



Universiteit  
Leiden  
The Netherlands

# Modern Astrostatistics

–The Latin of data analysis–

Dr Ln( $\alpha$ ) Sellentin

Sterrewacht, Universiteit Leiden

**Scientific Content:**

Dr. Elena Sellentin  
Sterrewacht Leiden  
Universiteit Leiden  
Postbus 9500  
2300 RA Leiden

**Latex Style File:**

The Style File is licensed under the Creative Commons Attribution-NonCommercial 3.0 Unported License (the “License”). You may not use this file except in compliance with the License. You may obtain a copy of the License at <http://creativecommons.org/licenses/by-nc/3.0>. Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

## Literature recommendations

- Phil Gregory, “Bayesian Logical Data Analysis for the Physical Sciences”. (Excellent all-rounder.)
- David MacKay, “Information Theory”. (Excellent for algorithms and sampling, linked [here](#).)
- Gelman, Carlin,..., Rubin, “Bayesian Data Analysis”, CRC Press. (Very basic, but very complete as well.)
- E. T. Jaynes, “Probability Theory: The Logic of Science”. (Pedagogical, sometimes dogmatic, linked [here](#).)
- J.L. Starck, F. Murtagh, J. Fadili, “Sparse Image and Signal Processing”. (Data-driven, data-manipulation, little on inference.)
- Roberto Trotta, “Bayes in the sky”, arXiv:0803.4089. (Research review from 2008.)
- T. W. Anderson, “An Introduction to Multivariate Statistical Analysis”. (Not so ‘introductory’, but mathematically very insightful.)
- [Analysis behind the first gravitational wave detection](#)
- [Surprises concerning Chi-squared as goodness of fit](#)

THIS SCRIPT SUPPORTS THE LECTURE AND SERVES AS A SOURCE FOR LATER REFERENCE.  
THE EXAM MAY CONTAIN PROBLEMS NOT DESCRIBED IN THE SCRIPT.



# Contents

## I Part One: Getting used to Noise

<b>1</b>	<b>Accepting Randomness</b>	<b>6</b>
1.1	Numerical Setup	6
1.2	Cross-Validation: Why care about noise at all?	7
1.3	Redundant encryption to guard against noise corruption	8
1.3.1	Noisy encryption	9
1.4	Basics of probability manipulations	10
1.4.1	Summarizing data sets: (sufficient) statistics	11
1.5	Propagating noise: From one distribution to another	12
1.5.1	One random variable	13
1.5.2	Two random variables	15

## II Part Two: Inferring a model from noisy data

<b>2</b>	<b>Frequentist versus Bayesian</b>	<b>19</b>
2.0.1	Bayes theorem	20
2.1	Many random variables: (Bayesian) Hierarchical Models	21
2.2	The BHM Cheat Sheet	21
2.2.1	Parameter degeneracies	22
2.3	Selected probability distributions	23
2.4	Priors	24
2.4.1	Improper priors	24

2.4.2	Uninformative priors . . . . .	24
<b>2.5</b>	<b>Model selection</b>	<b>25</b>
2.5.1	The Bayesian evidence . . . . .	25
<b>2.6</b>	<b>The Devil's staircase and singular probability distributions</b>	<b>25</b>
<b>2.7</b>	<b>Missing data</b>	<b>26</b>

## III

## Part Three: Filtering, optimization, and sparsity

<b>2.8</b>	<b>Filtering</b>	<b>28</b>
<b>2.9</b>	<b>Optimization</b>	<b>29</b>
2.9.1	Lagrange multipliers . . . . .	29
2.9.2	Cauchy-Schwarz inequality . . . . .	30
<b>2.10</b>	<b>The matched filter</b>	<b>30</b>
<b>2.11</b>	<b>The Wiener filter</b>	<b>31</b>
<b>2.12</b>	<b>Weighting schemes</b>	<b>32</b>
<b>2.13</b>	<b>Creating your own optimal filters</b>	<b>34</b>
<b>2.14</b>	<b>Wavelets</b>	<b>34</b>

## IV

## Part Four: Numerical techniques

<b>3</b>	<b>Sampling Methods . . . . .</b>	<b>37</b>
3.1	The Gibbs sampler	37
3.2	Detailed Balance	38
3.3	Metropolis(-Hastings) algorithm	38
3.4	Hamilton Monte Carlo	39
3.5	Convergence	40
3.6	Manipulations of MCMC chains	40
<b>4</b>	<b>Basics of Machine Learning . . . . .</b>	<b>41</b>
<b>4.1</b>	<b>PAC learnable and finite VC-dimension</b>	<b>42</b>
4.1.1	No Free-Lunch Theorem . . . . .	42
<b>4.2</b>	<b>Artificial Neural Networks (ANN)</b>	<b>42</b>
<b>4.3</b>	<b>Linear Algebra and Matrix Manipulations</b>	<b>42</b>
4.3.1	Singular value decomposition . . . . .	42



# Part One: Getting used to Noise

<b>1</b>	<b>Accepting Randomness</b> .....	<b>6</b>
1.1	Numerical Setup	
1.2	Cross-Validation: Why care about noise at all?	
1.3	Redundant encryption to guard against noise corruption	
1.4	Basics of probability manipulations	
1.5	Propagating noise: From one distribution to another	



An abstract background image featuring a dense, glowing network of blue lines and nodes, resembling a complex web or a data visualization of connections. The lines are of varying thickness and brightness, creating a sense of depth and movement. The nodes are small, bright blue spheres that serve as connection points. The overall color palette is dominated by shades of blue, with some lighter, almost white highlights where the lines and nodes are most concentrated.

# 1. Accepting Randomness

Most students of statistics describe an initial confusion and frustration, independent of their prior academic record: The random events just don't seem to obey the rules which held so far, and seemingly 'logical' solutions turned out to work sometimes, and sometimes not. This initial confusion is absolutely normal, and I can assure you that after a certain time it will give way to a steady feeling of tolerance ('Yep, it's noisy!'). Often a feeling of inquisitive curiosity then sets in ('I wonder what happens this time?'), and many statisticians also develop a surprising degree of self-irony ('How do I fool myself best today?'). Students also often describe that many 'patterns' which they saw in their lives so far, vanished from their perception, whereas they suddenly began to perceive new patterns around themselves.

All this happens, because learning statistics increases your tolerance of randomness. It teaches you to distinguish correlation from causation, and it teaches you that one-and-the-same event can suddenly appear very differently if it is realized in a noisy process. The former will remodel your assessment of 'significance' and it will teach you not to over-interpret. The latter will change your perception of 'similarity'. In research, having been trained in statistics has not only the advantage of possessing the necessary numerical skills to analyze data, but it also helps you understand information flows, and think critically about whether you just discovered a wonderful Nobel-prize worthy effect, or whether it may just be... noise.

In this lecture, we shall be focusing on astronomy-relevant situations, ranging from manipulations of noisy images, to inferring e.g. the cosmological model from data of the large sky surveys. Most methods have however a much wider scope: Optimizing the scientific return of a satellite mission is mathematically equivalent to e.g. optimizing the financial returns of a company. Numerical skills will be a key asset throughout this course. Even though you are fully free to choose your favourite programming language, I can recommend the following setups:

## 1.1 Numerical Setup

1. C/C++, with the Gnu Scientific Library (GSL), Armadillo, and maybe BOOST.  
If you intend to go towards analyses of huge data sets, then these elements will become your friends.

2. Python, with scipy, numpy, cython, and Jupyter-Notebook.

More user-friendly, and easy plotting, but it is not a compiled language, so in large codes you might dearly miss the feedback from your compiler.

## 1.2 Cross-Validation: Why care about noise at all?

Consider the situation in Fig. 1.1. A number of data points is given, and we are supposed to fit a line to them. We already know from hearsay that the best-fitting line is not supposed to hit all data points, but rather miss most points and lie in the middle of the general trend. Why is the line which hits all data points not the best explanation for the data set?

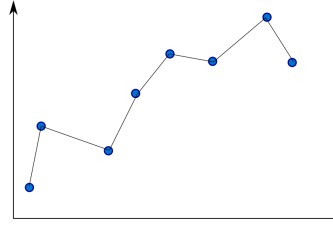


Figure 1.1: Why noise matters.

We investigate this as follows. Let each of the  $n$  data points be a tuple  $\mathbf{x}_i = (x_i, y_i)$ , which we collect in a data matrix  $D = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)$ . If one of the two components is fully noise-free, e.g. all ordinate values  $x_i$  had no noise at all, then we could equally replace the ordinates by indices, and instead use a data vector  $\mathbf{y} = \{y_1, y_2, \dots, y_n\}$ . We shall do this here for simplicity. Let us denote the curve by  $f(x, \theta)$ , where  $\theta$  are parameter which modify the shape of the curve.

We now need a measure which quantifies what we mean by *best* when speaking of best-fitting. Let us chose the  $L_1$ -norm, and demand the best-fitting curve to be the one which minimizes the  $L_1$  norm:

$$\text{best } f(x, \hat{\theta}) = \operatorname{argmin}(L_1[f(x, \theta), \mathbf{y}]), \quad (1.1)$$

i.e. the curve parameter  $\theta$  has to be chosen such that

$$L_1 = \sum_i^n |f(x_i, \theta) - y_i|, \quad (1.2)$$

is minimized. Clearly the minimal value of  $L_1$  is zero, meaning the best-fitting curve would then be the one which indeed hits all data points perfectly. Why is the connect-the-points solution then not the one we want?

This becomes evident by imagining that the data points given are but a subset of all data points that actually existed. Imagine a new set of equally good data points  $\tilde{\mathbf{y}}$  is added to the first set  $\mathbf{y}$ . Let now  $f(x, \theta_c)$  be the connect-the-points solution, and  $f(x, \theta_s)$  be a smooth solution which misses most points. Then we have

$$L_1(f_c, \mathbf{y}) = 0, \quad L_1(f_s, \mathbf{y}) > 0 \quad (1.3)$$

but importantly

$$L_1(f_c, \tilde{\mathbf{y}}) = H, \quad L_1(f_s, \tilde{\mathbf{y}}) = h, \quad (1.4)$$

with typically  $H \gg h$ . This means the connect-the-points solution fails to represent an equivalent data set; it only fitted a single noise realized but it did not recover an underlying truth. Due to  $h < H$ , the smooth curve was closer to the second data set, and hence represented better the underlying truth (which might be fully recovered for  $n \rightarrow \infty$ ).

**Corollary 1.2.1** Best-fitting solutions must always miss most of the data points, since they are meant to not only explain the data set you have, but also equivalent data sets which you never got.

**Exercise 1.1** Sample this cross-validation.

1. Pick the curve  $f(x) = ax^2 + bx^3$ , with  $a = 1, b = 4$ .
2. Create a noisy realization  $\mathbf{y}$  of 10 data points with  $x_i = 1, 2, \dots, 10$  and  $y_i \sim \mathcal{G}(f(x_i), 0.1)$ .
3. Evaluate  $L_1$  for a smooth curve through the points, e.g. vary the prefactors of the polynomial by hand and fit ‘by eye’.
4. Evaluate  $L_1$  for the connect-the-points solution.
5. Create a second realization  $\mathbf{y}_2$  of the noisy process.
6. Evaluate  $L_1$  with respect to  $\mathbf{y}_2$ .
7. Repeat 1000 times, make a histogram of  $L_1(f_c, \mathbf{y}_i), L_1(f_s, \mathbf{y}_i)$
8. Go back to Step 2 and now fit by picking the (a,b) combination which minimizes  $L_1$ . Then repeat the remaining steps.

### 1.3 Redundant encryption to guard against noise corruption

Imagine a satellite, which took an image of a galaxy. This image is our raw data, and for the sake of the argument we here assume it to be noise free. However, the downlink from the satellite will not be noise free, and hence a noisy image will arrive here on Earth. How can the space agencies guard themselves against the noise from the downlink?

The answer is redundant encryption. Let us represent the image by a  $p \times p$  matrix  $M$ , such that the elements  $M_{ij}$  are the pixels of the image. To transform it into a data vector, we define a matrix vectorization operation

$$\text{vec}(M) = \mathbf{d}, \quad \text{such that } M_{ij} = d_{i^*p+j}. \quad (1.5)$$

The vector  $\mathbf{d}$  now has  $p^2$  elements. We then represent a (here linear) redundant encryption by an operator

$$\mathbf{E}\mathbf{d} = \mathbf{e}, \quad \text{with } \dim(\mathbf{e}) = r > p^2. \quad (1.6)$$

meaning the encrypted vector  $\mathbf{e}$  now holds more elements than the original vector  $\mathbf{d}$ . The operator  $E$  has hence  $p^2$  columns (across) and  $r$  rows (down). If the encryption by itself is lossless, then there exists a corresponding decryption operator  $D$  such that

$$\mathbf{D}\mathbf{e} = \mathbf{d} \quad (1.7)$$

where  $D$  has now  $p^2$  rows and  $r$  columns. I.e. it takes us back to the original data vector. An



example could be the operators

$$E = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad D = \frac{1}{2} \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix} \quad (1.8)$$

where  $E$  enforces that the input data vector is repeated 2 times, and  $D$  undoes this operation.

If noise now occurs during transmission of the encrypted vector  $\mathbf{e}$ , then it might e.g. change certain elements of it,

$$\forall i: e_i \rightarrow e_i + n_i. \quad (1.9)$$

Decrypting the noisy data vector

$$D(\mathbf{e} + \mathbf{n}) = \hat{\mathbf{d}} \quad (1.10)$$

will then produce an *estimator* of the true original message. The double repetition will then have suppressed the noise in  $\hat{\mathbf{d}}$  as compared to how strongly  $\mathbf{d}$  would have been affected by noise, if  $\mathbf{d}$  had been transmitted directly.

Clearly, the aim of decryption is to use any source of information available, in order to either facilitate the decryption, accelerate it, or improve its faithfulness. If we had more information on how noise arises during transmission, we could redefine the decryption operator  $D \rightarrow D'$ , such that  $D'$  handles noise more accurately than  $D$  which uses no information at all about the noise pattern.

### 1.3.1 Noisy encryption

It is often illustrative to think of statistics as a noisy encryption problem, which is to be undone as well as possible. Imagine there are parameters  $\vec{\theta} = (\vartheta_1, \vartheta_2, \dots, \vartheta_n)$  which are of physical interest, for example the parameters of the cosmological standard model. Let us assume  $\vec{\theta}$  is indeed the combination of parameters that ‘the Universe’ uses (i.e. the correct ones, but in reality we would never know.) Then we can represent astronomical data as

$$\mathbf{d} = \mathbf{E}\vec{\theta}, \quad (1.11)$$

where  $\mathbf{E}$  is now a redundant encryption operator, which encrypts the  $n$  parameter values in astronomical data sets  $\mathbf{d}$ , where  $\mathbf{d}$  has dimension  $N \gg n$ . The boldfont  $\mathbf{E}$  thereby indicates that the encryption operator is now random itself, and hence produced random data  $\mathbf{d}$  from the classical (i.e. non-random) parameter vector  $\vec{\theta}$ .

This though experiment teaches us the following vital insights

1. Redundancy in data sets helps us suppress noise. We always have to search for it.
2. Since  $\mathbf{E}$  is random, its corresponding decryption operator  $\mathbf{D}$  will also be random. We might however only be able to write down a classical decryption operator  $D$ . In either case the output  $\theta = D\mathbf{d}$  or  $\theta = \mathbf{D}\mathbf{d}$  will now be random (because  $\mathbf{d}$  and  $\mathbf{D}$  are random). This means *estimated* parameters will also be *random variables*. If we succeed, then  $\theta$  will scatter around  $\vec{\theta}$ .
3. If we are irresponsible during the decryption, e.g. through excessive modelling or oversimplifications of  $\mathbf{D}$ , then the output of our decryption will be *another* message than the originally encrypted one.

**Definition 1.3.1 — Notation.**

- Random vectors will be indicated in boldfont, e.g.  $\mathbf{d}$ .
- Data are sometimes called  $\mathbf{x}$ , sometimes  $\mathbf{d}$ , depending which notation minimizes confusion.
- Random matrices will be indicated in boldfont, e.g.  $\mathbf{D}$ .
- Classical (non-random) vectors will be indicated with vector signs,  $\vec{x}$ .
- Probability distributions will be indicated in curly fonts  $\mathcal{G}$ ,  $\mathcal{P}$ .
- We write  $\mathbf{x} \sim \mathcal{D}(\mathbf{x})$  if the random variable  $\mathbf{x}$  follows the distribution  $\mathcal{D}$ .

**1.4 Basics of probability manipulations**

Random variables are drawn from probability distributions, and we write ‘drawn from’ as ‘ $\sim$ ’. We denote simultaneous occurrence of two random events  $A$  and  $B$  as  $\mathcal{P}(A, B)$ . Conditional occurrence is denoted with a  $|$  instead, e.g.  $\mathcal{P}(A|B)$  is the probability of  $A$  occurring, given that  $B$  has occurred. The basics of probability manipulations follow from the Kolmogorov axioms.

**Theorem 1.4.1 Kolmogorov axioms**

1.  $P \geq 0$
2.  $\sum_i P_i = 1$ , especially  $P(A) + P(\bar{A}) = 1$ .
3.  $P(A, B|C) = P(A|C)P(B|A, C) = P(B|C)P(A|B, C)$ .

We then have, also for distributions

$$\mathcal{P}(A, B) = \mathcal{P}(A|B)\mathcal{P}(B) = \mathcal{P}(B|A)\mathcal{P}(A). \quad (1.12)$$

From this we see that we can exchange the order of conditionality via

$$\mathcal{P}(A|B) = \frac{\mathcal{P}(B|A)\mathcal{P}(A)}{\mathcal{P}(B)}. \quad (1.13)$$

This is called *Bayes’ theorem* but it does not yet have anything to do with what is today known as ‘Bayesian statistics’. It simply follows from the laws of probability manipulations.

We now apply Bayes theorem to data  $\mathbf{x}$ , which are a noisy realization of a theory with parameters  $\theta$ . To minimize the number of identical symbols floating around, we switch to the following notation. The probability of parameter values producing data  $\mathbf{x}$  is denoted as

$$\mathcal{P}(\mathbf{x}|\theta) = \mathcal{L}(\mathbf{x}|\theta), \quad (1.14)$$

and is called *the likelihood*. It describes the order in which the noise process occurs in nature: Given yet undiscovered parameter values  $\theta$ ,  $\mathcal{P}(\mathbf{x}|\theta)$  produces a noisy realization of data. In contrast  $\mathcal{P}(\mathbf{x})$  is simply a distribution of noisy data, independent of a theory.

We call

$$\mathcal{P}(\theta|\mathbf{x}), \quad (1.15)$$

the *posterior likelihood*, or *posterior* for short. It describes the human attempt of inferring the parameter values  $\theta$  from observed data  $\mathbf{x}$ . The workflow is now the inverse of  $\mathcal{L}(\mathbf{x}|\theta)$ , which is why inference problems are also often named ‘inverse problems’. To relate the posterior and the likelihood, we require the probabilities  $\mathcal{P}(\mathbf{x}) = \pi(\mathbf{x})$  and  $\mathcal{P}(\theta) = \pi(\theta)$ . These are called *priors*, as they describe an a-priori probability of data  $\mathbf{x}$  or parameter values  $\theta$ . According to Bayes’ theorem we then have

$$\mathcal{P}(\theta|\mathbf{x}) = \frac{\mathcal{L}(\mathbf{x}|\theta)\pi(\theta)}{\pi(\mathbf{x})}. \quad (1.16)$$

**Distribution tails and p-values**

Let  $x \sim \mathcal{D}(x)$ , i.e. the random variable follows the distribution  $\mathcal{D}$ . Distributions typically have tails, which indicate realizations of the data which are rather improbable. We call

$$P(x \geq t) = \int_t^{\infty} \mathcal{D}(x) dx, \quad (1.17)$$

the *tail-probability*. It describes the probability that one realization  $x$  comes from the tail beyond the threshold  $t$ . If the density  $\mathcal{D}$  also depends on a hypothesis  $H$ , we write  $\mathcal{D}(\mathbf{x}|H)$ . The probability

$$P(x \geq t|H) = \int_t^{\infty} \mathcal{D}(x|H) dx, \quad (1.18)$$

is a sensible thing to calculate: it describes that *if*  $H$  is true, then  $x$  will be larger than the threshold a certain fraction of the times. However, if you have data  $x$  and  $P(x \geq t|H)$  is low, this does not automatically mean the hypothesis  $H$  is wrong. Sometimes it does, sometimes it doesn't! This idea is called 'p-values for hypothesis rejection', and we will study why it is so tricky.

**Exercise 1.2 'Proving' the pope ain't human.** There are about  $8 \cdot 10^9$  people in the world, but only one pope. Given that you are human ( $h$ ), the probability of you being the pope ( $p$ ) is

$$P(p|h) = 1/(8 \cdot 10^9) = 1.25 \cdot 10^{-10}.$$

If you are the pope, the probability that you are human is then

$$P(h|p) = 1.25 \cdot 10^{-10}.$$

This p-value is smaller than 0.01, hence we reject the null-hypothesis of the pope being human. **What went wrong?**

■

**1.4.1 Summarizing data sets: (sufficient) statistics**

'A' statistic  $T$  is a function  $T = f(x_1, x_2, \dots, x_n)$  of the random sample  $\mathbf{x} = (x_1, \dots, x_n)$ . Statistics  $T$  are random variables, because the input to the function  $f$  are random variables. The function  $f$  will often be a *lossy* operation, and  $T$  will then contain less information than the original data set  $\mathbf{x}$ . In certain special situations,  $f$  will not be lossy – this then results in *sufficient statistics* which contain the same information as the original data set  $\mathbf{x}$ , but can be handled more easily than the often large and hence unwieldy  $\mathbf{x}$ . Amongst the sufficient statistics, those are most important for which their entire distribution function (meaning their entire statistical behaviour) is known.

Examples of sufficient statistics are:

- $x_i \sim \mathcal{G}(\mu, \sigma) \rightarrow T = \bar{x} = \frac{1}{N} \sum_{i=1}^N x_i \sim \mathcal{G}(\mu, \sigma/\sqrt{N})$ .
- For a random Gaussian field with  $\mu = 0$ ,  $\hat{\xi}(r)$  (the real-space correlation function) or equivalently  $\hat{C}(\ell)$  (the estimated harmonic power spectrum on the sphere), or  $\hat{P}(k)$  (the estimated power spectrum of Fourier modes in Euclidean space).

Other interesting statistics are

- $T = \max(x_1, \dots, x_n)$  or  $T = \min(x_1, \dots, x_n)$ , leading to *extreme value statistics*. Astronomy is full of extreme value statistics: galaxy clusters are *the most massive* objects, cosmic voids are the *least dense* regions, telescopes preferentially see *the brightest* objects, the perpetual chase after the 'galactic distance record holder' is the chase of 'the brightest of the earliest galaxies', etc.

**How to find sufficient statistics**

**Theorem 1.4.2** Sufficient statistics can be found iff the joint density of random variables  $x_1, \dots, x_n$  can be factorized as

$$\mathcal{P}(x_1, \dots, x_n | \theta) = \mathcal{U}(x_1, \dots, x_n) \mathcal{V}[T(x_1, \dots, x_n), \theta] \quad (1.19)$$

where  $\mathcal{U}$  and  $\mathcal{V}$  are positive-semidefinite functions.  $\mathcal{U}$  mustn't depend on  $\theta$  for  $T$  to be a sufficient statistic with respect to  $\theta$ .  $\mathcal{U}$  can depend on all of the  $x_i$  individually, but  $\mathcal{V}$  is only allowed to depend on the statistic  $T$ , instead of the individual  $x_i$ . For inference of the parameter  $\theta$ ,  $\mathcal{V}$  should obviously depend on  $\theta$ .

**Proof of the mean of Gaussian rvs being a sufficient statistic**

We assume that we are given a data set  $\mathbf{x} = (x_1, \dots, x_n)$ , where all  $x_i \sim \mathcal{G}(\mu | \sigma)$ , meaning the mean  $\mu$  is to be estimated/inferred, but the standard deviation  $\sigma$  is known.

We set up the joint density of the random variates, in order to proof that it factorizes as required. The joint density is

$$\mathcal{P}(x_1, \dots, x_n | \mu) = \frac{1}{\sqrt{(2\pi\sigma^2)^n}} \exp \left( -\frac{1}{2\sigma^2} \sum_{i=1}^n (x_i - \mu)^2 \right). \quad (1.20)$$

We now work out the square

$$\mathcal{P}(x_1, \dots, x_n | \mu) = \frac{1}{\sqrt{(2\pi\sigma^2)^n}} \exp \left( -\frac{1}{2\sigma^2} \sum_{i=1}^n x_i^2 + \frac{\mu}{\sigma^2} \sum_{i=1}^n x_i - \frac{n\mu^2}{2\sigma^2} \right). \quad (1.21)$$

We want to infer  $\mu$ , hence we need to collect all terms which contain  $\mu$  and hope that they factorize out in a positive-semi-definite function  $\mathcal{V}[T, \mu]$ . And indeed, we can define

$$T = \bar{x} = \frac{1}{n} \sum_{i=1}^n x_i, \quad (1.22)$$

and then have the  $\mu$ -dependent part

$$\mathcal{V}[T(x_1, \dots, x_n), \mu] = \exp \left( -\frac{n\mu^2}{2\sigma^2} + \frac{n\mu}{\sigma^2} T(x_1, \dots, x_n) \right). \quad (1.23)$$

The remaining factor is then

$$\mathcal{U}(x_1, \dots, x_n) = \frac{1}{\sqrt{(2\pi\sigma^2)^n}} \exp \left( -\frac{1}{2\sigma^2} \sum_{i=1}^n x_i^2 \right). \quad (1.24)$$

Both,  $\mathcal{V}$  and  $\mathcal{U}$  fulfill the requirements set to them.

This thus proves that  $T = \bar{x}$  is a sufficient statistic for Gaussian rv with known  $\sigma$ . In fact, the entire distribution function of  $\bar{x}$  is also known, which makes it even more useful:

$$\bar{x} \sim \mathcal{G}(\mu, \sigma/\sqrt{n}) \quad (1.25)$$

**1.5 Propagating noise: From one distribution to another**

We imagine the situation where we have a random variable  $x'$  whose probability distribution we know to be  $\mathcal{P}(x')$ . Maybe we can however only observe/predict a function of  $f(x') = y'$ , or  $y'$  is for any other reason of interest. All information about the random  $y'$  is in its probability distribution  $\mathcal{P}(y')$ , so it is essential to derive  $\mathcal{P}(y')$ . In this section, we learn different ways of achieving this.

### 1.5.1 One random variable

In this section we imagine  $y'$  to only depend on a single other variable  $y' = f(x')$ . In the next section, we will progress to  $y'$  being a function of two random variables  $u, v$ .

#### Via the cumulative distribution function

In the case of integer-valued random variables, or finite support of the probability density function, or non-differentiable functions, it can be tricky to derive new distribution functions. Most proofs then run via the cumulative distribution function. (This applies for example to X-ray astronomy, where photons can be extremely rare, such that no continuum-approximation works well. One is then often stuck with Poisson distributions and other distributions of random integers.)

We now denote random variables with a prime, like  $x', y'$ , because we have to introduce associated thresholds  $x, y$ .

The cumulative distribution function of  $x'$  is

$$\mathcal{C}_{x'}(x) = P(x' \leq x) = \int_{x'_{\min}}^x \mathcal{P}(x') dx'. \quad (1.26)$$

The subscript  $x'$  indicates that it is the cumulative distribution function (cdf) of the random variable  $x'$ . The cdf is a function of the variable  $x$ , which is the threshold in the first equality, and the upper boundary in the second equality. The variable  $x$  is not random: the threshold is a classical variable.

Likewise, the cdf of  $y'$  is

$$\mathcal{C}_{y'}(y) = P(y' \leq y) = \int_{y'_{\min}}^y \mathcal{P}(y') dy'. \quad (1.27)$$

$\mathcal{P}(y')$  is now what we wish to find, given that we know  $\mathcal{P}(x')$ , and given the constraint  $y' = f(x')$ , with  $f$  known.

We hence plug the relation  $f$  into (1.27)

$$\mathcal{C}_{y'}(y) = P(f(x') \leq y). \quad (1.28)$$

It is important to notice that we have substituted the *random* variable, not the classical threshold. The aim is now to manipulate the inequality  $f(x') \leq y$  until it reads  $x' \leq \text{expr}(y)$ . The reason for doing so, is that the term after the equality then reads

$$P(x' \leq \text{expr}(y)) \equiv \mathcal{C}_{x'}(\text{expr}(y)), \quad (1.29)$$

because the probability of  $x'$  being smaller than some threshold is by definition the cdf of  $x'$ . Combining (1.29) with (1.28), we have that the cdf of  $y'$  is

$$\mathcal{C}_{y'}(y) = P(x' \leq \text{expr}(y)) = \mathcal{C}_{x'}(\text{expr}(y)), \quad (1.30)$$

where the last equality is something we can evaluate, since we know  $\mathcal{C}_{x'}$  and  $\text{expr}$  is an expression which results from the function  $f$ . The sought pdf is then

$$\mathcal{P}(y) = \frac{d}{dy} \mathcal{C}_{x'}(\text{expr}(y)). \quad (1.31)$$

#### Via the Jacobian

In many fortunate situations (1.31) can be further simplified: if  $f$  is invertible then we will have  $\text{expr} \equiv f^{-1}$ , the inverse function. If the inverse is also differentiable, then we can carry out the



differentiation from (1.31)

$$\begin{aligned}
 \mathcal{P}(y) &= \frac{d}{dy} \mathcal{C}_{x'}(f^{-1}(y)) \\
 &= \mathcal{P}_{x'}(x' = f^{-1}(y)) \frac{d}{dy} f^{-1}(y) \\
 &= \mathcal{P}_{x'}(x' = f^{-1}(y)) \left( \frac{df(x)}{dx} \right)^{-1}
 \end{aligned} \tag{1.32}$$

where we used the chain rule of differentiation. The result is intuitive since a transformation of variables needs to satisfy

$$\mathcal{P}_y(y)dy = \mathcal{P}_x(x)dx, \tag{1.33}$$

which is

$$\mathcal{P}_y(y) = \mathcal{P}_x(x) \frac{dx}{dy} = \mathcal{P}_x(x) \left( \frac{dy}{dx} \right)^{-1}. \tag{1.34}$$

Since  $y = f(x)$  the last fraction is again the reciprocal of the derivative  $f'(x)$ . Eliminating  $x$  in  $\mathcal{P}_x(x)$  by  $f^{-1}(y)$  we arrive again at

$$\mathcal{P}_y(y) = \mathcal{P}_x(f^{-1}(y)) \left( \frac{df(x)}{dx} \Big|_{x=f^{-1}(y)} \right)^{-1}. \tag{1.35}$$

### Via the moment-generating function

Finding a probability distribution  $\mathcal{P}(y)$  by indentifying its moment-generating function relies on a uniqueness theorem:

**Theorem 1.5.1** Let  $\mathcal{P}(x)$  be a probability density function and let  $\mathbb{E}(\cdot)$  be the associated expectation operator, such that  $\mathbb{E}(\cdot) = \int(\cdot)\mathcal{P}(x)dx$ . Then, the moment-generating function is defined to be the Laplace transform

$$M_x(t) = \mathbb{E}(e^{tx}). \tag{1.36}$$

By substituting  $t \rightarrow it$ , we can also define the Fourier transform

$$C_x(t) = \mathbb{E}(e^{\pm itx}), \tag{1.37}$$

where the rotation direction  $\pm$  of the Fourier phase is up to convention.  $C_x$  is called the characteristic function. These integrals do not necessarily exist, but if they do for a  $\mathcal{P}(x)$ , then the associated moment-generating function/characteristic function are *unique*.

Therefore, many probability distributions were *identified* by computing the moment-generating function and noticing that this function is already known in the literature. The technique of moment-generating functions is more important to identify multivariate extensions of univariate distributions, and not as important in a natural science such as physics.

### Examples

Imagine we know  $x \sim \mathcal{P}(x) = 3x^2$ , and we are interested in the distribution of  $y = x^2$ . We assume  $x \in [0, 1]$  for reasons of normalization.

We first derive  $\mathcal{P}(y)$  via the cdf. We have

$$\mathcal{C}_{x'}(x) = \int_0^x 3x'^2 dx' = x^3. \tag{1.38}$$

Since  $y = x^2, x \geq 0$ , we have  $x^3 = \sqrt{y^3}$ , and the cdf of  $y$  is then

$$\mathcal{C}_{y'}(y) = y^{3/2}, \quad (1.39)$$

and by differentiating we get

$$\mathcal{P}(y) = \frac{d}{dy} \mathcal{C}_{y'}(y) = \frac{3}{2} y^{1/2}. \quad (1.40)$$

Similarly, we could have gone via the Jacobian

$$\mathcal{P}(y) = \mathcal{P}(x) \frac{dx}{dy}. \quad (1.41)$$

Since  $dx/dy = \frac{1}{2} y^{-1/2}$ , we have

$$\mathcal{P}(y) = (3y) \left( \frac{1}{2} y^{-1/2} \right) = \frac{3}{2} y^{1/2}, \quad (1.42)$$

as before.

### 1.5.2 Two random variables

We now generalize to the situation where the random variable of interest is a function of *two* other random variables,  $y = f(u, v)$ , and we seek  $\mathcal{P}(y)$ , knowing  $\mathcal{P}(u, v)$ . If  $u, v$  are independent, then it is sufficient to know  $\mathcal{P}(u), \mathcal{P}(v)$ . Computing  $\mathcal{P}(y)$  then always consists of two steps:

- Eliminating one variable, e.g.  $u$ , by rewriting it as a function of  $y, v$ .
- Marginalizing over it, such that it also disappears, and only  $y$  remains.

#### Example: Ratio distributions

Let  $y = u/v$ , and we know  $\mathcal{P}(u, v)$ . We have

$$\int \mathcal{P}(u, v) du dv = 1. \quad (1.43)$$

Now we eliminate  $u$  via  $u = yv$ , and have

$$\int \mathcal{P}(yv, v) d(yv) dv = 1. \quad (1.44)$$

We introduce a one by writing out the trivial operation of differentiation and integration

$$\int \mathcal{P}(yv, v) \frac{d(yv)}{dy} dy dv = 1, \quad (1.45)$$

since we want our upcoming distribution to be normalized with respect to  $y$ , so we need  $dy$ . This brings us to

$$\int \mathcal{P}(yv, v) v dy dv = 1, \quad (1.46)$$

the unity on the right hand side can now be replaced by  $1 = \int \mathcal{P}(y) dy$ ,

$$\int \mathcal{P}(yv, v) v dy dv = \int \mathcal{P}(y) dy, \quad (1.47)$$

since  $\mathcal{P}(y)$  also needs to be normalized. By comparing the left and right-hand side, we identify our sought distribution as

$$\mathcal{P}(y) = \int \mathcal{P}(yv, v) v dv. \quad (1.48)$$

(Some absolute values went missing.)

**Product distributions**

This time we set  $y = uv$  and seek  $\mathcal{P}(y)$ . We proceed as before

$$\begin{aligned}\mathcal{P}(u, v) &= \mathcal{P}(y/v, v), \\ \int \mathcal{P}(u, v) \, du dv &= 1.\end{aligned}\tag{1.49}$$

Introducing the  $dy/dy$  we have

$$\begin{aligned}\int \mathcal{P}(y/v, v) \frac{d(y/v)}{dy} dy dv \\ = \int \mathcal{P}(y/v, v) \frac{1}{v} dv dy \\ = \int \mathcal{P}(y) dy,\end{aligned}\tag{1.50}$$

hence

$$\mathcal{P}(y) = \int \mathcal{P}(y/v, v) \frac{1}{v} dv.\tag{1.51}$$

If  $u$  and  $v$  were independent, then the joint distribution factorizes and we have

$$\mathcal{P}(y) = \int \mathcal{P}_u(y/v) \mathcal{P}_v(v) \frac{1}{v} dv.\tag{1.52}$$

**Distributions of sums**

Finally, we investigate the case  $y = u + v$ , and eliminate  $u = y - v$ . We then have

$$\begin{aligned}\int \mathcal{P}(u, v) \, du dv &= \int \mathcal{P}(y - v, v) \, d(y - v) dv \\ &= \int \mathcal{P}(y - v, v) \, dy dv.\end{aligned}\tag{1.53}$$

The last variable to be removed is  $v$ , so we marginalize it out, and arrive at

$$\mathcal{P}(y) = \int \mathcal{P}(y - v, v) \, dv.\tag{1.54}$$

If  $u, v$  were independent, their distribution factorizes again, where upon we arrive at

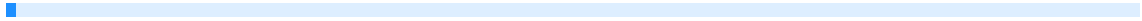
$$\mathcal{P}(y) = \int \mathcal{P}_u(y - v) \mathcal{P}_v(v) \, dv.\tag{1.55}$$

This integral is a *convolution*. It can be extended to many summands,  $y = \sum_i u_i$ , which is why one often hears that the Central Limit Theorem arises from ‘an infinite repetition of convolutions’: repeatedly convolving bumpy or edgy distributions with each other will smooth out all edges, until one arrives at a Gaussian distribution, which is the eigenfunction to the convolutional operator: convolving a Gaussian with a Gaussian again produces a Gaussian, so the infinite repetition of convolutions homes in on a stable solution (under mild prerequisites, e.g. existence of the integral in Eq. 1.55).

**Exercise 1.3**

1. For  $u, v$  independent, give the distribution of  $y = u^a v^b$ , for  $a, b$  non-random.
2. For  $u, v$  independent, give the distribution of  $y = u + v + u^a + v^b$  with  $a, b$  non-random.
3. For  $u \sim \mathcal{G}$  and independently  $v \sim \text{Uniform}$ , give the distribution of  $y = u/v$ .
4. Derive the Cauchy distribution for  $y = u/v$ ,  $u \sim \mathcal{G}(0, 1)$ ,  $v \sim \mathcal{G}(0, 1)$ ,  $u, v$  independent.

■





# Part Two: Inferring a model from noisy data

<b>2</b>	<b>Frequentist versus Bayesian</b>	<b>19</b>
2.1	Many random variables: (Bayesian) Hierarchical Models	
2.2	The BHM Cheat Sheet	
2.3	Selected probability distributions	
2.4	Priors	
2.5	Model selection	
2.6	The Devil's staircase and singular probability distributions	
2.7	Missing data	



## 2. Frequentist versus Bayesian

Statistics can be broadly split up into three schools: The first two schools are frequentist statisticians and Bayesian statisticians, both of which accept the introduction of models to explain data. The third school is composed of statisticians who seek ground truths, independent of models. Knowing when to use apply methods from which school is a hallmark of qualitative data analyses.

In the previous chapter, we dealt with basics of random processes and descriptive statistics. There has been no need for the introduction of a physical model in order to succeed in such manipulations of data sets. Typically however, astronomers do not only seek to *summarize* data, but to also *explain* them in a second step. This second step will necessarily introduce a *model* to explain the data. This makes the inference step model-dependent, whereas the former data-manipulation step can be conducted in model-independent manners. Transiting too quickly from model-independent preparations to model-dependent analyses is one of the mortal sins of statistics (because it quickly limits or biases your possibilities to answer whether your model is credible or not). Moreover, ‘the model’ is often an intricate compound of a *statistical model* on the one hand, and a *theoretical/physical model* on the other hand. Differentiating between the two can be difficult.

Mathematically, this can be seen as follows. Imagine there are noisy data  $\mathbf{x}$ , which follow a distribution

$$\mathcal{P}(\mathbf{x}). \quad (2.1)$$

For many manipulations of  $\mathbf{x}$ , knowing  $\mathcal{P}(\mathbf{x})$  is fully sufficient. If the distribution  $\mathcal{P}(\cdot)$  is *not* known, but assumed, then this imposes a *statistical model*. Imagine for example that  $\mathcal{P}(\cdot)$  is in reality a  $\chi^2$ -distribution, but the data analyst assumes it is a Gaussian instead. Then the statistical model does not reflect reality, and if the experiment can be repeated sufficiently many times, then the inadequacy of the statistical model can be detected.

A *theoretical/physical* model instead assumes that the data can be explained by theoretical physics or other scientific insights. It introduces the update

$$\mathcal{P}(\mathbf{x}) \rightarrow \mathcal{P}(\mathbf{x}|M, \theta). \quad (2.2)$$

Here, the introduction of the conditional statement is the (dangerous, if incorrect) novelty. The letter  $M$  abbreviates the ‘model’, and  $\theta$  describes free parameters of the model. Whether or not the

model is convincing, depends on the argumentation line of theoretical physics, *and* the model's compatibility with the data.

The order  $\mathcal{P}(\mathbf{x}|M, \theta)$  reads 'If  $M$  is the correct model, and if  $\theta$  are the correct parameter values, then  $\mathcal{P}(\mathbf{x}|M, \theta)$  describes how often nature will generate data  $\mathbf{x}$ .' The *if*-statement is binding: if any of the if-statements is broken, so are its consequences.

In a *frequentist* science, like particle physics, the order  $\mathcal{P}(\mathbf{x}|M, \theta)$  is the natural order in which to think. Often it is reasoned that particle physics is a frequentist science because it is a laboratory science, where experiments can be repeated until manpower, time and money run out. This is a superficial reason only.

The much more compelling reason for why particle physics is frequentist, is the quantum nature of its theoretical backbone: QFT, QED and QCD all operate with transition *probabilities*. The theory predicts branching *ratios*, event *rates*, it works with transition matrices/mixing matrices, the multitude of Feynman diagrams indicate possible transitions, etc. Whether or not a particle can be detected is then a fight against background and rare chances, hence being frequentist is natural.

In many branches of astronomy, generating new, fully independent data, would ultimately require observing further, independent universes. We do not have these at our disposal and the lack thereof forces many branches of astronomy to invert the order of the conditional statement, resulting in

$$\mathcal{P}(M, \theta|\mathbf{x}). \quad (2.3)$$

The new statement now reads 'Given that we observed the data  $\mathbf{x}$ , what have I learned about  $M$  and  $\theta$ , can I constrain which values the free parameters can credibly take?' This makes astronomy a *Bayesian* science. Historically, Frequentists, Bayesians and model-independence-seekers were often talking down to the respective 'mere adherents of the competing camps'. This was before certain problems were discovered to be naturally frequentist or naturally Bayesian. Thanks to further development of prior theory in the 1980s-2010s, the disputes between Frequentists and Bayesians have been largely arbitrated. Today, an epidemiologist who insists on infecting children in frequentist numbers with Ebola, is regarded as lacking statistical literacy, and a few other virtues.

We will now focus on Bayesian inference, due to its natural prevalence in astronomy.

### 2.0.1 Bayes theorem

In the introduction, we already had Bayes' theorem

$$\mathcal{P}(\theta|\mathbf{x}) = \frac{\mathcal{L}(\mathbf{x}|\theta)\pi(\theta)}{\pi(\mathbf{x})}, \quad (2.4)$$

where  $\mathcal{P}(\theta|\mathbf{x})$  is the posterior/the solution to the inverse problem: 'Given the data and the model, what are credible values for the parameters?'. The term  $\mathcal{L}(\mathbf{x}|\theta)$  is the likelihood/the forward workflow from theory to data: 'Given these parameter values, how often do we get which data set?'. The term  $\mathcal{P}(\theta)$  is a prior on the parameters, and was historically the cause of much unease between frequentists and Bayesians. The term  $\mathcal{P}(\mathbf{x})$  could be seen as a prior on the data. Its meaning becomes more transparent when written as

$$\pi(\mathbf{x}) = \int \pi(\mathbf{x}|\theta) d^n \theta. \quad (2.5)$$

This means  $\pi(\mathbf{x})$  is the marginal of the data over all possible values of the parameters. It describes the assumed model's capability to produce the data  $\mathbf{x}$  *at all*. If Eq. (2.5) holds, then  $\pi(\mathbf{x})$  is also often called 'marginal likelihood' or 'evidence'. If the equality in Eq. (2.5) does not hold, then one has a *missing data problem*, because the data depend on something else as well, which is omitted, but destroys the equality in Eq. (2.5).

## 2.1 Many random variables: (Bayesian) Hierarchical Models

Hierarchical Models are needed, if one is interested in multivariate probability distributions, or if one random variable is a function of many other random variables, all of which might be correlated, drawn from different distributions, or otherwise be statistically different.

Hierarchical Models allow one disentangle ‘error bars’, and optimize the individual components of the entire uncertainty budget. They are very much in the spirit of ‘Engineering 101: Dissect a complicated problem into feasible sub-problems’.

A Hierarchical Model is *Bayesian* if one tries to infer hidden parameters, and hence tries to do an inverse problem whereupon Bayes’ theorem is at least used once, and priors will appear. If one does not try to infer hidden parameters, but instead forward-propagates noise of e.g. data, then the model is often simply Hierarchical, without being Bayesian.

Hierarchical models

- Are at first difficult to set up, but one gets used to it.
- Allow you to pin-point problems in complex analyses, since the components of the model can be investigated individually.
- Allow to optimize individual contributions to the total uncertainty budget.
- Are extremely powerful when combined with physics, such that the ‘modelling’ aspect is ‘not just a model’ but ‘a mathematical description of nature’ instead.

Questions to be asked when setting up a Hierarchical Model:

- Do I have a physical understanding of the process?
- What is the set of all random variables, which I am interested in?
- Which of those are independent of each other?
- How many of them am I not interested in, such that I will later marginalize over them? How many integrals will this be, and how will I do those integrals?

## 2.2 The BHM Cheat Sheet

### 2.2.1 Cheat sheet for lengthy BHM derivations

1. We always want the a posteriori probability of the parameters, conditional on all data points that we have

$$\mathcal{P}(\text{parameters} | \text{all observables} = \text{all data points}), \quad (2.6)$$

meaning we always want

$$\mathcal{P}(\theta | \mathbf{x}). \quad (2.7)$$

2. To compute this, we will have to accept that we will pick up priors. We will introduce them here formally as distributions  $\pi$ . Some of them can cancel in the calculations, and others will remain, and we shall think about their meaning later on.
3. The first step is always to use **Bayes Theorem**, to access the likelihood, i.e. to tap into the noise process.
4. Further calculation then depends on the problem, but in general the following manipulations are helpful.
  - The *compound-block rule*: Compound events transform as blocks. Here we consider

the compound event  $x, y$

$$\mathcal{P}(m|x, y) = \frac{\mathcal{P}(x, y|m)\pi(m)}{\pi(x, y)}. \quad (2.8)$$

Note that  $(x, y)$  has changed sides of the conditional statement.

- The *expansion rule*: joint probabilities can be expanded in a sequence of sequential occurrences

$$\mathcal{P}(A, B, C) = \underbrace{\mathcal{P}(A)\mathcal{P}(B|A)}_{\mathcal{P}(A, B)}\mathcal{P}(C|A, B). \quad (2.9)$$

In words, this equation reads ‘First A occurs, then B occurs given that A occurred, then C occurs given that A and B have already occurred’.

- The *splitting compound-events rule*: How to move just one element across the conditional bar.

$$\mathcal{P}(A, B|C) = \mathcal{P}(A|C, B) \frac{\mathcal{P}(C, B)}{\mathcal{P}(C)}, \quad (2.10)$$

where we moved  $B$  across the bar.

5. Apart from model-averaging applications, it makes no sense to marginalize over variables who occur ‘*purely* on the *right* of the conditional bar  $|$ ’. For example

$$\int \mathcal{P}(u|v)dv, \quad (2.11)$$

is a none-sensical thing to do. The quantity  $v$  here can be any conditional statement;  $v$  isn’t even necessarily random here. Since we can only marginalize *random* variables, they need to occur at least once on the left of the conditional sign, such as in

$$\int \mathcal{P}(u|v)\mathcal{P}(v)dv. \quad (2.12)$$

Eq. (2.12) has a meaningful marginal over  $v$ :  $\mathcal{P}(v)$  indicates that  $v$  is a random variable. (In fact, the integrand of Eq. (2.12) is simply  $\mathcal{P}(u, v)$ , which can of course be marginalized over  $v$ .)

These are only auxilliary rules, and in general it is easiest to always go via the joint distribution, if any conditional statement is to be altered. For example, the *splitting compound-events rule* arises because

$$\mathcal{P}(A|C, B)\mathcal{P}(C, B) = \mathcal{P}(A, B|C)\mathcal{P}(C) = \mathcal{P}(A, B, C), \quad (2.13)$$

and similar for more than three random variables.

### 2.2.1 Parameter degeneracies

A parameter degeneracy occurs, if a data set  $\mathbf{x}$  does not react to each parameter  $\theta_i$  of a multivariate parameter vector  $\theta$  individually. Instead, the data  $\mathbf{x}$  may react only to certain combinations of multiple parameters. For example, assume a data set  $\mathbf{x}$  is a noisy realization of some signal shape  $\mathbf{s}(\theta)$  times a scalar amplitude  $A(\theta)$ :  $\langle \mathbf{x} \rangle = A(\hat{\theta})\mathbf{s}(\hat{\theta})$ . Assume now that the parameter vector is 5-dimensional, and the shape depends on the three even parameters  $\mathbf{s}(\theta_0, \theta_2, \theta_4)$ , whereas the amplitude is  $A = \theta_1 \cdot \theta_3$ . Then you cannot measure  $\theta_1$  and  $\theta_3$  individually, because the data only



constrain their product. Then, for all values of  $\theta_1$ , there will be a solution for  $\theta_3$ , namely  $\theta_3 = A/\theta_1$  which fits the data. The parameters  $\theta_1$  and  $\theta_3$  are hence said to be *perfectly degenerate*. An imperfect degeneracy occurs if one parameter can compensate for nearly all values of another parameter.

Parameter degeneracies are often hard-wired into a physical model. Sometimes they can be broken by combining with an auxiliary (sometimes non-astronomical) data set.

## 2.3 Selected probability distributions

**Theorem 2.3.1 — The Central Limit Theorem, and Limits to the Central Limit Theorem.**  
CLT

### The Chi-squared distribution

The  $\chi^2$ -distribution arises frequently in the ‘goodness of fit’ argumentation line. Its distribution is given by

$$\chi^2 \sim \frac{x^{n/2-1} \exp(-x/2)}{2^{n/2} \Gamma(n/2)} \quad (2.14)$$

### The Cauchy distribution

If  $x$  follows a general Cauchy distribution with scale parameter  $s$  and center  $\mu$ , then its density is given by

$$x \sim \frac{1}{\pi s} \left( \frac{s^2}{(x - \mu)^2 + s^2} \right). \quad (2.15)$$

The ratio of normal distributed variables follows the standard Cauchy distribution with  $s = 1$ ,  $\mu = 0$ .

### The t-distribution

The  $t$ -distribution is often mistaken for a Gaussian distribution, since it is symmetric, and has indeed a Gaussian limit. Its tails are however decidedly different before it Gaussianizes. Its density is given by

$$x \sim \frac{\Gamma(\frac{v+1}{2})}{\Gamma(v/2)} \frac{1}{\sqrt{v\pi}} \left( 1 + \frac{x^2}{v} \right)^{-(v+1)/2}. \quad (2.16)$$

### The F-distribution

The  $F$ -distribution arises frequently when Gaussian random variables are added in quadrature, and the thus resulting  $\chi^2$ -distributed variables are divided by each other. It has the density

$$x \sim \frac{\left( \frac{n_1}{n_2} \right)^{\frac{n_1}{n_2}}}{B\left( \frac{n_1}{2}, \frac{n_2}{2} \right)} x^{n_1/2-1} \left( 1 + \frac{n_1}{n_2} x \right)^{-\frac{n_1+n_2}{2}} \quad (2.17)$$

### The Beta-distribution

The Beta distribution generates random variables on the interval  $x \in (0, 1)$  and has parameters  $\alpha, \beta$  and density

$$x \sim \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1} (1-x)^{\beta-1}. \quad (2.18)$$



### The Gamma-distribution

The Gamma distribution generalizes the  $\chi^2$ -distribution and frequently occurs when Gaussian random variables are added in quadrature. Its density is

$$\text{Gamma}(x, n, s) = \frac{1}{\Gamma(n)} \frac{x^{n-1}}{s^n} \exp\left(-\frac{x}{s}\right). \quad (2.19)$$

### The Poisson distribution

The Poisson distribution generates random integers, it is given by

$$\mathcal{P}(n) = e^{-\mu} \frac{\mu^n}{n!}, \quad (2.20)$$

its mean is  $\langle n \rangle = \mu$ , which can be non-integer. Its variance is also  $\mu$ .

## 2.4 Priors

Priors are inevitable in Bayesian inference, and at the same time the prime target of criticism. Let us therefore spend a few words on priors.

### 2.4.1 Improper priors

If a prior  $\pi(x)$  is not normalizable, it est there is no real-valued scalar  $N$  such that

$$\frac{1}{N} \int \pi(x) dx = 1, \quad (2.21)$$

or if a prior can diverge for certain values of its argument

$$\pi(x) \rightarrow \infty, \text{ as } x \rightarrow x_0, \quad (2.22)$$

then a prior is said to be *improper*.

Interestingly, certain improper priors can still be used for inference. In order for an improper prior to still be admissible, it must guarantee a proper posterior. A proper posterior  $\mathcal{P}(\theta|\mathbf{x}) \propto \pi(\cdot)L(\mathbf{x}|\theta)$  is one where a constant  $N$  can be found such that

$$\frac{1}{N(\mathbf{x}, \theta)} \int \mathcal{P}(\theta|\mathbf{x}) d^n \theta = 1, \quad \forall \theta, \quad \forall \mathbf{x}, \quad (2.23)$$

and

$$\nexists(\mathbf{x}, \theta) \text{ for which } \mathcal{P}(\theta|\mathbf{x}) \rightarrow \infty. \quad (2.24)$$

The difficulty to overcome when working with improper priors, is to show that *for all* outcomes of the data  $\mathbf{x}$  and through the *entire* parameter space  $\theta$ , the posterior will always be proper.

### 2.4.2 Uninformative priors

Often, one wishes to adopt an ‘uninformative’ prior, but casting the human notion of ‘uninformativeness’ into mathematical shape is one of the major discussions in statistics. For example, a flat prior

$$\pi(x) = \text{Uniform}[a, b], \quad (2.25)$$

could be regarded as uninformative. However, it does not stay flat under non-linear transformations:  $\sqrt{x}, x^2$ , etc will have nonflat distributions, if  $x$  is uniformly distributed.

Similarly, if  $a \sim \text{Uniform}$  and  $b \sim \text{Uniform}$ , then  $ab$  will not be uniformly distributed anymore. If  $a$  and  $b$  are physically meaningful parameters (with sensible units), but  $ab$  is the quantity which a measurement reacts to, then it will become difficult to select an uninformative prior.

## 2.5 Model selection

One peculiarity of physics is that we usually have a mathematical *model framework* within which we wish to infer parameters. Not all natural sciences have such a mathematical framework. In our case, one often has the situation of different models still using the same parameters. In cosmology for example, one model might be Newtonian gravity, and its competing model might be General Relativity (or even an extension of General Relativity). Another example of model selection is whether neutrinos follow the normal or inverted hierarchy. A third example might be whether a badly resolved object of peculiar shape is one abnormal galaxy, or two blended galaxies.

In the sections above, we learned how to infer parameters, such as Newton's constant,  $G$ , or the age of the Universe – but how do we quantify which *model* the data prefer, Newtonian gravity, or General Relativity, or any other theory? For this, statistical *model comparison* is needed. The following quantities are often computed when the compatibility of a model with the data is investigated.

### 2.5.1 The Bayesian evidence

The evidence is an *integral* measure for a model's suitability. Within a model framework, it runs over the entire parameter space and accumulates in a weighted manner whether these values of the parameters are likely to produce the observed data. Since it runs over the entire parameter space, it thereby measures how likely the model is to produce the observed data *at all*.

Mathematically, the evidence is the normalization constant in Bayes' theorem, which we neglected during parameter inference,

$$\mathcal{P}(\theta, \mathcal{M} | \mathbf{x}) = \frac{L(\mathbf{x} | \theta, \mathcal{M}) \pi(\theta)}{\varepsilon(\mathbf{x} | \mathcal{M})} \quad (2.26)$$

$$\begin{aligned} \varepsilon(\mathbf{x} | \mathcal{M}) &= L(\mathbf{x} | \mathcal{M}) \\ &= \int L(\mathbf{x} | \theta) \pi(\theta) d^n \theta \end{aligned} \quad (2.27)$$

We will now focus on three so-called *information criteria*. In comparison to the evidence, the information criteria are *point estimates*: They base their analysis on a single point in parameter space, instead of including the entire possible parameter space.

## 2.6 The Devil's staircase and singular probability distributions

The above examples of probability densities in closed-form expressions, together with the law of how variable transformations change probability density functions, and together with BHM's could evoke the impression that probability densities could be written down for all random variables. This is not true. Often, the statistical behaviour of random variables is understood to a large degree, *despite* it being impossible to write down their probability density.

Often, a probability density can be shown to *exist*, and a subset of those existing distributions can be *constructed*, and a subset of the constructable distributions exist in a *closed-form* expression. Hence a gigantic literature on the study of different random variables and literature on the study of properties of their distributions exist. When constructing a BHM, for example, it is hence highly advisable to first check whether a solution, or all needed elements for the different hierarchical levels, exist at all.

One example is the Wishart distribution with  $n < d - 1$ . In this setting, the Wishart distribution is singular, where  $n$  is the number of summed up matrices  $\mathbf{x}\mathbf{x}^\top$ , with  $\mathbf{x} \sim \mathcal{G}(\cdot, \cdot)$ , and  $d$  the matrix

dimension. Random matrices

$$S = \sum_{i=1}^{n < d} \mathbf{x}_i \mathbf{x}_i^\top, \quad (2.28)$$

can be generated, but their density cannot be written down (despite some properties of it being known).

**Another Famous example: The Cantor density function.** The Cantor density function is singular with respect to the Lebesgue measure; it exists, and some properties of it are known, for example that it is a symmetric distribution. Nonetheless, no closed-form expression can be given, and the Cantor density function needs to be constructed recursively instead. It's cumulative distribution lacks absolute continuity and has been dubbed *Devil's staircase* due to its appearance and mathematical properties.

## 2.7 Missing data

Astronomy is notoriously plagued by *selection effects*: brighter objects are easier to detect, and therefore often numerically dominate over faint objects in catalogues. Further selection effects occur because astronomers tend to request additional observing time for interesting objects, and relatively few observing time is spent on statistical studies. At the time of writing, this applies for example to ALMA observations of protoplanetary disks, where disks with pronounced spirals are repeatedly imaged (since found interesting), yet it is to date unclear whether disks with spirals are intrinsically typical, or simply easier to detect due to the enhanced flux from the dense spirals, or which impact the communities preferential observing wishes have. (The decision committees for granting the observing time are the most likely to answer this question first.)

Further selection effects occur due to astronomical objects being variable on timescales not observable by humans. One well-understood example of this is the Hertzsprung-Russell diagram, when combined with models of stellar evolution: any arbitrary star on the sky is most likely one from the main sequence. But this is only because the main sequence is a long-lived stability strip.

Statistically, selection effects can be studied via the concept of *missing data*. To be made precise in the following, we split the data set into

$$\text{complete data} = \text{observed data} + \text{missing data}. \quad (2.29)$$

In a Bayesian manner, we can call the missing-data process  $\mathcal{I}$ . The missing-data process is said to be *ignorable*, if we have

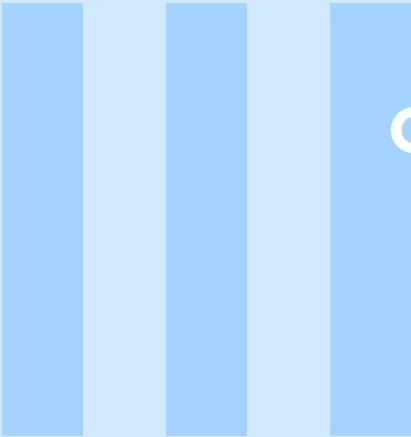
$$\mathcal{P}(\boldsymbol{\theta} | \mathbf{x}, \mathbf{y}_{\text{obs}}, \mathcal{I}) = \mathcal{P}(\boldsymbol{\theta} | \mathbf{x}, \mathbf{y}_{\text{obs}}). \quad (2.30)$$

This means the posterior of the parameters  $\boldsymbol{\theta}$  is independent of the missing-data process  $\mathcal{I}$ . The missing-property  $\mathcal{I}$  could thereby be an intrinsic property of the objects under study.

Otherwise, we could also have that a data-collection process  $\mathcal{C}$  influences the posterior, in the sense of

$$\mathcal{P}(\boldsymbol{\theta} | \mathbf{y}_{\text{obs}}, \mathcal{C}) \neq \mathcal{P}(\boldsymbol{\theta} | \mathbf{y}_{\text{obs}}). \quad (2.31)$$

This would imply that even though the data *themselves* have no intrinsic tendency to go missing, the observer's data collection process has produced missing data, by either ignoring or discarding data, or by not taking data, or by not recognizing that the taken data still depend on some outer effect. In astronomy, this easily happens if a catalogue of objects is retrospectively pruned: cuts in fluxes, cuts in magnitudes, or cuts in colour-space are an example of this. Cuts in signal-to-noise, or running an algorithm with false classification rates are another example.



# Part Three: Filtering, optimization, and sparsity

- 2.8 Filtering
- 2.9 Optimization
- 2.10 The matched filter
- 2.11 The Wiener filter
- 2.12 Weighting schemes
- 2.13 Creating your own optimal filters
- 2.14 Wavelets

## 2.8 Filtering

Filtering is a common technique in astronomy, with applications including gravitational wave detection, reconstruction of cosmic density fields, image denoising, object detection, data compression, and many more. How filters are used, depends on the specific field of astronomical research. Some fields use filters and report their *detection rate* and the detected signals. Other fields use filters to denoise e.g. and image, which is then analyzed further. When linearly compressing data, multiple filters are combined into a compression matrix, and the word ‘filter’ is then not used anymore, since the matrix is the new object of primary interest. Filters also appear when astronomical data are transformed between different coordinate systems, for example from Cartesian  $\mathbb{R}^3$  to the sphere  $S_2$  and a radial distance  $r$ , or from spherical harmonic space  $a_{lm}$  to real-space  $\vec{x}$ .

Filters are functions applied to noisy data; examples will be given in this chapter. Many filters can be shown to be *optimal*. Optimality is here to be understood in a strictly mathematical sense: the optimization is carried out by solving a maximization or minimization problem. For this, a loss function or an information measure must be defined. An optimal filter then minimizes the loss, or maximizes the information. Since loss and information are always measured *with respect to something*, a huge range of filters exist.

As an astronomer, one often encounters the situation that a certain filter is optimal with respect to something that one is not interested in. After learning about commonly used filters in astronomy, we will hence learn how to construct our own optimal filters.

All filters take the data as input. A *linear* filter produces a linear combination of the data as output. A *non-linear* filter produces a non-linear combination of the data as output.

Examples of linear filters are the following,

$$x_F(t) = \int_a^b F(t')x(t-t')dt', \quad (2.32)$$

or

$$x_F(t) = \int_a^b F(t)x(tk)dk, \quad (2.33)$$

where the integrals arise when  $x(t)$  is a continuous function. If the data are instead discrete  $x_i$ , one often sees filters written as an inner product

$$x_F = \langle \vec{F} | \mathbf{x} \rangle. \quad (2.34)$$

We write the filter as a vector with arrows  $\vec{F}$  in this script, to emphasize that it is a non-random vector. The bold  $\mathbf{x}$  is the random data vector. The  $\langle | \rangle$  above denotes an inner product, as the bra-ket notation of quantum mechanics. Eq. (2.33) above has the same structure as e.g. a Fourier transform

$$\hat{f}(\omega) = \int f(x)e^{i\omega x}dx, \quad (2.35)$$

or other basis transformations. This is why linear filters are sometimes said to project the data on a new, better suited basis.

Filters will perform inaccurately when they are applied to different noisy processes than the noise type they were created for. Famous noise types include (but are not limited to) the following:

### Famous noise types

1. Additive Gaussian noise: zero mean, and known covariance.
2. Additive Gaussian noise: non-zero mean, maybe covariance not known.
3. Additive non-Gaussian noise, etc...
4. Multiplicative noise (detector non-linearities; amplification; speckles)
5. Convolutional noise



## 2.9 Optimization

### 2.9.1 Lagrange multipliers

The method of Lagrange multipliers is very useful to optimize a system under additional constraints.

We imagine the function  $F(\vec{x})$  with  $\vec{x} \in \mathbb{R}^n$ . We furthermore impose  $p$  constraints of the form  $G_i(\vec{x}) = 0$ . Of all points in  $\mathbb{R}^n$ , these constraints pick out those which we are actually interested in. If the functions  $G_i(\vec{x}) = 0$  are continuous, then these constraints determine trajectories in  $\mathbb{R}^n$ . Along these trajectories, we now wish to find the maximum of  $F(\vec{x})$ .

We can achieve this as follows. If the function  $F(\vec{x})$  has maxima and/or minima, then there will be isocontours around these maxima/minima. When starting in a region where  $F(\vec{x})$  is particularly low, and then following the trajectories  $G_i$  ‘uphill’, then these trajectories will intersect a range of increasingly higher isocontours, until one is reached which is only tangentially touched. This is the highest isocontour of  $F$  reached by  $G_i$ . Following  $G_i$  further will lead to a decent on  $F(\vec{x})$ .

At the highest point, the  $G_i$  and the isocontour of  $F$  will be tangential. Since the gradient of  $F$  is perpendicular to the isocontours, this is equivalent to stating that the gradient of  $F$  and the gradients of the  $G_i$  must be parallel

$$\nabla F(\vec{x}) = - \sum_{i=1}^p \lambda_i \nabla G_i(\vec{x}) \quad \forall i. \quad (2.36)$$

The gradient is the vector

$$\nabla F(\vec{x}) = \begin{pmatrix} \frac{\partial F}{\partial x_1} \\ \frac{\partial F}{\partial x_2} \\ \cdot \\ \cdot \\ \frac{\partial F}{\partial x_n} \end{pmatrix}. \quad (2.37)$$

The scalars  $\lambda_i$  express the freedom that vectors of different lengths can still be parallel, and neither the gradients of  $F$  and the  $G_i$ , nor the gradients of all  $G_i$  amongst themselves, are necessarily equal.

Finding the constraint optimum is hence equal to solving the system

$$\begin{aligned} \nabla F(\vec{x}) &= - \sum_{i=1}^p \lambda_i \nabla G_i(\vec{x}) \quad \forall i. \\ G_i(\vec{x}) &= 0 \quad \forall i. \end{aligned} \quad (2.38)$$

This system is sometimes succinctly written as a *Lagrange function*,

$$\begin{aligned} \mathcal{L}(\vec{x}, \lambda_1, \dots, \lambda_p) &= F(\vec{x}) - \sum_{i=1}^p \lambda_i G_i(\vec{x}), \\ \nabla_{\vec{x}, \lambda_1, \dots, \lambda_p} \mathcal{L}(\vec{x}, \lambda_1, \dots, \lambda_p) &= \vec{0}. \end{aligned} \quad (2.39)$$

Note the derivatives wrt  $\lambda_i$  in the last line. Eq. (2.38) and Eq. (2.39) are precisely the same: carrying out the derivative with respect to  $\vec{x}$  of  $\mathcal{L}$  reproduces the first line of Eq. (2.38). Since the  $\lambda_i$  are linear, carrying out the derivatives wrt to  $\lambda_i$  simply reproduces the constraints  $G_i(\vec{x}) = 0$ . In the end, the renaming to a Lagrange function is not necessary; what is necessary, is the insight that Lagrange *multipliers* appear, since the gradients are only required to be parallel, and different lengths are still possible.

### 2.9.2 Cauchy-Schwarz inequality

Since we frequently deal with vectors, optimal bounds can also often be found by searching for the bound which saturates the Cauchy-Schwarz inequality. It reads

$$|\vec{x}^\top \vec{y}|^2 \leq (\vec{x}^\top \vec{x}) (\vec{y}^\top \vec{y}), \quad (2.40)$$

or equivalently

$$|\vec{x}^\top \vec{y}| \leq |\vec{x}| |\vec{y}|. \quad (2.41)$$

This induces the triangle-inequality

$$|\vec{x} + \vec{y}|^2 \leq (|\vec{x}| + |\vec{y}|)^2. \quad (2.42)$$

## 2.10 The matched filter

The matched filter applies in case of additive noise, and optimizes the expected signal-to-noise ratio  $\langle S/N \rangle$ .

If  $\vec{s}$  is the signal, and  $\mathbf{n}$  the noise, the matched filter makes the assumptions

$$\begin{aligned} \mathbf{x} &= \vec{s} + \mathbf{n} \\ \langle \mathbf{n} \rangle &= 0 \\ \langle \mathbf{n} \mathbf{n}^\top \rangle &= \mathbf{C}. \end{aligned} \quad (2.43)$$

We now wish to apply a linear filter  $\langle \vec{f} | \mathbf{x} \rangle = \langle \vec{f} | \vec{s} \rangle + \langle \vec{f} | \mathbf{n} \rangle$ , subject to the constraint that the  $S/N$

$$\langle S/N \rangle = \frac{\langle \vec{f} | \vec{s} \rangle^2}{\mathbb{E}(\langle \vec{f} | \mathbf{n} \rangle^2)} \quad (2.44)$$

is on average maximized. To optimize it, we first adapt the units of noise, and then demand in the second line that the Cauchy-Schwarz inequality be saturated

$$\begin{aligned} \langle S/N \rangle &= \frac{(\vec{f}^\top \mathbf{C}^{-\frac{1}{2}}) \mathbb{I}(\mathbf{C}^{-\frac{1}{2}} \vec{s})^2}{(\vec{f}^\top \mathbf{C}^{\frac{1}{2}}) \mathbb{I}(\mathbf{C}^{\frac{1}{2}} \vec{f})} \\ &\stackrel{!}{=} \frac{(\mathbf{C}^{\frac{1}{2}} \vec{f})^\top (\mathbf{C}^{\frac{1}{2}} \vec{f}) (\mathbf{C}^{-\frac{1}{2}} \vec{s})^\top (\mathbf{C}^{-\frac{1}{2}} \vec{s})}{(\mathbf{C}^{\frac{1}{2}} \vec{f})^\top (\mathbf{C}^{\frac{1}{2}} \vec{f})} \\ &= \vec{s}^\top \mathbf{C}^{-1} \vec{s}. \end{aligned} \quad (2.45)$$

We hence see that in order for

$$\frac{|\vec{f}^\top \vec{s}|^2}{\vec{f}^\top \mathbf{C} \vec{f}} = \vec{s}^\top \mathbf{C}^{-1} \vec{s} \quad (2.46)$$

to be possible, we need

$$\vec{f} = \mathbf{C}^{-1} \vec{s}. \quad (2.47)$$

Eq. (2.47) is the solution for the matched filter  $\vec{f}$ . The reason it is called a matched filter, is that it equals the inverse-variance weighted signal. It is therefore also often describes as ‘correlating the data with a signal template’.

## 2.11 The Wiener filter

In the matched filter above, the signal  $\vec{s}$  was a non-stochastic quantity. We will now promote it to a random variable  $\mathbf{s}$ . In astronomy, examples for  $\mathbf{s}$ -type signals are random fields or random signals as a function of time, e.g. fluctuating flux with a given spectrum. On top of these anyway already random signals, random noise  $\mathbf{n}$  may add. If both signal and noise are stationary, then the Wiener filter can be applied. It estimates which component of the sum is due to the random signal. We shall see that it is optimal in the sense of being a minimum-variance filter.

We write

$$\mathbf{x} = \mathbf{s} + \mathbf{n}, \quad \langle \mathbf{s}\mathbf{s}^\top \rangle = \mathbf{C}, \quad \langle \mathbf{n}\mathbf{n}^\top \rangle = \mathbf{N}. \quad (2.48)$$

The Wiener filter applies to a situation where  $\mathbf{s}$  is random, Gaussianly distributed with covariance  $\mathbf{C}$

$$\mathcal{P}(\mathbf{s}|\mathbf{C}) \propto \exp\left(-\frac{1}{2}\mathbf{s}^\top \mathbf{C}^{-1}\mathbf{s}\right). \quad (2.49)$$

It also assumes that Gaussian random noise then adds onto this anyways already random signal, and that the noise has covariance  $\mathbf{N}$

$$\mathcal{P}(\mathbf{x}|\mathbf{s}, \mathbf{N}) \propto \exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{s})^\top \mathbf{N}^{-1}(\mathbf{x} - \mathbf{s})\right). \quad (2.50)$$

In this setup we now wish to suppress the noise  $\mathbf{n}$  and thereby get an estimator  $\hat{\mathbf{s}}$  which is closer to  $\mathbf{s}$  than  $\mathbf{x}$  is. We know however only  $\mathbf{N}$ , not  $\mathbf{n}$ . To solve the problem, we take the route as outlined in the BHM-cheat-sheet. We want

$$\mathcal{P}(\mathbf{s}|\mathbf{x}, \mathbf{C}, \mathbf{N}) \quad (2.51)$$

and in order to get it, we start from the joint distribution

$$\mathcal{P}(\mathbf{s}|\mathbf{x}, \mathbf{C}, \mathbf{N}) = \frac{\mathcal{P}(\mathbf{s}, \mathbf{x}, \mathbf{C}, \mathbf{N})}{\pi(\mathbf{x}, \mathbf{C}, \mathbf{N})}. \quad (2.52)$$

We ponder the denominator first

$$\pi(\mathbf{x}, \mathbf{C}, \mathbf{N}) = \underbrace{\pi(\mathbf{N}|\mathbf{C}, \mathbf{x})}_{=\pi(\mathbf{N})} \underbrace{\pi(\mathbf{C}, \mathbf{x})}_{\pi(\mathbf{x}|\mathbf{C})\pi(\mathbf{C})}. \quad (2.53)$$

The first term  $\pi(\mathbf{N})$  arises since the covariance  $\mathbf{N}$  of the additive noise does not causally depend on the drawn data vector  $\mathbf{x}$  or the signal covariance  $\mathbf{C}$  (or even the signal  $\mathbf{s}$ ). In the second term, we meet the conditional dependency  $\pi(\mathbf{x}|\mathbf{C})$ . To make progress, we make the potentially idealizing assumption  $\pi(\mathbf{x}|\mathbf{C}) = \pi(\mathbf{x})$ . A counter-example of this would be that the power  $\mathbf{C}$  might be so large that the signal saturates the detector, which would cause a clipping problem.

We then arrive at

$$\mathcal{P}(\mathbf{s}|\mathbf{x}, \mathbf{C}, \mathbf{N}) = \frac{\mathcal{P}(\mathbf{s}, \mathbf{x}, \mathbf{C}, \mathbf{N})}{\pi(\mathbf{x})\pi(\mathbf{C})\pi(\mathbf{N})}, \quad (2.54)$$

and can now care about the top part of the fraction. We have

$$\mathcal{P}(\mathbf{s}, \mathbf{x}, \mathbf{C}, \mathbf{N}) = \underbrace{\mathcal{P}(\mathbf{x}|\mathbf{s}, \mathbf{C}, \mathbf{N})}_{\mathcal{P}(\mathbf{x}|\mathbf{s}, \mathbf{N})} \underbrace{\mathcal{P}(\mathbf{s}, \mathbf{C}, \mathbf{N})}_{\mathcal{P}(\mathbf{s}, \mathbf{C})}. \quad (2.55)$$

The first term simplifies as shown since the data  $\mathbf{x}$  can be fully generated, once  $\mathbf{s}$  and  $\mathbf{N}$  are given. The second term simplifies as shown since no knowledge on  $\mathbf{N}$  is needed to put  $\mathbf{s}$  and  $\mathbf{C}$  into relation to each other. Putting the numerator back into the fraction, we have

$$\begin{aligned}\mathcal{P}(\mathbf{s}|\mathbf{x}, \mathbf{C}, \mathbf{N}) &= \frac{\mathcal{P}(\mathbf{x}|\mathbf{s}, \mathbf{N}) \mathcal{P}(\mathbf{s}, \mathbf{C})}{\pi(\mathbf{x})\pi(\mathbf{C})\pi(\mathbf{N})} \\ &= \frac{\mathcal{P}(\mathbf{x}|\mathbf{s}, \mathbf{N}) \mathcal{P}(\mathbf{s}|\mathbf{C}) \cancel{\pi(\mathbf{C})}}{\pi(\mathbf{x}) \cancel{\pi(\mathbf{C})} \pi(\mathbf{N})}\end{aligned}\quad (2.56)$$

We now assume that  $\pi(\mathbf{x})$  and  $\pi(\mathbf{N})$  are flat, i.e. take some constant value no matter what  $\mathbf{x}$  and  $\mathbf{N}$  are. For  $\mathbf{N}$  this must indeed be the case, according to our assumptions where we claimed to ‘know’  $\mathbf{N}$  (which would correspond to a delta-function).

The only remaining distributions are then  $\mathcal{P}(\mathbf{x}|\mathbf{s}, \mathbf{N})$  and  $\mathcal{P}(\mathbf{s}|\mathbf{C})$ , both of which are the Gaussians given above. Plugging the Gaussians in, the distribution of  $\mathbf{s}$  is then

$$\mathcal{P}(\mathbf{s}|\mathbf{x}, \mathbf{C}, \mathbf{N}) \propto \mathcal{P}(\mathbf{x}|\mathbf{s}, \mathbf{N}) \mathcal{P}(\mathbf{s}|\mathbf{C}) \propto \exp\left(-\frac{1}{2}\left[(\mathbf{x} - \mathbf{s})^\top \mathbf{N}^{-1}(\mathbf{x} - \mathbf{s}) + \mathbf{s}^\top \mathbf{C}^{-1}\mathbf{s}\right]\right). \quad (2.57)$$

We already see that  $\mathbf{s}$  will therefore again follow a Gaussian distribution. If we hence collect all terms of  $\mathbf{s}$ , with the aim to rewrite the Gaussian as a function of  $\mathbf{s} - f(\mathbf{x})$ , then  $f(\mathbf{x})$  will be the maximum a posteriori solution for  $\mathbf{s}$ . We hence multiply out Eq. (2.57) and reorder terms:

$$\mathcal{P}(\mathbf{s}|\mathbf{x}, \mathbf{C}, \mathbf{N}) \propto \exp\left(-\frac{1}{2}\left[\mathbf{x}^\top \mathbf{N}^{-1}\mathbf{x} - \mathbf{x}^\top \mathbf{N}^{-1}\mathbf{s} - \mathbf{s}^\top \mathbf{N}^{-1}\mathbf{x} + \mathbf{s}^\top (\mathbf{N}^{-1} + \mathbf{C}^{-1})\mathbf{s}\right]\right). \quad (2.58)$$

The distribution  $\mathcal{P}(\mathbf{s}|\mathbf{x}, \mathbf{C}, \mathbf{N})$  needs to be normalized when integrated over  $\mathbf{s}$ , hence the term  $\mathbf{x}^\top \mathbf{N}^{-1}\mathbf{x}$  is not of interest to us, since it doesn’t depend on  $\mathbf{s}$ . The term  $\mathbf{s}^\top (\mathbf{N}^{-1} + \mathbf{C}^{-1})\mathbf{s}$  shows us that the new covariance matrix must be  $\mathbf{C}_W = (\mathbf{N}^{-1} + \mathbf{C}^{-1})^{-1}$ , where the inversion appears because the scalar product uses the inverse covariance matrix as metric. But if  $(\mathbf{N}^{-1} + \mathbf{C}^{-1})^{-1}$  is the new covariance matrix, then we need to adapt the terms  $\mathbf{x}^\top \mathbf{N}^{-1}\mathbf{s}$  and its transpose, to have the same covariance matrix. Otherwise we cannot arrive at a form  $\mathbf{s} - f(\mathbf{x})$ . We hence demand to achieve this by introducing a linear filter  $\mathbf{W}$  of the data  $\mathbf{x}$ :  $f(\mathbf{x}) \stackrel{!}{=} \mathbf{W}\mathbf{x}$ . Putting this into Eq. (2.58) we get

$$(\mathbf{W}\mathbf{x})^\top (\mathbf{C}^{-1} + \mathbf{N}^{-1})\mathbf{s} \stackrel{!}{=} \mathbf{x}^\top \mathbf{N}^{-1}\mathbf{s} \Rightarrow \mathbf{W}(\mathbf{C}^{-1} + \mathbf{N}^{-1}) = \mathbf{N}^{-1}. \quad (2.59)$$

Therefore our sought solution for the filter is

$$\mathbf{W} = (\mathbf{C}^{-1} + \mathbf{N}^{-1})\mathbf{N}^{-1}. \quad (2.60)$$

The full posterior normalized is then

$$\mathcal{P}(\mathbf{s}|\mathbf{x}, \mathbf{C}, \mathbf{N}) = \frac{1}{\sqrt{|2\pi(\mathbf{C}^{-1} + \mathbf{N}^{-1})^{-1}|}} \exp\left(-\frac{1}{2}(\mathbf{s} - \mathbf{W}\mathbf{x})^\top (\mathbf{C}^{-1} + \mathbf{N}^{-1})(\mathbf{s} - \mathbf{W}\mathbf{x})\right). \quad (2.61)$$

## 2.12 Weighting schemes

Filters applied to discrete data are related to weighting schemes. We show this on the example of *minimum-variance weighting*.

Minimum-variance weighting poses the question of how to optimally combine repeated measurements of the same quantity. Let

$$x_1, x_2, \dots, x_n \quad (2.62)$$

be independent estimates of a true quantity  $M$ . Each  $x_i$  shall thereby have

$$\text{var}(x_i) = \sigma_i^2, \quad (2.63)$$

with the  $\sigma_i^2$  differing from each other. We now wish to generate a weighted average which is optimal in the sense of having a minimum variance

$$m = \sum_{i=1}^n w_i x_i, \quad (2.64)$$

$$\text{var}(m) = \text{minimal}.$$

For starters, let us minimize the variance in the traditional way

$$\text{var}(m) = \sum_{i=1}^n w_i^2 \text{var}(x_i) \Rightarrow \frac{d}{dw_i} \text{var}(m) = 2w_i \sigma_i^2 = 0. \quad (2.65)$$

We hence arrive at the solution

$$w_i = 0 \quad \forall i. \quad (2.66)$$

This is the trivial solution, and not what we want, since it multiplies the data set with zero. Obviously, zero has indeed the lowest possible variance of all. This is why it is rather common in statistics that the ‘usual’ way of optimizing a problem simply defaults in the result ‘Multiply your data by zero.’. What we require to evade this situation, is *constraint optimization*.

To keep our data set from disappearing, we impose the constraint

$$\sum_i w_i = 1. \quad (2.67)$$

This leads to a constraint  $c(w_i) = (\sum_i w_i) - 1 = 0$  which has the right form to be added in the Lagrangian

$$\mathcal{L} = \text{var}(m) - \lambda c(w_i). \quad (2.68)$$

We then have the system

$$\begin{aligned} \frac{\partial}{\partial w_i} \mathcal{L} &= 2w_i \sigma_i^2 - \lambda \stackrel{!}{=} 0, \\ \sum_i w_i &= 1, \end{aligned} \quad (2.69)$$

to solve. Resolving for  $w_i$  in the first line and putting it into the sum on the second line results in

$$\frac{\lambda}{2} = \frac{1}{\sum_i \frac{1}{\sigma_i^2}}. \quad (2.70)$$

Eliminating  $\lambda$ , the weights are then

$$w_i = \frac{1}{\sigma_i^2} \frac{1}{\sum_i \frac{1}{\sigma_i^2}} \quad (2.71)$$

which gives us the (under this constraints) optimally weighted mean

$$m = \frac{\sum_i \frac{x_i}{\sigma_i^2}}{\sum_i \frac{1}{\sigma_i^2}}. \quad (2.72)$$

The weighted mean  $m$  is now the estimate of  $M$  which will scatter the least about  $M$ .

The weights  $w_i$  could now be collected into a filter  $\vec{f}$  such that

$$f_i = \frac{\frac{1}{\sigma_i^2}}{\sum_{i=1}^N \frac{1}{\sigma_i^2}} \quad (2.73)$$

and the weighted data vector is then  $\mathbf{x}_F = \langle \vec{f} | \mathbf{x} \rangle = \vec{f}^\top \mathbf{x}$ , as before.

## 2.13 Creating your own optimal filters

To create an optimal filter, two inputs are needed

- The correct noise type: is it additive, multiplicative, quadratic, pink, etc?
- A loss function or information measure: What are you trying to measure, and which quantity do you wish to maximize/minimize?

Famous loss functions include

1.  $\chi^2$
2.  $L_1, L_2$  norms
3. The total (scalar) variance (e.g. in inverse variance weighting)
4. Negative log-likelihoods
5. Kullback-Leibler divergence
6. Total Bias+Variance

but when you create your own filter, you will typically have a detailed understanding of your particular situation, so you can define your own loss function.

Famous information measures include

1. Fisher information
2. Shannon information
3. Entropy

where it is again likely that you might want to define your own measure with respect to which you wish to optimize.

## 2.14 Wavelets

Fourier trafo	Physical signal	Wavelets
Unbounded infinite interval $-\infty, +\infty$	often compact, bounded interval	finite, compact interval $[a, b]$
orthonormal	hopefully square integrable	orthonormal
periodic basis functions	signal not periodic	unperiodic basis functions
If $f$ local in real space, $\hat{f}$ mostly not local in Fourier space. And vice versa.	Information often local in real space and Fourier space	Made to be local in real space and Fourier space.
Age: $\approx$ since 1820s.	—	Age: $\approx$ 150 years younger.
$e^{ikx}, \cos(\omega x), \sin(\omega x), \delta(x)$	GW example:	e.g. Haar wavelet and GM-wavelet

More to come, for now kindly refer to the exercise sheet on gravitational waves.

# IV

## Part Four: Numerical techniques

<b>3</b>	<b>Sampling Methods .....</b>	<b>37</b>
3.1	The Gibbs sampler	
3.2	Detailed Balance	
3.3	Metropolis(-Hastings) algorithm	
3.4	Hamilton Monte Carlo	
3.5	Convergence	
3.6	Manipulations of MCMC chains	
<b>4</b>	<b>Basics of Machine Learning .....</b>	<b>41</b>
4.1	PAC learnable and finite VC-dimension	
4.2	Artificial Neural Networks (ANN)	
4.3	Linear Algebra and Matrix Manipulations	



Given a scientific question, answering it may mean digging through high-dimensional data sets, which in turn may imply that answering the question is numerically impossible without refuge to the correct numerical methods. In modern astronomy, the struggle for numerical feasibility is indeed common, and this chapter hence presents the most common techniques to render complex astronomical data analyses feasible. We first turn to sampling methods which are e.g. relevant when a highly-dimensional posterior is to be represented, and then to examples of machine learning methods which are relevant for e.g. astronomical object classification, or transient detection in the large sky surveys.

### 3. Sampling Methods

In high dimensions, it becomes increasingly inefficient to compute likelihoods or posteriors on a grid. This is because the volume of the empty edges (regions of low likelihood) become comparatively ever larger (see also the exercise on the unit sphere in high dimensions).

**Definition 3.0.1 Markov Chain:** A sequence of random variables  $x_1, x_2, \dots, x_t, \dots, x_N$  where  $\forall t$  the *distribution* of  $x_{t+1}$  depends only on  $x_t$ . The distribution may change for each  $t$ : A Markov Chain uses a *transition distribution*  $T_t(x_t|x_{t-1})$  and  $T_t$  may change as a function of  $t$ , but it cannot depend on earlier states than  $x_{t-1}$ .

For special choices of the transition distribution the Markov Chain converges towards the sought posterior distribution  $\mathcal{P}(\theta|\mathbf{x})$ , in the sense that it then generates samples which have a local density  $n$

$$n(\theta) \propto \mathcal{P}(\theta|\mathbf{x}). \quad (3.1)$$

#### 3.1 The Gibbs sampler

Gibbs sampling is also known as ‘alternate conditional sampling’ and builds up a Markov Chain where the transition distributions are directly visible: The Gibbs sampler iterates over conditional draws.

Let  $\theta$  be the random vector whose distribution we wish to approximate through sampling. Then  $\theta$  can be split up in  $d$  subcomponents, which can either be its elements, or subvectors. We write

$$\theta = \theta_d \cup \theta_{\bar{d}}, \quad (3.2)$$

where  $\theta_d$  is a subvector of  $\theta$ , and  $\theta_{\bar{d}}$  is its complement, in the sense that it contains in the right order all elements of  $\theta$  which are not in  $\theta_d$ .

Gibbs sampling now employs an outer loop, with index  $t$  which denotes the iterations, and an inner loop with index  $d$  where it cycles over the sub-components  $\theta_d$ .

For each inner loop, the conditional probabilities  $\mathcal{P}(\theta_d|\theta_{\bar{d}})$  must be known. This is often the case in data space, not so often in parameter space. Before the next iteration  $t$  is started, the order of drawing subcomponents  $d$  is typically jumbled.

The Gibbs sampling algorithm has the following pseudo code

**Gibbs pseudo code**

1. For int  $i = 0, \dots, N_{\text{samples}}$ 
  - (a) For all  $d$  draw  $\theta_d \sim \mathcal{P}(\theta_d | \theta_{\bar{d}})$
2. Randomize ordering of  $d$ .

The Gibbs sampler has no explicit 'rejection' step as the algorithms below, which can make it highly efficient – however, drawing from the conditionals forces the sampler to walk in a zig-zag like manner through the full multivariate distribution. For highly correlated variables, the sampler will then be inefficient.

### 3.2 Detailed Balance

Monte Carlo Markov Chains borrow the idea of transition probabilities between different states as known from thermodynamics or quantum mechanics. If there exist multiple discrete probability levels  $\mathcal{P}_i$ , and the possibility of a transition from state  $\mathcal{P}_i$  to state  $\mathcal{P}_j$  has the rate probability  $r_{i \rightarrow j}$ , then an equilibrium between the occupation of the different states  $\mathcal{P}_i$  is reached if

$$r_{i \rightarrow j} \mathcal{P}_i = r_{j \rightarrow i} \mathcal{P}_j, \quad (3.3)$$

In equilibrium, the density  $n$  of accepted points is then proportional to the posterior

$$n(\theta) \propto \mathcal{P}(\theta), \quad (3.4)$$

### 3.3 Metropolis(-Hastings) algorithm

The Metropolis-Hastings algorithm combines a jumping distribution  $J_t(\theta_a | \theta_b)$  and an acceptance/rejection rule. For the Metropolis-setup, the jumping distribution needs to be symmetric in the sense of  $J_t(\theta_a | \theta_b) = J_t(\theta_b | \theta_a)$ , for Metropolis-Hastings a change in the acceptance/rejection rule can also allow for non-symmetric jumping distributions. We here describe the Metropolis algorithm with a Gaussian jumping distribution.

Prerequisites to reach equilibrium (i.e. prerequisites for the chain to 'converge') are

- For  $t \rightarrow \infty$  the jumping distribution  $J_t(\theta_a | \theta_b)$  must be able to reach all possible states with finite probability. Otherwise certain areas of the target distribution cannot be reached.
- The chain must be aperiodic (and non-transient, which is anyway the case apart from artificial counter examples).

1. Come up with a guess for a Gaussian approximation  $\mathcal{G}_P(\theta)$  to the posterior. A crudely estimated parameter covariance matrix from potential previous runs serves this purpose.
2. FOR  $i = 0$  TO  $N_{MCMC}$   
if  $i = 0$  evaluate the posterior  $\mathcal{P}$  at some point  $\theta_0$  in parameter space, preferentially one with a presumed high likelihood, else use the current  $\theta_i$  of the chain.
3. Draw a random step in parameter space  $\Delta\theta_i \sim \mathcal{G}_P(\theta)$ .
4. Calculate  $\mathcal{P}(\theta_i + \Delta\theta_i)$  and  $R = \frac{\mathcal{P}(\theta_i + \Delta\theta_i)}{\mathcal{P}(\theta_i)}$ .
5. IF  $R > 1$ , then the posterior probability at the new point  $\theta_i + \Delta\theta_i$  is larger than the old probability; the new point is then accepted as  $\theta_{i+1} = \theta_i + \Delta\theta_i$ .
6. IF  $R < 1$ , then draw  $\alpha \sim \text{Uniform}[0, 1]$ .  
IF  $\alpha > R$ , then  $\theta_{i+1} = \theta_i$ , i.e. the point  $\theta_i + \Delta\theta_i$  is rejected because it has too low a probability.  
IF, however,  $\alpha < R$ , then  $\theta_{i+1} = \theta_i + \Delta\theta_i$ , i.e. the trial point is accepted because it has still a fairly high probability.
7. Store all points  $\theta_i$ . These then build up the Monte Carlo Markov Chain.

### 3.4 Hamilton Monte Carlo

Hamilton-Monte-Carlo (HMC) sampling essentially implements a Metropolis-Hasting sampler, but additionally increases the distance between accepted samples in a way which generalizes better to high dimensions.

HMC introduces a potential energy  $U(\theta)$  as the logarithm of the posterior to be sampled

$$U(\theta) = -\log \mathcal{P}(\theta). \quad (3.5)$$

It then introduces a kinetic energy

$$K(\mathbf{u}) = \mathbf{u}^T \mathbf{u} / 2, \quad \mathbf{u} \sim \mathcal{N}(0, \mathbb{I}), \quad (3.6)$$

where the velocity  $\mathbf{u}$  is a random variable drawn from the multivariate normal distribution. Kinetic and potential energy are then combined into a Hamiltonian

$$H(\theta, \mathbf{u}) = U(\theta) + K(\mathbf{u}). \quad (3.7)$$

The exponential of the Hamiltonian then relates to the posterior  $\mathcal{P}(\theta)$  as

$$\exp(-H(\theta, \mathbf{u})) = \mathcal{P}(\theta) \mathcal{N}(0, \mathbb{I}) \quad (3.8)$$

If the auxiliary velocities  $\mathbf{u}$  are marginalized over, then sampling  $\exp(-H)$  samples  $\mathcal{P}(\theta)$ .

The HMC algorithm increases the distance between two Metropolis-Hastings steps by walking along trajectories which solve the Hamiltonian equations of motion. These are

$$\dot{\theta} = \mathbf{u}, \quad \dot{u}_i = -\frac{\partial H}{\partial \theta_i}. \quad (3.9)$$

These can be solved numerically and as they will later only provide auxiliary steps, it is sufficient to apply the leapfrog algorithm

$$\begin{aligned} u_i(t + \frac{\epsilon}{2}) &= u_i(t) - \frac{\epsilon}{2} \left( \frac{\partial U}{\partial \theta_i} \right)_{\theta(t)} \\ \theta_i(t + \epsilon) &= \theta_i(t) + \epsilon u_i(t + \epsilon/2) \\ u_i(t + \frac{\epsilon}{2}) &= u_i(t) - \frac{\epsilon}{2} \left( \frac{\partial U}{\partial \theta_i} \right)_{\theta(t+\epsilon)}. \end{aligned} \quad (3.10)$$

The HMC sampler now alternates between leapfrog steps (to increase the distance between samples) and Metropolis-Hastings steps (to satisfy the detailed balance). The HMC algorithm is given by

#### Hamilton Monte Carlo algorithm

1. FOR  $i = 0$  TO  $N_{MCMC}$   
     if  $i = 0$ , choose a starting point  $\theta_0$ ,  
     else use the current  $\theta_i$  of the chain.
2. Draw a random velocity  $\mathbf{u}_i \sim \mathcal{N}(0, \mathbb{I})$ .  
     **Leapfrog loop**  
     (a) Use  $\theta_i$  and  $\mathbf{u}_i$  as initial conditions for the Hamiltonian equations of motions.  
     (b) For  $j = 0$  to  $N_L$   
         make leapfrog steps that update  $(\theta_j, \mathbf{u}_j) \rightarrow (\theta_{j+1}, \mathbf{u}_{j+1})$
3. Having arrived at  $(\theta_{N_L}, \mathbf{u}_{N_L})$ , calculate  $R = \exp[-H(\theta_i, \mathbf{u}_i) + H(\theta_{N_L}, \mathbf{u}_{N_L})]$ .
4. IF  $R > 1$ , the new point is accepted,  $\theta_{i+1} = \theta_i$ .
5. IF  $R < 1$ , draw  $\alpha \sim \text{Uniform}[0, 1]$ .  
     IF  $\alpha > R$ , then  $\theta_{i+1} = \theta_i$ , i.e. the trial point  $\theta_{N_L}$  is rejected.  
     IF  $\alpha < R$ , then  $\theta_{i+1} = \theta_{N_L}$ , i.e. the trial point is accepted.

### 3.5 Convergence

In real life, none of the above algorithms guarantees by default that samples generated from it are automatically drawn from the posterior density. The Gibbs-sampler can be very slow or inefficient for strongly correlated parameters, and the Metropolis-Hastings and Hamilton Monte Carlo sampler have free parameters which need to be tuned to enhance the sampler's efficiency. For inconvenient settings of those parameters, the samplers will not converge towards the posterior distribution in a humanly acceptable time.

### 3.6 Manipulations of MCMC chains

Kindly refer to the exercise sheet on MCMC sampling.





## 4. Basics of Machine Learning

In the past chapters we have gathered experience with randomness. One of the revelations was that seemingly very different data sets can still come from the same underlying model, and the seeming difference is then due to the noise realizations. In the exercises we have additionally gained experience in how to code up algorithms which are able to generate random samples, or conduct inference based on random samples.

Built on these prerequisites, we now transit to a further idea. So far, we wrote a static programs, which deterministically mapped input to output. To the C/C++ programmers this deterministic nature is evident, since they need to set the random seeds by hand, and hence indeed observe that they can perfectly reproduce the output of the program infinitely many times. To the Python programmers, the question might impose itself whether their programs were deterministic, since on each execution the histograms, the noise, and the likelihood peaks will have shifted. This is however due to Python initializing the random seed on its own (if you wish to convince yourself that your programs were deterministic, you can also provide fixed seeds to Python).

We now wonder, in which sense we can go beyond writing deterministic programs. Is it possible to write an algorithm, which right from the start understands that its input is just one random realization out of many, and which is able to abstract beyond the realization given? In other words, we wish to create an algorithm which generalizes from experience, and which recognizes that statistically similar (but not identical) situations should prompt it to take similar (but not identical) actions. This brings us to the realm of *machine learning*.

Most machine learning algorithms have the following common elements

1. A training phase, during which training data is either provided to the computer, or it is given the possibility to actively generate its own experience.
2. Tasks are specified, which the computer has to accomplish.
3. A performance measure is imposed, to describe the desired result. This can e.g. be a loss function, or a reward/punishment-system.

For the purpose of this astronomy-focused lecture we distinguish the following categories of machine learning

- Supervised Learning: Training data are provided as input, together with the desired output.

Feedback is provided through e.g. a loss function, and the algorithm has to learn how to map statistically similar input to the equivalent output.

- **Unsupervised Learning:** The algorithm has to find structure in the data, e.g. for data mining and feature learning.
- **Reinforcement Learning:** The algorithm is not only a passive structure which is fed input data, but instead an active agent itself, who interacts with the environment. Training is conducted by assigning reward & punishment points, depending on the actions taken by the computer. Random trials are then conducted by the algorithm, and refined, based on the reward system. For example, to train an animated figure to walk, the reward can linearly increase with ‘the distance to be crossed’.

All insights which we gained from studying selected elements of statistics in detail will now remain true, even when we put them together into a complex machine learning algorithm. For example, if a machine-learning algorithm overfits the training data, then it will have problems in generalizing beyond the training data set, and then produce unintended results when applied to real data.

## 4.1 PAC learnable and finite VC-dimension

In preparation.

### 4.1.1 No Free-Lunch Theorem

In preparation.

## 4.2 Artificial Neural Networks (ANN)

In preparation.

### Theorem 4.2.1 — The Universal Representation Theorem.

There will exist an architecture for an ANN to approximate any continuous function on a compact set.

Smallprint: In the sense of  $\hat{f}(x) = \sum_{i=1}^N v_i a(w_i^\top x + b_i)$  with  $|f(x) - \hat{f}(x)| < \varepsilon$ , for activation function  $a(\cdot)$  monotonically increasing, continuous and bounded.

The loss function is the distance measure which judges statistical compatibility. It is also often the thing to be optimized (e.g. an information measure). In the end, it is the teacher, who decided what precisely is learned from the labelled data.

## 4.3 Linear Algebra and Matrix Manipulations

### 4.3.1 Singular value decomposition

Let  $M$  be a potentially non-square matrix,  $M \in \mathbb{R}^{m,n}$ , and let  $s > 0$  be a real-valued scalar. A pair of normalized vectors  $\vec{v} \in \mathbb{R}^n$  and  $\vec{u} \in \mathbb{R}^m$  is then called right- and left-singular vectors of  $M$  if they satisfy

$$M\vec{v} = s\vec{u}, \text{ and } M^\top \vec{u} = s\vec{v}. \quad (4.1)$$

The scalar  $s$  is then called a *singular* value of  $M$ . If  $M$  has rank  $r$ , then there will exist  $r$  such singular values. The matrix  $M$  can then be decomposed as

$$M = \sum_{i=1}^r s_i \vec{u}_i \vec{v}_i^\top. \quad (4.2)$$



Likewise, if a matrix  $U$  is constructed which has the vectors  $\vec{u}_i$  as columns, and a matrix  $V$  which has the vectors  $\vec{v}_i$  as columns, then

$$M = USV^T, \quad (4.3)$$

where

$$S = \text{diag}(s_1, \dots, s_r). \quad (4.4)$$

Furthermore,

$$UU^T = \mathbb{I}, \quad VV^T = \mathbb{I}. \quad (4.5)$$

