they should follow.

6.1 Monte Carlo integration

Let us begin with the integration problem itself and assume, for the moment, that we already know how to draw independent samples $\boldsymbol{\theta}_i \sim p$ from some distribution $p(\boldsymbol{\theta})$. Any integral of the form of the first expression in Equation (6.1) is then an expectation, and we can estimate it by the *Monte Carlo estimator*

$$\hat{I}_N := \frac{1}{N} \sum_{i=1}^{N} h(\boldsymbol{\theta}_i), \qquad \boldsymbol{\theta}_i \sim p. \tag{6.2}$$

This is nothing but the sample mean of Section 3 applied to the random variable $h(\boldsymbol{\theta})$, and all the machinery we developed there can be applied in exactly the same way. In particular, the estimator is unbiased,

$$\langle \hat{I}_N \rangle = \frac{1}{N} \sum_{i=1}^{N} \langle h(\boldsymbol{\theta}_i) \rangle = \langle h \rangle = I, \tag{6.3}$$

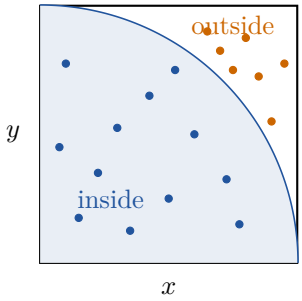
and, because the samples are independent, its variance follows the $1/N$ law of Equation (3.22),

$$\mathrm{Var}(\hat{I}_N) = \frac{\sigma_h^2}{N}, \qquad \sigma_h^2 = \langle h^2 \rangle - \langle h \rangle^2. \tag{6.4}$$

The Monte Carlo error therefore scales as $\sigma_h/\sqrt{N}$, and by the central limit theorem (see again Figure 3.6): The estimator is asymptotically Gaussian.

It cannot be stressed enough what is absent from Equation (6.4): the dimension d of the parameter space. The error of a Monte Carlo estimate falls as $N^{-1/2}$ regardless of how many parameters we have. We can compare this with a deterministic quadrature on a grid. A one-dimensional rule of order r has an error $\sim N^{-r}$ in the number of evaluation points N. In d dimensions, maintaining the same per-axis resolution costs exponentially many points: with N total evaluations only $N^{1/d}$ lie along each axis, so the error degrades to $\sim N^{-r/d}$. For the twenty- or hundred-parameter spaces typical of cosmological analyses, this becomes completely unfeasible: to halve the error of a product Simpson rule in $d = 20$, one would need $2^{20/4} \approx 32$ times as many points. This is the curse of dimensionality, and it is the single most

important reason why stochastic methods dominate high-dimensional inference. An example for Monte Carlo integration is shown in Figure 6.1. Here, we sample points uniformly in a square and then use the indicator function $\mathbb{I}$, which only selects those points satisfying $x^2 + y^2 \leq 1$.



$$\frac{\pi}{4} = \int_0^1 \int_0^1 \mathbb{I}\left[x^2 + y^2 \leq 1\right] \, \mathrm{d}x \, \mathrm{d}y$$

$$\hat{I}_N = \frac{N_{\text{in}}}{N} \xrightarrow{N \to \infty} \frac{\pi}{4}$$

$$\text{error} \propto 1/\sqrt{N}$$

Figure 6.1: *Monte Carlo integration as dart throwing. Points are drawn uniformly in the unit square, and the fraction landing inside the quarter circle (blue) estimates its area $\pi/4$. The estimator is the sample mean of the indicator function $h(x, y) = \mathbb{I}[x^2 + y^2 \leq 1]$, and its error shrinks as $1/\sqrt{N}$ independently of the number of dimensions.*

As mentioned above, however, this construction assumes that we can actually draw samples $\boldsymbol{\theta}_i \sim p$. Of course, drawing random numbers in a multidimensional cube is trivial; the hard part is the posterior. We typically know $p(\boldsymbol{\theta} \mid \boldsymbol{x})$ only up to the intractable normalisation $p(\boldsymbol{x})$, and the distribution lives in a high-dimensional space with a complicated shape which is only given numerically. We therefore need a way to generate samples from a distribution we can only evaluate pointwise (and only up to a constant).

6.2 Sampling from a known distribution

Let us start with the easier sub-problem in which the target is simple: a low-dimensional, properly normalised density. The techniques here are the foundation of the more powerful methods that follow.

6.2.1 Inverse transform sampling

Almost every sampling method ultimately relies on a stream of uniform random numbers $u \sim \mathcal{U}(0, 1)$, which pseudo-random number generators provide (recall exercise sheet 1 for random number generation and its pitfalls). So, how do we convert these into draws from a target distribution? The most direct route uses the CDF of the target PDF. First, we note that the CDF $F_X(x) = \mathbb{P}(X \leq x)$ is non-decreasing and maps the real line onto $[0, 1]$. Its inverse, F_X^{-1} is the quantile function, and we define it as:

$$F_X^{-1}(u) = \inf\{x : F_X(x) \geq u\}. \tag{6.5}$$

Now let us assume $u \sim \mathcal{U}(0, 1)$, then $X = F_X^{-1}(u)$ has CDF F_X:

$$\mathbb{P}(X \leq x) = \mathbb{P}\Big(F_X^{-1}(u) \leq x\Big) = \mathbb{P}(u \leq F_X(x)) = F_X(x). \tag{6.6}$$

Geometrically, we draw a height u uniformly on the vertical axis and read off the corresponding x from the graph of the CDF, as shown in Figure 6.2.

As an example, let us consider the exponential distribution

$$f(x) = \lambda \, \mathrm{e}^{-\lambda x}, \quad F_X(x) = 1 - \mathrm{e}^{-\lambda x} \tag{6.7}$$

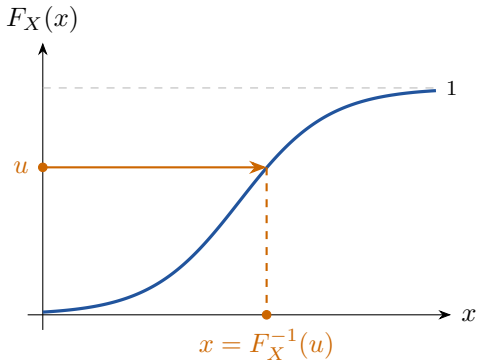


Figure 6.2: *Inverse transform sampling. A uniform draw $u \in [0, 1]$ on the vertical axis is mapped through the inverse CDF to a sample $x = F_X^{-1}(u)$ on the horizontal axis. Where the CDF is steep (high density), many values of u map into a narrow band of x, so those x are sampled more often.*

The inverse of the CDF is:

$$X = -\ln(1 - U)/\lambda. \tag{6.8}$$

The method is exact and efficient when the quantile function is available in closed form. For most distributions of interest, including the Gaussian, it is not, and even when it is, the approach does not generalise to the high-dimensional, unnormalised posteriors we actually care about. It nonetheless remains the workhorse for sampling the elementary distributions.

6.2.2 Rejection sampling

Let us make it one step more complicated and assume that we can evaluate the target $p(x)$ but cannot invert its CDF. If we have a proposal (or envelope) density $q(x)$ that we can sample, together with a constant M such that

$$p(x) \le M\, q(x) \qquad \forall x, \tag{6.9}$$

then we can sample p by the following recipe: draw a candidate $x \sim q$ and an independent $u \sim \mathcal{U}(0, 1)$, and accept x if

$$u \le \frac{p(x)}{M\, q(x)}, \tag{6.10}$$

otherwise reject it and try again. The accepted values are exact draws from p. The geometric interpretation is shown in Figure 6.3: we sample uniformly in the area under the envelope $Mq(x)$ and keep only the points that also fall under $p(x)$. Since uniform points under a curve are distributed in x according to that curve, the survivors follow p.

The overall acceptance probability is the ratio of the two areas, $1/M$ for normalised p and q. This exposes the fatal weakness of the method in high dimensions: to satisfy Equation (6.9) everywhere, M must exceed the largest density ratio, and for a d-dimensional Gaussian target sampled with a slightly broader Gaussian envelope M grows exponentially with d. You can also think of a sphere in d dimensions: as $d \to \infty$, most of the volume lies on the surface and will therefore be rejected. The acceptance rate then becomes arbitrarily small, and rejection sampling is useless for realistic posteriors.

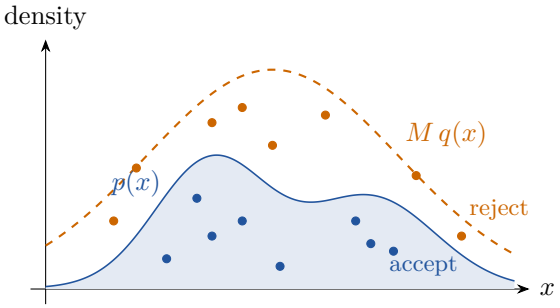


Figure 6.3: *Rejection sampling. Candidates are drawn under the envelope $M\,q(x)$ (orange dashed). A candidate at position x is accepted with probability $p(x)/[Mq(x)]$, i.e. kept if its uniform height lands below $p(x)$ (blue points) and discarded otherwise (orange points). The accepted points are exact samples from p.*

6.2.3 Importance sampling

Since rejection sampling throws away most points, especially in high dimensions, importance sampling aims to keep every draw by reweighting it. Writing the target expectation as an expectation under the proposal q,

$$\langle h \rangle_p = \int h(\boldsymbol{\theta})\, p(\boldsymbol{\theta})\, \mathrm{d}\boldsymbol{\theta} = \int h(\boldsymbol{\theta})\, \frac{p(\boldsymbol{\theta})}{q(\boldsymbol{\theta})}\, q(\boldsymbol{\theta})\, \mathrm{d}\boldsymbol{\theta} = \big\langle h(\boldsymbol{\theta})\, w(\boldsymbol{\theta}) \big\rangle_q, \tag{6.11}$$

with the importance weights $w(\boldsymbol{\theta}) = p(\boldsymbol{\theta})/q(\boldsymbol{\theta})$. Drawing $\boldsymbol{\theta}_i \sim q$, which we know how to do, gives the estimator

$$\hat{I}_N = \frac{1}{N} \sum_i h(\boldsymbol{\theta}_i)\, w(\boldsymbol{\theta}_i). \tag{6.12}$$

When the target is known only up to a constant, as is the case for the posterior, one uses a normalised estimator

$$\hat{I}_N = \frac{\sum_{i=1}^{N} h(\boldsymbol{\theta}_i)\, w(\boldsymbol{\theta}_i)}{\sum_{i=1}^{N} w(\boldsymbol{\theta}_i)}, \qquad w(\boldsymbol{\theta}_i) = \frac{\mathcal{L}(\boldsymbol{x} \mid \boldsymbol{\theta}_i)\, \pi(\boldsymbol{\theta}_i)}{q(\boldsymbol{\theta}_i)}, \tag{6.13}$$

in which the unknown normalisation cancels between the numerator and the denominator. An important feature of importance sampling is that it can estimate the evidence with relative ease. With the unnormalised weights of Equation (6.13), $w(\boldsymbol{\theta}) = \mathcal{L}(\boldsymbol{x} \mid \boldsymbol{\theta})\, \pi(\boldsymbol{\theta})/q(\boldsymbol{\theta})$, the identity Equation (6.11) with $h = 1$ gives the evidence directly:

$$p(\boldsymbol{x}) = \int_{\Theta} \mathcal{L}(\boldsymbol{x} \mid \boldsymbol{\theta})\, \pi(\boldsymbol{\theta})\, \mathrm{d}\boldsymbol{\theta} = \big\langle w(\boldsymbol{\theta}) \big\rangle_q \approx \frac{1}{N} \sum_i w(\boldsymbol{\theta}_i). \tag{6.14}$$

The quality of the estimate is entirely determined by how well q matches the target: if q has thinner tails than p a few samples acquire enormous weights and the variance explodes. A useful (but not sufficient) diagnostic is the effective sample size

$$N_{\mathrm{eff}} = \frac{\left(\sum_i w_i\right)^2}{\sum_i w_i^2}, \tag{6.15}$$

which counts how many equally-weighted samples the weighted set is worth; $N_{\mathrm{eff}} \ll N$ signals a poor proposal distribution. Importance sampling is invaluable for reusing an existing set of samples under a modified likelihood or prior. However, like rejection sampling, it requires a good global proposal and therefore does not, by itself, solve the high-dimensional problem.

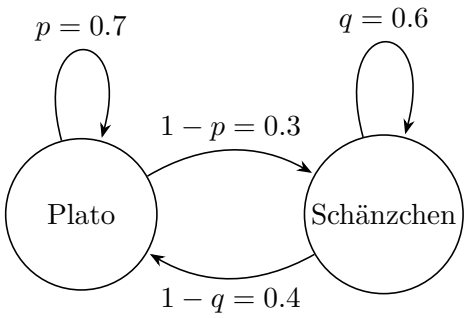


Figure 6.4: *Let us assume that there are only two bars in Bonn (I am not paid by any of these). Every bar visitor chooses a bar at random each week, depending only on their current state, i.e., the bar they were at the week before.*

6.3 Markov chains and stationary distributions

The idea of Markov Chain Monte Carlo (MCMC) is to give up on generating independent samples and instead construct a stochastic process that moves through parameter space, visiting each region in proportion to its posterior probability. This no longer requires a global envelope distribution. We only need to know how to take a good *local* step.

A sequence of random variables $\boldsymbol{\theta}_0, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots$ is a *Markov chain* if the next state depends only on the current one and not on the full history,

$$p(\boldsymbol{\theta}_{t+1} \mid \boldsymbol{\theta}_t, \boldsymbol{\theta}_{t-1}, \ldots, \boldsymbol{\theta}_0) = p(\boldsymbol{\theta}_{t+1} \mid \boldsymbol{\theta}_t) =: T(\boldsymbol{\theta}_{t+1} \mid \boldsymbol{\theta}_t). \tag{6.16}$$

The conditional density $T(\boldsymbol{\theta}' \mid \boldsymbol{\theta})$ is the transition probability. It tells us the probability of moving to $\boldsymbol{\theta}'$ given that we are at $\boldsymbol{\theta}$. A distribution $p^\star$ is called *stationary* (or invariant) for the chain if running one step of the chain leaves it unchanged,

$$\int_\Theta T(\boldsymbol{\theta}' \mid \boldsymbol{\theta})\, p^\star(\boldsymbol{\theta})\, \mathrm{d}\boldsymbol{\theta} = p^\star(\boldsymbol{\theta}'). \tag{6.17}$$

Our goal, therefore, is to design a transition probability whose stationary distribution is the posterior, then run the chain and collect its states. This might sound a bit obscure, so let us make a simple example illustrated in Figure 6.4 and construct the transition probability between the two states:

$$\boldsymbol{T} = \begin{pmatrix} \mathbb{P}(P \mid P) & \mathbb{P}(S \mid P) \\ \mathbb{P}(P \mid S) & \mathbb{P}(S \mid S) \end{pmatrix} = \begin{pmatrix} 0.7 & 0.3 \\ 0.4 & 0.6 \end{pmatrix}. \tag{6.18}$$

So, suppose that initially the visitors are split evenly between the two bars, $\mathbb{P}(P_0) = \mathbb{P}(S_0) = 1/2$. Labelling week zero and one with a subscript, respectively, after a week, we have:

$$\mathbb{P}(P_1) = \mathbb{P}(P \mid P)\mathbb{P}(P_0) + \mathbb{P}(P \mid S)\mathbb{P}(S_0) \tag{6.19}$$

and similarly for Schänzchen. We can write this as a matrix multiplication, so after one week

$$\begin{pmatrix} \mathbb{P}(P_1) \\ \mathbb{P}(S_1) \end{pmatrix} = (\boldsymbol{T}^\top) \begin{pmatrix} \mathbb{P}(P_0) \\ \mathbb{P}(S_0) \end{pmatrix} = \begin{pmatrix} 0.55 \\ 0.45 \end{pmatrix} \tag{6.20}$$

For n weeks, we would just want to calculate $(\boldsymbol{T}^\top)^n$ instead. Is this a stationary state? According to Equation (6.17), we need for a stationary state:

$$\begin{pmatrix} \mathbb{P}^*(P_{i+1}) \\ \mathbb{P}^*(S_{i+1}) \end{pmatrix} = (\boldsymbol{T}^\top) \begin{pmatrix} \mathbb{P}^*(P_i) \\ \mathbb{P}^*(S_i) \end{pmatrix}, \tag{6.21}$$

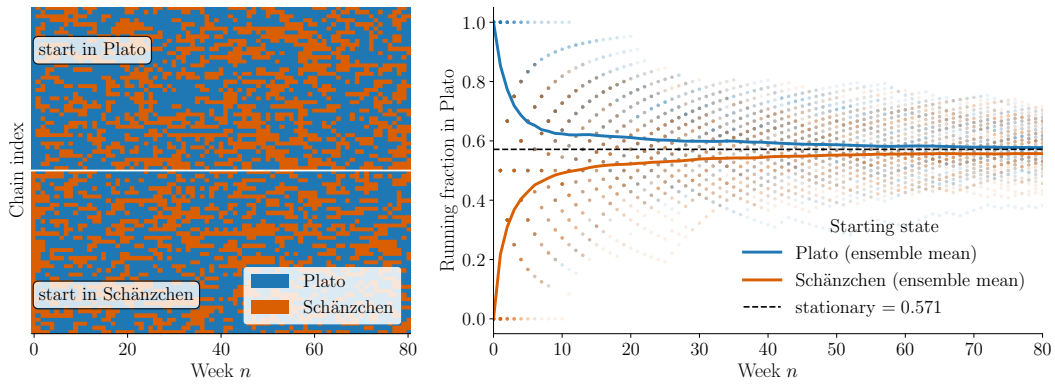


Figure 6.5: *Left: State sequences of $N = 80$ independent chains (blue = Plato, orange = Schänzchen); the top half are initialised in Plato, the bottom half in Schänzchen. Mixing is evident within ∼20 weeks, after which the two groups become visually indistinguishable. Right: Running time-average $\hat{\pi}_0(n) = (n+1)^{-1} \sum_{k=0}^{n} \mathbf{1}[X_k = 0]$ for 50 individual chains per starting state (dots) and the ensemble mean (thick lines). Regardless of the initial state, all chains converge to the same stationary value π_0 (dashed line), illustrating the ergodic theorem.*

which can be solved to find

$$\begin{pmatrix} \mathbb{P}^*(P_i) \\ \mathbb{P}^*(S_i) \end{pmatrix} = \begin{pmatrix} \frac{4}{7} \\ \frac{3}{7} \end{pmatrix}. \tag{6.22}$$

Alternatively, one can investigate the long-term behaviour of the process; thus, an eventual steady state is obtained by considering the situation as n tends to infinity. If the steady-state exists and the Markov Chain converges to it, we have

$$p^* = \lim_{n\to\infty} (\boldsymbol{T}^\top)^n p_0. \tag{6.23}$$

This statement holds for any initial state p_0, and hence we get

$$p_j^* = \lim_{n\to\infty} \sum_i (\boldsymbol{T}^\top)_{ji}^n \, p_{0,i}, \tag{6.24}$$

provided the steady-state exists. Therefore, the Markov chain converges to the stationary distribution as $n \to \infty$ independent of its starting point. Lastly, in this limit, $\lim_{n\to\infty}(T^n)_{ij}$ will have all rows equal to the steady-state distribution.

The two-bar example already contains, in miniature, every feature we will need. The visitor's choice each week depends only on where they are now and not on the entire history of their nights out, so the process is Markovian in the sense of Equation (6.16). Iterating the transition matrix drives the population towards a unique distribution $p^\star = (4/7, 3/7)^\top$, and, crucially, it does so from any starting point, as Figure 6.5 shows. The chain forgets where it started and remembers only the stationary distribution.

6.4 Conditions for sampling the posterior

In the bar example, finding the stationary distribution was almost trivial. With only two states, $\boldsymbol{T}$ was a 2×2 matrix, Equation (6.17) reduced to a pair of linear equations, and we could either solve them by hand or simply raise $\boldsymbol{T}^\top$ to a high power and read off the limit. It is natural to ask why we do not proceed in the same way for a real posterior: write down a kernel, impose stationarity, and solve.

First, the problem is posed backwards. In the bar example, the kernel $\boldsymbol{T}$ was given, and the stationary distribution was the unknown. For the posterior, it is the other way round. The stationary distribution is known: the posterior we wish to sample from. However, it is the kernel that we lack. Our task is hence to construct a transition rule that keeps it invariant. Importantly, infinitely many kernels share the same stationary distribution, so there is nothing to solve for in the first place.

Second, the state space is no longer two bars but a continuous, high-dimensional parameter space. The transition matrix becomes an integral kernel $T(\boldsymbol{\theta}' \mid \boldsymbol{\theta})$, and Equation (6.17) turns from a finite linear system into a homogeneous integral equation. Solving it head-on would mean discretising Θ onto a grid: the very strategy whose N^d cost we are trying to solve to begin with. Related, we cannot even write $p^\star$ down in a closed form over Θ. We are able to evaluate the posterior only pointwise, and only up to its normalisation $p(\boldsymbol{x})$, whereas any global solution of Equation (6.17) would require the normalised density everywhere at once.

The way out of this issue has already been spelt out in the definition of the Markov Chain Equation (6.16): we only need local information, in other words, information about a previous step. Hence, the idea is to impose local conditions that a candidate (stationary) kernel either satisfies or does not. We will find that two conditions are enough: the first, Section 6.4.1, guarantees that our chosen kernel really does leave the posterior invariant; the second, Section 6.4.2, makes sure that the resulting chain actually reaches the posterior from any starting point, and that averages taken along it converge.

6.4.1 Detailed balance

Detailed balance is a sufficient condition for stationarity that is far easier to enforce in practice than Equation (6.17) itself. A chain satisfies detailed balance (or is *reversible*) with respect to $p^\star$ if

$$p^\star(\boldsymbol{\theta})\, T(\boldsymbol{\theta}' \mid \boldsymbol{\theta}) = p^\star(\boldsymbol{\theta}')\, T(\boldsymbol{\theta} \mid \boldsymbol{\theta}'). \tag{6.25}$$

This says that, in equilibrium, the probability flux from $\boldsymbol{\theta}$ to $\boldsymbol{\theta}'$ exactly balances the reverse flux. Integrating Equation (6.25) over $\boldsymbol{\theta}$ and using the fact that the kernel integrates to unity in its first argument

$$\int_\Theta p^\star(\boldsymbol{\theta})\, T(\boldsymbol{\theta}' \mid \boldsymbol{\theta})\, \mathrm{d}\boldsymbol{\theta} = \int_\Theta p^\star(\boldsymbol{\theta}')\, T(\boldsymbol{\theta} \mid \boldsymbol{\theta}')\, \mathrm{d}\boldsymbol{\theta} = p^\star(\boldsymbol{\theta}') \underbrace{\int_\Theta T(\boldsymbol{\theta} \mid \boldsymbol{\theta}')\, \mathrm{d}\boldsymbol{\theta}}_{=1} = p^\star(\boldsymbol{\theta}'), \tag{6.26}$$

yields the sufficiency. It is worth noting again that stationarity, Equation (6.17), is a global balance condition: it demands only that the total probability flowing into any state from everywhere else equal the total flowing out. It permits steady circulating currents, provided they cancel when summed over the whole space. Detailed balance forbids these loops: it asks that every pair of states be in balance on its own. To verify the detailed balance, we need only inspect one pair of states at a time, and because only the ratio of densities appears, the unknown normalisation $p(\boldsymbol{x})$ cancels.

6.4.2 Ergodicity

The second condition, ergodicity, guarantees that stationarity is the chain's unique destination. We call a chain *ergodic* if it satisfies the following two properties:

i) Irreducibility (any region of positive posterior measure is reachable from any starting point in finite time). In Figure 6.6, two chains are shown with three

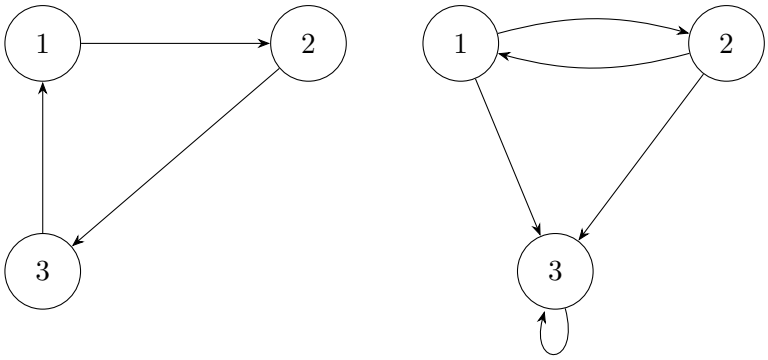


Figure 6.6: *On the left is an irreducible Markov chain since every state can eventually be reached. The chain on the right is reducible because state 3 is closed; once it is reached, it will never be left.*

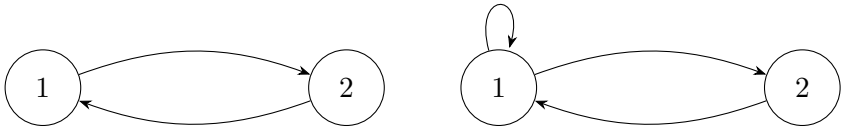


Figure 6.7: *On the left, there is a periodic Markov chain, while the chain on the right is aperiodic.*

states. The chain on the left can and will reach every state after a finite time; it cannot be further reduced. On the right-hand side, however, starting in state 3 will always keep you there. Also, there is a probability of getting trapped in state 3 even if one starts at 1 or 2. Therefore, this chain is reducible.

ii) Aperiodicity (it does not get trapped in deterministic cycles). In Figure 6.7 we show a periodic chain on the left and a non-periodic chain on the right. Note that the left chain in Figure 6.6 is also periodic and hence non-ergodic.

It is worth seeing what each one rules out. Without irreducibility, the chain can be trapped in a subregion forever, and the time average Equation (6.27) converges to an integral over that subregion rather than over the whole posterior. Without aperiodicity, the distribution of $\boldsymbol{\theta}_t$ never settles but cycles indefinitely, so there is no single limit to converge to.

In realistic inference, the danger is rarely a strictly reducible chain as a sensible proposal can usually reach anywhere in principle. The problem is that a chain is rather nearly reducible. A multimodal posterior whose peaks are well separated and connected only through deep, low-probability valleys is ergodic on paper, yet a local sampler may need a long time to cross from one mode to another. On any finite run it then behaves as though it were reducible. This is one of the central practical difficulties of MCMC.

For an ergodic chain, the stationary distribution is unique, the chain converges to it from any starting point, and, most importantly for us, the sample average of Equation (6.2) is replaced with a time average along a single chain:

$$\frac{1}{N}\sum_{t=1}^{N} h(\boldsymbol{\theta}_t) \xrightarrow{N\to\infty} \int_{\Theta} h(\boldsymbol{\theta})\, p^{\star}(\boldsymbol{\theta})\, \mathrm{d}\boldsymbol{\theta}. \tag{6.27}$$

There are two distinct notions of convergence at play here, and it is important, but also illustrative to keep them apart. The first is that, as t grows, the updating

law approaches $p^\star$ in distribution. This is the continuous analogue of the rows of $(\boldsymbol{T}^\top)^n$ becoming equal in the bar example. Early states, drawn before the chain has forgotten its starting point, are not representative of $p^\star$. Those states can be seen again in Figure 6.5, where only after some weeks the picture in both halves settles. In practice one discards an initial segment of the chain as burn-in. The second notion is the ergodic theorem Equation (6.27) itself: a law of large numbers for time averages along a single realisation, which is what allows us to estimate posterior expectations by summing along the chain. The first tells us when the samples become usable and the second tells us what we may do with them.

One might now say, hang-on if we have to remove a burn-in, is it not true that any point in the chain remembers a certain amount from its past? This is true, and the samples are now correlated rather than independent, which, as we shall see in Section 6.6, inflates the variance relative to Equation (6.4). But the estimator is still consistent. It remains only to find a kernel whose stationary distribution is the posterior.

6.5 The Metropolis–Hastings algorithm

The Metropolis–Hastings (MH) algorithm constructs a reversible kernel for *any* target $p(\boldsymbol{\theta})$ we can evaluate up to a constant. Given a current state $\boldsymbol{\theta}_t$, one cycle proceeds as follows:

1. Draw a candidate $\boldsymbol{\theta}'$ from a proposal distribution $q(\boldsymbol{\theta}' \mid \boldsymbol{\theta}_t)$.

2. Compute the acceptance probability

$$a(\boldsymbol{\theta}', \boldsymbol{\theta}_t) = \min\!\left(1,\; \frac{p(\boldsymbol{\theta}')\, q(\boldsymbol{\theta}_t \mid \boldsymbol{\theta}')}{p(\boldsymbol{\theta}_t)\, q(\boldsymbol{\theta}' \mid \boldsymbol{\theta}_t)}\right). \tag{6.28}$$

3. Draw $u \sim \mathcal{U}(0,1)$. If $u \le a$, accept and set $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}'$, otherwise reject and set $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t$.

The most important practical feature is visible in Equation (6.28): the target enters only through the ratio $p(\boldsymbol{\theta}')/p(\boldsymbol{\theta}_t)$. For the posterior, this is

$$\frac{p(\boldsymbol{\theta}' \mid \boldsymbol{x})}{p(\boldsymbol{\theta}_t \mid \boldsymbol{x})} = \frac{\mathcal{L}(\boldsymbol{x} \mid \boldsymbol{\theta}')\, \pi(\boldsymbol{\theta}')}{\mathcal{L}(\boldsymbol{x} \mid \boldsymbol{\theta}_t)\, \pi(\boldsymbol{\theta}_t)}, \tag{6.29}$$

in which the evidence $p(\boldsymbol{x})$ cancels exactly. Let us highlight again that this is why MCMC sidesteps the very integral that made the posterior hard to normalise in Section 5: we never need $p(\boldsymbol{x})$ to explore $p(\boldsymbol{\theta} \mid \boldsymbol{x})$. When the proposal is symmetric

$$q(\boldsymbol{\theta}' \mid \boldsymbol{\theta}) = q(\boldsymbol{\theta} \mid \boldsymbol{\theta}'), \tag{6.30}$$

as in the original Metropolis algorithm, the proposal ratio drops out too and the acceptance probability reduces to

$$a = \min(1, p(\boldsymbol{\theta}')/p(\boldsymbol{\theta}_t)). \tag{6.31}$$

Therefore, in plain words: always move to a higher density, and move to a lower density with a probability equal to the density ratio.

Mathematical background: Metropolis–Hastings satisfies detailed balance

For $\boldsymbol{\theta}' \neq \boldsymbol{\theta}$ the MH transition kernel is the product of proposing and accepting, $T(\boldsymbol{\theta}' \mid \boldsymbol{\theta}) = q(\boldsymbol{\theta}' \mid \boldsymbol{\theta})\, a(\boldsymbol{\theta}', \boldsymbol{\theta})$. Using $\min(1, r) = r \min(1, 1/r)$ we verify Equation (6.25) with $p^\star = p$:

$$\begin{aligned} p(\boldsymbol{\theta})\, T(\boldsymbol{\theta}' \mid \boldsymbol{\theta}) &= p(\boldsymbol{\theta})\, q(\boldsymbol{\theta}' \mid \boldsymbol{\theta}) \min\!\left(1, \frac{p(\boldsymbol{\theta}')q(\boldsymbol{\theta} \mid \boldsymbol{\theta}')}{p(\boldsymbol{\theta})q(\boldsymbol{\theta}' \mid \boldsymbol{\theta})}\right) \\ &= \min\big(p(\boldsymbol{\theta})q(\boldsymbol{\theta}' \mid \boldsymbol{\theta}),\; p(\boldsymbol{\theta}')q(\boldsymbol{\theta} \mid \boldsymbol{\theta}')\big), \end{aligned}$$

which is manifestly symmetric under $\boldsymbol{\theta} \leftrightarrow \boldsymbol{\theta}'$ and therefore equals $p(\boldsymbol{\theta}')\, T(\boldsymbol{\theta} \mid \boldsymbol{\theta}')$. The rejection part of the kernel (the chain staying put) is trivially in detailed balance. Hence, the posterior is stationary; with a proposal that can reach the whole space, the chain is also ergodic, and Equation (6.27) applies.

The most common choice is the *random-walk Metropolis* sampler, with a Gaussian proposal $\boldsymbol{\theta}' = \boldsymbol{\theta}_t + \boldsymbol{\epsilon}$, $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, s^2 \boldsymbol{\Sigma}_q)$, centred on the current state. The single tuning knob is the step scale s, and its effect is illustrated in Figure 6.8. Too small a step gives a high acceptance rate, but a chain that crawls and explores the posterior agonisingly slowly; too large a step proposes wildly into the tails, is almost always rejected, and the chain stands still. The art of running MH is to balance these failure modes.

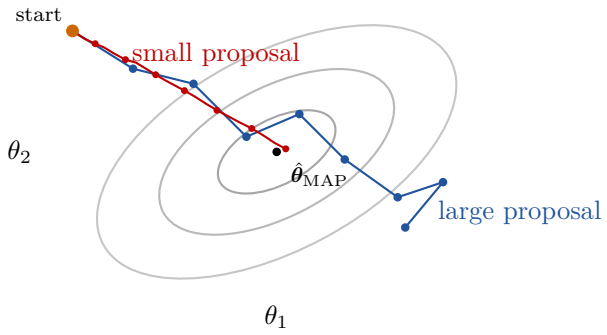


Figure 6.8: *A random-walk Metropolis chain (blue) exploring a correlated two-dimensional posterior (grey contours). From the starting point (orange) the chain takes local steps; where a node is repeated the proposal was rejected and the chain stayed in place. After an initial transient (burn-in) the chain settles into the bulk of the posterior and its states become representative samples.*

Mathematical background: the optimal acceptance rate

How aggressively should one tune the step size? For a random-walk proposal on a d-dimensional target, scaling analysis shows that the chain explores most efficiently when the proposal scale is $s \simeq 2.38/\sqrt{d}$ times the target scale, corresponding to an asymptotic acceptance rate of approximately 0.234 as $d \to \infty$ (and around 0.44 in one dimension). The intuition is a trade-off: at very high acceptance, the steps are too timid, while at very low acceptance, too much computation is wasted on rejected moves. Adaptive schemes take the running acceptance rate into account and adjust s towards this target, though adaptation must be diminished over time to preserve the stationary distribution.

6.6 Convergence diagnostics

A Markov chain produces correct samples only in the limit, and only after it has forgotten its starting point. Two practical consequences follow, and neither can be ignored when reporting results.

First, the chain must reach its stationary distribution before its states are representative of the underlying distribution. The early portion, during which the chain travels from an arbitrary starting point into the bulk of the posterior (the transient visible at the start of Figure 6.8), is called the burn-in and is discarded. Its length is judged in practice from the trace plot. One drops up to the point where the chain stops drifting and begins to fluctuate stably. Or, more defensibly, from the point at which several independently started chains become statistically indistinguishable. Second, consecutive states are correlated. This is quantified by the autocorrelation function of a scalar quantity $h(\boldsymbol{\theta})$ along the chain,

$$\rho(\tau) = \frac{\big\langle (h_t - \langle h \rangle)(h_{t+\tau} - \langle h \rangle) \big\rangle}{\sigma_h^2}, \tag{6.32}$$

which decays from $\rho(0) = 1$ over a characteristic lag. The relevant summary is the integrated autocorrelation time

$$\tau_{\mathrm{int}} = 1 + 2 \sum_{\tau=1}^{\infty} \rho(\tau), \tag{6.33}$$

in terms of which the variance of a chain-based estimator is no longer σ_h^2/N but

$$\mathrm{Var}(\hat{I}_N) \approx \frac{\sigma_h^2}{N}\, \tau_{\mathrm{int}} = \frac{\sigma_h^2}{N_{\mathrm{eff}}}, \qquad N_{\mathrm{eff}} = \frac{N}{\tau_{\mathrm{int}}}. \tag{6.34}$$

The effective sample size N_{eff} is the number of independent samples the correlated chain is worth. A chain of a million states with $\tau_{\mathrm{int}} = 10^3$ carries only a thousand independent samples. Reporting N_{eff}, not the raw chain length, is what honestly quantifies the Monte Carlo error.

The quantity $h(\boldsymbol{\theta})$ here is any scalar summary of the chain, and because these diagnostics are cheap one monitors several and takes the worst case: most commonly each parameter θ_j in turn. Neither the autocorrelation $\rho(\tau)$ of Equation (6.32) nor the time τ_{int} of Equation (6.33) is known in advance; both must be estimated from the single finite chain at hand. The expectation in Equation (6.32) is replaced by its sample analogue, the empirical autocovariance at lag τ,

$$\hat{c}(\tau) = \frac{1}{N} \sum_{t=1}^{N-\tau} (h_t - \hat{h})(h_{t+\tau} - \hat{h}), \qquad \hat{h} = \frac{1}{N} \sum_{t=1}^{N} h_t, \tag{6.35}$$

from which $\hat{\rho}(\tau) = \hat{c}(\tau)/\hat{c}(0)$ follows. The normalisation by $1/N$ rather than $1/(N-\tau)$ is deliberate: it biases $\hat{c}(\tau)$ low at large lag but sharply reduces its variance, damping the noisy tail at $\tau \gg 1$, that would otherwise dominate the sum. Even with this, the integrated time then follows from Equation (6.33), but the sum cannot be carried to infinity: the high-lag terms $\hat{\rho}(\tau)$ are pure noise, and adding them makes the estimate grow without bound. One therefore truncates at a window M,

$$\hat{\tau}_{\mathrm{int}}(M) = 1 + 2 \sum_{\tau=1}^{M} \hat{\rho}(\tau), \tag{6.36}$$

chosen self-consistently as the smallest M satisfying $M \geq c\, \hat{\tau}_{\mathrm{int}}(M)$ with $c \approx 5$. This is a convergence requirement in disguise: τ_{int} can be estimated reliably only from a

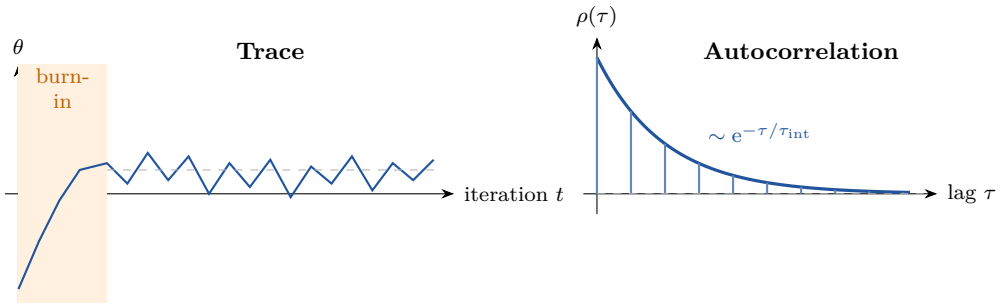


Figure 6.9: *Two MCMC diagnostics. Left: a trace plot of one parameter against iteration; the orange band is the burn-in transient, discarded before analysis, after which the chain fluctuates stably about the posterior mean. Right: the autocorrelation function decays over a characteristic lag set by the integrated autocorrelation time τ_{int}, which determines the effective sample size $N_{\mathrm{eff}} = N/\tau_{\mathrm{int}}$.*

chain that is itself many autocorrelation times long. As a rule of thumb, the chain should be at least several tens of τ_{int} long. The two convergence effects, burn-in and autocorrelation, are shown in Figure 6.9.

Trace plots and autocorrelation times diagnose a single chain, but they do not reveal a chain that is, for example, stuck in a local maximum of the posterior. The standard way to deal with this is to run several independent chains from different starting points and compare them. The Gelman–Rubin statistic $\hat{R}$ formalises this by comparing the variance between chains to the variance within them: if the chains have mixed and forgotten their starting points, the two estimates of the posterior variance agree and $\hat{R} \to 1$. In practice one requires $\hat{R}$ to be below a small threshold (commonly 1.01) for every parameter before trusting the result. Convergence can never be proven; passing several complementary diagnostics, however, is the best one can do.

Mathematical background: the Gelman–Rubin statistic

Run M chains of length N and let h_{mt} be the value of a scalar quantity in chain m at step t, with chain means $\bar{h}_m$ and grand mean $\bar{h}$. Define the *between-chain* and *within-chain* variances

$$B = \frac{N}{M-1}\sum_{m=1}^{M}(\bar{h}_m - \bar{h})^2, \qquad W = \frac{1}{M}\sum_{m=1}^{M} s_m^2, \qquad s_m^2 = \frac{1}{N-1}\sum_{t=1}^{N}(h_{mt} - \bar{h}_m)^2.$$

The marginal posterior variance is estimated by a weighted combination of the two, and the *potential scale reduction factor* is the ratio of that estimate to the within-chain variance,

$$\widehat{\mathrm{Var}}(h) = \frac{N-1}{N}W + \frac{1}{N}B, \qquad \hat{R} = \sqrt{\frac{\widehat{\mathrm{Var}}(h)}{W}}.$$

While the chains are still finding the posterior, B overestimates the variance (the chains sit in different places) and W underestimates it (each chain has explored only locally), so $\hat{R} > 1$. As the chains mix, both tend to the same true variance and $\hat{R} \to 1$ from above.

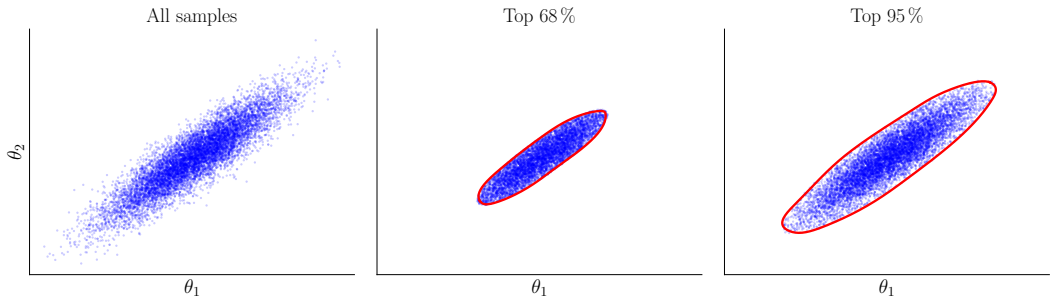


Figure 6.10: *All samples of a chain (after burn-in removal and thinning) in the very left. The two panels on the right only show those samples with the* 68(95)*% HPD. The red lines show the convex hull around the cloud of points. You can generate this plot with the notebook on GitHub.*

6.7 Visualising the posterior

The output of a converged, burn-in-trimmed run is nothing more than a set of samples $\{\boldsymbol{\theta}^{(i)}\}_{i=1}^{N}$ drawn from the posterior. The payoff of the whole sampling enterprise is that every quantity we might want from the posterior is now a simple operation on this sample set, with no further integral to evaluate.

The most important such operation is marginalisation, i.e. integrating out nuisance parameters as in Equation (5.25). With samples, it costs nothing: writing each draw as $\boldsymbol{\theta}^{(i)} = (\boldsymbol{\psi}^{(i)}, \boldsymbol{\eta}^{(i)})$, the sub-vectors $\{\boldsymbol{\psi}^{(i)}\}$ are distributed according to Equation (5.25). We marginalise simply by ignoring the columns we do not care about. This is what makes MCMC so well-suited to models with many nuisance parameters: the nuisance parameters are integrated out for free.

A one-dimensional marginal is summarised by a histogram or kernel density estimate of a single column, from which we read a point estimate (mean, median or mode) and a credible interval (Section 5.3), typically the HPD. For a pair of parameters, the joint marginal is a two-dimensional density, estimated from the two corresponding columns and displayed as credible contours. The creation of the contours is shown in Figure 6.10: we take the chain, sort it by log-posterior, and select the fraction $1-\alpha$ of points with the highest values. Computing the hull around these points yields the contour. Alternatively, one can simply compute a kernel density estimate.

The standard way to present a multi-dimensional posterior gathers all of these into a single corner plot (or triangle plot), shown in Figure 6.11: a lower-triangular grid whose diagonal panels are the 1D marginals of each parameter and whose off-diagonal panels are the 2D marginal contours for each pair. It compresses every one- and two-parameter marginal into one figure and makes degeneracies immediately apparent. In cosmology, these are produced with dedicated packages such as `getdist` and `corner`, which handle the density estimation, contour levels and labelling.
Both the diagonal densities and the off-diagonal contours are estimated from a finite, correlated sample, almost always by kernel density estimation. If you have too few effective samples, the contours are jagged and the tails poorly determined. Furthermore, aggressive smoothing washes out genuine structure and biases the apparent widths. Because the outer (95%) contour is fixed by the sparsely populated tails, it needs considerably more effective samples than the 68% contour to be trustworthy. This is the practical link back to Section 6.6: it is N_{eff}, and not the raw chain length, that controls how smooth and reliable these plots actually are.

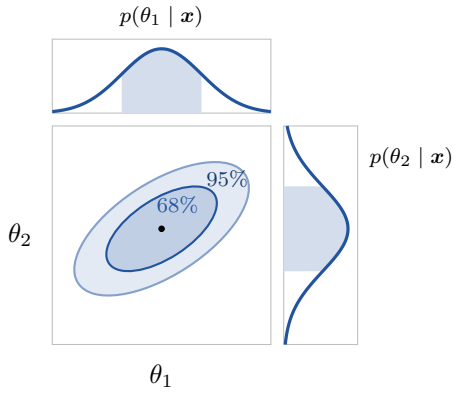


Figure 6.11: *A corner (or triangle) plot of a two-parameter posterior, the standard way to present MCMC output. The lower-left panel is the joint marginal of (θ_1, θ_2) with its 68% and 95% credible contours; the tilt encodes the degeneracy between the two parameters. The diagonal panels are the one-dimensional marginals obtained by projecting onto each parameter, with the shaded band marking the central 68% credible interval. For d parameters the same layout becomes a $d \times d$ lower triangle of all pairwise marginals.*

6.8 Gibbs sampling

Metropolis–Hastings makes no use of the structure of the posterior beyond our ability to evaluate it. Often, though, a model has structure worth exploiting: even when the full posterior is intractable, its conditional distributions for one parameter or a block of parameters may be standard distributions we can sample directly. For an illustration, see Figure 6.12.

Write the parameter vector as $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_d)$ and let $\boldsymbol{\theta}_{-\alpha}$ denote all components except the α-th. The full conditional of θ_α is

$$p(\theta_\alpha \mid \boldsymbol{\theta}_{-\alpha}, \boldsymbol{x}) \propto p(\boldsymbol{\theta} \mid \boldsymbol{x}), \tag{6.37}$$

the posterior read as a function of θ_α alone with the rest held fixed. A Gibbs sweep updates each parameter in turn by drawing from its full conditional, conditioning always on the most recent values of the others,

$$\begin{aligned} \theta_1^{(t+1)} &\sim p(\theta_1 \mid \theta_2^{(t)}, \ldots, \theta_d^{(t)}, \boldsymbol{x}), \\ \theta_2^{(t+1)} &\sim p(\theta_2 \mid \theta_1^{(t+1)}, \theta_3^{(t)}, \ldots, \theta_d^{(t)}, \boldsymbol{x}), \\ &\vdots \\ \theta_d^{(t+1)} &\sim p(\theta_d \mid \theta_1^{(t+1)}, \ldots, \theta_{d-1}^{(t+1)}, \boldsymbol{x}). \end{aligned} \tag{6.38}$$

There is no accept/reject step: every draw is kept. When the conditionals are standard distributions, as they are for hierarchical models, where they often come out Gaussian, inverse-gamma, etc., each draw is cheap and exact. In those cases, Gibbs sampling is coming to wreck house!

Its weakness is exactly the other side of the coin from its strength: it moves only along coordinate directions. When parameters are strongly correlated, the conditionals are narrow, and the chain inches along the degeneracy in small axis-aligned steps that mix slowly, as in Figure 6.12. A remedy is to block correlated parameters together, drawing them jointly from their multivariate conditional.

Mathematical background: Gibbs sampling is Metropolis–Hastings with acceptance one

A Gibbs update of θ_α is a Metropolis–Hastings step whose proposal is the full conditional itself, $q(\theta'_\alpha \mid \boldsymbol{\theta}) = p(\theta'_\alpha \mid \boldsymbol{\theta}_{-\alpha}, \boldsymbol{x})$, leaving $\boldsymbol{\theta}_{-\alpha}$ unchanged. Writing $p(\boldsymbol{\theta} \mid \boldsymbol{x}) = p(\theta_\alpha \mid \boldsymbol{\theta}_{-\alpha}, \boldsymbol{x})\, p(\boldsymbol{\theta}_{-\alpha} \mid \boldsymbol{x})$ and noting that $\boldsymbol{\theta}_{-\alpha}$ is common to both states, the Metropolis–Hastings ratio (Equation 6.28) collapses,

$$\frac{p(\theta'_\alpha, \boldsymbol{\theta}_{-\alpha} \mid \boldsymbol{x})}{p(\theta_\alpha, \boldsymbol{\theta}_{-\alpha} \mid \boldsymbol{x})} \frac{q(\theta_\alpha \mid \boldsymbol{\theta}')}{q(\theta'_\alpha \mid \boldsymbol{\theta})} = \frac{p(\theta'_\alpha \mid \boldsymbol{\theta}_{-\alpha}, \boldsymbol{x})}{p(\theta_\alpha \mid \boldsymbol{\theta}_{-\alpha}, \boldsymbol{x})} \frac{p(\theta_\alpha \mid \boldsymbol{\theta}_{-\alpha}, \boldsymbol{x})}{p(\theta'_\alpha \mid \boldsymbol{\theta}_{-\alpha}, \boldsymbol{x})} = 1.$$

Every proposal is therefore accepted, which is why Gibbs has no reject step. The same calculation exposes its weakness: the moves are precisely the conditional ones, taken along the coordinate axes.

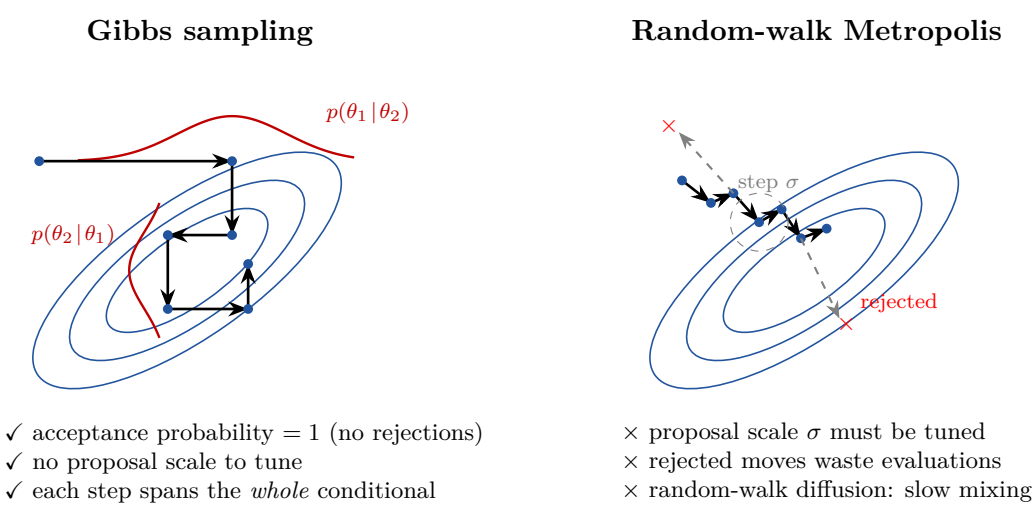


Figure 6.12: *When every full conditional $p(\theta_\alpha \mid \theta_{-\alpha})$ is available in closed form, Gibbs turns sampling into a sequence of* exact *one-dimensional draws with acceptance* 1 *and zero tuning. A general-purpose sampler must spend proposals and rejections just to imitate what Gibbs obtains for free.*

6.9 Gradient-based and ensemble samplers

Random-walk Metropolis works, but it explores by diffusion: each step is an uninformed local perturbation, which is deemed good only after the step has been proposed. In d dimensions, the step must shrink to keep the acceptance rate reasonable (recall the 0.234 guideline), while the number of steps needed to traverse the posterior grows. For high-dimensional posteriors, the diffusive scaling becomes very slow. Two families of samplers do better by using more information about the posterior than its value at a single point.

6.9.1 Hamiltonian Monte Carlo

Hamiltonian Monte Carlo (HMC) uses the gradient of the log-posterior to propose distant moves that nevertheless remain in regions of high probability. The idea is borrowed from Hamiltonian mechanics. We augment the parameters $\boldsymbol{\theta}$ (now thought of as positions) with auxiliary momentum variables $\boldsymbol{p}$ of the same dimension, and

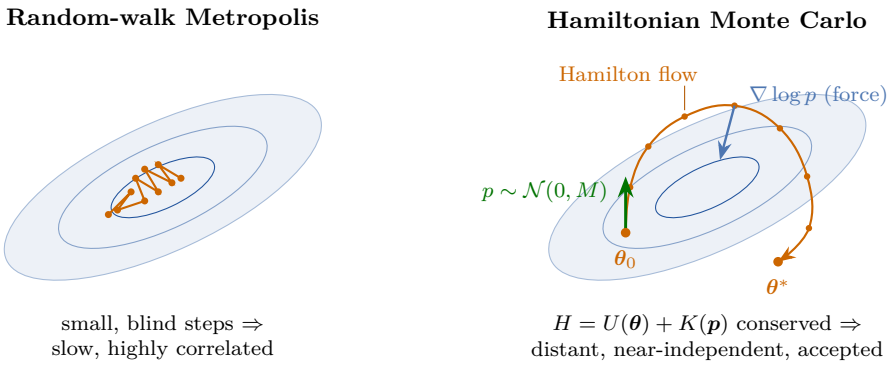


Figure 6.13: *Why gradient information helps. Left: random-walk Metropolis perturbs the state blindly; in a correlated posterior, the steps must be small, so the chain diffuses slowly and takes many highly correlated steps to cross the distribution. Right: Hamiltonian Monte Carlo follows the gradient of the log-posterior along a simulated trajectory, landing at a distant, near-independent proposed step that is almost always accepted.*

define a Hamiltonian

$$H(\boldsymbol{\theta}, \boldsymbol{p}) = U(\boldsymbol{\theta}) + K(\boldsymbol{p}), \quad U(\boldsymbol{\theta}) = -\ln p(\boldsymbol{\theta} \mid \boldsymbol{x}), \quad K(\boldsymbol{p}) = \frac{1}{2}\,\boldsymbol{p}^{\top}\boldsymbol{M}^{-1}\boldsymbol{p}, \tag{6.39}$$

a potential energy equal to the negative log-posterior, and a kinetic energy set by a symmetric, positive-definite mass matrix $\boldsymbol{M}$. The corresponding canonical distribution factorises,

$$\pi(\boldsymbol{\theta}, \boldsymbol{p}) \propto \mathrm{e}^{-H(\boldsymbol{\theta},\boldsymbol{p})} = p(\boldsymbol{\theta} \mid \boldsymbol{x})\,\mathcal{N}(\boldsymbol{p}\,;\,\mathbf{0}, \boldsymbol{M}), \tag{6.40}$$

so its marginal in $\boldsymbol{\theta}$ is exactly the posterior: if we can sample $(\boldsymbol{\theta}, \boldsymbol{p})$ from π and simply discard the momenta, the positions are posterior draws. To produce a proposal we draw a fresh momentum $\boldsymbol{p} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{M})$ and evolve the system under Hamilton's equations,

$$\frac{\mathrm{d}\boldsymbol{\theta}}{\mathrm{d}t} = \boldsymbol{M}^{-1}\boldsymbol{p}, \qquad \frac{\mathrm{d}\boldsymbol{p}}{\mathrm{d}t} = -\nabla U(\boldsymbol{\theta}) = \nabla \ln p(\boldsymbol{\theta} \mid \boldsymbol{x}), \tag{6.41}$$

for some fictitious time. These dynamics conserve H and follow the contours of the posterior, so the state moves along ridges and around degeneracies rather than diffusing across them. This is what allows a single proposal to travel far while remaining plausible. At the endpoint of the trajectory, a standard Metropolis accept/reject is taken

$$\alpha = \min\!\Big(1,\ \mathrm{e}^{-H(\boldsymbol{\theta}',\boldsymbol{p}')+H(\boldsymbol{\theta},\boldsymbol{p})}\Big), \tag{6.42}$$

which makes the posterior the exact stationary distribution. Because accurate integration keeps H nearly constant, the acceptance rate stays high even for long trajectories that reach near-independent proposals. The contrast with the random walk is shown in Figure 6.13.

The price we pay is that HMC needs the gradient, which restricts it to differentiable models, though automatic differentiation and emulators make this far less of a restriction than it once was. Furthermore, HMC has tuning knobs, the step size ϵ, the trajectory length L and the mass matrix $\boldsymbol{M}$, poor choices of which spoil its efficiency. The widely used No-U-Turn Sampler (NUTS) removes the most awkward of these by choosing L automatically, terminating a trajectory once it begins to double back on itself.

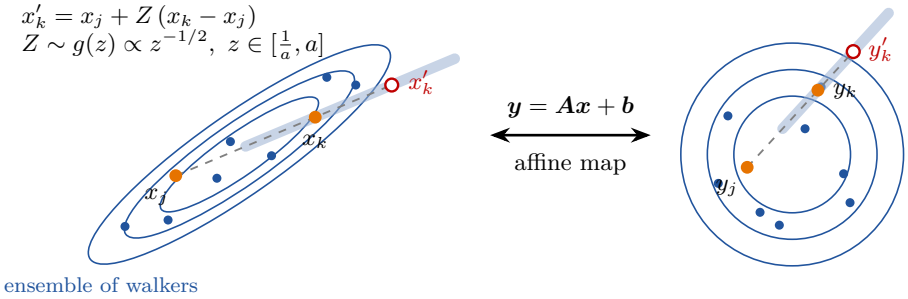


Figure 6.14: *The stretch move of the. To update walker $\boldsymbol{x}_k$, a second walker $\boldsymbol{x}_j$ is drawn at random from the ensemble and a candidate is proposed along the line joining the two, $\boldsymbol{y} = \boldsymbol{x}_j + z\,(\boldsymbol{x}_k - \boldsymbol{x}_j)$. The candidate therefore lies between $z = 1/a$ and $z = a$ on the connecting line, with $Z = 1$ recovering the current position. We show the strongly correlated, anisotropic target distribution the same ensemble and the same move viewed in whitened coordinates $\boldsymbol{y} = A^{-1}(\boldsymbol{x} - \boldsymbol{\mu})$, in which the target is isotropic. Since the proposal depends on the walkers only through the difference $\boldsymbol{x}_k - \boldsymbol{x}_j$, which transforms covariantly under the affine map, the identical scalar z generates the proposal in both frames and the acceptance ratio is unchanged. The sampler thus behaves identically on p and on any affine image of it, so its performance is insensitive to the linear transformation of the target.*

6.9.2 Affine-invariant ensemble samplers

When gradients are unavailable or expensive, a popular gradient-free alternative is the affine-invariant ensemble sampler of Goodman & Weare, packaged in the corresponding Python package `emcee`. Instead of a single chain, it evolves an ensemble of walkers $\{\boldsymbol{\theta}_k\}$ at once, and proposes a move for each walker using the current positions of the others. In the stretch move, walker k is updated by picking another walker $j \neq k$ at random and proposing a point along the line joining them,

$$\boldsymbol{\theta}'_k = \boldsymbol{\theta}_j + Z\,(\boldsymbol{\theta}_k - \boldsymbol{\theta}_j), \tag{6.43}$$

with the stretch factor Z drawn from a density $g(z) \propto z^{-1/2}$ on $[1/a, a]$ (typically $a = 2$). The proposal is accepted with probability

$$\min\left(1,\ Z^{\,d-1}\,\frac{p(\boldsymbol{\theta}'_k \mid \boldsymbol{x})}{p(\boldsymbol{\theta}_k \mid \boldsymbol{x})}\right), \tag{6.44}$$

where d is the dimension of the parameter space; the factor Z^{d-1} is what makes the move exactly reversible. The idea is sketched in Figure 6.14.

The defining property is affine invariance: the sampler behaves identically on a posterior and on any linearly transformed version of it. A long, thin, tilted degeneracy is internally equivalent to an isotropic one, so the ensemble handles it without retuning. That, together with needing no gradients and almost no hand-tuning, accounts for its popularity in astrophysics. Its main limitation is dimensionality: the ensemble must hold more walkers than parameters (in practice, a few times d), and its efficiency degrades in very high dimensions, where gradient-based methods such as HMC take over.

6.10 An illustrative example: Bayesian linear regression

Before turning a sampler loose on a problem we cannot solve any other way, it is worth meeting one we can solve. Linear regression with Gaussian noise and a

Gaussian prior: the posterior, the evidence and the predictive distribution are all available in closed form and can be easily compared to results from the sampling techniques discussed in this section.

We model a scalar target y as a linear combination of fixed basis functions $\boldsymbol{\phi}(x) = (\phi_1(x), \dots, \phi_d(x))^\top$ with weights $\boldsymbol{w}$ (the parameters):

$$y_i = \boldsymbol{w}^\top \boldsymbol{\phi}(x_i) + \varepsilon_i, \qquad \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{C}_\mathrm{N}), \tag{6.45}$$

the straight-line fit $\boldsymbol{\phi}(x) = (1, x)^\top$ being the simplest instance. Collecting the data into a vector $\boldsymbol{y}$ and the basis values into the design matrix $\boldsymbol{\Phi}_{ij} = \phi_j(x_i)$, the likelihood:

$$p(\boldsymbol{y} \mid \boldsymbol{w}) = \mathcal{N}(\boldsymbol{y}\,;\, \boldsymbol{\Phi}\boldsymbol{w},\, \boldsymbol{C}_\mathrm{N}) \propto \exp\!\left[-\frac{1}{2}(\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{w})^\top \boldsymbol{C}_\mathrm{N}^{-1}(\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{w})\right], \tag{6.46}$$

with noise covariance $\boldsymbol{C}_\mathrm{N}$ (diagonal, $\boldsymbol{C}_\mathrm{N} = \mathrm{diag}(\sigma_i^2)$, for independent errors). We complete the model with a Gaussian prior on the weights,

$$p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}\,;\, \boldsymbol{m}_0,\, \boldsymbol{S}_0), \tag{6.47}$$

which encodes whatever we believe about the weights before seeing the data. A broad, zero-centred choice $\boldsymbol{m}_0 = \mathbf{0}$, $\boldsymbol{S}_0 = \tau^2 \boldsymbol{I}$ is the usual default and is all we need here. Because both prior and likelihood are Gaussian and all parameters are linear, the posterior is also a Gaussian. One can see this as follows: The log-posterior is, up to a constant independent of $\boldsymbol{w}$,

$$\ln p(\boldsymbol{w} \mid \boldsymbol{y}) = -\frac{1}{2}(\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{w})^\top \boldsymbol{C}_\mathrm{N}^{-1}(\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{w}) - \frac{1}{2}(\boldsymbol{w} - \boldsymbol{m}_0)^\top \boldsymbol{S}_0^{-1}(\boldsymbol{w} - \boldsymbol{m}_0). \tag{6.48}$$

Both terms are quadratic in $\boldsymbol{w}$, so the sum is too. Collecting the quadratic and linear pieces,

$$\ln p(\boldsymbol{w} \mid \boldsymbol{y}) = -\frac{1}{2}\,\boldsymbol{w}^\top(\boldsymbol{S}_0^{-1} + \boldsymbol{\Phi}^\top \boldsymbol{C}_\mathrm{N}^{-1}\boldsymbol{\Phi})\boldsymbol{w} + \boldsymbol{w}^\top(\boldsymbol{S}_0^{-1}\boldsymbol{m}_0 + \boldsymbol{\Phi}^\top \boldsymbol{C}_\mathrm{N}^{-1}\boldsymbol{y}). \tag{6.49}$$

Comparing with the canonical Gaussian form

$$-\frac{1}{2}\boldsymbol{w}^\top \boldsymbol{S}_\mathrm{N}^{-1}\boldsymbol{w} + \boldsymbol{w}^\top \boldsymbol{S}_\mathrm{N}^{-1}\boldsymbol{m}_\mathrm{N}, \tag{6.50}$$

identifies the precision and mean by inspection:

$$\begin{aligned} p(\boldsymbol{w} \mid \boldsymbol{y}) &= \mathcal{N}(\boldsymbol{w}\,;\, \boldsymbol{m}_\mathrm{N},\, \boldsymbol{S}_\mathrm{N}), \\ \boldsymbol{S}_\mathrm{N}^{-1} &= \boldsymbol{S}_0^{-1} + \boldsymbol{\Phi}^\top \boldsymbol{C}_\mathrm{N}^{-1}\boldsymbol{\Phi}, \\ \boldsymbol{m}_\mathrm{N} &= \boldsymbol{S}_\mathrm{N}(\boldsymbol{S}_0^{-1}\boldsymbol{m}_0 + \boldsymbol{\Phi}^\top \boldsymbol{C}_\mathrm{N}^{-1}\boldsymbol{y}). \end{aligned} \tag{6.51}$$

The structure is worth reading off directly: precisions (inverse covariances) add, so the posterior precision is the prior precision plus the data precision $\boldsymbol{\Phi}^\top \boldsymbol{C}_\mathrm{N}^{-1}\boldsymbol{\Phi}$, and the posterior mean is a precision-weighted average of the prior mean and the data. Because the distribution is Gaussian, the posterior mean, mode and median coincide: the single best-fit weight vector $\boldsymbol{m}_\mathrm{N}$ is unambiguous.

We now have the complete posterior Equation (6.51) at hand and sampling is unnecessary in practice: a Gaussian can be sampled directly. Let us define the Cholesky decomposition of (any positive definite) $\boldsymbol{S}_\mathrm{N} = \boldsymbol{L}\boldsymbol{L}^\top$, then the quadratic form in the exponential of the Gaussian takes the following form:

$$(\boldsymbol{w} - \boldsymbol{m}_\mathrm{N})^\top (\boldsymbol{L}^\top)^{-1}\boldsymbol{L}^{-1}(\boldsymbol{w} - \boldsymbol{m}_\mathrm{N}) \tag{6.52}$$

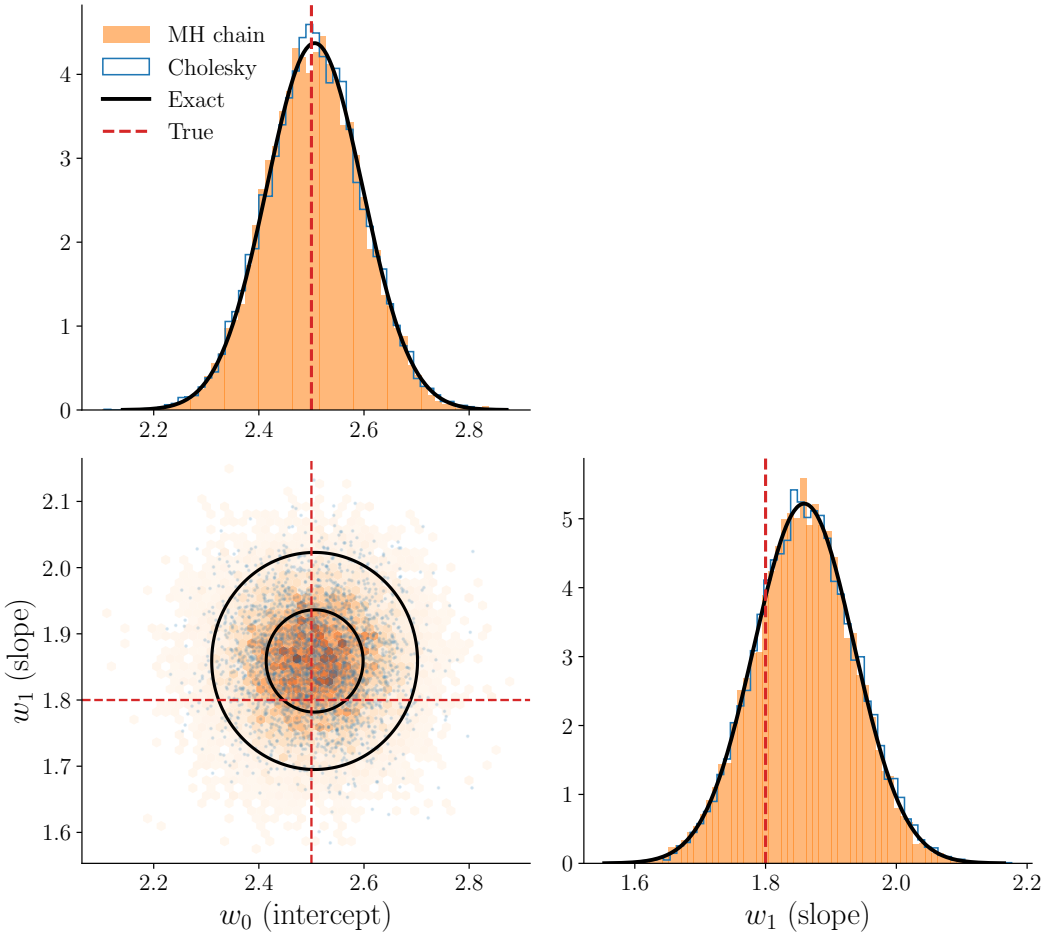


Figure 6.15: *Corner plot for the straight-line fit, intercept w_0 against slope w_1. The solid curve overlaid on the w_0 histogram is the exact marginal posterior from Equation (6.51). You can find the notebook to generate this plot and the convergence tests on GitHub.*

so if we choose

$$\boldsymbol{z} = \boldsymbol{L}^{-1}(\boldsymbol{w} - \boldsymbol{m}_{\mathrm{N}}), \tag{6.53}$$

it follows a standard Gaussian distribution, that is uncorrelated and unit variance. This means that for d parameters, we can simply sample d independent standard normal variables with and map them to $\boldsymbol{w}$ via:

$$\boldsymbol{w} = \boldsymbol{m}_{\mathrm{N}} + \boldsymbol{L}z. \tag{6.54}$$

The recipe for the sampler is that of Section 6.5. The log-posterior we hand the sampler is just the sum of the log-likelihood Equation (6.46) and the log-prior Equation (6.47), and the normalisation we never computed cancels in the acceptance ratio Equation (6.28). We then apply the full diagnostic suite of Section 6.6: several overdispersed chains, burn-in removal, the integrated autocorrelation time, and the Gelman–Rubin $\hat{R}$. The pooled sample mean must reproduce $\boldsymbol{m}_{\mathrm{N}}$, the sample covariance must reproduce $\boldsymbol{S}_{\mathrm{N}}$, and the marginals, obtained as in Section 6.7 by reading off columns, must reproduce the exact Gaussian marginals, as Figure 6.15 shows.

It is worth noting in passing that this is also the textbook case for Gibbs sampling (Section 6.8): the full conditional of any weight given the others is, like every

conditional of a multivariate Gaussian, again Gaussian and exactly known, so a Gibbs sampler can be written down with no rejection step at all.

6.11 Summary and some practical remarks

We have introduced various algorithms that enable sampling of the high-dimensional, unnormalised posteriors in astrophysics and cosmology. As before, we close with practical advice:

1. **Choose the sampler to fit the problem**: tractable conditionals favour Gibbs sampling; moderate dimensions without gradients favour affine-invariant ensemble samplers; high-dimensional differentiable models favour HMC.

2. **Always discard burn-in and report** N_{eff}: the raw chain length overstates the information content, unless you seriously oversampled (which is an option when computing time is not a problem). Estimate the integrated autocorrelation time (Equation 6.33) and quote the effective sample size, which sets the true Monte Carlo error via Equation (6.34).

3. **Run multiple chains**: convergence to a single mode is not convergence to the posterior. Use overdispersed starting points and require the Gelman–Rubin $\hat{R}$ should be close to one for every parameter before believing the result.

4. **Marginalise for free, and visualise honestly**: any marginal posterior is obtained by simply keeping the columns of interest. Summarise each parameter with a point estimate and a credible interval, and present the joint result as a corner plot of the 1D marginals and 68%/95% contours (Section 6.7). Make sure N_{eff} is large enough that the contours, the outer one especially, are not dominated by sampling noise.

5. **Use the Laplace approximation as a sanity check**: before launching an expensive run, the Gaussian approximation of Section 5.4 predicts roughly where the posterior sits and how wide it is. A converged chain whose mean and covariance disagree wildly with the Laplace prediction is a signal either of strong non-Gaussianity or of a sampling problem.

7 Machine learning

Everything so far has assumed that we can write down a model: a likelihood $\mathcal{L}(\boldsymbol{x} \mid \boldsymbol{\theta})$ relating the data to a handful of physically meaningful parameters, which we then estimate by maximum likelihood or sample from their posterior. Machine learning addresses a different question, one in which the relationship between inputs and outputs is generally unknown and possibly very complicated. The latter is, of course, also true for many physical models. However, we can generally write them down abstractly. Instead, the model is to be learned from data using a flexible family of functions rather than something derived from first principles.

This might seem a departure from the physics-driven approach of the preceding chapters, but machine learning is increasingly indispensable in physics because of the large amount of data involved. Tasks such as classifying galaxy morphologies, estimating photometric redshifts, separating signals from instrumental systematics, or compressing simulation outputs into fast emulators all involve mappings of high complexity that are difficult to model from first principles but can be learned efficiently from data. Note that data does not have to mean observed data in the real world; it can be anything, an output of a simulation or a function which is numerically expensive to compute.

The standard non-astrophysical example clarifies the definition above. Suppose we want to predict the sale price y of a house from a feature vector $\boldsymbol{x}$ containing floor area, number of rooms, distance to the city centre, year of construction, and so on. There is no physical law that tells us how these features combine to determine a price. The relationship is shaped by market dynamics, local demand, and countless other factors. What we do have, however, is a large dataset of past sales, giving labelled pairs $(\boldsymbol{x}_i, y_i)$ for $i = 1, \ldots, N$. The machine learning approach is to use these pairs to learn a predictor $h(\boldsymbol{x})$ that generalises to unseen houses, without ever writing down a specific model.

The boundary between classical inference and machine learning is not sharp (a straight-line fit is both at once, with a not-very-complicated functional family). But once again, it is more a change in perspective: from a few interpretable parameters to many uninterpretable ones, and from "what is the value of $\boldsymbol{\theta}$?" to "what function predicts y from $\boldsymbol{x}$?". The statistical foundations laid in the preceding sections are identical. Empirical risk minimisation generalises maximum likelihood estimation (Section 4), regularisation is a prior (Section 5), and the optimisation algorithm used to train deep networks often borrows aspects from Hamiltonian Monte Carlo (Section 6).

Machine learning can be divided into three categories. Supervised learning the data outcome in input–output pairs $(\boldsymbol{x}, y)$ and the goal is to predict y from $\boldsymbol{x}$. This splits further into regression, where y is continuous, and classification, where y is discrete. In unsupervised learning there are no targets, only inputs, and the aim is to uncover structure such as clusters, a low-dimensional representation, or the density itself. In reinforcement learning, the learning process is done by trial and reward. Most of this section covers supervised learning, which is closest to the estimation problems in the previous sections. We then discuss some instances of unsupervised learning and return to density estimation via normalising flows in Section 7.9, which forms the foundation of the simulation-based inference methods of Section 8.

7.1 The learning problem

Let us set the stage for a supervised learning problem. The data are a finite sample $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$, which we call the training data (or training set), assumed to be

drawn independently from some unknown joint distribution $f(\boldsymbol{x}, y)$. Here $\boldsymbol{x}_i \in \mathbb{R}^d$ is a vector of d input features (e.g. pixel values, spectral channels, or simulation parameters) and $y_i \in \mathbb{R}$ is a scalar target (e.g. a flux, a redshift, or a class label). Note that there is an asymmetry here: the inputs are typically multi-dimensional, while the output is a single number for regression or a single discrete label for classification. This can be trivially generalised so that y_i would also be a vector, but the scalar case suffices to develop all the key ideas. The hypothesis class $\mathcal{H}$ is the family of candidate functions $h : \mathcal{X} \to \mathcal{Y}$ we want to consider: lines, polynomials, neural networks. The loss $\ell(y, \hat{y})$ measures how bad a prediction $\hat{y} = h(\boldsymbol{x})$ is when the truth is y. And the optimisation searches $\mathcal{H}$ for a good h. This is made precise by the risk, the expected loss over the data-generating distribution,

$$R(h) = \big\langle \ell(y, h(\boldsymbol{x})) \big\rangle_{f(\boldsymbol{x},y)} = \int \ell(y, h(\boldsymbol{x}))\, f(\boldsymbol{x}, y)\, \mathrm{d}\boldsymbol{x}\, \mathrm{d}y. \tag{7.1}$$

The function we want is the one minimising this risk over $\mathcal{H}$. However, once again, $f(\boldsymbol{x}, y)$ is unknown, and all we have is the sample, so Equation (7.1) cannot be evaluated. We replace it with its Monte Carlo estimate over the training data, exactly the device of Section 6, giving the empirical risk

$$\hat{R}(h) = \frac{1}{N} \sum_{i=1}^{N} \ell(y_i, h(\boldsymbol{x}_i)), \tag{7.2}$$

and minimise that instead. This is called empirical risk minimisation (ERM),

$$\hat{h} = \arg\min_{h \in \mathcal{H}} \hat{R}(h). \tag{7.3}$$

We can ask again why $\hat{R}(h)$ is a good substitute for $R(h)$? Since the training pairs are iid, the individual losses $\ell(y_i, h(\boldsymbol{x}_i))$ are iid random variables with expectation $R(h)$, so $\hat{R}(h)$ is their sample mean. By the law of large numbers and the central limit theorem (Sections 3 and 6.1), it converges to $R(h)$ as $1/\sqrt{N}$. The subtlety, addressed in Section 7.3, is that minimising over all of $\mathcal{H}$ can make $\hat{R}(\hat{h})$ optimistically biased relative to $R(\hat{h})$ even when N is large.

Almost every supervised method is a choice of $\mathcal{H}$, of ℓ, and of how the minimisation Equation (7.3) is carried out. The whole difficulty of the subject is that minimising the empirical risk is only a proxy for what we actually care about, the true risk Equation (7.1). The gap between the two is the subject of Section 7.3. Before doing so, let us set aside the finite sample and the restriction to a particular $\mathcal{H}$ and ask what the best possible predictor would be if $h(\boldsymbol{x})$ could be chosen freely at every point $\boldsymbol{x}$. For the squared loss $\ell(y, \hat{y}) = (y - \hat{y})^2$, factorise the joint density as $f(\boldsymbol{x}, y) = f(\boldsymbol{x})\, f(y \mid \boldsymbol{x})$, so that the risk Equation (7.1) becomes a nested average,

$$R(h) = \Big\langle \big\langle (y - h(\boldsymbol{x}))^2 \mid \boldsymbol{x} \big\rangle \Big\rangle_{f(\boldsymbol{x})}, \tag{7.4}$$

where the inner average $\langle \cdot \mid \boldsymbol{x} \rangle \equiv \int (\cdot)\, f(y \mid \boldsymbol{x})\, \mathrm{d}y$ is over y at fixed $\boldsymbol{x}$, and the outer average $\langle \cdot \rangle_{f(\boldsymbol{x})}$ is over $\boldsymbol{x}$. Since $h(\boldsymbol{x})$ is a free choice at each $\boldsymbol{x}$ independently of every other point, minimising the outer average reduces to minimising the inner one pointwise. In other words, fix $\boldsymbol{x}$, write $c = h(\boldsymbol{x})$ for the value to be chosen there, and minimise

$$g(c) = \big\langle (y - c)^2 \mid \boldsymbol{x} \big\rangle = \big\langle y^2 \mid \boldsymbol{x} \big\rangle - 2c \big\langle y \mid \boldsymbol{x} \big\rangle + c^2 \tag{7.5}$$

over $c \in \mathbb{R}$. Setting $g'(c) = 2c - 2\langle y \mid \boldsymbol{x} \rangle = 0$ gives the unique minimiser $c^\star = \langle y \mid \boldsymbol{x} \rangle$, the conditional mean of y given $\boldsymbol{x}$ under $f(y \mid \boldsymbol{x})$. Repeating this at every $\boldsymbol{x}$ shows

that the risk-minimising predictor is the conditional mean function, also called the regression function,

$$h^\star(\boldsymbol{x}) = \langle y \mid \boldsymbol{x} \rangle. \tag{7.6}$$

For any predictor h, add and subtract $h^\star(\boldsymbol{x})$ inside the squared loss and expand:

$$\begin{aligned} \langle (y - h(\boldsymbol{x}))^2 \rangle = \langle (y - h^\star(\boldsymbol{x}))^2 \rangle + 2 \langle (y - h^\star(\boldsymbol{x}))(h^\star(\boldsymbol{x}) - h(\boldsymbol{x})) \rangle \\ + \langle (h^\star(\boldsymbol{x}) - h(\boldsymbol{x}))^2 \rangle. \end{aligned} \tag{7.7}$$

The cross term vanishes by iterated expectations. Since $h^\star(\boldsymbol{x}) - h(\boldsymbol{x})$ depends only on $\boldsymbol{x}$ (not on y), it can be pulled out of the inner average over y at fixed $\boldsymbol{x}$:

$$\begin{aligned} \langle (y - h^\star(\boldsymbol{x}))(h^\star(\boldsymbol{x}) - h(\boldsymbol{x})) \rangle_{f(\boldsymbol{x},y)} &= \Big\langle (h^\star(\boldsymbol{x}) - h(\boldsymbol{x})) \, \langle y - h^\star(\boldsymbol{x}) \mid \boldsymbol{x} \rangle \Big\rangle_{f(\boldsymbol{x})} \\ &= 0, \end{aligned} \tag{7.8}$$

since $\langle y - h^\star(\boldsymbol{x}) \mid \boldsymbol{x} \rangle = \langle y \mid \boldsymbol{x} \rangle - h^\star(\boldsymbol{x}) = 0$ by Equation (7.6). Substituting Equation (7.8) into Equation (7.7) gives the two-term risk split

$$\langle (y - h(\boldsymbol{x}))^2 \rangle = \underbrace{\langle (y - h^\star(\boldsymbol{x}))^2 \rangle}_{\text{irreducible}} + \langle (h^\star(\boldsymbol{x}) - h(\boldsymbol{x}))^2 \rangle. \tag{7.9}$$

The first term, the intrinsic scatter of y about its conditional mean, can never be removed by any choice of h; learning can only reduce the second. Explicitly, the irreducible term is

$$\langle (y - h^\star(\boldsymbol{x}))^2 \rangle = \Big\langle \langle (y - \langle y \mid \boldsymbol{x} \rangle)^2 \mid \boldsymbol{x} \rangle \Big\rangle_{f(\boldsymbol{x})} = \langle \mathrm{Var}(y \mid \boldsymbol{x}) \rangle_{f(\boldsymbol{x})} \equiv \sigma^2, \tag{7.10}$$

the average, over $\boldsymbol{x}$, of the conditional variance of y given $\boldsymbol{x}$. No choice of h can touch it, because $h(\boldsymbol{x})$ is a single number at each $\boldsymbol{x}$ and Equation (7.5) already showed that the conditional mean is the best such number. Any deviation from it only adds to the second, reducible term.

This does not mean that σ^2 is a fundamental floor for prediction accuracy. However, it is irreducible only given the chosen inputs $\boldsymbol{x}$. If $\boldsymbol{x}$ does not capture everything that determines y, part of what looks like noise is really just unmodelled signal.

7.2 Loss functions and the link to maximum likelihood

In the abstract formulation of Section 7.1, the hypothesis class $\mathcal{H}$ was left unspecified. In practice, we almost always restrict ourselves to a parametric family, writing the predictor as $h_{\boldsymbol{\theta}}(\boldsymbol{x})$ with a learnable parameter vector $\boldsymbol{\theta} \in \mathbb{R}^p$. For a linear model $h_{\boldsymbol{\theta}}(\boldsymbol{x}) = \boldsymbol{\theta}^\top \boldsymbol{x}$; for a neural network, $\boldsymbol{\theta}$ collects all the weights and biases across the layers (see Section 7.5). In either case the output $h_{\boldsymbol{\theta}}(\boldsymbol{x})$ is the model's prediction for target y given input $\boldsymbol{x}$, and the abstract search over functions Equation (7.3) becomes a search over parameters,

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta} \in \Theta} \frac{1}{N} \sum_{i=1}^{N} \ell(y_i, \, h_{\boldsymbol{\theta}}(\boldsymbol{x}_i)). \tag{7.11}$$

The choice of loss ℓ encodes an assumption about how targets scatter around the prediction. The most systematic way to specify this is to write down a conditional PDF $f(y \mid \boldsymbol{x}, \boldsymbol{\theta})$ in which $h_{\boldsymbol{\theta}}(\boldsymbol{x})$ appears as a location parameter, i.e. the predicted

mean or probability. Taking the loss to be the negative log-likelihood (NLL) of a single observation,

$$\ell(y, h_{\boldsymbol{\theta}}(\boldsymbol{x})) = -\ln f(y \mid \boldsymbol{x}, \boldsymbol{\theta}), \tag{7.12}$$

the empirical risk Equation (7.11) becomes $-\frac{1}{N}\ln \mathcal{L}(\boldsymbol{\theta})$, and minimising it is exactly the maximum-likelihood estimation of Section 4. Every choice of noise model thus induces a loss function, and the two most common choices are Gaussian and Bernoulli noise.

7.2.1 Regression: squared loss from Gaussian noise

The natural noise model for a continuous target is Gaussian. We write $y = h_{\boldsymbol{\theta}}(\boldsymbol{x}) + \varepsilon$ with $\varepsilon \sim \mathcal{N}(0, \sigma^2)$ independent of $\boldsymbol{x}$, so the predictor is the conditional mean. In the house price example, $h_{\boldsymbol{\theta}}(\boldsymbol{x})$ is the model's price prediction and ε captures all market factors not encoded in $\boldsymbol{x}$. The conditional distribution is then $f(y \mid \boldsymbol{x}, \boldsymbol{\theta}) = \mathcal{N}(y;\, h_{\boldsymbol{\theta}}(\boldsymbol{x}), \sigma^2)$ (cf. Section 4.3.1), and the NLL loss Equation (7.12) becomes

$$\ell(y, h_{\boldsymbol{\theta}}(\boldsymbol{x})) = \frac{\big(y - h_{\boldsymbol{\theta}}(\boldsymbol{x})\big)^2}{2\sigma^2} + \text{const.} \tag{7.13}$$

Minimising the empirical risk with this loss is least-squares regression, and we see that it is Gaussian maximum likelihood. The factor $1/(2\sigma^2)$ does not affect the location of the minimum with respect to $\boldsymbol{\theta}$ and is usually absorbed into the learning rate.

7.2.2 Classification: Bernoulli log-likelihood loss

For binary classification, $y \in \{0, 1\}$, the target is a class label rather than a continuous value. The appropriate noise model is the Bernoulli distribution (see Section 4): we model the conditional probability of class $y = 1$ as $h_{\boldsymbol{\theta}}(\boldsymbol{x}) \in [0, 1]$. To ensure the output lies in the unit interval, a linear score $\boldsymbol{w}^\top \boldsymbol{x}$ is passed through the logistic (sigmoid) function,

$$h_{\boldsymbol{\theta}}(\boldsymbol{x}) = \operatorname{sig}(\boldsymbol{w}^\top \boldsymbol{x}), \qquad \operatorname{sig}(t) = \frac{1}{1 + e^{-t}}, \tag{7.14}$$

which maps any real number into $[0, 1]$. The conditional distribution of y given $\boldsymbol{x}$ is then $\mathrm{Ber}(h_{\boldsymbol{\theta}}(\boldsymbol{x}))$, with probability mass function

$$p(y \mid \boldsymbol{x}) = h_{\boldsymbol{\theta}}(\boldsymbol{x})^y \left(1 - h_{\boldsymbol{\theta}}(\boldsymbol{x})\right)^{1-y}, \qquad y \in \{0, 1\}. \tag{7.15}$$

Taking $-\ln$ of Equation (7.15) gives the NLL loss,

$$\ell(y, h_{\boldsymbol{\theta}}(\boldsymbol{x})) = -\,y \ln h_{\boldsymbol{\theta}}(\boldsymbol{x}) - (1 - y) \ln\big(1 - h_{\boldsymbol{\theta}}(\boldsymbol{x})\big). \tag{7.16}$$

This penalises confident wrong predictions severely: if $y = 1$ but $h_{\boldsymbol{\theta}}(\boldsymbol{x}) \to 0$, the first term diverges, forcing the model to assign high probability to the correct class. In the literature, this loss is commonly called the cross-entropy loss, a name that will be justified when we discuss information theory in **??**.

7.3 Generalisation: overfitting and the bias–variance trade-off

A model is useful only if it predicts well on data it has not seen. The empirical risk on the training set is an optimistically biased estimate of the true risk, because the very same data were used to choose h. The honest measure is the risk on an independent test set, and the difference between training and test error is the generalisation gap.

If $\mathcal{H}$ is too small (too few parameters, too rigid), no member of it can capture the structure in the data, both training and test error are large, and the model underfits (Figure 7.1, left). This corresponds to high bias: the average fit over many training sets is systematically offset from the true $h^\star$, regardless of which sample is drawn. If $\mathcal{H}$ is too large, the model has enough freedom to fit not only the signal but the noise in the particular training sample: the training error becomes tiny while the test error blows up, and the model overfits (right). This corresponds to high variance: small changes in the training data lead to very different fitted functions. The art is to match the capacity of the model to the information in the data.

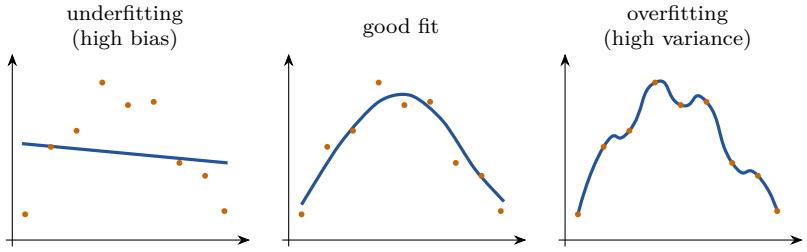


Figure 7.1: *The same nine noisy data points fit by models of increasing capacity. A straight line (left) is too rigid to follow the trend and underfits. A smooth low-order curve (middle) captures the signal while ignoring the scatter. A highly flexible curve (right) passes through every point, chasing the noise, and overfits: it will generalise poorly to new data.*

This tension is made quantitative by the bias–variance decomposition. In Equation (3.16) we already split the mean squared error of a point estimator $\hat{\boldsymbol{\theta}}$ into a squared bias and a variance term. The same split applies here, with two differences: the object being estimated is now a function $\hat{h}(\boldsymbol{x})$ rather than a scalar, and the irreducible noise σ^2 of the target y adds a third, irreducible term that no predictor can remove. Averaging over training sets $\mathcal{D}$ of size N, the expected test error at a point $\boldsymbol{x}$ is

$$\big\langle (y - \hat{h}(\boldsymbol{x}))^2 \big\rangle = \sigma^2 + \underbrace{\big(\langle \hat{h}(\boldsymbol{x}) \rangle_{\mathcal{D}} - h^\star(\boldsymbol{x})\big)^2}_{\text{bias}^2} + \underbrace{\big\langle \big(\hat{h}(\boldsymbol{x}) - \langle \hat{h}(\boldsymbol{x}) \rangle_{\mathcal{D}}\big)^2 \big\rangle_{\mathcal{D}}}_{\text{variance}}, \tag{7.17}$$

where the irreducible σ^2 is the intrinsic noise of Section 7.1. Equations (7.7) and (7.8) hold for any predictor, including a data-dependent $\hat{h}(\,\cdot\,; \mathcal{D})$: averaging additionally over $\mathcal{D}$, the cross term still vanishes by the same iterated-expectation argument, since $\hat{h}(\boldsymbol{x})$ is independent of y at fixed $\boldsymbol{x}$ and $\mathcal{D}$. The first term is σ^2 by Equation (7.10). The remaining approximation term splits by introducing $\bar{h}(\boldsymbol{x}) \equiv \langle \hat{h}(\boldsymbol{x}) \rangle_{\mathcal{D}}$ and adding and subtracting $\bar{h}$:

$$\big\langle \big(h^\star(\boldsymbol{x}) - \hat{h}(\boldsymbol{x})\big)^2 \big\rangle_{\mathcal{D}} = \underbrace{\big(h^\star(\boldsymbol{x}) - \bar{h}(\boldsymbol{x})\big)^2}_{\text{bias}^2} + \underbrace{\big\langle \big(\bar{h}(\boldsymbol{x}) - \hat{h}(\boldsymbol{x})\big)^2 \big\rangle_{\mathcal{D}}}_{\text{variance}}, \tag{7.18}$$

since the cross term $2\,(h^\star - \bar{h})\,\langle \bar{h} - \hat{h} \rangle_{\mathcal{D}} = 0$ by definition of $\bar{h}$. Together these give Equation (7.17).

The bias is the error of the average fit: a rigid model is biased because it cannot reach $h^\star$. The variance is the sensitivity of the fit to the particular sample: a flexible model has high variance because it chases the noise. Raising the capacity lowers the bias but raises the variance, and the test error, their sum, is minimised at intermediate capacity (Figure 7.2). This is the curse of dimensionality of Section 6 in

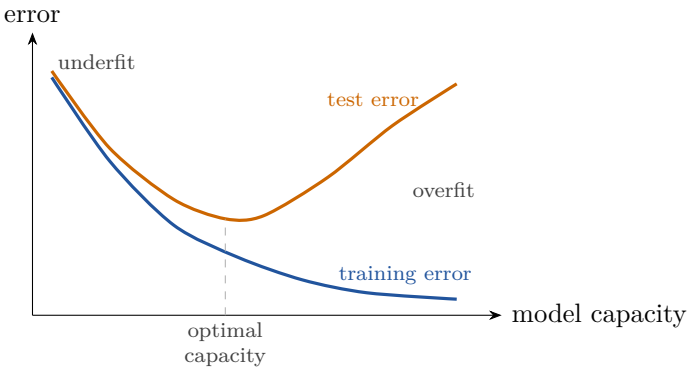


Figure 7.2: *Training and test error as functions of model capacity. The training error falls monotonically as the model gains freedom, but the test error is U-shaped: dominated by bias on the left (underfitting) and by variance on the right (overfitting), with a minimum at intermediate capacity. Only the test error reflects true performance.*

different clothes: in high dimensions a flexible model has enormous variance unless it is heavily constrained.

Because the test set must not be touched during training, in practice, one splits the data three ways: a training set to fit h, a validation set to choose between models and tune knobs such as the capacity, and a test set, inspected once, to report the final performance.

7.4 Regularisation and the Bayesian connection

The lesson of Section 7.3 is that raw empirical-risk minimisation with a flexible model overfits. Regularisation counters this by adding a penalty that discourages complexity, minimising

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \Big[\hat{R}(\boldsymbol{\theta}) + \lambda\, \Omega(\boldsymbol{\theta}) \Big], \tag{7.19}$$

where Ω measures the "size" of the parameters and the hyperparameter $\lambda > 0$ sets how hard the penalty is, itself tuned on the validation set. The two standard choices are the squared (L_2) penalty $\Omega = \|\boldsymbol{\theta}\|_2^2$, called ridge regression, which shrinks all parameters smoothly towards zero, and the L_1 penalty $\Omega = \|\boldsymbol{\theta}\|_1$, called the lasso.

This step is simply the Bayesian picture of training and risk and therefore not ad hoc. Well, at least, not more ad hoc than choosing a prior in the first place. With the negative-log-likelihood loss Equation (7.12), the objective Equation (7.19) reads, up to an additive constant,

$$\sum_{i=1}^{N} \big[-\ln f(y_i \mid \boldsymbol{x}_i, \boldsymbol{\theta}) \big] + \lambda\, \Omega(\boldsymbol{\theta}) = -\ln \Big[\mathcal{L}(\boldsymbol{\theta})\, \pi(\boldsymbol{\theta}) \Big] + \text{const}, \qquad \pi(\boldsymbol{\theta}) \propto \mathrm{e}^{-\lambda \Omega(\boldsymbol{\theta})}. \tag{7.20}$$

Minimising this is therefore equivalent to maximising the posterior $p(\boldsymbol{\theta} \mid \mathcal{D}) \propto \mathcal{L}(\boldsymbol{\theta})\, \pi(\boldsymbol{\theta})$, i.e. to MAP estimation as introduced in Section 5. The ridge penalty $\lambda \|\boldsymbol{\theta}\|_2^2$ corresponds to a Gaussian prior $\pi \propto \exp(-\lambda \|\boldsymbol{\theta}\|_2^2)$; the lasso penalty corresponds to a Laplace prior. The strength λ acts as an inverse prior variance: a larger λ encodes a stronger prior belief that the parameters are small. The Bayesian and machine learning vocabularies are thus exactly the same. Other regularisers act less explicitly. Halting the optimisation before it converges (early stopping), injecting

noise, or randomly disabling parts of a network during training (dropout), all limit the effective capacity and combat overfitting.

7.5 From linear regression to neural networks

The simplest useful hypothesis class is linear in its parameters. We already studied this model in detail in Section 6.10: choosing fixed basis functions $\boldsymbol{\phi}(\boldsymbol{x}) = (\phi_1(\boldsymbol{x}), \dots, \phi_M(\boldsymbol{x}))^\top$, the predictor is

$$h_{\boldsymbol{w}}(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{\phi}(\boldsymbol{x}), \tag{7.21}$$

linear in $\boldsymbol{w}$ but not necessarily in $\boldsymbol{x}$: polynomial bases $\phi_j(x) = x^j$ or Fourier bases fit nonlinear data without any problems. Under the squared loss, minimising the empirical risk Equation (7.11) is ordinary least squares, and the solution is the closed-form result already derived in Section 6.10; the only difference is that here we take the limit of a flat prior. Ridge regression adds $\lambda \boldsymbol{I}$ to $\boldsymbol{\Phi}^\top \boldsymbol{\Phi}$ before inversion, corresponding to the Gaussian prior of Section 7.4, and cures the ill-conditioning that arises when the design matrix has near-collinear columns.

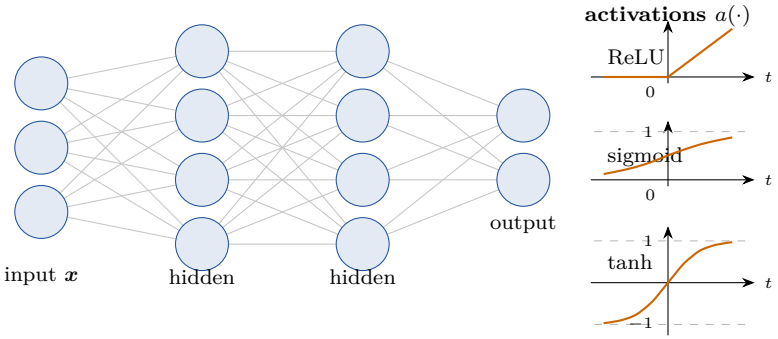


Figure 7.3: *Left: a feed-forward neural network with two hidden layers. Each node applies an affine combination of the previous layer's outputs followed by a pointwise nonlinear activation (Equation 7.22). Right: three common activation functions plotted on a common output scale. ReLU outputs are non-negative (range $[0, \infty)$, the t-axis marks the minimum). Sigmoid maps to $(0, 1)$ and is always positive; the dashed line shows the upper asymptote at 1. Tanh maps to $(-1, 1)$ and can produce negative outputs; the dashed lines mark the asymptotes at ± 1. The output range determines which function suits a given layer or output head.*

The limitation of Equation (7.21) is that the basis functions are fixed in advance: the model can only combine features we have chosen by hand. A neural network removes this restriction by learning the basis functions as well. A feed-forward network composes simple layers, each an affine map followed by a nonlinear activation a applied component-wise,

$$\boldsymbol{h}^{(l)} = a(\boldsymbol{W}^{(l)} \boldsymbol{h}^{(l-1)} + \boldsymbol{b}^{(l)}), \qquad \boldsymbol{h}^{(0)} = \boldsymbol{x}, \tag{7.22}$$

with the final layer $\boldsymbol{h}^{(L)}$ the network output. Each weight matrix $\boldsymbol{W}^{(l)} \in \mathbb{R}^{d_l \times d_{l-1}}$ and bias $\boldsymbol{b}^{(l)} \in \mathbb{R}^{d_l}$ specify layer l, where d_l is its width. Because the product $\boldsymbol{W}^{(l)} \boldsymbol{h}^{(l-1)}$ can have any number of rows, consecutive layers can freely expand or contract the representation. For the network in Figure 7.3 the layer widths are $(d_0, d_1, d_2, d_3) = (3, 4, 4, 2)$: the input vector $\boldsymbol{x} \in \mathbb{R}^3$ is first widened to four hidden features, kept at four in the second hidden layer, and then projected down to a

two-dimensional output. The full forward pass reads

$$\begin{aligned} \boldsymbol{h}^{(0)} &= \boldsymbol{x} \in \mathbb{R}^3, \\ \boldsymbol{h}^{(1)} &= a\Big(\boldsymbol{W}^{(1)}\boldsymbol{h}^{(0)} + \boldsymbol{b}^{(1)}\Big) \in \mathbb{R}^4, \quad \boldsymbol{W}^{(1)} \in \mathbb{R}^{4\times 3}, \\ \boldsymbol{h}^{(2)} &= a\Big(\boldsymbol{W}^{(2)}\boldsymbol{h}^{(1)} + \boldsymbol{b}^{(2)}\Big) \in \mathbb{R}^4, \quad \boldsymbol{W}^{(2)} \in \mathbb{R}^{4\times 4}, \\ h_{\boldsymbol{\theta}}(\boldsymbol{x}) &= \boldsymbol{W}^{(3)}\boldsymbol{h}^{(2)} + \boldsymbol{b}^{(3)} \in \mathbb{R}^2, \quad \boldsymbol{W}^{(3)} \in \mathbb{R}^{2\times 4}. \end{aligned} \tag{7.23}$$

Two points deserve attention. First, the activation a is essential: without it, every layer would be a linear map, the composition of all layers would collapse to a single affine map $\boldsymbol{W}\boldsymbol{x} + \boldsymbol{b}$, a sad lasagna: many layers, no more expressive than one. This also has a clean interpretation in terms of the Fisher information (Section 4.5). For any number of consecutive linear layers:

$$\boldsymbol{W}^{(L)} \dots \boldsymbol{W}^{(1)}\boldsymbol{x} = \boldsymbol{W}\boldsymbol{x}. \tag{7.24}$$

Because the prediction is unchanged, so is the log-likelihood, and therefore the Fisher information is invariant under the entire family of such transformations. The individual weight matrices $\boldsymbol{W}^{(1)} \dots \boldsymbol{W}^{(L)}$ are not identifiable from data: infinitely many pairs yield the same prediction and the same likelihood. This invariance is a structural property of the model: the extra parameters introduced by stacking linear layers carry no new information about the data. Second, the output layer $l = L$ is typically kept linear (no activation), so the network can produce any real-valued vector for regression. For binary classification, one appends a sigmoid to obtain a probability in $(0, 1)$. Common activations for the hidden layers are shown in Figure 7.3. The full parameter vector $\boldsymbol{\theta} = \{\boldsymbol{W}^{(l)}, \boldsymbol{b}^{(l)}\}_{l=1}^{L}$ is learned by minimising the loss. The reach of this architecture is captured by the universal approximation theorem: for any continuous function $f : K \to \mathbb{R}$ on a compact set $K \subset \mathbb{R}^{d_0}$ and any $\epsilon > 0$, there exists a single-hidden-layer network $h_{\boldsymbol{\theta}}$ such that

$$\sup_{\boldsymbol{x}\in K} \big|h_{\boldsymbol{\theta}}(\boldsymbol{x}) - f(\boldsymbol{x})\big| < \epsilon, \tag{7.25}$$

provided the activation a is not a polynomial. The result extends to virtually all activations used in practice (sigmoid, tanh, and rectified linear unit (ReLU)).

Three caveats are, however, in order. First, the theorem guarantees the existence of weights achieving the bound but says nothing about how to find them. Second, the hidden-layer width required for a given ϵ can grow exponentially with d_0, a manifestation of the curse of dimensionality. Deep networks of many layers with moderate width can represent the same functions with far fewer total parameters, which is one practical motivation for depth. Third, approximating f on the training inputs does not imply good generalisation to new data: a network wide enough to fit $\mathcal{D}$ exactly may still overfit badly (Section 7.3). The theorem therefore confirms that expressiveness is not the bottleneck. What limits learning in practice is data, computation, and regularisation.

7.6 Training: gradient descent and backpropagation

Only for the linear model Equation (7.21) is the minimum available in closed form. For a neural network, the loss is a complicated, non-convex function of the parameters, and we minimise it iteratively. The basic method is gradient descent: repeatedly step downhill along the negative gradient of the empirical risk,

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta\, \nabla_{\boldsymbol{\theta}} \hat{R}(\boldsymbol{\theta}_t), \tag{7.26}$$

where the learning rate $\eta > 0$ controls the step size. Its choice matters: if η is too large the iterate overshoots the minimum, and the loss diverges or oscillates without converging; if η is too small, the update is so cautious that convergence requires a prohibitive number of iterations (Figure 7.4). A further difficulty arises in ill-conditioned regions of the loss surface, where the curvature differs greatly between directions. In these elongated valleys the gradient points mostly across the valley rather than along it, so the plain gradient descent zig-zags and makes very slow net progress toward the minimum (Figure 7.5).

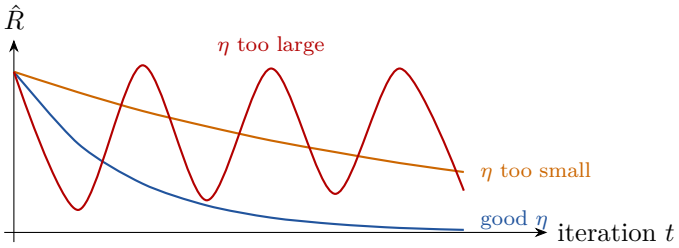


Figure 7.4: *Effect of the learning rate η on gradient descent. A well-chosen η (blue) gives smooth, fast convergence. Too small an η (orange) converges but requires far more iterations. Too large an η (red) causes the iterate to overshoot the minimum repeatedly: the loss oscillates and makes no net progress.*

Two practical extensions address these problems. The first is momentum: rather than stepping in the direction of the current gradient alone, one maintains a running weighted average of past gradients called the velocity,

$$\begin{aligned} \boldsymbol{v}_{t+1} &= \beta\, \boldsymbol{v}_t + \nabla_{\boldsymbol{\theta}} \hat{R}(\boldsymbol{\theta}_t), \\ \boldsymbol{\theta}_{t+1} &= \boldsymbol{\theta}_t - \eta\, \boldsymbol{v}_{t+1}, \end{aligned} \tag{7.27}$$

with momentum coefficient $\beta \in [0, 1)$, typically $\beta = 0.9$. Gradient components that oscillate across the valley cancel in the running average $\boldsymbol{v}$, while the component pointing consistently along the valley accumulates: the optimiser accelerates in directions of sustained descent and damps the cross-valley oscillations.

The second extension addresses the fact that different parameters may have very different gradient magnitudes: a parameter that governs a rarely-active feature may receive tiny gradients most of the time, yet benefit from a large step when it does. Adaptive learning rates give each parameter its own effective step size, normalised by the history of its gradients. The Adam optimiser combines momentum with this normalisation by maintaining exponential moving averages of the gradient $\boldsymbol{g}_t = \nabla_{\boldsymbol{\theta}} \hat{R}(\boldsymbol{\theta}_t)$ and its element-wise square,

$$\begin{aligned} \boldsymbol{m}_{t+1} &= \beta_1\, \boldsymbol{m}_t + (1 - \beta_1)\, \boldsymbol{g}_t, \\ \boldsymbol{s}_{t+1} &= \beta_2\, \boldsymbol{s}_t + (1 - \beta_2)\, \boldsymbol{g}_t \odot \boldsymbol{g}_t, \end{aligned} \tag{7.28}$$

where $\odot$ is the Hadamard product, $[\boldsymbol{u} \odot \boldsymbol{v}]_i = u_i v_i$, so $\boldsymbol{g}_t \odot \boldsymbol{g}_t$ is the vector of squared gradient components. The parameter update is then

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta\, \frac{\hat{\boldsymbol{m}}_{t+1}}{\sqrt{\hat{\boldsymbol{s}}_{t+1}} + \varepsilon}, \tag{7.29}$$

where $\hat{\boldsymbol{m}}$ and $\hat{\boldsymbol{s}}$ are bias-corrected versions of the two averages (divided by $1 - \beta_1^{t+1}$ and $1 - \beta_2^{t+1}$ respectively, to remove the transient bias from initialising at zero), the square root is taken component-wise, and $\varepsilon > 0$ prevents division by zero. Parameters

with a large squared-gradient history (large $\hat{\boldsymbol{s}}$) receive a smaller effective step; those with a small history receive a larger one. The defaults $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\varepsilon = 10^{-8}$ work well across a wide range of problems, making Adam the de facto standard optimiser for deep learning.

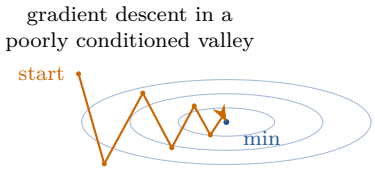


Figure 7.5: *Gradient descent on a poorly conditioned loss surface (elongated contours). The gradient points mostly across the narrow valley, so the iterates overshoot and zig-zag, making slow progress along it. Momentum and adaptive step sizes, or better conditioning of the problem, alleviate this.*

Evaluating $\nabla \hat{R}$ requires the full training set at every step, which is wasteful when N is large. Stochastic gradient descent instead estimates the gradient from a small random mini-batch $\mathcal{B}$ at each step,

$$\nabla_{\boldsymbol{\theta}} \hat{R} \;\approx\; \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \nabla_{\boldsymbol{\theta}}\, \ell\big(y_i, h_{\boldsymbol{\theta}}(\boldsymbol{x}_i)\big), \tag{7.30}$$

a noisy but unbiased estimate, Monte Carlo once more applied now to the gradient itself. The noise is not purely a nuisance: it helps the iteration escape poor local minima and saddle points, and a full pass over the data (an epoch) costs as much as a single full-batch step yet yields many cheap updates.

The remaining question is how to compute $\nabla_{\boldsymbol{\theta}} \hat{R}$ for a network with potentially millions of weights buried in the composition Equation (7.22). The answer is back-propagation, the chain rule applied systematically from the output layer backwards. It helps to split each layer into a pre-activation and an activation step. Write

$$\boldsymbol{z}^{(l)} = \boldsymbol{W}^{(l)} \boldsymbol{h}^{(l-1)} + \boldsymbol{b}^{(l)}, \qquad \boldsymbol{h}^{(l)} = a\Big(\boldsymbol{z}^{(l)}\Big), \tag{7.31}$$

and define the error signal at layer l as

$$\boldsymbol{\delta}^{(l)} \equiv \frac{\partial \ell}{\partial \boldsymbol{z}^{(l)}}, \tag{7.32}$$

the gradient of the per-sample loss with respect to the pre-activation. At the output layer with a linear activation (regression), this is simply $\boldsymbol{\delta}^{(L)} = \partial \ell / \partial \boldsymbol{h}^{(L)}$. Applying the chain rule one layer at a time from $l = L - 1$ back to $l = 1$ gives the backward recursion

$$\boldsymbol{\delta}^{(l)} = \Big[\big(\boldsymbol{W}^{(l+1)}\big)^{\top} \boldsymbol{\delta}^{(l+1)}\Big] \odot a'\Big(\boldsymbol{z}^{(l)}\Big), \tag{7.33}$$

where a' is the derivative of the activation applied component-wise and $\odot$ is the element-wise product defined in Equation (7.28). Once all error signals are known, the gradients with respect to the weights and biases of layer l follow from a single outer product,

$$\frac{\partial \ell}{\partial \boldsymbol{W}^{(l)}} = \boldsymbol{\delta}^{(l)} \big(\boldsymbol{h}^{(l-1)}\big)^{\top}, \qquad \frac{\partial \ell}{\partial \boldsymbol{b}^{(l)}} = \boldsymbol{\delta}^{(l)}. \tag{7.34}$$

The crucial observation is that each step of Equation (7.33) requires exactly one matrix–vector product $(\boldsymbol{W}^{(l+1)})^{\top} \boldsymbol{\delta}^{(l+1)}$, mirroring the product $\boldsymbol{W}^{(l)} \boldsymbol{h}^{(l-1)}$ in the forward pass. The backward pass therefore has the same computational cost as

the forward pass, so the full gradient over a mini-batch costs only a small constant multiple of evaluating the loss itself. This is the same gradient computation that underpins Hamiltonian Monte Carlo (Section 6); modern automatic-differentiation frameworks (PyTorch, JAX) implement it for arbitrary computation graphs without the user deriving a single partial derivative by hand.

7.7 Gaussian processes: Bayesian learning of functions

The models of the previous section are fitted by point estimation of the parameters $\boldsymbol{\theta}$, which gives a single function $\hat{h}$ but no measure of uncertainty in that function. A Gaussian process (GP) provides a fully Bayesian treatment: it places a prior directly over functions, and the posterior after seeing the data is again a distribution over functions, quantifying which fits are plausible given the observations. This is the function-space analogue of Bayesian inference for finite-dimensional parameter vectors introduced in Section 5.

Formally, a GP is a collection of random variables $\{g(\boldsymbol{x})\}_{\boldsymbol{x}\in\mathcal{X}}$ such that any finite subcollection $(g(\boldsymbol{x}_1),\dots,g(\boldsymbol{x}_n))$ has a joint Gaussian distribution. It is fully specified by its mean function

$$m(\boldsymbol{x}) = \langle g(\boldsymbol{x}) \rangle \tag{7.35}$$

and its covariance kernel

$$k(\boldsymbol{x},\boldsymbol{x}') = \langle (g(\boldsymbol{x}) - m(\boldsymbol{x}))(g(\boldsymbol{x}') - m(\boldsymbol{x}')) \rangle, \tag{7.36}$$

and we write $g \sim \mathcal{GP}(m,k)$. The mean function encodes prior beliefs about the average level of g; it is commonly set to zero for simplicity. The kernel $k(\boldsymbol{x},\boldsymbol{x}')$ controls how correlated $g(\boldsymbol{x})$ and $g(\boldsymbol{x}')$ are expected to be before seeing data, and therefore how smooth the sampled functions will be. The most widely used choice is the squared-exponential (or radial basis function) kernel

$$k_{\mathrm{SE}}(\boldsymbol{x},\boldsymbol{x}') = \sigma_f^2 \exp\!\left(-\frac{\|\boldsymbol{x}-\boldsymbol{x}'\|^2}{2\ell^2}\right), \tag{7.37}$$

whose hyperparameters are the signal variance σ_f^2 and the length-scale ℓ. Points separated by much more than ℓ are nearly uncorrelated; points much closer are almost perfectly correlated, so sample paths are infinitely differentiable. Other kernels encode different smoothness assumptions, and kernels can be combined by addition or multiplication to build richer priors.

7.7.1 GP regression

Given noisy observations $y_i = g(\boldsymbol{x}_i) + \varepsilon_i$ with $\varepsilon_i \overset{\text{iid}}{\sim} \mathcal{N}(0,\sigma_n^2)$, Bayesian inference proceeds exactly as in Section 5: update the GP prior with the Gaussian likelihood. Because both are Gaussian, the posterior is again a GP, yielding a closed-form Gaussian update analogous to the conjugate Bayesian analysis of Section 5. Collecting the training inputs $\boldsymbol{X} = (\boldsymbol{x}_1,\dots,\boldsymbol{x}_N)^\top$, targets $\boldsymbol{y} = (y_1,\dots,y_N)^\top$, and a new test point $\boldsymbol{x}_* \in \mathcal{X}$, define the kernel matrices

$$[\boldsymbol{K}]_{ij} = k(\boldsymbol{x}_i,\boldsymbol{x}_j), \qquad [\boldsymbol{k}_*]_i = k(\boldsymbol{x}_i,\boldsymbol{x}_*), \qquad k_{**} = k(\boldsymbol{x}_*,\boldsymbol{x}_*).$$

The predictive distribution at $\boldsymbol{x}_*$ is then Gaussian,

$$g(\boldsymbol{x}_*) \mid \boldsymbol{X},\boldsymbol{y},\boldsymbol{x}_* \;\sim\; \mathcal{N}(\mu_*,\,\sigma_*^2), \tag{7.38}$$

with

$$\mu_* = m(\boldsymbol{x}_*) + \boldsymbol{k}_*^\top (\boldsymbol{K} + \sigma_n^2 \boldsymbol{I})^{-1} (\boldsymbol{y} - \boldsymbol{m}), \tag{7.39}$$

$$\sigma_*^2 = k_{**} - \boldsymbol{k}_*^\top (\boldsymbol{K} + \sigma_n^2 \boldsymbol{I})^{-1} \boldsymbol{k}_*, \tag{7.40}$$

where $\boldsymbol{m} = (m(\boldsymbol{x}_1), \ldots, m(\boldsymbol{x}_N))^\top$. The posterior mean Equation (7.39) interpolates smoothly between the training points, and the posterior variance Equation (7.40) shrinks to σ_n^2 at training inputs and grows back towards the prior variance k_{**} far from them. This is illustrated in Figure 7.6: the shaded band narrows wherever data are available and widens in unexplored regions.

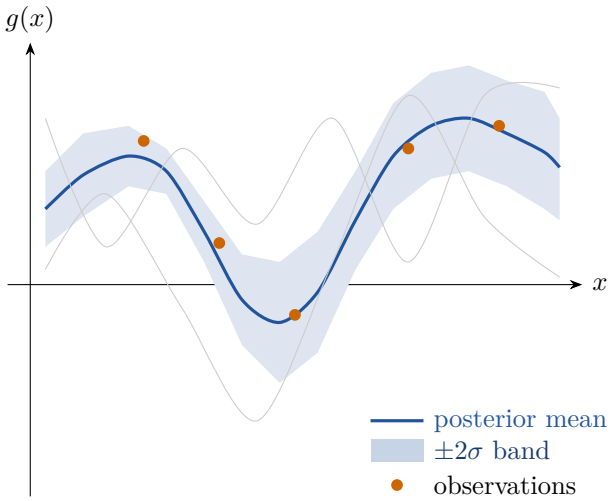


Figure 7.6: *Gaussian process regression. The shaded region is the posterior $\pm 2\sigma$ predictive band; it contracts near the five training points (orange) and expands in unobserved regions. Two faint curves show prior samples before conditioning on data.*

The kernel hyperparameters $(\ell, \sigma_f, \sigma_n)$ are themselves unknown and are typically fitted by maximising the marginal likelihood (also called the evidence):

$$\ln f(\boldsymbol{y} \mid \boldsymbol{X}) = -\tfrac{1}{2}(\boldsymbol{y}-\boldsymbol{m})^\top (\boldsymbol{K}+\sigma_n^2\boldsymbol{I})^{-1}(\boldsymbol{y}-\boldsymbol{m}) - \tfrac{1}{2}\ln\det(\boldsymbol{K}+\sigma_n^2\boldsymbol{I}) - \tfrac{N}{2}\ln 2\pi, \tag{7.41}$$

which rewards good fit through the first term while penalising unnecessary complexity through the log-determinant term, an automatic Occam's razor. The dominant computational cost is the matrix inversion in Equations (7.39) to (7.41), which scales as $O(N^3)$, making exact GP inference infeasible for more than a few thousand training points. Sparse and inducing-point approximations reduce this to $O(NM^2)$ with $M \ll N$ inducing points. In astrophysics GPs are used as emulators for expensive cosmological simulations: trained on a few hundred simulator runs, a GP can predict the output at new parameter values almost instantaneously, together with a calibrated uncertainty that grows in unexplored regions of parameter space.

7.8 Unsupervised learning: dimensionality reduction

High-dimensional data often contain far less independent variation than their nominal dimension suggests. A handwritten digit is a point in $\mathbb{R}^{784}$ (a 28×28 pixel image), yet all images of the digit seven differ from one another only in a handful of ways: stroke thickness, slant, and overall size. The data therefore cluster near a low-dimensional surface within the ambient 784-dimensional space, and a projection onto that surface loses almost no information while dramatically reducing storage

and computation. The same situation arises in many astrophysical settings: galaxy spectra with thousands of wavelength channels, weak-lensing power-spectrum vectors, or cosmological simulation snapshots at many redshifts all exhibit strong correlations that concentrate the relevant variation into a much smaller effective dimension. Dimensionality reduction seeks a low-dimensional representation $\boldsymbol{z} \in \mathbb{R}^k$ (with $k \ll d$) that captures most of the structure in the original d-dimensional data $\boldsymbol{x}$.

7.8.1 Principal component analysis

The linear solution is principal component analysis (PCA), which finds the k orthogonal directions in $\mathbb{R}^d$ that maximise the variance of the projected data. Centering the N training points as $\tilde{\boldsymbol{x}}_i = \boldsymbol{x}_i - \bar{\boldsymbol{x}}$, the sample covariance matrix is

$$\hat{\boldsymbol{\Sigma}} = \frac{1}{N-1} \sum_{i=1}^{N} \tilde{\boldsymbol{x}}_i \tilde{\boldsymbol{x}}_i^{\top}. \tag{7.42}$$

Its eigendecomposition $\hat{\boldsymbol{\Sigma}} = \boldsymbol{U} \boldsymbol{\Lambda} \boldsymbol{U}^{\top}$, with $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1 \geq \cdots \geq \lambda_d \geq 0)$, yields the principal components: the columns $\boldsymbol{u}_j$ of $\boldsymbol{U}$ are the eigenvectors, i.e. the directions of maximum variance. The score of data point $\boldsymbol{x}$ projected onto the first k components is

$$\boldsymbol{z} = \boldsymbol{U}_k^{\top} \tilde{\boldsymbol{x}}, \qquad \boldsymbol{U}_k = (\boldsymbol{u}_1, \ldots, \boldsymbol{u}_k), \tag{7.43}$$

and reconstruction uses $\hat{\boldsymbol{x}} = \bar{\boldsymbol{x}} + \boldsymbol{U}_k \boldsymbol{z}$. The mean squared reconstruction error is $\sum_{j=k+1}^{d} \lambda_j$, minimised over all rank-k linear projections by the choice Equation (7.43). The fraction of variance retained is $\sum_{j=1}^{k} \lambda_j / \sum_{j=1}^{d} \lambda_j$, which indicates how many components are sufficient.

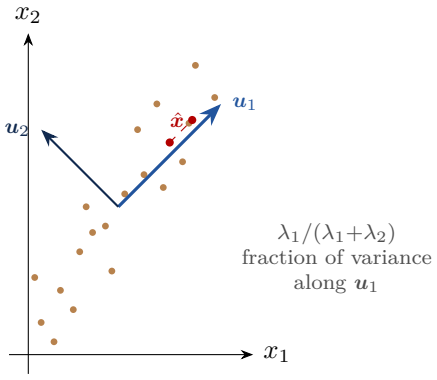


Figure 7.7: *PCA on two-dimensional data. The first principal component $\boldsymbol{u}_1$ (solid arrow) points in the direction of maximum variance; $\boldsymbol{u}_2$ is orthogonal. A point (red dot) is projected onto $\boldsymbol{u}_1$ to give the rank-1 reconstruction $\hat{\boldsymbol{x}}$ (open dot); the projection error is the dashed segment.*

PCA has an elegant probabilistic interpretation. A latent variable model $\boldsymbol{x} = \boldsymbol{W}\boldsymbol{z} + \boldsymbol{\mu} + \boldsymbol{\varepsilon}$ with $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ and isotropic noise $\boldsymbol{\varepsilon} \sim \mathcal{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I})$ is called probabilistic PCA. Marginalising out $\boldsymbol{z}$ gives $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{W}\boldsymbol{W}^{\top} + \sigma^2 \boldsymbol{I})$, and maximising the marginal likelihood over $\boldsymbol{W}$ recovers exactly the standard PCA solution in the limit $\sigma^2 \to 0$. This probabilistic form enables principled treatment of missing data and connects PCA to the family of variational autoencoders and other latent variable models widely used in deep learning.

In cosmology, PCA is used to compress galaxy spectra, reduce the dimensionality of weak-lensing power-spectrum vectors, and build data-compression schemes for efficient likelihood evaluation. Its main limitation is linearity: if the data lie on

a curved low-dimensional manifold, the variance captured per component falls off slowly and many components are needed.

7.8.2 Autoencoders

A natural nonlinear generalisation uses the neural network architecture of Section 7.5. An autoencoder consists of two networks composed in series: an encoder $e_{\boldsymbol{\phi}} : \mathbb{R}^d \to \mathbb{R}^k$ that maps each observation to a low-dimensional code, and a decoder $d_{\boldsymbol{\psi}} : \mathbb{R}^k \to \mathbb{R}^d$ that reconstructs the original from the code. Both are trained jointly to minimise the reconstruction loss

$$\hat{R}(\boldsymbol{\phi}, \boldsymbol{\psi}) = \frac{1}{N} \sum_{i=1}^{N} \left\| \boldsymbol{x}_i - d_{\boldsymbol{\psi}}(e_{\boldsymbol{\phi}}(\boldsymbol{x}_i)) \right\|^2. \tag{7.44}$$

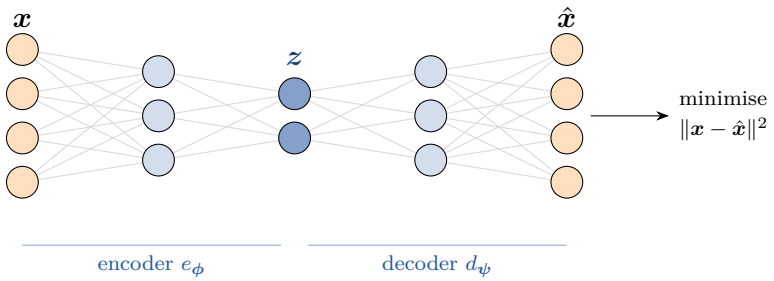


Figure 7.8: *Autoencoder architecture. The encoder $e_{\boldsymbol{\phi}}$ progressively compresses the input $\boldsymbol{x} \in \mathbb{R}^d$ into a low-dimensional bottleneck code $\boldsymbol{z} \in \mathbb{R}^k$; the decoder $d_{\boldsymbol{\psi}}$ reconstructs $\hat{\boldsymbol{x}} \in \mathbb{R}^d$ from $\boldsymbol{z}$. Training minimises the mean squared reconstruction error Equation (7.44).*

The bottleneck layer $\boldsymbol{z} = e_{\boldsymbol{\phi}}(\boldsymbol{x}) \in \mathbb{R}^k$ is forced to contain all information needed to reconstruct $\boldsymbol{x}$, so the network learns a compressed representation of the data manifold. With linear activations and k hidden units, the autoencoder recovers exactly the first k principal components; nonlinear activations allow it to capture curved structure that PCA misses.

A probabilistic extension is the variational autoencoder (VAE), which imposes a prior $\boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ on the code and replaces the deterministic encoder with a distribution $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{\phi}}(\boldsymbol{x}), \mathrm{diag}(\boldsymbol{\sigma}^2_{\boldsymbol{\phi}}(\boldsymbol{x})))$. The training objective becomes the evidence lower bound (ELBO),

$$\mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\psi}) = \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}[\ln d_{\boldsymbol{\psi}}(\boldsymbol{x}|\boldsymbol{z})] - D_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) \,\|\, \mathcal{N}(\mathbf{0}, \boldsymbol{I})), \tag{7.45}$$

which balances reconstruction quality against the Kullback-Leibler (KL) divergence $D_{\mathrm{KL}}(q\|p) = \mathbb{E}_q[\ln q - \ln p]$ between the approximate posterior and the prior. The KL term regularises the code space and prevents the encoder from collapsing to a delta function, while the Gaussian prior ensures the latent space is smooth and can be sampled from at generation time. VAEs are widely used in astrophysics for anomaly detection in survey data and for learning compact representations of simulation outputs.

7.9 Normalising flows and density estimation

A central task in cosmological inference is density estimation: given samples from an unknown distribution, learn to evaluate its density at arbitrary points. For a Gaussian posterior this is trivial, but for the complex, multimodal posteriors common

in high-dimensional cosmology it requires flexible learned models. Normalising flows are the most tractable such models: they parameterise a complex distribution as the pushforward of a simple base distribution through an invertible neural network.

Formally, let $f_{\boldsymbol{z}}(\boldsymbol{z})$ be a base distribution (typically $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$) on $\mathbb{R}^d$. A normalising flow is a bijective, differentiable map $T_{\boldsymbol{\theta}} : \mathbb{R}^d \to \mathbb{R}^d$ with differentiable inverse. Setting $\boldsymbol{x} = T_{\boldsymbol{\theta}}(\boldsymbol{z})$ and applying the change-of-variables formula gives the density of $\boldsymbol{x}$,

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = f_{\boldsymbol{z}}\left(T_{\boldsymbol{\theta}}^{-1}(\boldsymbol{x})\right) \left| \det \frac{\partial T_{\boldsymbol{\theta}}^{-1}}{\partial \boldsymbol{x}} \right|. \tag{7.46}$$

Taking logarithms gives

$$\ln f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \ln f_{\boldsymbol{z}}\left(T_{\boldsymbol{\theta}}^{-1}(\boldsymbol{x})\right) + \ln \left| \det \frac{\partial T_{\boldsymbol{\theta}}^{-1}}{\partial \boldsymbol{x}} \right|, \tag{7.47}$$

which is tractable whenever the log-determinant of the Jacobian can be computed cheaply. This is the key design constraint on flow architectures: the transformation $T_{\boldsymbol{\theta}}$ must be expressive, invertible, and have a cheap Jacobian determinant. Flows are trained by maximising the log-likelihood over the dataset $\{\boldsymbol{x}_i\}_{i=1}^N$,

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \sum_{i=1}^{N} \ln f_{\boldsymbol{\theta}}(\boldsymbol{x}_i), \tag{7.48}$$

which is exactly the maximum-likelihood estimation of Section 4 applied to an implicit model whose density is evaluated via Equation (7.47). Once trained, sampling is achieved by drawing $\boldsymbol{z} \sim f_{\boldsymbol{z}}$ and evaluating $\boldsymbol{x} = T_{\boldsymbol{\theta}}(\boldsymbol{z})$.

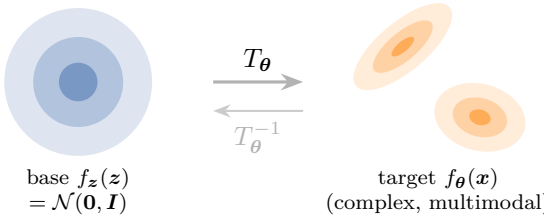


Figure 7.9: *A normalising flow maps a simple base distribution $f_{\boldsymbol{z}}$ (Gaussian, left) to a complex target distribution $f_{\boldsymbol{\theta}}(\boldsymbol{x})$ (right) via an invertible transformation $T_{\boldsymbol{\theta}}$. The inverse $T_{\boldsymbol{\theta}}^{-1}$ maps data back to the base, enabling exact density evaluation via the change-of-variables formula Equation (7.47).*

7.9.1 Architecture

A practical flow is built by composing L invertible layers $T_{\boldsymbol{\theta}} = T_L \circ \cdots \circ T_1$. Each layer must have a tractable Jacobian determinant, and the composition of invertible maps is invertible. A particularly elegant design is: split $\boldsymbol{x}$ into two halves $(\boldsymbol{x}_A, \boldsymbol{x}_B)$ and transform only $\boldsymbol{x}_B$ while conditioning on $\boldsymbol{x}_A$,

$$\boldsymbol{x}'_A = \boldsymbol{x}_A, \qquad \boldsymbol{x}'_B = \boldsymbol{x}_B \odot \exp(s_{\boldsymbol{\theta}}(\boldsymbol{x}_A)) + t_{\boldsymbol{\theta}}(\boldsymbol{x}_A), \tag{7.49}$$

where $s_{\boldsymbol{\theta}}$ and $t_{\boldsymbol{\theta}}$ are arbitrary neural networks (the scale and translation networks). The Jacobian of Equation (7.49) is triangular, so its determinant is simply $\exp(\sum_j [s_{\boldsymbol{\theta}}(\boldsymbol{x}_A)]_j)$, computed in one forward pass. Alternating which half is transformed at each layer ensures all dimensions are eventually mixed. Masked autoregressive flows achieve a similar effect by imposing an autoregressive structure $x'_j = x_j \exp(s_j(x_{<j})) + t_j(x_{<j})$, which also yields a triangular Jacobian.

7.9.2 Conditional flows

A conditional normalising flow extends Equation (7.46) to model a conditional density $f_{\boldsymbol{\theta}}(\boldsymbol{x} \mid \boldsymbol{c})$ by making the transformation depend on a conditioning variable $\boldsymbol{c}$: replace $s_{\boldsymbol{\theta}}(\boldsymbol{x}_A)$ with $s_{\boldsymbol{\theta}}(\boldsymbol{x}_A; \boldsymbol{c})$ in Equation (7.49). Trained on pairs $(\boldsymbol{x}, \boldsymbol{c})$, the flow learns to produce the distribution of $\boldsymbol{x}$ for any given $\boldsymbol{c}$. Setting $\boldsymbol{c} = \boldsymbol{d}$ (observed data) and $\boldsymbol{x} = \boldsymbol{\theta}$ (model parameters), the conditional flow directly learns the posterior $p(\boldsymbol{\theta} \mid \boldsymbol{d})$ from simulated $(\boldsymbol{\theta}, \boldsymbol{d})$ pairs, without ever evaluating the likelihood. This is the neural posterior estimation variant of simulation-based inference, which we develop in Section 8.

Mathematical background: flow log-likelihood as MLE plus regularisation

By the chain rule applied to the composition $T_{\boldsymbol{\theta}} = T_L \circ \cdots \circ T_1$, the log-likelihood Equation (7.47) accumulates contributions from each layer,

$$\ln f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \ln f_{\boldsymbol{z}}(\boldsymbol{z}_0) + \sum_{l=1}^{L} \ln \left| \det \frac{\partial T_l^{-1}}{\partial \boldsymbol{z}_l} \right|,$$

where $\boldsymbol{z}_0 = T_{\boldsymbol{\theta}}^{-1}(\boldsymbol{x})$ and $\boldsymbol{z}_l$ is the intermediate representation after layer l. Maximising this over the training set is maximum likelihood (Equation (7.48)). The log-determinant terms automatically penalise transformations that compress or expand volume too aggressively, playing a role analogous to regularisation: they prevent the flow from collapsing the base density onto a low-dimensional set to fit a few points.

7.10 Summary and practical remarks

Machine learning extends the estimation of the previous sections from a few physical parameters to flexible function families with very many. empirical risk minimisation generalises maximum likelihood, regularisation is a prior, and the central difficulty, generalisation, is the tension between bias and variance. Gaussian processes provide a fully Bayesian alternative that yields calibrated predictive uncertainties and is particularly suited to emulation tasks where the training set is small and expensive to generate. Normalising flows enable tractable density estimation for complex distributions and form the engine of simulation-based inference. We close with practical advice.

1. **Always separate training, validation and test data**: tune everything on the validation set and look at the test set only once. A model judged on data it was trained or tuned on will be rather useless.

2. **Start simple**: fit a linear or logistic model before reaching for a network. It is fast, often competitive, gives an interpretable baseline, and its failure tells you whether the extra capacity is actually needed.

3. **Diagnose with the train–test gap**: a large training error means underfitting, so raise the capacity. A small training error alongside a large test error indicates overfitting; regularise or gather more data.

4. **Standardise the inputs**: centring and scaling the features condition the loss surface and make both the fit and the optimisation better behaved, and many methods assume it implicitly.

5. **Regularise and tune λ honestly**: choose the penalty strength on the validation set, not by eye, and remember that it encodes a prior.

6. **Use a GP when the training set is small**: for emulation tasks with $N \lesssim 10^3$ points, a GP with a well-chosen kernel often outperforms neural networks while providing honest uncertainty estimates that grow in unexplored regions of parameter space.

7. **Do not extrapolate**: a learned function is an interpolation of the training distribution, and outside the region the training data cover, it is unconstrained. This matters acutely for emulators now common in cosmology, which must never be trusted beyond the parameter ranges on which they were trained. The GP posterior variance provides a built-in warning sign; neural networks do not.

These tools also point beyond classical inference. When the likelihood itself is intractable but we can simulate data, a network can learn to stand in for the likelihood or to compress the data into informative summaries; this is the subject of simulation-based inference in Section 8, where the machinery of this section and the inference of the earlier ones finally merge.

8 Simulation-based inference

Every inference method we have developed so far shares one silent assumption: that we can evaluate the likelihood $\mathcal{L}(\boldsymbol{\theta}) = p(\boldsymbol{d} \mid \boldsymbol{\theta})$ as a number, for a given parameter point and a given dataset. The maximum likelihood estimator of Section 4 maximises it, the Fisher matrix is built from its second derivatives, the Bayesian posterior of Section 5 multiplies it by a prior, and even the Monte Carlo samplers of Section 6, which never need the evidence, still require the likelihood ratio $\mathcal{L}(\boldsymbol{\theta}')/\mathcal{L}(\boldsymbol{\theta})$ at every step. Take away the ability to evaluate $\mathcal{L}$, and essentially the entire toolbox of the preceding chapters stops working.

In many modern astrophysical and cosmological problems, this is exactly what happens. We have a simulator: a piece of code that, given parameters $\boldsymbol{\theta}$, generates a synthetic dataset $\boldsymbol{d}$ with all the realism we can muster, but we cannot write down, let alone evaluate, the probability density of that dataset. The model is specified procedurally, by the steps of a computation, rather than analytically, by a formula for $p(\boldsymbol{d} \mid \boldsymbol{\theta})$. To see how quickly tractability is lost, consider the following simulator: we create a two-dimensional grid with L^2 sites. Each site is then populated with probability θ (i.e. it assumes value 1 with that probability). We then find the size largest connected cluster on this grid and report it:

```
from scipy.ndimage import label
import numpy as np

def simulator(theta, L=64):
    grid = np.random.rand(L, L) < theta
    lab, k = label(grid)
    return np.bincount(lab.ravel())[1:].max() if k else 0
```

What is the likelihood of this, well it is technically easy because it is simply a Poisson process over all possible configurations. However, there are 2^{L^2} configurations of the grid which all need to be checked. On the other hand, the simulator itself runs almost instantly.

Inference in this regime is the subject of simulation-based inference (SBI), also called likelihood-free inference or implicit-likelihood inference. The term "likelihood-free" is a slight misnomer: a likelihood always exists in principle, sometimes it is, however, simply inaccessible. The whole fuss about it is to do Bayesian inference while touching the likelihood only through samples.

8.1 The likelihood-free setting

Recall the generative picture of Section 5.7.1: the data are produced by descending a chain from parameters to a latent signal to observations. A simulator makes this literal. Internally it draws a long vector of random numbers $\boldsymbol{z}$, and then computes the data by a deterministic map,

$$\boldsymbol{d} = g(\boldsymbol{\theta}, \boldsymbol{z}), \qquad \boldsymbol{z} \sim p(\boldsymbol{z}). \tag{8.1}$$

Running the simulator once is therefore a single draw $\boldsymbol{d} \sim p(\boldsymbol{d} \mid \boldsymbol{\theta})$. The likelihood is what we obtain by marginalising over everything internal,

$$\mathcal{L}(\boldsymbol{\theta}) = p(\boldsymbol{d} \mid \boldsymbol{\theta}) = \int p(\boldsymbol{d}, \boldsymbol{z} \mid \boldsymbol{\theta})\, \mathrm{d}\boldsymbol{z} = \int \delta^{\mathrm{D}}(\boldsymbol{d} - g(\boldsymbol{\theta}, \boldsymbol{z}))\, p(\boldsymbol{z})\, \mathrm{d}\boldsymbol{z}\,, \tag{8.2}$$

which is exactly the latent marginalisation of Equation (5.35), written with the generative map made explicit. The delta distribution enforces that only those

internal draws $\boldsymbol{z}$ that happen to produce exactly the observed $\boldsymbol{d}$ contribute. For a realistic forward model $\boldsymbol{z}$ has millions of components and g is a full simulation pipeline, so this integral is hopeless: there is no closed form, and naive Monte Carlo fails because the delta distribution is satisfied with probability zero. We can, however, sample $p(\boldsymbol{d} \mid \boldsymbol{\theta})$ effortlessly by forward simulation, yet we cannot evaluate it at a point.

A model has an implicit (or intractable) likelihood if it is defined by a stochastic simulator $\boldsymbol{d} = g(\boldsymbol{\theta}, \boldsymbol{z})$, with $\boldsymbol{z} \sim p(\boldsymbol{z})$ the internal randomness, such that samples $\boldsymbol{d} \sim p(\boldsymbol{d} \mid \boldsymbol{\theta})$ are cheap to obtain but the density $p(\boldsymbol{d} \mid \boldsymbol{\theta})$ cannot be evaluated in closed form.

Mathematical background: when is a likelihood tractable?

It is instructive to see where tractability is lost. If the map Equation (8.1) is invertible in $\boldsymbol{z}$ at fixed $\boldsymbol{\theta}$, with $\boldsymbol{z} = g^{-1}(\boldsymbol{\theta}, \boldsymbol{d})$, the change-of-variables formula (the same one behind normalising flows, Equation 7.46) gives a closed likelihood

$$p(\boldsymbol{d} \mid \boldsymbol{\theta}) = p(g^{-1}(\boldsymbol{\theta}, \boldsymbol{d})) \left| \det \frac{\partial g^{-1}}{\partial \boldsymbol{d}} \right| .$$

This is exactly the tractable case: a single, invertible noise source with a computable Jacobian. Intractability arises the moment (i) $\boldsymbol{z}$ is higher-dimensional than $\boldsymbol{d}$, so the inverse is a high-dimensional integral rather than a point; (ii) g involves discrete or non-invertible steps (thresholding, selection, binning); or (iii) g is a black box whose Jacobian is unavailable. Realistic simulators violate all three at once. Normalising flows can be read as learning a tractable, invertible surrogate for an intractable g; this is the bridge to the neural methods of Section 8.3.

Some concrete examples from the field:

- **Cosmological large-scale structure**: A forward model takes cosmological parameters $\boldsymbol{\theta}$, runs an N-body or hydrodynamic simulation, applies a galaxy–halo connection, survey masks, redshift errors, and instrumental noise, and outputs a mock galaxy catalogue. Each stage is stochastic and the likelihood of the final catalogue is undefined analytically.
- **Gravitational-wave populations**: Given hyperparameters describing the black-hole mass, spin, and redshift distribution, a simulator draws a population, applies detector selection effects, and produces a catalogue of detections. The selection integral makes the population likelihood a high-dimensional intractable object.

In all cases the physics is encoded in code, not in a formula, and the central object of Sections 4 and 5, the likelihood, has quietly become inaccessible.

8.2 Approximate Bayesian computation

The oldest and most intuitive likelihood-free method is Approximate Bayesian Computation (ABC). Its logic is a direct translation of the definition of the posterior into a Monte Carlo procedure that needs only forward simulations. In its simplest, rejection form:

1. Draw $\boldsymbol{\theta}^\star \sim \pi(\boldsymbol{\theta})$ from the prior.
2. Simulate a mock dataset $\boldsymbol{d}^\star \sim p(\boldsymbol{d} \mid \boldsymbol{\theta}^\star)$ by running the simulator once.

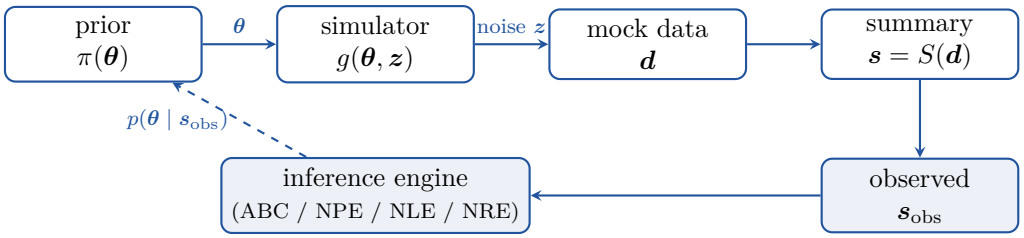


Figure 8.1: *The simulation-based inference loop. Parameters are drawn from the prior, pushed through a stochastic simulator to produce mock data, and compressed to summaries. Many such $(\boldsymbol{\theta}, \boldsymbol{s})$ pairs train an inference engine that, confronted with the observed summary $\boldsymbol{s}_{\mathrm{obs}}$, returns the posterior. The likelihood is never evaluated, only sampled.*

3. Accept $\boldsymbol{\theta}^\star$ if the mock data lie within a tolerance of the observed data, $\rho(\boldsymbol{d}^\star, \boldsymbol{d}_{\mathrm{obs}}) \leq \varepsilon$; otherwise reject.

Here ρ is a distance on data space and $\varepsilon \geq 0$ a tolerance. The retained parameters are distributed according to

$$p_\varepsilon(\boldsymbol{\theta} \mid \boldsymbol{d}_{\mathrm{obs}}) \;\propto\; \pi(\boldsymbol{\theta}) \int \mathbb{I}[\rho(\boldsymbol{d}, \boldsymbol{d}_{\mathrm{obs}}) \leq \varepsilon]\, p(\boldsymbol{d} \mid \boldsymbol{\theta})\, \mathrm{d}\boldsymbol{d}\,, \tag{8.3}$$

because a proposal survives only if both the prior draw and the acceptance test are passed. As $\varepsilon \to 0$ the indicator collapses onto $\boldsymbol{d} = \boldsymbol{d}_{\mathrm{obs}}$ and the integral tends to $p(\boldsymbol{d}_{\mathrm{obs}} \mid \boldsymbol{\theta})$, so Equation (8.3) becomes the exact posterior of Section 5. The procedure is nothing but the rejection sampling of Section 6, with the intractable likelihood replaced by the operational test "did the simulation land near the data?". The tolerance interpolates between the exact posterior ($\varepsilon \to 0$, zero acceptance) and the prior ($\varepsilon \to \infty$, everything accepted).

A sharp indicator wastes every simulation that just misses. A standard improvement replaces it by a smooth kernel K_ε of bandwidth ε,

$$p_\varepsilon(\boldsymbol{\theta} \mid \boldsymbol{d}_{\mathrm{obs}}) \;\propto\; \pi(\boldsymbol{\theta}) \int K_\varepsilon\big(\rho(\boldsymbol{d}, \boldsymbol{d}_{\mathrm{obs}})\big)\, p(\boldsymbol{d} \mid \boldsymbol{\theta})\, \mathrm{d}\boldsymbol{d}\,, \tag{8.4}$$

so that near-misses are down-weighted rather than discarded. This shows what ABC s: a kernel density estimate of the smeared likelihood, evaluated at $\boldsymbol{d}_{\mathrm{obs}}$.

The difficulty is the interplay between ε and the dimension of the object inside ρ. A small ε gives an accurate posterior but an exponentially small acceptance rate. A large ε accepts often but blurs the posterior. In a high-dimensional data space almost every simulation lands far from $\boldsymbol{d}_{\mathrm{obs}}$, a direct manifestation of the curse of dimensionality of Section 6.1, so ε must be relaxed to the point of uselessness. The universal remedy is to compress the data to a handful of summary statistics $\boldsymbol{s} = S(\boldsymbol{d})$ before comparing,

$$\rho(\boldsymbol{d}^\star, \boldsymbol{d}_{\mathrm{obs}}) \;\longrightarrow\; \rho\big(S(\boldsymbol{d}^\star),\, S(\boldsymbol{d}_{\mathrm{obs}})\big)\,, \tag{8.5}$$

so that closeness is judged in a low-dimensional space (for example moments). A sensible ρ also whitens by the covariance, $\rho(\boldsymbol{s}, \boldsymbol{s}')^2 = (\boldsymbol{s} - \boldsymbol{s}')^\top \boldsymbol{C}^{-1} (\boldsymbol{s} - \boldsymbol{s}')$, the Mahalanobis distance already familiar from the χ^2 of Equation (4.35), so that each summary is compared on the scale of its own scatter. Compression introduces a second approximation: unless the summaries are sufficient, that is, they retain all the information about $\boldsymbol{\theta}$ (cf. the Fisher discussion of Section 4.5), the ABC posterior is broader than the truth. ABC therefore trades one intractability for two tuning problems, the summaries and the tolerance.

Mathematical background: why ABC scales badly

Suppose the summaries are k-dimensional and, for a tolerance ε, we accept when each component matches to within ε. The acceptance probability is the fraction of simulations landing in that region,

$$P_{\text{acc}} = \int_{\|\boldsymbol{s}-\boldsymbol{s}_{\text{obs}}\|\le\varepsilon} p(\boldsymbol{s})\,\mathrm{d}\boldsymbol{s} \;\approx\; p(\boldsymbol{s}_{\text{obs}})\,V_k\,\varepsilon^k,$$

with V_k the volume of the unit k-ball, provided p is roughly constant across the region. To collect a fixed number M of accepted samples therefore costs $N \sim M/(p(\boldsymbol{s}_{\text{obs}})V_k\varepsilon^k) \propto \varepsilon^{-k}$ simulator calls. The cost grows exponentially in the number of summaries and polynomially in the inverse tolerance: even for modest $k \sim 10$ and a tolerance tight enough for a trustworthy posterior, this is millions to billions of simulations. This exponential wall, and the fact that each rejected simulation is simply thrown away, is precisely why the neural methods below, which learn a function from all simulations, have displaced ABC for all but the lowest-dimensional problems.

Before leaving ABC, note two standard improvements that mirror Section 6. MCMC-ABC runs a Markov chain in $\boldsymbol{\theta}$, proposing moves and accepting them with a rule that embeds the ABC test, so that the sampler dwells in high-acceptance regions instead of blindly drawing from the prior. Sequential / population Monte Carlo ABC shrinks ε over a series of rounds, using the accepted particles of one round as the proposal for the next, an idea we meet again, in learned form, in Section 8.5. Both mitigate, but do not remove, the exponential scaling with dimensionality.

8.3 Neural simulation-based inference

The modern approach replaces the accept/reject filter of ABC with a learned density. We generate a training set of simulations $\{(\boldsymbol{\theta}_i, \boldsymbol{d}_i)\}_{i=1}^N$ by repeatedly drawing $\boldsymbol{\theta}_i \sim \pi$ and simulating $\boldsymbol{d}_i \sim p(\boldsymbol{d} \mid \boldsymbol{\theta}_i)$, so that each pair is a draw from the joint $p(\boldsymbol{\theta}, \boldsymbol{d}) = \pi(\boldsymbol{\theta})p(\boldsymbol{d} \mid \boldsymbol{\theta})$. We then fit a flexible neural model, typically a conditional normalising flow from Section 7.9.2, to one of the distributions appearing in Bayes' theorem. Two features distinguish this from ABC. First, every simulation contributes to the fit, none is thrown away, so the simulation budget is used far more efficiently. Second, the result is amortised: once the network is trained it returns the answer for any observation by a single forward pass, with no further simulation. There are three canonical targets, distinguished by what is learned. We compare them in Figure 8.2 and table 3.

8.3.1 Neural posterior estimation (NPE)

Neural posterior estimation learns the posterior directly. A conditional flow $q_\phi(\boldsymbol{\theta} \mid \boldsymbol{d})$ with weights ϕ is trained by maximising the total log-density of the true parameters given their simulated data,

$$\hat{\phi} = \arg\max_{\phi} \sum_{i=1}^{N} \ln q_\phi(\boldsymbol{\theta}_i \mid \boldsymbol{d}_i)\,. \tag{8.6}$$

This is exactly the flow training objective Equation (7.48) with the conditioning variable set to the data, as anticipated at the end of Section 7.9.2. At inference time one simply evaluates or samples $q_{\hat{\phi}}(\boldsymbol{\theta} \mid \boldsymbol{d}_{\text{obs}})$. There is no sampling loop and no tolerance. Because the flow supplies both a density and a sampler, the point

estimates and credible regions of Section 5.3 follow immediately, and a corner plot is produced by drawing a few thousand samples in milliseconds.

8.3.2 Neural likelihood estimation (NLE)

Neural likelihood estimation instead learns the likelihood as a conditional flow $q_\phi(\boldsymbol{d} \mid \boldsymbol{\theta})$, trained with the roles of the two variables in Equation (8.6) swapped,

$$\hat{\phi} = \arg\max_{\phi} \sum_{i=1}^{N} \ln q_\phi(\boldsymbol{d}_i \mid \boldsymbol{\theta}_i)\,. \tag{8.7}$$

By the same argument as above this converges to the true likelihood $p(\boldsymbol{d} \mid \boldsymbol{\theta})$. We have thus manufactured an evaluable surrogate for the object that was intractable in Equation (8.2), and can now feed it straight into the standard machinery: form $q_{\hat{\phi}}(\boldsymbol{d}_{\mathrm{obs}} \mid \boldsymbol{\theta})\,\pi(\boldsymbol{\theta})$ and sample it with the MCMC methods of Section 6. The advantage over NPE is flexibility in the prior: because only the likelihood was learned, one may change $\pi(\boldsymbol{\theta})$ after training, or combine the learned likelihood with an independent one, without re-simulating. The price is that a sampling step, with its burn-in and convergence diagnostics from Section 6.6, is reintroduced, and that a good density model for $\boldsymbol{d}$ may be harder to learn than one for $\boldsymbol{\theta}$ when the data are higher-dimensional than the parameters.

8.3.3 Neural ratio estimation (NRE)

Neural ratio estimation sidesteps density estimation altogether and learns the likelihood-to-evidence ratio

$$r(\boldsymbol{d}, \boldsymbol{\theta}) = \frac{p(\boldsymbol{d} \mid \boldsymbol{\theta})}{p(\boldsymbol{d})} = \frac{p(\boldsymbol{\theta} \mid \boldsymbol{d})}{\pi(\boldsymbol{\theta})} = \frac{p(\boldsymbol{\theta}, \boldsymbol{d})}{\pi(\boldsymbol{\theta})\,p(\boldsymbol{d})}\,, \tag{8.8}$$

where the three forms are equal by Bayes' theorem. This ratio is all one needs to do inference: the posterior is $p(\boldsymbol{\theta} \mid \boldsymbol{d}) = r(\boldsymbol{d}, \boldsymbol{\theta})\,\pi(\boldsymbol{\theta})$, and the Metropolis–Hastings acceptance probability of Equation (6.28) depends only on ratios of r, so the unknown evidence cancels just as it did there. The elegant part is how r is obtained: by training a classifier. Label pairs drawn jointly from the simulator, $(\boldsymbol{\theta}, \boldsymbol{d}) \sim p(\boldsymbol{\theta}, \boldsymbol{d})$, as class $y = 1$ ("dependent"), and pairs formed by shuffling parameters against unrelated data, $(\boldsymbol{\theta}, \boldsymbol{d}) \sim \pi(\boldsymbol{\theta})\,p(\boldsymbol{d})$, as class $y = 0$ ("independent"). A network trained to tell the two apart, using the Bernoulli/cross-entropy loss of Section 7.2.2, learns a function of exactly the ratio Equation (8.8). SBI thereby reduces to supervised binary classification, one of the best-understood and best-tooled tasks in machine learning.

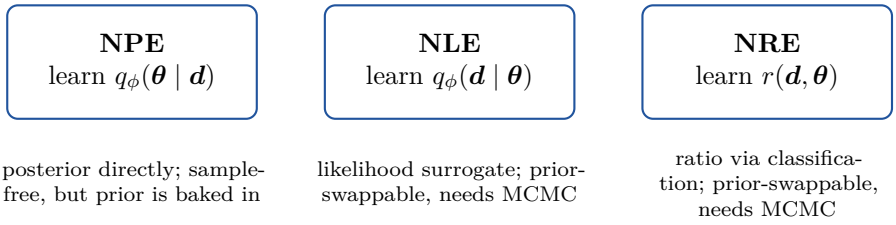


Figure 8.2: *The three canonical neural SBI targets. All are trained on the same set of simulated $(\boldsymbol{\theta}, \boldsymbol{d})$ pairs and differ only in which quantity of Bayes' theorem the network represents, and hence in what must be done at inference time.*

Table 3: The neural SBI methods at a glance. They are all amortised, which means that a single trained network serves any future observation without re-simulation. Sequential variants (Section 8.5) trade this away for simulation efficiency.

	NPE	**NLE**	**NRE**
Learns	$p(\boldsymbol{\theta} \mid \boldsymbol{d})$	$p(\boldsymbol{d} \mid \boldsymbol{\theta})$	$p(\boldsymbol{d} \mid \boldsymbol{\theta})/p(\boldsymbol{d})$
Model type	cond. flow	cond. flow	classifier
Inference step	direct eval / sample	MCMC	MCMC
Change prior later?	no (retrain)	yes	yes
Density needed?	of $\boldsymbol{\theta}$	of $\boldsymbol{d}$	none

8.4 Learned summaries and data compression

Raw simulated data $\boldsymbol{d}$ are usually far too high-dimensional to feed to a flow or classifier directly, so, exactly as in ABC, a compression step $\boldsymbol{s} = S(\boldsymbol{d})$ is required, only now we can learn it rather than guess it. The goal is a compression that is asymptotically sufficient: one that loses no information about $\boldsymbol{\theta}$. The natural yardstick is the Fisher information of Section 4.5: a good S preserves $\boldsymbol{F}$, so that the SBI posterior is no broader than a full analysis would allow.

A classical and still widely used construction compresses the data down to exactly the dimension of the parameter vector using the score. Around a fiducial model $\boldsymbol{\theta}_{\mathrm{fid}}$, define

$$\boldsymbol{s} = \boldsymbol{\theta}_{\mathrm{fid}} + \boldsymbol{F}^{-1}\, \nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta})\big|_{\boldsymbol{\theta}_{\mathrm{fid}}}\,, \tag{8.9}$$

which turns the whole dataset into a d-dimensional "pseudo-estimate" of $\boldsymbol{\theta}$. By the Fisher argument of Section 4.5 these summaries are asymptotically sufficient and their covariance saturates the Cramér–Rao bound of Section 4.6: the score is exactly the quantity whose variance is $\boldsymbol{F}$, so no Fisher information is discarded. Modern variants replace the hand-built Equation (8.9) by a neural compressor trained jointly with the inference network, either to maximise the Fisher information of its outputs or, more directly, to minimise the same posterior loss Equation (8.6). The decisive advantage over ABC is that the quality of the compression is checkable: if the SBI error bars agree with the Fisher forecast of Section 4.5, essentially no information was lost. If they are much wider, the summaries are lossy and should be enriched. In ABC the same information loss occurs silently.

8.5 Sequential schemes and amortisation

A central design choice is between amortised and sequential inference. In the amortised mode the network is trained once on prior-drawn simulations and thereafter returns the posterior for any observation instantly; the up-front simulation cost is amortised over all future analyses. This is transformative when many datasets share the same model, for instance a posterior for every gravitational-wave event from a single trained network. The drawback is statistical inefficiency: simulations are scattered across the whole prior, and if the observation is informative, almost all of them land where the posterior has negligible support and thus teach the network little about the region that matters.

When simulations are expensive and only one observation is of interest, sequential methods win. They proceed in rounds: infer a preliminary posterior $\tilde{p}(\boldsymbol{\theta})$, use it as the proposal for the next batch of simulations, and refit. Iterating concentrates simulations in the high-posterior region and sharpens the estimate with far fewer

calls. The subtlety is that drawing $\boldsymbol{\theta} \sim \tilde{p}$ instead of $\boldsymbol{\theta} \sim \pi$ changes what the network learns, and the objective must be corrected accordingly.

8.6 Validation and calibration

Because an SBI posterior is the output of a trained network rather than an exact computation, it must be validated. A flow can be overconfident if it is undertrained, or biased if the training simulations are too few, and no internal quantity announces the failure. Fortunately. the generative setting hands us an endless supply of labelled test cases: we can draw a parameter, simulate data, infer, and check the answer against the known truth, as many times as we like.

The basic diagnostic is a coverage test. For a genuinely $100(1-\alpha)\%$ credible region, the true parameter should fall inside it in a fraction $1-\alpha$ of test simulations, exactly the frequentist coverage statement of Equation (4.42), now turned around and used as a check on a Bayesian method. Under-coverage (the truth lands outside too often) signals an overconfident posterior, over-coverage signals an unnecessarily broad one. The systematic, all-at-once version is simulation-based calibration (SBC), which examines not one credibility level but the whole posterior shape.

Mathematical background: why SBC ranks are uniform

Fix a scalar summary of $\boldsymbol{\theta}$ (say one component). Generate a test case by drawing $\boldsymbol{\theta}_0 \sim \pi$ and $\boldsymbol{d} \sim p(\boldsymbol{d} \mid \boldsymbol{\theta}_0)$, then draw L posterior samples $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_L \sim p(\boldsymbol{\theta} \mid \boldsymbol{d})$ from the (correct) inference engine. Conditioned on $\boldsymbol{d}$, the truth $\boldsymbol{\theta}_0$ is itself a draw from $p(\boldsymbol{\theta} \mid \boldsymbol{d})$, because the pair was generated from the joint $p(\boldsymbol{\theta}, \boldsymbol{d})$ and conditioning on $\boldsymbol{d}$ leaves $\boldsymbol{\theta}_0 \sim p(\boldsymbol{\theta} \mid \boldsymbol{d})$. Hence $\boldsymbol{\theta}_0$ and the L samples are $L+1$ iid draws from the same distribution, so the rank of $\boldsymbol{\theta}_0$ among them is uniformly distributed on $\{0, 1, \ldots, L\}$. Averaging over $\boldsymbol{d}$ preserves the uniformity. A histogram of these ranks over many test cases must therefore be flat; systematic departures diagnose specific pathologies:

- $\cup$-shaped (ranks pile up at the extremes): the posterior is too narrow, i.e. overconfident;
- $\cap$-shaped (ranks pile up in the middle): the posterior is too wide;
- sloped or shifted: the posterior is biased, its mean systematically off.

SBC thus converts calibration into a simple, interpretable uniformity test.

Two further checks are standard. First, posterior predictive checks push posterior samples back through the simulator and verify that the resulting mock data resemble the real data, a generative echo of the goodness-of-fit ideas of Section 4.9. And a final, inescapable caveat: the whole pipeline is only as trustworthy as the simulator. If the forward model omits a systematic that is present in the real Universe, the inference will be confidently wrong, and no amount of calibration on simulations can reveal a defect that the simulations do not contain. Detecting such model misspecification, by comparing the observed summaries to the distribution of simulated ones, is an active area.

8.7 Summary and practical remarks

Simulation-based inference extends the Bayesian odeas of Section 5 to the vast class of models defined by a simulator rather than a formula. It keeps Bayes' theorem

and every downstream notion, posteriors, credible regions, marginalisation, model comparison, completely intact, and replaces only the single ingredient that has become inaccessible, the likelihood, with a quantity learned from simulated data. Figure 8.3 summarises the end-to-end workflow, mirroring those of Figure 4.9 and Figure 5.7.

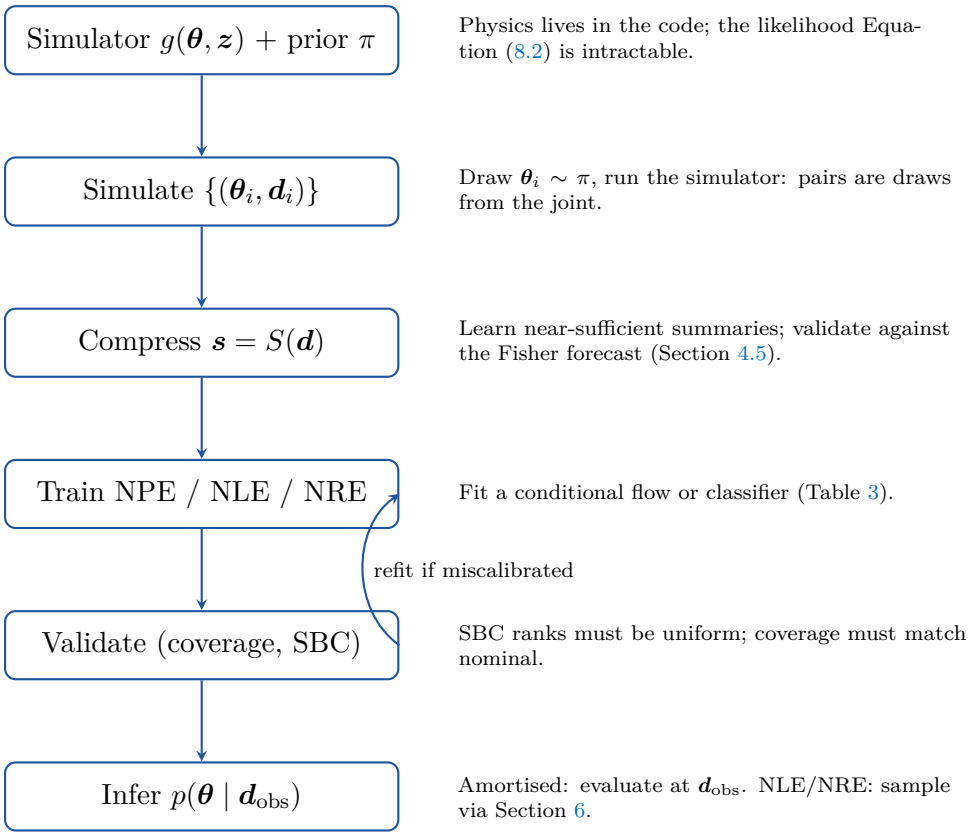


Figure 8.3: *End-to-end simulation-based inference workflow, with pointers to the relevant results. The dashed feedback loop stresses that validation is not optional: a miscalibrated network is retrained (more simulations, richer summaries, or a more flexible model) before any result is quoted.*

Choosing an SBI method

1. **ABC**: conceptually transparent and assumption-light, but scales exponentially with the number of summaries and discards most simulations. Use for very low-dimensional problems, quick baselines, or pedagogy.
2. **NPE**: learns the posterior directly and evaluates it without sampling. The default when the prior is fixed and many observations share the same model.
3. **NLE / NRE**: learn the likelihood or its ratio, so the prior can be changed after training, at the cost of an MCMC step. NRE reduces inference to classification and tends to be robust when the data are high-dimensional.

A few practical points to close:

1. **Compress before you infer**: Learn near-sufficient summaries and validate them against the Fisher forecast of Section 4.5; do not feed raw high-dimensional data to the network unless it is small.

2. **Always run coverage tests and SBC**: An SBI posterior is a trained approximation; quoting it without a calibration check invites an over- or under-confident result. Uniform SBC ranks are necessary, not sufficient, but their failure is a red flag.

3. **Budget your simulations**: Amortised inference pays off when many datasets share a prior; sequential inference is far more simulation-efficient for a single expensive analysis, provided the proposal correction is applied.

4. **The simulator is the model**: SBI is only as faithful as the forward model. Model misspecification, a simulator that fails to reproduce the real data, is the dominant risk and cannot be cured by more training or better networks.

5. **Cross-check against a tractable case**: Wherever a Gaussian likelihood is defensible, compare the SBI posterior to the explicit result of Section 5; agreement in the regime where both apply builds confidence in the pipeline before it is trusted where only SBI can go.